

République Algérienne Démocratique et Populaire

Université A. MIRA de Béjaia
Faculté des Sciences Exactes
Département de Recherche Opérationnelle



Mémoire présenté pour l'obtention du diplôme de master
Spécialité : Sciences de Données et Aide à la Décision

**Implémentation d'un modèle machine learning pour la
prédiction de congestion dans les équipements d'antennes**

Présenté par :
DEBBOUZ Sami Zakaria

Soutenu le 19/09/2024, devant le jury composé de :

Dr B. BRAHMI	Président du jury	UAMB - Bejaia
Dr H. EL BOUHISSI	Encadrante	UAMB - Bejaia
Dr S. TOUATI	Examineur	UAMB - Bejaia

Année Universitaire 2023 2024

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à Dieu, le Tout-Puissant et Miséricordieux, qui m'a doté de la force, de la volonté et de la patience nécessaires pour mener à bien ce travail.

J'adresse mes remerciements les plus sincères à mon encadrante, Madame Houda El Bouhissi, dont les conseils avisés, les orientations éclairées et la disponibilité constante ont été essentiels à la réalisation de cette étude.

J'exprime également ma reconnaissance envers mes professeurs, qui m'ont offert un cursus universitaire de qualité, ainsi que leur soutien continu et leur temps précieux.

Mes remerciements vont également aux membres du jury, qui me font l'honneur d'évaluer ce travail.

Enfin, je tiens à manifester ma gratitude la plus sincère à tous ceux qui ont, de près ou de loin, contribué à la réalisation de ce projet.

Dédicaces

Ce travail est dédié à ma famille bien-aimée, en particulier à mes chers parents, à ma sœur Yasmine et son époux Abderrahmane, ainsi qu'à mes deux frères, Amine et Khalil, dont le soutien indéfectible et les encouragements ont illuminé chaque étape de mon parcours universitaire.

Je le dédie également à mes adorables neveux, Anes et Iskandar, ainsi qu'à ma charmante nièce, Rym, pour leur présence chaleureuse et leur affection.

À tous mes amis et à chaque personne avec qui j'ai eu le privilège d'entretenir une relation précieuse, je leur adresse cette dédicace avec tout mon amour et ma reconnaissance.

Table des matières

Liste des figures	6
Liste des tableaux	9
Liste des abréviations	10
Introduction générale	13
1 Notions fondamentales	14
1.1 Introduction	14
1.2 Les réseaux mobiles	14
1.2.1 Définition d'un réseau mobile	14
1.2.2 Le concept cellulaire	14
1.2.3 Architecture d'un réseau mobile	16
1.2.4 Les différentes générations des réseaux mobiles	16
1.2.4.1 La première génération (1G)	17
1.2.4.2 La deuxième génération (2G)	17
1.2.4.3 La troisième génération (3G)	17
1.2.4.4 La quatrième génération (4G)	17
1.2.5 La congestion	19
1.2.5.1 Facteurs contribuant à la congestion des réseaux	20
1.3 Indicateur Clé de Performance (KPI)	20
1.3.1 Les KPI dans les réseaux mobiles	21
1.3.1.1 Mobilité	21
1.3.1.2 Intégrité	21
1.3.1.3 Disponibilité	22

TABLE DES MATIÈRES

1.3.2	Importance de la surveillance continue des KPI	22
1.4	Séries temporelles	22
1.4.1	Composantes d'une série temporelle	23
1.4.1.1	Tendance	23
1.4.1.2	Saisonnalité	24
1.4.1.3	Cycle	25
1.4.1.4	Résidu (aléa, bruit)	25
1.4.2	Types des séries temporelles	25
1.4.3	Stationnarité des séries temporelles	26
1.4.4	Modèles classiques des séries temporelles	26
1.4.4.1	Modèle autorégressif (AR)	26
1.4.4.2	Modèle de la moyenne mobile (MA)	27
1.4.4.3	Modèle de moyenne mobile autorégressive (ARMA)	27
1.4.4.4	Moyenne Mobile AutoRégressive Intégrée (Modèle ARIMA)	28
1.5	Détection et prédiction de congestions	28
1.5.1	La détection de congestions	28
1.5.2	La prédiction de congestions	29
1.6	Machine Learning	29
1.6.1	Définition	29
1.6.2	Types d'apprentissage automatique	29
1.6.2.1	Apprentissage supervisé	30
1.6.2.2	Apprentissage non supervisé	30
1.6.2.3	Apprentissage par renforcement	31
1.7	Deep Learning	32
1.8	Conclusion	33
2	Etat de l'art	34
2.1	Introduction	34
2.2	Travaux connexes	34
2.3	Analyse et discussion	40
2.4	Conclusion	41
3	Approche proposée	42
3.1	Introduction	42
3.2	Approche proposée	42
3.3	Vue détaillée du bloc de prédiction	46
3.3.1	Extraction des données	46
3.3.2	Préparation des données	46
3.3.3	Processus de prédiction	47

TABLE DES MATIÈRES

3.3.3.1	Présentation du modèle CatBoost	47
3.3.3.2	Principaux hyperparamètres de CatBoost	48
3.4	Conclusion	50
4	Expérimentation et évaluation	51
4.1	Introduction	51
4.2	Description du dataset	51
4.2.1	Description	51
4.2.2	Préparation de données	52
4.3	Environnement de développement	56
4.3.1	Environnement logiciel	56
4.3.2	Bibliothèques Utilisées	57
4.3.3	Environnement Matériel	58
4.4	Construction et évaluation des modèles de prédiction	58
4.4.1	Construction des modèles	58
4.4.2	Métriques d'évaluation utilisées	60
4.4.3	Évaluation des modèles construits	62
4.5	Identification et visualisation des appareils congestionnés	66
4.6	Conclusion	67
	Conclusion générale	69

Table des figures

1.1	Figure représentant un motif élémentaire (à gauche) et un ensemble de motifs dans un réseau (à droite).	15
1.2	Structure d'un réseau de téléphonie mobile.	16
1.3	Architecture du réseau 4G	18
1.4	Exemple de série temporelle montrant l'évolution du trafic aérien mondial .	24
1.5	Exemple de la saisonnalité du marché de l'immobilier	24
1.6	Les deux Types d'apprentissage supervisé	30
1.7	Le clustering et les règles d'association	31
1.8	Processus d'apprentissage par renforcement	32
1.9	Relation entre le Machine Learning et le Deep Learning	32
3.1	Architecture globale de l'approche	43
3.2	Architecture globale du bloc de prédiction du pourcentage d'utilisation des équipements	44
3.3	Architecture globale d'identification des cas suspects de congestions	45
3.4	Schéma de fonctionnement du modèle CatBoost	48
4.1	Aperçu du dataset	52
4.2	Nombre d'observations après la première et la deuxième étape de suppression des doublons	53
4.3	Nombre de valeurs manquantes par équipement	54
4.4	Nombre de valeurs manquantes par équipement après l'imputation	55
4.5	Aperçu des données Agrégées par Heure	56
4.6	Extrait des données après l'ajout des caractéristiques temporelles	56
4.7	Boîte à moustache de la MAE des modèles CatBoost pour la direction réception(RX)	62
4.8	Boîte à moustache de la RMSE des modèles CatBoost pour la direction réception(RX)	63

TABLE DES FIGURES

4.9	Boîte à moustache du R^2 des modèles CatBoost pour la direction réception(RX)	63
4.10	Boîte à moustache de la MAE des modèles CatBoost pour la direction transmission(TX)	64
4.11	Boîte à moustache de la RMSE des modèles CatBoost pour la direction transmission(TX)	65
4.12	Boîte à moustache du R^2 des modèles CatBoost pour la direction transmission(TX)	65
4.13	Tableau de bord de prédiction des congestions	66

Liste des tableaux

2.1	Comparaison des approches de prédiction du trafic réseau et des congestions	38
4.1	Valeurs des hyperparamètres possibles pour les modèles de prédiction . . .	60

Liste des abréviations

Abréviation	Signification
RAN	Radio Access Network
CN	Core Network
UE	User Equipment
LTE	Long Term Evolution
eNodeB	evolved Node B
HSS	Home Subscriber Server
MME	Mobility Management Entity
S-GW	Serving Gateway
P-GW	PDN Gateway (Packet Data Network Gateway)
QoS	Quality of Service
KPI	Key Performance Indicator
IA	Intelligence Artificielle
ARIMAX	AutoRegressive Integrated Moving Average with Exogenous Variables
MLP	Multi-Layer Perceptron
LSTM	Long Short-Term Memory
ARIMA	Autoregressive Integrated Moving Average model
GRU	Gated Recurrent Units
FbProphet	Facebook Prophet
CNN	Convolutional Neural Network
GMDH	Group Method of Data Handling
ANN	Artificial Neural Network
ANFIS	Adaptive Neuro-Fuzzy Inference System
FFNN	Feedforward Neural Network
CDR	Call Detail Record
IDS	Insights Data Storage
R²	Coefficient of Determination

Abréviation	Signification
MSE	Mean Squared Error
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
CSV	Comma Separated Values
ML	Machine Learning
MAE	Mean Absolute Error
BATS	Exponential smoothing state space model with Box-Cox transformation, ARMA errors,
TBATS	Trigonometric Exponential smoothing state space model with Box-Cox transformation,

Introduction Générale

I. Contexte

À l'ère de la connectivité omniprésente, la qualité et la fiabilité des réseaux de télécommunication mobile sont devenues essentielles. Les utilisateurs dépendent de plus en plus de leurs appareils mobiles pour rester connectés, accomplir des tâches diverses et accéder à l'information en temps réel. Cette dépendance accrue met les réseaux sous une pression constante pour supporter un volume croissant de données, rendant la surveillance de la qualité du réseau plus complexe et critique. Optimum Telecom Algérie "**Djezzy**", conscient de ces enjeux, s'efforce de maintenir une performance optimale malgré les défis grandissants.

II. Problématique

La congestion des réseaux est l'un des principaux obstacles affectant la qualité des services de télécommunication aujourd'hui. Elle se manifeste par des ralentissements de la transmission, des latences accrues, et parfois des pertes de connexion, ce qui détériore l'expérience utilisateur. Dans un contexte où la demande en données explose, les méthodes traditionnelles de surveillance des performances des équipements radio montrent rapidement leurs limites. Elles sont souvent réactives et peinent à prévoir les congestions avant qu'elles ne se produisent, rendant la gestion de la congestion complexe et peu efficace.

L'incapacité à anticiper les congestions peut entraîner des interruptions de service, des dégradations de la qualité, et nuire à la satisfaction des clients ainsi qu'à l'image de l'opérateur. Pour des opérateurs comme **DJEZZY**, il est crucial de passer à des approches plus avancées et automatisées, capables de prédire les congestions avant qu'elles ne surviennent. Cela permettrait une gestion proactive de la congestion, réduisant ainsi les risques d'interruption de service et garantissant une qualité de service optimale pour les utilisateurs.

III. Objectifs

L'objectif de mon travail est de prédire les congestions futures du réseau en utilisant des techniques de machine learning. Ce projet vise à développer des modèles prédictifs capables d'identifier les signes avant-coureurs de la congestion, permettant ainsi à **Djezzy** d'anticiper ces problèmes et de prendre des mesures préventives pour maintenir la qualité du service.

IV. Organisation du mémoire

Tout d'abord, nous avons commencé notre mémoire par une introduction générale du projet en présentant le contexte, la problématique à laquelle il fait face ainsi que les objectifs à atteindre. Le reste du mémoire est organisé comme suit :

- **Chapitre 1** : Ce chapitre présente des notions et des généralités relatifs à notre mémoire, notamment les réseaux de télécommunication, la congestion réseau, ainsi que l'apprentissage automatique.
- **Chapitre 2** : Ce chapitre explore diverses recherches portant sur la prédiction de la congestion réseau. Nous expliquons les approches des études et leurs résultats en développant un état de l'art.
- **Chapitre 3** : Ce chapitre se concentre sur notre approche proposée pour la prédiction de la congestion dans les réseaux de télécommunication. Nous fournissons une description détaillée de la méthodologie et de ses techniques. Cette section représente notre contribution à la prédiction de la congestion réseau.
- **Chapitre 4** : Ce chapitre couvre à la fois la phase expérimentale et l'évaluation de notre modèle de prédiction. Il comprend une description détaillée des données utilisées, de l'environnement matériel et logiciel, ainsi qu'une explication complète des phases impliquées dans la mise en œuvre de notre approche.

Chapitre 1

Notions fondamentales

1.1 Introduction

Dans le cadre de notre mémoire, il est essentiel de bien comprendre les concepts fondamentaux qui sous-tendent la problématique de la prédiction des congestions dans les réseaux téléphoniques. Ce chapitre fournira un aperçu global des concepts de base nécessaires à cette compréhension, notamment les réseaux téléphoniques, la congestion, les Indicateurs Clés de Performance, les séries temporelles, ainsi que les approches de machine learning et de deep learning. Ces éléments sont cruciaux pour appréhender les défis liés à la gestion et à la prédiction des congestions.

1.2 Les réseaux mobiles

1.2.1 Définition d'un réseau mobile

Un réseau mobile est un système de communication cellulaire qui permet l'utilisation simultanée d'un grand nombre de téléphones sans fil, qu'ils soient immobiles ou en mouvement, y compris lors de déplacements à grande vitesse sur de longues distances. Ces réseaux fonctionnent grâce à l'allocation de spectres hertziens, utilisés pour véhiculer l'information sous forme d'ondes radio. La technologie radio employée permet de partager ce spectre entre de multiples usagers, répartis dans différentes cellules radio selon leur position géographique [1].

1.2.2 Le concept cellulaire

Les premières générations de réseaux mobiles étaient caractérisées par de grandes cellules, chacune centrée autour d'une station de base qui servait de point d'émission. Initia-

lement, ces réseaux attribuaient une bande de fréquence fixe à chaque utilisateur dans la cellule, indépendamment de son besoin réel, limitant ainsi le service au nombre de bandes de fréquence disponibles. Pour augmenter le nombre potentiel d'abonnés, une première évolution consistait à attribuer un canal uniquement lorsqu'un utilisateur en avait besoin. Néanmoins, cette méthode exigeait des stations mobiles de forte puissance d'émission, ce qui se traduisait par des dispositifs mobiles lourds et volumineux. En outre, pour prévenir les interférences, il était nécessaire que deux cellules adjacentes n'utilisent pas les mêmes fréquences, entraînant une utilisation peu optimale du spectre fréquentiel.

Le concept des réseaux cellulaires a été introduit comme solution à ces problèmes. Il repose sur la division du territoire en zones plus petites, nommées cellules, au sein desquelles les fréquences radio sont partagées. Chaque cellule comprend une station de base connectée au Réseau Téléphonique Commuté (RTC), et lui sont associés plusieurs canaux de fréquences à bande étroite. Pour éviter les interférences, les mêmes fréquences ne sont pas utilisées dans des cellules adjacentes. Ce système permet une utilisation plus efficace du spectre fréquentiel grâce à la création de motifs (clusters) de cellules, où chaque fréquence est employée une fois par motif [2].

Traditionnellement, les cellules sont représentées par des hexagones en raison de leur capacité à fournir une couverture sans interruption du plan. Cette modélisation permet une simulation optimale de la couverture radioélectrique.

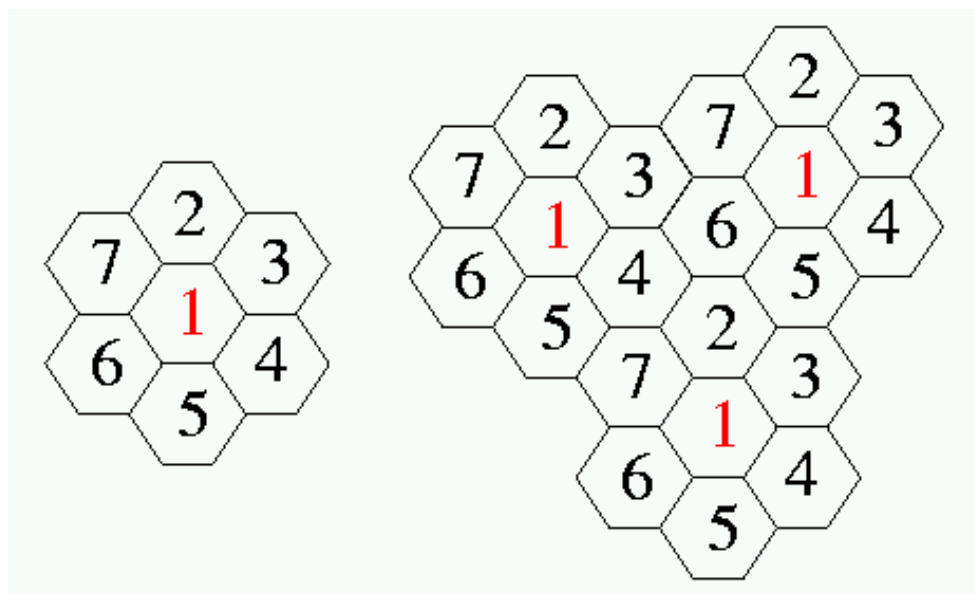


FIGURE 1.1 – Figure représentant un motif élémentaire (à gauche) et un ensemble de motifs dans un réseau (à droite).

Dans le contexte cellulaire, le système se distingue par le transfert intercellulaire, aussi appelé « handover », assurant la continuité des services lorsque les utilisateurs se déplacent

d'une cellule à une autre. Durant ces transitions, les fréquences sont automatiquement ajustées pour maintenir la communication et assurer la connectivité des utilisateurs.

1.2.3 Architecture d'un réseau mobile

L'architecture d'un réseau mobile comprend trois entités fonctionnelles :

- Le terminal mobile, appelé aussi équipement utilisateur (ou usager), abrégé en UE (User Equipment).
- Le réseau d'accès ou RAN (Radio Access Network).
- Le réseau cœur ou CN (Core Network).

Deux domaines distincts sont également identifiés :

- Le domaine de l'utilisateur (UE), comprenant les équipements spécifiques à l'utilisateur.
- Le domaine de l'infrastructure, constitué des équipements propres à l'opérateur.

La figure 1.2 illustre la structure d'un réseau mobile. L'UE fait partie du domaine de l'équipement utilisateur, et il est connecté au réseau d'accès via l'interface radio. La station de base agit en tant qu'élément d'interconnexion entre le réseau d'accès et l'interface radio. Le réseau d'accès et le réseau cœur font partie du domaine de l'infrastructure et sont reliés par une ou plusieurs interfaces terrestres [1].

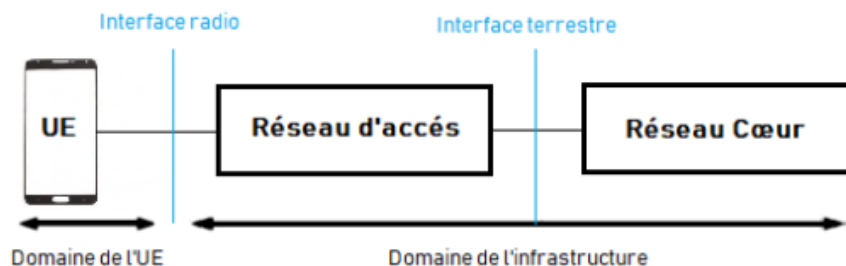


FIGURE 1.2 – Structure d'un réseau de téléphonie mobile.

1.2.4 Les différentes générations des réseaux mobiles

Au fil du temps, plusieurs générations se sont succédé, marquant des avancées technologiques significatives. Ces avancées ont considérablement amélioré les capacités de connectivité, offrant des débits toujours plus élevés. Dans les sections suivantes, nous survolerons

brèvement chaque génération de réseau mobile, en mettant particulièrement l'accent sur l'architecture de la 4G qui joue un rôle central dans notre étude.

1.2.4.1 La première génération (1G)

La première génération de téléphones portables a vu le jour au début des années 1980, offrant exclusivement un service de transmission vocale, caractérisé par une qualité médiocre et des coûts élevés. Cette première génération, connue sous le nom de 1G, présentait plusieurs inconvénients majeurs, tels que l'incompatibilité des normes entre différentes régions, ce qui entravait la communication à l'échelle internationale. De plus, la 1G était caractérisée par une transmission analogique vulnérable, exposant les communications à des écoutes [3].

1.2.4.2 La deuxième génération (2G)

La deuxième génération a été introduite dans les années 90 avec l'apparition du GSM. Fondée sur des transmissions numériques sécurisées par cryptage, elle a également standardisé le roaming international. Cette technologie permettait l'envoi de SMS et s'appuyait sur une connexion orientée circuit pour passer des appels, similaire à celle utilisée en téléphonie fixe. Cette méthode créait un canal dédié entre l'émetteur et le destinataire pour la durée de l'appel, garantissant l'ordre et l'intégrité de la communication, malgré l'inconvénient de monopoliser un canal même lors des silences [4].

1.2.4.3 La troisième génération (3G)

La troisième génération (3G) a été développée principalement pour supporter les applications vidéo sur les appareils mobiles et pour améliorer la qualité de service (QoS) des contenus multimédias. Elle visait à permettre aux utilisateurs de profiter de services tels que le visionnage de vidéos sur YouTube et les appels vidéo. En plus de fournir une augmentation significative de la vitesse de connexion, un défi majeur de cette transition était de passer d'un système de téléphonie traditionnel, basé sur des circuits, à un système de transmission de données en mode paquets [4].

1.2.4.4 La quatrième génération (4G)

La 4G ou Long Term Evolution (LTE) a marqué une évolution significative dans le domaine des télécommunications sans fil, étant principalement conçue pour répondre à la demande croissante de services de données à haute vitesse et de qualité supérieure pour les applications multimédia. Introduite dans les années 2010, la 4G utilise des technologies entièrement basées sur le protocole IP (Internet Protocol), permettant non seulement des

vitesses de transmission de données considérablement accrues par rapport à la 3G, mais aussi une meilleure efficacité du spectre et une connectivité améliorée.

La figure 1.3 ci-dessous représente l'architecture du réseau 4G [5] :

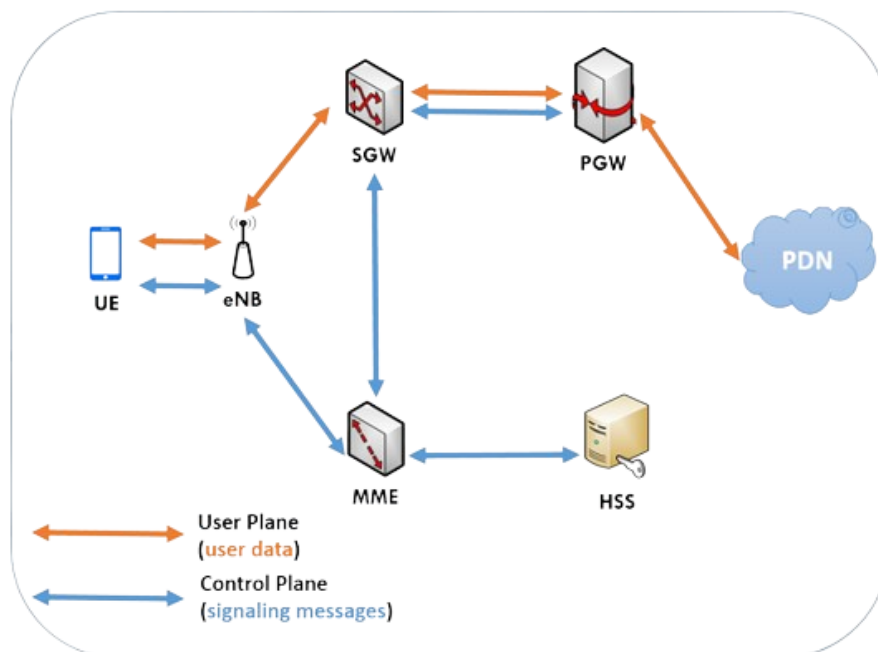


FIGURE 1.3 – Architecture du réseau 4G

Le réseau LTE comprend les équipements suivants [6] :

- **UE** (User Equipment) : le dispositif mobile tel que le smartphone ou la tablette utilisé par l'abonné pour se connecter au réseau LTE et accéder aux services de communication sans fil.
- **eNodeB** (evolved Node B) : est une station de base dans les réseaux LTE. Il gère les fonctions de liaison radio et de communication avec les équipements utilisateur (UE), assurant ainsi la connectivité sans fil pour les appareils mobiles.
- **MME** (Mobility Management Entity) : Le MME agit comme le cerveau du réseau LTE. Il est responsable de la gestion des procédures de signalisation liées à la mobilité des utilisateurs, à l'authentification, à la sélection du réseau et à la sécurité. Le MME gère les sessions, l'authentification des utilisateurs, le choix du S-GW pour un utilisateur donné, et assure la gestion des clés de sécurité.
- **HSS (Home Subscriber Server)** : Le HSS est une base de données centrale qui contient les informations d'abonnement des utilisateurs. Il supporte l'authentification et l'autorisation des utilisateurs, et stocke des informations telles que les profils

d'utilisateur, les détails de l'authentification, et les services auxquels l'utilisateur a accès. Le HSS est crucial pour la gestion de l'identité et des sessions des utilisateurs.

- **S-GW (Serving Gateway)** : Le S-GW agit comme un nœud de routage et de commutation pour les données utilisateurs. Il sert de passerelle entre le réseau radio et le réseau cœur. Le S-GW maintient également les données de session des utilisateurs en cas de mobilité entre les eNodeBs.
- **P-GW (PDN Gateway)** : Le P-GW fournit la connectivité avec le réseau externe (PDN - Packet Data Network), typiquement Internet ou des réseaux privés. Il effectue l'allocation d'adresses IP pour les terminaux, le filtrage des paquets pour chaque utilisateur, et l'application de politiques de facturation et QoS. Le P-GW est le point d'entrée pour le trafic entrant et sortant du réseau LTE.

Fonctionnement d'un réseau 4G

Dans un réseau 4G lorsqu'un utilisateur active son appareil, le processus de connexion débute par la recherche de signaux radio émis par les stations de base les plus proches, les eNodeB. Une fois le signal capté, l'appareil envoie une demande de connexion à l'eNodeB, qui la retransmet au MME. Le MME joue un rôle crucial dans la gestion de la mobilité et de la sécurité, il initie le processus d'authentification en interrogeant le HSS, qui contient toutes les informations d'abonnement et de profil de l'utilisateur. Une fois l'authentification réussie, le MME communique avec le SGW pour établir un tunnel pour le trafic de données. Le SGW sert de point de passage pour les données entre l'appareil et le réseau, agissant également comme un point d'ancrage de mobilité lorsque l'utilisateur se déplace entre différentes zones couvertes par les eNodeB. Pour accéder à des services Internet, les données passent ensuite par le PGW, qui attribue une adresse IP publique à l'appareil et assure la liaison entre le réseau de l'opérateur mobile et l'Internet. Ce chemin de données, de l'appareil à l'eNodeB, puis à travers le MME, le SGW, et enfin le PGW, permet à l'utilisateur de naviguer sur Internet, d'envoyer des emails, de streamer du contenu vidéo, ou d'utiliser des applications en ligne, avec les données voyageant à travers le réseau 4G de manière sécurisée et efficace.

1.2.5 La congestion

La congestion dans les réseaux se réfère à une situation où la demande de ressources réseau dépasse la capacité disponible, entraînant une dégradation des performances. Cela se produit lorsque trop d'utilisateurs ou de dispositifs tentent de transmettre des données en même temps, ce qui peut mener à une saturation des liens de communication. Bien que la congestion du réseau soit généralement temporaire, elle peut entraîner des problèmes de réseau gênants affectant la performance, tels que des niveaux élevés de gigue, des pertes de paquets, une latence accrue, ainsi qu'une diminution du débit [7].

1.2.5.1 Facteurs contribuant à la congestion des réseaux

Il est essentiel de comprendre les causes de la congestion réseau afin de pouvoir prendre des mesures pour y remédier. Voici quelques-unes des raisons les plus courantes de la congestion réseau.

- **Bande passante limitée** : La bande passante fait référence à la quantité maximale de données pouvant être transmises à travers le réseau pendant une période donnée. Si cette bande passante est insuffisante pour accommoder tout le trafic simultanément, des problèmes surviennent, entraînant une congestion du réseau.
- **Mauvaise planification du réseau** : Un déploiement inadéquat des infrastructures réseau peut entraîner des goulets d'étranglement et des zones mal desservies, provoquant ainsi des congestions localisées.
- **Interférences radio** : Les signaux peuvent être perturbés par des interférences radio causées par divers facteurs, comme des dispositifs électroniques ou des conditions météorologiques, ce qui peut provoquer des ralentissements et des coupures de connexion.
- **Mauvaise gestion du trafic** : Une mauvaise gestion du trafic, notamment des politiques de qualité de service inappropriées, peut conduire à une utilisation inefficace des ressources réseau et aggraver la congestion. Cela se produit lorsque les priorités de traitement du trafic ne sont pas correctement définies, ce qui peut entraîner une allocation inéquitable de la bande passante et une utilisation inefficace des capacités du réseau.
- **Défauts d'équipement ou de logiciel** : Des pannes matérielles ou des bugs logiciels peuvent provoquer des dysfonctionnements du réseau, entraînant des ralentissements et des interruptions de service.

1.3 Indicateur Clé de Performance (KPI)

En des termes simples, un KPI est une mesure quantitative utilisée de manière stratégique pour évaluer les performances et les succès d'un projet, d'une entreprise ou d'un effort spécifique. Ces indicateurs sont soigneusement sélectionnés pour correspondre aux objectifs globaux et offrir un aperçu de l'efficacité et de l'efficience des différents aspects opérationnels. En surveillant activement les KPI, les organisations peuvent recueillir des informations approfondies sur les tendances de performance, identifier les secteurs nécessitant des améliorations, et prendre des décisions éclairées pour avancer [8].

1.3.1 Les KPI dans les réseaux mobiles

Dans le secteur des télécommunications mobiles, les KPI jouent un rôle essentiel pour garantir une expérience utilisateur de qualité. Ils permettent de suivre la performance du réseau en temps réel, d'optimiser les ressources, et d'identifier rapidement les défaillances potentielles. Trois aspects fondamentaux à surveiller sont la mobilité, l'intégrité et la disponibilité du réseau.

1.3.1.1 Mobilité

La mobilité est un aspect clé des réseaux mobiles, car elle garantit une connectivité fluide lorsque l'utilisateur se déplace entre différentes cellules ou zones de couverture. Les principaux KPI associés à la mobilité sont :

Taux de transfert intercellulaire réussi (Handover Success Rate) : Ce KPI mesure la capacité du réseau à transférer un appel ou une session de données d'une cellule à une autre sans interruption. Un taux de succès élevé garantit une expérience continue lors de la mobilité.

Taux de chute des appels (Call Drop Rate - CDR) : Ce KPI évalue le pourcentage d'appels interrompus involontairement, souvent causés par une transition ratée entre les cellules. Une gestion efficace du handover permet de minimiser ce taux.

Temps d'établissement de connexion (Connection Setup Time) : Il s'agit du temps nécessaire pour établir une connexion lors de la demande d'un utilisateur. Des connexions rapides et stables sont essentielles pour les utilisateurs en déplacement.

1.3.1.2 Intégrité

L'intégrité concerne la qualité et la fiabilité des transmissions de données et de voix sur le réseau. Les KPI d'intégrité garantissent que les utilisateurs bénéficient d'une expérience de qualité, sans interruption ni dégradation perceptible.

Débit de données (Throughput) : Il mesure la quantité de données pouvant être transmises par unité de temps. Un débit élevé est essentiel pour offrir des services tels que le streaming et les appels vidéo.

Latence : Ce KPI mesure le temps écoulé entre l'envoi d'une requête et la réception de la réponse. Une faible latence est critique pour les applications en temps réel, comme les jeux en ligne et la vidéoconférence.

Taux de perte de paquets (Packet Loss) : Cet indicateur mesure la proportion de paquets de données perdus pendant la transmission. Des pertes minimales sont nécessaires pour maintenir une transmission fluide, surtout pour les appels VoIP.

Jitter (Gigue) : Ce KPI évalue les variations dans la latence. Une faible gigue est nécessaire pour garantir la stabilité des flux audio et vidéo en temps réel.

1.3.1.3 Disponibilité

La disponibilité évalue la capacité du réseau à rester opérationnel et accessible aux utilisateurs. Elle est essentielle pour garantir une expérience utilisateur fiable et réduire les interruptions de service.

Taux de disponibilité du réseau : Ce KPI mesure le pourcentage de temps pendant lequel le réseau est accessible et fonctionnel. Un taux de disponibilité proche de 100 % est nécessaire pour assurer une expérience optimale.

Temps moyen de rétablissement (Mean Time to Repair - MTTR) : Cet indicateur évalue le temps nécessaire pour rétablir un service après une panne. Un MTTR faible réduit l'impact des interruptions sur les utilisateurs.

Taux d'utilisation du réseau (Network Utilization Rate) : Il mesure le pourcentage de la capacité totale du réseau utilisé à un moment donné. Un taux d'utilisation optimisé permet de prévenir les surcharges et les congestions, garantissant ainsi la disponibilité des services.

1.3.2 Importance de la surveillance continue des KPI

Une surveillance continue de ces KPI permet aux opérateurs de :

- Améliorer l'expérience utilisateur en garantissant une mobilité fluide, une intégrité élevée et une disponibilité constante.
- Identifier et résoudre les problèmes proactivement. Une dégradation de l'intégrité ou une chute de la disponibilité peut être corrigée rapidement grâce à la surveillance en temps réel.
- Optimiser les ressources réseau en analysant les tendances des KPI. Les opérateurs peuvent ajuster la capacité et l'allocation des ressources pour répondre à la demande croissante.
- Planifier l'expansion du réseau. Les tendances de mobilité et d'utilisation permettent d'identifier les zones nécessitant des améliorations, comme l'ajout de nouvelles antennes ou l'extension de la couverture.

1.4 Séries temporelles

Une série temporelle (ou encore une série chronologique) est un ensemble d'observations faites séquentiellement dans le temps. La principale caractéristique des séries chronologiques est la dépendance temporelle, c'est-à-dire la relation entre les observations dans le temps. Cette dépendance est un aspect important dans l'analyse des séries temporelles, et son interprétation est cruciale [9].

Mathématiquement, une série temporelle peut être représentée comme suit :

$$X(t) = \{X_1, X_2, \dots, X_n\} \quad (1.1)$$

où $X(t)$ désigne la séquence des valeurs de la série temporelle à différents moments dans le temps t , et X_1, X_2, \dots, X_n représentent les valeurs numériques correspondantes de la série à chaque instant t . La notation $\{\}$ indique que les valeurs sont ordonnées selon le temps. Ainsi, la séquence $X(t)$ peut être considérée comme une fonction qui associe à chaque instant t une valeur spécifique de la série temporelle.

1.4.1 Composantes d'une série temporelle

Une série temporelle est le résultat de différentes composantes qui servent à comprendre les différentes sources de variation présentes dans les données temporelles. Dans ce qui suit nous présenterons les principales composantes fondamentales.

1.4.1.1 Tendances

Lorsque les valeurs d'une série temporelle diminuent ou augmentent dans un intervalle à long terme, on peut dire que la série temporelle présente une tendance. La tendance d'une série temporelle peut être linéaire ou non linéaire, positive ou négative, et peut prendre différentes directions au fil du temps [10]. La figure 1.4 montre un graphique de l'évolution du trafic aérien mondial qui est une série temporelle avec une tendance positive.

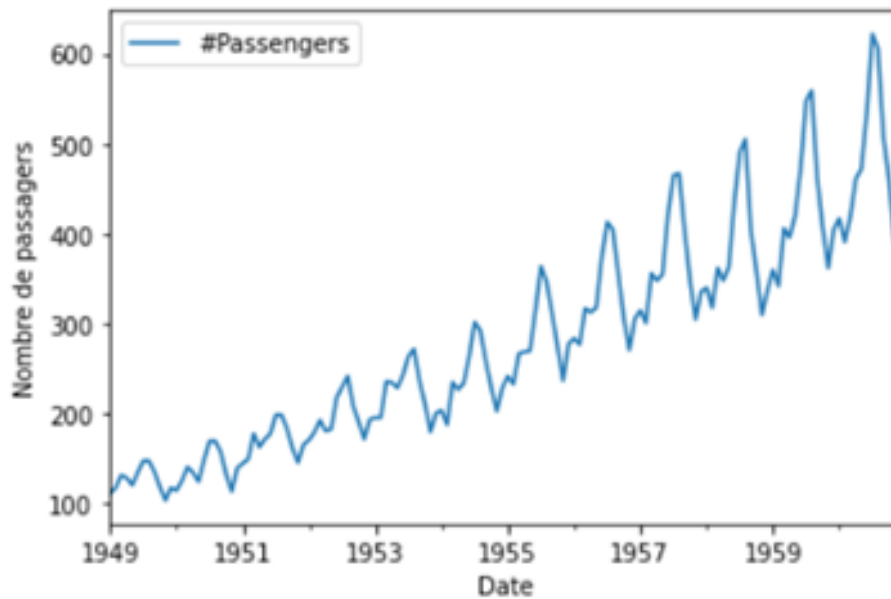


FIGURE 1.4 – Exemple de série temporelle montrant l'évolution du trafic aérien mondial

1.4.1.2 Saisonnalité

La saisonnalité se manifeste par la récurrence périodique d'un cycle sur toute la durée de la série temporelle. Ainsi, une série temporelle est considérée comme saisonnière lorsqu'elle est influencée par des facteurs saisonniers, qu'il s'agisse de l'année ou d'une autre période spécifique [11]. Un exemple de ce paramètre est montré dans la figure 1.5, qui représente le marché immobilier, à savoir qu'il n'est pas actif au début de l'année, mais il augmente au milieu de l'année et revient à diminuer à sa fin.



FIGURE 1.5 – Exemple de la saisonnalité du marché de l'immobilier

1.4.1.3 Cycle

Dans les séries chronologiques, le terme « cycle » fait référence à un modèle d'oscillations répétées dans le temps. Les changements saisonniers, les cycles économiques et les catastrophes naturelles peuvent générer ces schémas. Par exemple, les données de température et les valeurs boursières, qui fluctuent tout au long de l'année, sont des séries temporelles avec des cycles [12].

Les modèles saisonniers et cycliques dans les données de séries temporelles impliquent tous les deux des modèles périodiques récurrents, mais ils diffèrent de manière significative :

- Les modèles saisonniers se manifestent à des périodes particulières de l'année et sont impactés par des éléments comme les conditions météorologiques, les vacances et les calendriers scolaires. Par exemple, les ventes au détail peuvent connaître un pic pendant la période de Noël, et ces modèles sont généralement prévisibles et se répètent chaque année [12].
- Les modèles cycliques, quant à eux, ont une durée plus longue et n'adhèrent pas à des cycles annuels fixes. Ils peuvent être déterminés par divers facteurs, notamment des cycles économiques et commerciaux, ainsi que des événements naturels. Par exemple, le taux de chômage peut présenter des fluctuations cycliques sur plusieurs années [13].

Il est crucial, lors de l'analyse des séries chronologiques, de différencier les modèles saisonniers des modèles cycliques. Cette distinction est primordiale car elle a des répercussions spécifiques sur les processus de prédiction et de modélisation.

1.4.1.4 Résidu (aléa, bruit)

Correspondent aux fluctuations qui apparaissent de manière irrégulière et généralement sur de courtes périodes. En pratique, toutes les composantes des séries temporelles qui ne peuvent être attribuées à l'influence des fluctuations cycliques ou des variations ou tendances saisonnières sont classées comme irrégulières.

1.4.2 Types des séries temporelles

Les séries temporelles sont classifiées selon le nombre de variables et les dépendances entre elles en deux classes :

- **Séries temporelles univariées** : Une série temporelle univariée est une séquence de mesures d'une seule variable collectée au fil du temps [14].
- **Séries temporelles multivariées** : Une série temporelle multivariée comporte plusieurs variables dépendantes du temps. Chaque variable dépend non seulement de ses valeurs précédentes, mais aussi d'autres variables [15].

1.4.3 Stationnarité des séries temporelles

On dit qu'une série chronologique est stationnaire si ses caractéristiques statistiques, telles que sa moyenne et sa variance, restent constantes dans le temps. En d'autres termes, une série chronologique stationnaire est une série dans laquelle les modèles et les fluctuations des données ne changent pas au fil du temps. Une série chronologique faiblement stationnaire est parfois appelée série chronologique stationnaire. La stationnarité stricte est un type plus important de stationnarité qui exige que la distribution conjointe de toute collection d'observations reste constante dans le temps [10].

1.4.4 Modèles classiques des séries temporelles

Les modèles classiques des séries temporelles, tels que AR, MA, ARMA et ARIMA, sont largement utilisés pour analyser et prédire des phénomènes évoluant dans le temps. Chacun de ces modèles offre une approche spécifique pour capturer les relations temporelles dans les données, en tenant compte de la dépendance des valeurs passées, des erreurs résiduelles, ou de la stationnarité.

1.4.4.1 Modèle autorégressif (AR)

Un "modèle autorégressif (AR)" est utilisé pour prédire le comportement futur d'une variable en utilisant une combinaison linéaire des valeurs historiques de la variable. Le terme "autorégression" indique que la variable est régressée contre elle-même. On peut le voir comme une fonction des valeurs antérieures. Le modèle $AR(p)$ représente le modèle autorégressif qui est influencé par " p " de ses valeurs précédentes [16].

$$y_t = f(y_{t-1}, y_{t-2}, y_{t-3}, \dots, \epsilon_t) \quad (1.2)$$

Dans le paradigme $AR(p)$, p est le paramètre dont les valeurs peuvent varier comme suit :

$$y_t = \sum_{i=1}^p a_i \cdot y_{t-i} + c + \epsilon_t \quad (1.3)$$

Où ϵ_t est le terme d'erreur à l'instant t , a_i est le coefficient associé à la valeur précédente y_{t-i} , et c est une constante.

Dans le modèle $AR(p)$, p est le paramètre dont les valeurs peuvent changer comme suit [16] [17] :

- Lorsque $p = 0$, le modèle $AR(0)$ devient $y_t = c$.
- Lorsque $p = 1$, le modèle $AR(1)$ devient $y_t = c + a_1 y_{t-1}$.

Généralement, les modèles autorégressifs sont contraints à des données stationnaires. Cependant, lorsque $p \geq 3$, les contraintes deviennent significativement plus complexes, ce qui pose des défis. Il est donc nécessaire de déterminer la valeur optimale de p pour les prévisions [17].

1.4.4.2 Modèle de la moyenne mobile (MA)

Au lieu d'utiliser les valeurs antérieures d'une variable pour les prévisions, comme le fait le modèle AR(p), le "modèle de moyenne mobile (MA)" utilise les termes d'erreur antérieurs pour les prévisions [18]. Le modèle MA peut être considéré comme une fonction des termes d'erreur comme suit :

$$y_t = f(\epsilon_t, \epsilon_{t-1}, \epsilon_{t-2}, \dots, \epsilon_{t-q}) \quad (1.4)$$

Lorsqu'une série est régressée contre ses valeurs passées, on obtient un terme d'erreur, comme suit [18] :

$$y_t = a_1 \cdot \epsilon_{t-1} + \mu + \epsilon_t \quad (1.5)$$

Où y_t est la valeur de la série temporelle à l'instant t que l'on souhaite prédire, μ est une constante, a_1 est le paramètre du modèle qui détermine l'influence des termes d'erreur antérieurs sur la valeur actuelle de la série, et ϵ_t représente l'erreur [18].

Lorsque nous régressons la série avec différentes valeurs, nous obtenons d'autres termes d'erreur tels que ϵ_2 , ϵ_3 , et ainsi de suite. Dans ce modèle, par conséquent, nous utilisons les termes d'erreur plutôt que les valeurs historiques [17]. Par conséquent, le "modèle de moyenne mobile" peut être présenté comme suit [18] :

$$y_t = \sum_{i=1}^p a_i \cdot \epsilon_{t-i} + \mu + \epsilon_t \quad (1.6)$$

Où y_t est la valeur de la série temporelle à l'instant t que l'on souhaite prédire, μ représente la "moyenne de la série", a_i sont les paramètres du modèle qui déterminent l'influence des termes d'erreur antérieurs sur la valeur actuelle de la série, et ϵ_t désigne l'erreur. On suppose que les termes d'erreur ϵ_t sont des processus de bruit blanc, c'est-à-dire ceux ayant une "moyenne nulle" et une "variance constante" σ^2 [16].

Le modèle de moyenne mobile (MA) est plus difficile à ajuster à une série temporelle que le modèle autorégressif (AR), car les termes d'erreur aléatoires ne sont pas prévisibles [16].

1.4.4.3 Modèle de moyenne mobile autorégressive (ARMA)

Les modèles "Autoregressive" (AR) ainsi que les modèles "Moving Average" (MA) sont utilisés conjointement pour obtenir une nouvelle unité de modèles de séries temporelles

appelés "modèles ARMA" (c'est-à-dire AR + MA = modèle ARMA) [16]. La notation générale pour le "modèle ARMA" est la suivante :

$$y_t = \sum_{i=1}^p a_i \cdot y_{t-i} + c + \epsilon_t + \sum_{i=1}^q a_i \cdot \epsilon_{t-i} \quad (1.7)$$

Où y_t est la valeur de la série temporelle à l'instant t que l'on souhaite prédire, ϵ_t est le terme d'erreur à l'instant t , a_i est le coefficient de la valeur précédente y_{t-i} , et c est une constante. p est l'ordre de la composante autorégressive (AR) du modèle, et q est l'ordre de la composante moyenne mobile (MA) du modèle.

1.4.4.4 Moyenne Mobile AutoRégressive Intégrée (Modèle ARIMA)

Le modèle ARMA peut être utilisé avec succès sur des données de séries temporelles stationnaires. Cependant, la grande majorité des données de séries temporelles que nous observons dans le monde réel ont un comportement non stationnaire [18]. Ce problème est résolu par le modèle ARIMA, qui transforme les données non stationnaires en données stationnaires grâce à la méthode de différenciation. Cela consiste à remplacer les valeurs des données de séries temporelles par les différences entre ces valeurs et leurs valeurs précédentes aux pas de temps antérieurs [17].

L'équation ARIMA est une équation linéaire dans laquelle les prédicteurs sont composés de retards de la variable dépendante et/ou de retards des facteurs d'erreur. L'équation peut être écrite comme suit :

La valeur prédite de Y est une constante et/ou une somme pondérée d'une ou plusieurs valeurs récentes de Y et/ou une somme pondérée d'une ou plusieurs valeurs récentes des erreurs.

La formule commune d'un modèle "ARIMA" pour y_t est donnée comme suit [17] :

$$y_t = \sum_{i=1}^p a_i \cdot y_{t-i} + c + \epsilon_t + \sum_{i=1}^q a_i \cdot \epsilon_{t-i} \quad (2.7) \quad (1.8)$$

Où y_t est une "série temporelle différenciée" qui peut avoir été différenciée une fois ou plusieurs fois.

1.5 Détection et prédiction de congestions

1.5.1 La détection de congestions

La détection de congestions implique l'identification de leurs symptômes. Dans un contexte réseau, ces symptômes incluent les pertes de paquets d'information, un délai de transmission plus lent et donc une vitesse de fonctionnement du réseau plus faible. Pour

détecter ces congestions, on surveille des indicateurs clés de performance (KPIs) qui quantifient ces symptômes spécifiques [19]. Plusieurs techniques sont employées pour analyser ces KPIs, notamment des méthodes statistiques pour l'analyse de tendances, des techniques d'apprentissage automatique pour la reconnaissance de motifs, et des systèmes d'alerte basés sur des seuils prédéfinis.

1.5.2 La prédiction de congestions

Les congestions, détectables par l'analyse de l'évolution du trafic réseau au fil du temps, peuvent souvent être anticipées en prédisant ce dernier. Comme le suggère son nom, la prédiction de trafic consiste à estimer la quantité de données qui circuleront dans le réseau à un moment futur $t+1$, en se basant sur l'analyse de l'historique du trafic entre les instants 0 et t . Les techniques les plus couramment employées pour aborder ce type de problématiques sont les méthodes de prédiction des séries temporelles, fréquemment utilisées dans le secteur économique, où elles reposent largement sur des approches statistiques [20], parmi les méthodes les plus reconnues figurent les modèles autorégressifs (AR) ainsi que leurs dérivés tels qu'ARMA, ARIMA, et SARIMA. Ensuite, des modèles d'apprentissage automatique et d'apprentissage profond ont émergé, offrant une précision prédictive accrue grâce à des architectures de réseaux de neurones artificiels telles que le Perceptron Multicouches (MLP) ou le réseau de neurones récurrents (RNN).

1.6 Machine Learning

1.6.1 Définition

Le Machine Learning ou apprentissage automatique est un domaine de l'intelligence artificielle (IA) qui repose sur des approches mathématiques et statistiques pour permettre aux machines d'apprendre à partir de données manière autonome. Comme l'a souligné Arthur Samuel, l'un des pionniers de l'apprentissage automatique, dans sa définition de 1959, c'est "le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés" [21].

1.6.2 Types d'apprentissage automatique

Il existe aujourd'hui une grande variété d'algorithmes d'apprentissage automatique, souvent regroupés en trois grandes catégories qui sont détaillées ci-dessous :

1.6.2.1 Apprentissage supervisé

Dans l'apprentissage supervisé, un algorithme est entraîné sur des données étiquetées, ce qui implique que les données d'entrée sont associées à une sortie correspondante ou à une variable cible. Cette approche permet au système de prédire avec précision la valeur de la variable cible. Les données fournies sont utilisées pour apprendre à l'algorithme à effectuer des prédictions sur la variable à produire [22]. L'apprentissage supervisé peut être divisé en deux types de problèmes :

- **Classification** : assigne des étiquettes prédéfinies à des données.
- **Régression** : prédit des valeurs numériques basées sur des données d'entrée.

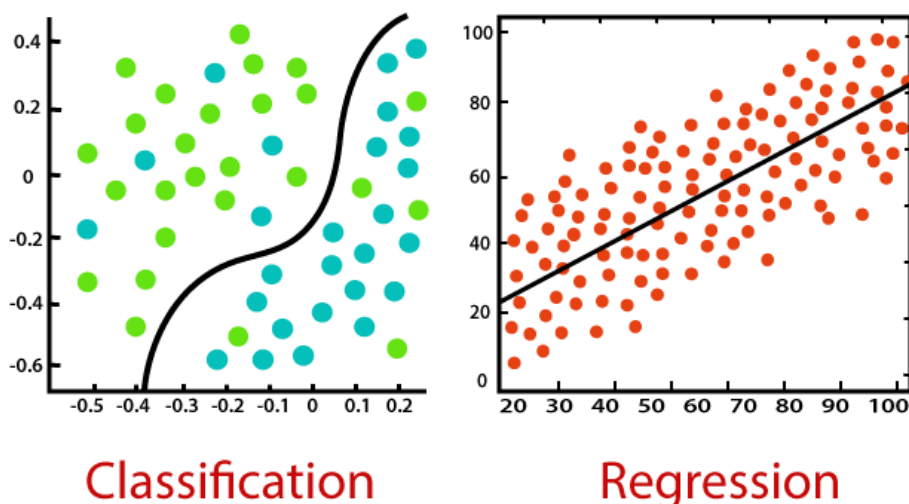


FIGURE 1.6 – Les deux Types d'apprentissage supervisé

La régression linéaire, la régression logistique, les arbres de décision, les forêts aléatoires et les machines à vecteurs de support sont autant d'exemples de techniques d'apprentissage supervisé.

1.6.2.2 Apprentissage non supervisé

Dans le cadre du machine learning non supervisé, l'algorithme détermine lui-même la structure de l'entrée (aucune étiquette n'est appliquée à l'algorithme). Cette approche peut être un but en soi (qui permet de découvrir des structures enfouies dans les données) ou un moyen d'atteindre un certain but. Cette approche est également appelée « apprentissage des caractéristiques » (feature learning) [23].

Il existe deux types de machine learning non supervisé :

- **Clustering** : l'objectif consiste à trouver des regroupements dans les données.

- **Association** : L'objectif est d'identifier des règles qui révèlent des relations significatives entre différents éléments au sein de vastes ensembles de données.

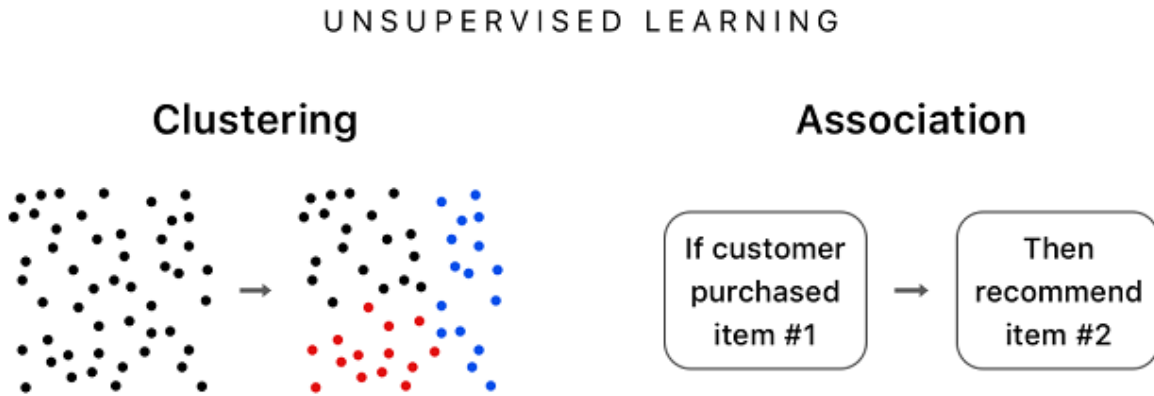


FIGURE 1.7 – Le clustering et les règles d'association

Les principaux algorithmes du machine learning non supervisé sont les suivants : K-Means, clustering/regroupement hiérarchique et réduction de la dimensionnalité .

1.6.2.3 Apprentissage par renforcement

L'apprentissage par renforcement ou Reinforcement Learning (RL) est une technique de machine learning qui diffère des autres méthodes citées précédemment. Dans le cadre du RL, un agent interagit avec son environnement en entreprenant des actions et en recevant un retour d'informations sous forme de récompenses ou de pénalités. L'objectif est de développer une stratégie qui maximisera la somme des récompenses futures, souvent appelée récompense cumulée [24].

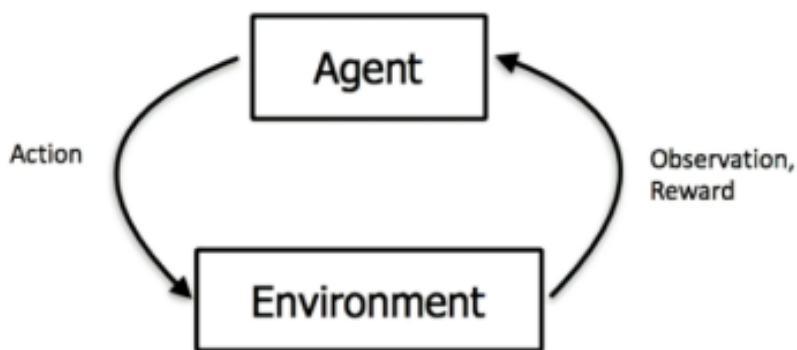


FIGURE 1.8 – Processus d'apprentissage par renforcement

Les principaux algorithmes d'apprentissage par renforcement sont les suivants : Q-learning, Deep Q Network (DQN) et SARSA (State-Action-Reward-State-Action).

1.7 Deep Learning

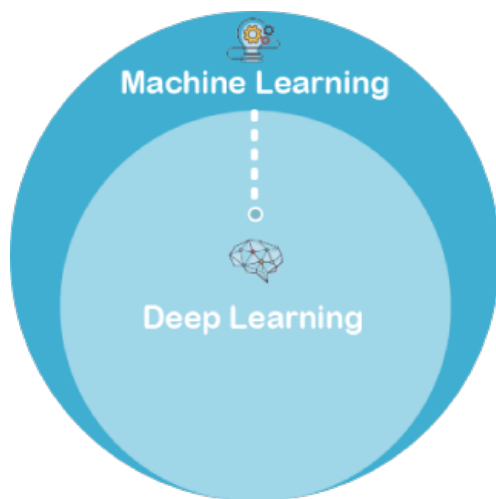


FIGURE 1.9 – Relation entre le Machine Learning et le Deep Learning

L'apprentissage profond, qui est une sous-catégorie de l'apprentissage automatique, se caractérise par sa capacité à extraire automatiquement des caractéristiques pertinentes à partir des données, sans nécessiter une expertise humaine approfondie en ingénierie des caractéristiques. Les modèles d'apprentissage profond sont capables d'apprendre de manière autonome des représentations hiérarchiques des données, leur permettant ainsi de

résoudre des problèmes complexes. Ce processus constitue l'un des principaux avantages de l'apprentissage profond. De plus, ces modèles peuvent traiter d'immenses volumes de données non structurées, telles que des images, des enregistrements sonores et des textes [22].

1.8 Conclusion

Dans ce chapitre, nous avons examiné les principes fondamentaux des réseaux mobiles, en mettant un accent particulier sur la congestion qui peut les affecter. Nous avons également présenté le concept des Indicateurs Clés de Performance (KPI), qui sont essentiels pour mesurer l'efficacité et les performances dans divers domaines, y compris les réseaux mobiles. Nous avons ensuite abordé les séries temporelles, cruciales pour la prédiction des congestions dans les réseaux. Enfin, nous avons défini les concepts généraux du machine learning et du deep learning qui jouent un rôle crucial dans la résolution des défis liés à la prédiction des séries temporelles. Le chapitre suivant sera consacré à une revue des travaux de littérature sur la prédiction de congestions.

Chapitre 2

Etat de l'art

2.1 Introduction

Dans l'univers des réseaux de télécommunications, la congestion représente un défi majeur qui impacte directement la qualité du service et l'expérience utilisateur. À l'ère du numérique, où la demande de données est en croissance exponentielle, alimentée par l'essor des applications mobiles, des services de streaming et de l'internet des objets, les opérateurs de réseaux sont confrontés à la nécessité impérieuse d'optimiser la gestion du trafic pour anticiper et résoudre les problèmes de congestion. L'intelligence artificielle (IA) et l'apprentissage automatique émergent comme des alliés puissants dans cette quête.

Dans ce chapitre, nous présentons les principaux travaux connexes dans le domaine de la prédiction du trafic réseau et des congestions, avant d'élaborer un tableau comparatif pour analyser les différents travaux abordés.

2.2 Travaux connexes

Torres et al. [23], Dans cette étude, les auteurs se concentrent sur la prédiction du trafic cellulaire à court terme pour anticiper les congestions futures des cellules dans les réseaux LTE. L'approche proposée s'appuie sur un modèle composé d'une couche LSTM (Long Short-Term Memory) suivie d'une couche de mise en commun moyenne et d'une couche de régression logistique. Cette architecture permet de générer une séquence de représentations à partir des données d'entrée, laquelle est ensuite moyennée et transmise à la couche de régression pour prédire la classe associée à la séquence.

L'étude a utilisé des données collectées sur une période de quatre semaines, avec 75% des données destinées à l'entraînement du modèle et le reste réservé aux tests. Les performances du modèle de Deep Learning ont été comparées à celles des modèles ARIMAX et de persistance naïve. Les résultats montrent que le modèle basé sur l'apprentissage profond

a surpassé les approches statistiques traditionnelles en termes de précision prédictive, avec un MSE (Mean Squared Error) significativement plus faible.

Dans une phase ultérieure, les chercheurs ont identifié les périodes de congestion dans le trafic prédit, définissant la congestion comme une chute de plus de 50% du débit moyen de la cellule durant les heures de pointe. Ce travail montre que l'approche basée sur les réseaux de neurones profonds est prometteuse pour la gestion proactive des ressources réseau, permettant d'anticiper les congestions et d'agir en conséquence pour améliorer la qualité de service.

Markus et al. [24], ont utilisé un réseau de neurones à propagation avant à plusieurs couches (Multi-Layer Feed forward Neural Network) pour prédire la congestion du trafic téléphonique. Les données utilisées sont des données historiques du trafic issues d'enregistrements effectués par le commutateur réseau sur la route Kaduna-Abuja à des intervalles de 15 min. Le modèle comprend trois étapes : prétraitement, prédiction et post-traitement des données. Les données, collectées aléatoirement sur une période de sept mois, sont d'abord prétraitées avant d'être introduites dans un réseau neuronal à propagation avant pour la prédiction. Les valeurs prédites sont ensuite post-traitées pour comparaison avec les valeurs réelles. L'étude a démontré la capacité du modèle à prédire la congestion du trafic téléphonique avec un degré de précision élevé, comme le montre un coefficient de corrélation de 87% en analyse de régression. Les auteurs ont également comparé cette approche aux modèles de séries temporelles. Ces derniers fonctionnent bien pour des prévisions à court terme mais sont moins performants sur le long terme. En conclusion, cette étude montre que les réseaux de neurones feedforward à plusieurs couches, notamment le Multi-Layer Perceptron (MLP), constituent un outil performant pour prédire et gérer la congestion du trafic téléphonique.

Chmieliauskas et Gursnys [25], abordent la prédiction du trafic cellulaire LTE en utilisant l'algorithme FbProphet, développé par Facebook. Les chercheurs démontrent que FbProphet excelle particulièrement dans les cas où les données présentent des tendances non linéaires et des motifs saisonniers significatifs, ce qui est typique des données de trafic LTE. Les données utilisées pour entraîner le modèle et tester sa précision dans la prédiction couvrent 180 jours, avec une granularité horaire des observations, en se concentrant uniquement sur le trafic descendant en raison de sa prépondérance. Les résultats montrent une corrélation entre les prédictions de FbProphet et les tendances réelles du trafic avec un score de variance expliquée et un coefficient de détermination « R^2 » de 0.9. Cependant, à mesure que le temps avance, l'erreur augmente progressivement. En plus de la prédiction du trafic, l'étude utilise l'efficacité spectrale des cellules pour évaluer la capacité à supporter le trafic prédit sans entraîner de congestion. Les auteurs concluent que l'algorithme FbProphet est un outil prometteur pour la prévision du trafic cellulaire, offrant aux opé-

rateurs de télécommunications des insights précieux pour la gestion de la capacité réseau et la prévention de la congestion.

Do et al. [26], Ont eu recours à des modèles de réseaux de neurones profonds, en particulier les LSTM et les Gated Recurrent Units (GRU), pour prédire le volume du trafic de données avec une heure d'avance. Ces modèles ont été comparés à d'autres méthodes, telles que le système d'inférence neuro-flou adaptatif (ANFIS), les réseaux de neurones artificiels (ANN) et la méthode de traitement de données par groupe (GMDH). L'étude, utilisant un ensemble de données fourni par Viettel Telecom couvrant une période d'un an avec des mesures horaires, a démontré que les modèles LSTM et GRU surpassent les autres méthodes sur plusieurs indicateurs de performance, notamment l'erreur quadratique moyenne (RMSE), l'erreur absolue moyenne (MAE), l'erreur absolue moyenne en pourcentage (MAPE), le coefficient de corrélation et la statistique U de Theil.

Les résultats obtenus montrent que le modèle LSTM a atteint une RMSE de 13971.4670, une MAPE de 0.3745, une MAE de 10337.0889, un coefficient de corrélation de 0.8866, et une statistique U de Theil de 0.1350. De même, le modèle GRU a obtenu des performances comparables avec un RMSE de 13830.3706, un MAPE de 0.3839, un MAE de 10309.7391, un coefficient de corrélation de 0.8863, et une statistique U de Theil de 0.1323. Ces résultats surpassent ceux des autres modèles testés, comme le GMDH qui a montré un RMSE de 1920553.0813, un MAPE de 1025.9252, un MAE de 6404251.2836, un coefficient de corrélation RRR de 0.0271, et une statistique U de Theil de 0.9998.

Ces performances supérieures des modèles LSTM et GRU soulignent leur efficacité dans la prédiction précise du volume du trafic de données, une capacité cruciale pour la gestion optimale des ressources réseau. Ils offrent une approche robuste et fiable pour l'analyse et la prédiction du trafic dans les réseaux de télécommunications, facilitant ainsi la prise de décisions stratégiques pour l'allocation des ressources et la planification de la capacité réseau. La supériorité des LSTM et GRU dans cette étude met en évidence leur potentiel en tant qu'outils précieux pour les opérateurs de télécommunications dans la gestion de la capacité du réseau et la minimisation des risques de congestion.

Jaffry [27], explore l'utilisation de réseaux de neurones récurrents, en particulier les LSTM, pour prédire le trafic cellulaire futur. Cette étude compare la performance des modèles LSTM avec les modèles traditionnels tels que le modèle Autorégressif Intégré à Moyenne Mobile (ARIMA) et le réseau de neurones à propagation avant (FFNN). Les données utilisées proviennent des enregistrements d'appels (CDR) de Telecom Italia, collectées sur une période de 62 jours à Milan. Les résultats montrent que les modèles LSTM surpassent les modèles FFNN et ARIMA en termes de précision prédictive, même avec un ensemble de données d'entraînement réduit. En particulier, les LSTM démontrent une capacité d'apprentissage plus rapide par rapport aux FFNN, atteignant une convergence des

erreurs de validation après seulement 2 époques d'entraînement, contre 10 pour les FFNN. De plus, alors que les modèles FFNN ont montré une précision réduite avec un petit jeu de données d'entraînement, les modèles LSTM ont maintenu une haute précision, indiquant leur robustesse et leur efficacité dans des environnements de trafic cellulaire dynamiques. L'auteur conclut que les modèles LSTM sont des outils prometteurs pour la prévision du trafic cellulaire, offrant une solution rapide et précise pour aider les opérateurs de réseaux à gérer de manière proactive les ressources et à prévenir la surcharge du réseau. Le travail futur de l'auteur se concentrera sur l'intégration des LSTM dans les méthodes d'allocation de ressources pour les réseaux 6G.

Wa Nkongolo [28], Dans cette étude l'objectif principal était de prédire la croissance de l'utilisation des données par les abonnés. Les données utilisées dans cette étude comprennent 730 points de données provenant de la table des statistiques horaires du trafic dans le système Insights Data Storage (IDS).

Pour ce faire, le modèle ARIMA a été utilisé, démontrant une capacité notable à prédire avec précision les tendances de consommation des données. La performance supérieure de l'ARIMA a été confirmée en comparant ses résultats à ceux obtenus par un réseau de neurones convolutifs (CNN), où l'ARIMA a non seulement surpassé le CNN en termes d'exactitude, mais s'est également révélé être considérablement plus rapide, traitant les données 43 fois plus rapidement pour plus de 80 000 lignes. Outre les modèles ARIMA et CNN, cette étude a aussi pris en considération les modèles BATS et TBATS pour évaluer leurs efficacités prédictives le contexte de cet étude. Le modèle BATS a révélé une erreur moyenne absolue en pourcentage de 20,04, tandis que TBATS a affiché une MAPE de 32,63. Ces résultats indiquent que le modèle BATS a démontré une précision supérieure à celle de TBATS, suggérant une meilleure adéquation de BATS pour cette application spécifique. Néanmoins, les deux modèles ont affiché une précision inférieure comparée à celle du modèle ARIMA, qui a montré des performances encore meilleures sur l'ensemble des ensembles de données examinés.

Nous résumerons dans le tableau suivant l'ensemble des travaux traitant la prédiction de trafic réseau et de congestions présentés dans ce chapitre, où chaque ligne résume un des articles cités auparavant.

Le tableau comporte (05) cinq colonnes, expliquées comme suit :

- **Approche** : Cette colonne cite les études référencées, en indiquant les auteurs et l'année de publication.
- **Algorithme** : Ici sont listés les algorithmes ou les modèles de machine learning utilisés dans chaque étude pour détecter la congestion ou prédire le trafic.
- **Application** : Cette colonne décrit l'objectif spécifique de chaque étude, elle indique

l'usage pratique des algorithmes mentionnés.

- **Données** : Elle décrit les données utilisées pour entraîner et tester les modèles.
- **Résultats et Conclusion** : Cette colonne présente une synthèse des résultats clés de chaque étude, accompagnée des conclusions tirées de ces dernières.

TABLE 2.1: Comparaison des approches de prédiction du trafic réseau et des congestions

Approche	Algorith- mes	Application	Données	Résultats et Conclusion
(Torres et al. 2018)	LSTM + régression logistique	Prédiction du trafic réseau d'une cellule d'un réseau mobile, suivie par l'identification de congestions en utilisant la régression logistique.	Données de débit descendant collectées à partir d'une cellule d'un opérateur de réseau mobile, enregistrées chaque heure sur une période d'un mois.	- Montre de meilleures performances comparées aux approches traditionnelles (ARIMA et Persistence Naive) - Peut prédire les Congestions jusqu'à 30h à l'avance
(Markus, Okereke, et Agee 2011)	Multi- Layer Feed forward Neural Network	Prédiction de la congestion du trafic téléphonique.	Données historiques du trafic issues d'enregistrements effectués par le commutateur réseau sur la route Kaduna-Abuja à des intervalles de 15 minutes, collectées pendant une période de sept mois.	- Une corrélation de 87 % entre les volumes de congestion prédits et réels, démontrant l'efficacité des réseaux neuronaux, particulièrement du MLP, dans la prédiction du trafic.

Approche	Algorithmes	Application	Données	Résultats et Conclusion
(Chmieliauskas et Gursnys 2019)	FbProphet	Prédiction à long terme du débit des cellules LTE pour identifier les zones à fort trafic nécessitant une expansion de capacité, en évaluant l'efficacité spectrale.	Données de trafic de 100 cellules, choisies de manière aléatoire, ont été collectées sur une période de 180 jours, avec une granularité d'une heure.	<ul style="list-style-type: none"> - La précision des prédictions de FbProphet est validée avec des scores R^2 et de variance expliquée de 0.9, indiquant une corrélation forte avec les tendances réelles. - Détection de congestions en évaluant les ressources d'une cellule réseau en utilisant les fréquences radio disponibles comme mesure. - La qualité des prédictions diminue après une semaine de prédiction.
(Do et al. 2020)	<ul style="list-style-type: none"> - LSTM - GRU 	Prédiction du volume du trafic de données une heure à l'avance.	Ensemble de données d'activité télécom mesurée chaque heure sur une année fourni par Viettel Telecom.	<ul style="list-style-type: none"> - Les modèles LSTM et GRU surpassent les méthodes ANFIS, ANN et GMDH en termes de RMSE, MAE et coefficient de corrélation. - LSTM et GRU se révèlent être des outils robustes et fiables pour la prédiction du trafic de données, aidant à la gestion optimale des ressources réseau et à la prévention de la congestion

Approche	Algorithmes	Application	Données	Résultats et Conclusion
(Jaffry, 2020)	- LSTM - FFNN - ARIMA	Prédiction du trafic cellulaire pour une gestion proactive des ressources réseau afin de prévenir la congestion.	Enregistrements d'appels (CDR) collectés par Telecom Italia sur 62 jours à Milan.	- Les résultats montrent que les modèles LSTM et FFNN prédisent avec une précision élevée, mais LSTM se distingue par une vitesse d'entraînement supérieure, même avec un nombre réduit de données d'entraînement. - ARIMA, bien qu'utile comme base de comparaison, ne capture pas aussi bien les dynamiques complexes du trafic cellulaire.
(Wa Nkongolo 2023)	- ARIMA - CNN	Prédiction de la croissance de la consommation des utilisateurs d'un opérateur téléphonique pour anticiper les congestions.	730 points de données provenant de la table des statistiques horaires du trafic dans le système Insights Data Storage (IDS).	- ARIMA donne de meilleures performances que CNN. - Les performances des modèles de prédiction ont largement influencés par les données utilisées. Dans cette situation, le modèle statistique ARIMA a surpassé le CNN malgré la simplicité de son approche.

2.3 Analyse et discussion

L'examen des différentes méthodes de prédiction de trafic réseau et de congestions souligne la complexité de la congestion réseau en tant que problème multifacette, nécessi-

tant une compréhension nuancée pour être abordé efficacement. Nous avons également pu conclure que les solutions de prédiction sont choisies en fonction des besoins spécifiques : certaines visent à prévoir ces événements à long terme (plus d'une semaine), tandis que d'autres se concentrent sur le court terme (quelques heures à deux jours à l'avance).

Les problèmes de prédiction ont d'abord été résolus par des algorithmes statistiques tels qu'ARMA, bien connus pour leur efficacité dans la prédiction des séries temporelles. Cependant, les algorithmes d'apprentissage automatique ont largement surpassé les performances des méthodes statistiques, offrant une précision et une adaptabilité accrues dans des contextes variés. Il est toutefois important de noter que, dans le cas de petits ensembles de données, les approches statistiques peuvent parfois être plus efficaces que les méthodes de deep learning, offrant des résultats plus robustes et pertinents.

2.4 Conclusion

Dans ce chapitre, nous avons dressé un état de l'art de la prédiction des congestions dans les réseaux de télécommunications, en incluant à la fois le résumé des travaux clés des chercheurs dans ce domaine et une analyse détaillée des différentes méthodes, jeux de données et résultats de ces études. Cette approche nous a permis de développer une compréhension éclairée de la contribution potentielle des diverses techniques à la résolution de ce problème. Ainsi, cette section offre un aperçu complet et structuré des avancées dans ce domaine, préparant ainsi le terrain pour notre contribution. Dans le chapitre suivant, nous détaillerons notre approche en détail.

Chapitre 3

Approche proposée

3.1 Introduction

Dans le chapitre précédent, nous avons établi un état de l'art sur la prédiction du trafic dans les réseaux de télécommunications et sur l'identification future des congestions.

Dans ce chapitre, nous allons présenter en détail l'approche proposée pour résoudre notre problème, dont l'objectif est de prédire les congestions futures. Nous commencerons par une vue globale de la solution, puis nous expliquerons en détail chaque bloc qui la compose.

3.2 Approche proposée

Notre projet consiste à prédire les congestions au sein des équipements d'antenne en utilisant l'apprentissage automatique. Pour atteindre cet objectif, nous avons structuré notre solution en plusieurs étapes, utilisant en entrée les données de trafic enregistrées par les équipements réseau.

Les données de trafic passeront d'abord par un bloc dédié à la prédiction. Les résultats de cette prédiction seront ensuite fournis à un second bloc, qui effectuera une tâche de détection. Enfin, ces résultats seront transmis à un tableau de bord, chargé de les diffuser. Les fonctions des deux blocs mentionnés sont les suivantes :

- Le premier bloc sera chargé de prédire l'évolution des valeurs de trafic pour chaque équipement réseau sur une ou plusieurs journées.
- Le second bloc sera responsable d'identifier les appareils susceptibles de subir des congestions.

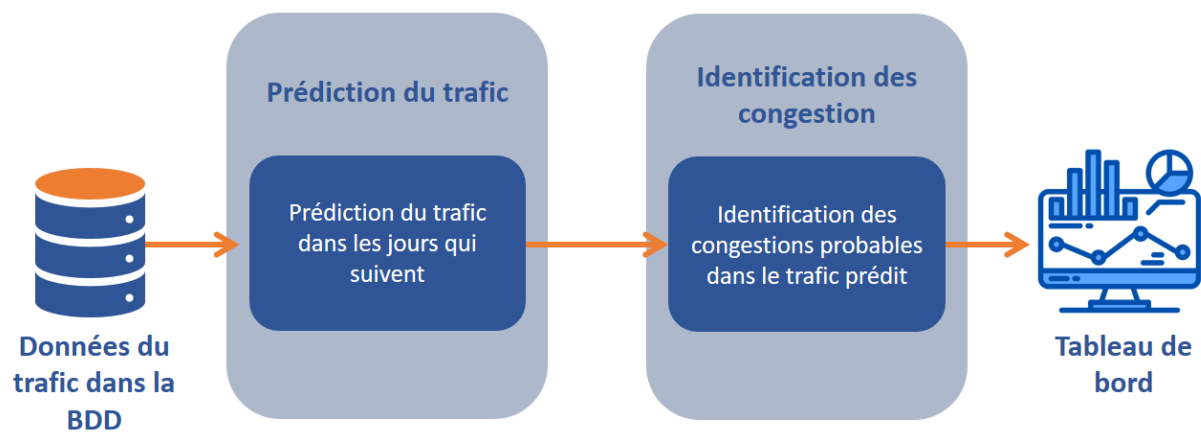


FIGURE 3.1 – Architecture globale de l'approche

Pour une meilleure compréhension de l'architecture présentée ci-dessus, nous allons détailler les étapes qui constituent chacun des deux blocs. Commençons par examiner de plus près le « Bloc de prédiction » :

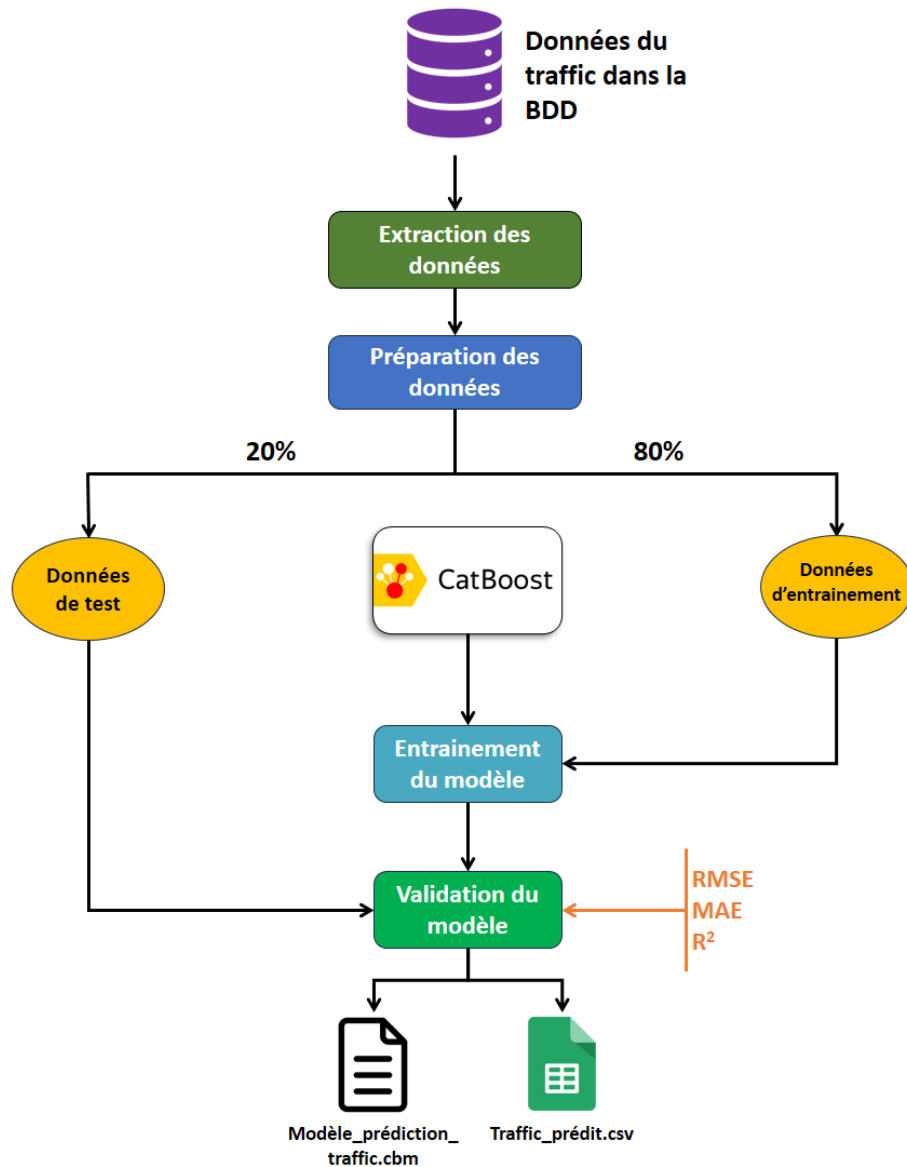


FIGURE 3.2 – Architecture globale du bloc de prédiction du pourcentage d'utilisation des équipements

La première étape de ce bloc consiste en l'extraction des données, c'est-à-dire la récupération des informations depuis les bases de données de Djazzy. Le résultat de cette extraction est un fichier CSV qui sert d'entrée à l'étape suivante, la préparation des données. Cette étape vise à effectuer les traitements nécessaires sur les données avant leur utilisation avec notre modèle d'apprentissage automatique. À l'issue de cette préparation, un fichier de données prétraitées est créé.

Ce fichier sert ensuite d'entrée pour l'étape d'entraînement et de validation du modèle. Ici, 80 % des données sont utilisées pour l'entraînement, tandis que les 20 % restants sont réservés pour la validation. Le modèle d'apprentissage automatique, une fois construit, sera capable de prédire les valeurs futures du trafic réseau pour tous les équipements.

En ce qui concerne le « Bloc d'identification des congestions », les étapes qui le composent sont illustrées dans la figure suivante :

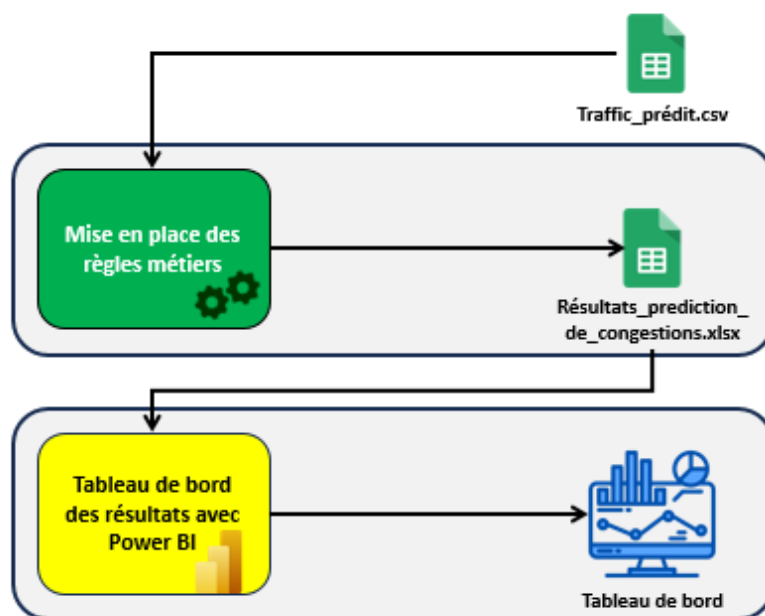


FIGURE 3.3 – Architecture globale d'identification des cas suspects de congestions

Comme illustré par la figure ci-dessus, Le bloc dit « identification des congestions probables » reçoit le trafic prédit par le modèle du « Bloc de prédiction ». Dans la première étape de ce bloc, nous mettons en œuvre une solution basée sur des règles métier pour déterminer si, d'après le trafic prédit, un appareil risque d'être congestionné. La liste des équipements identifiés sera sauvegardée dans un fichier Excel. Enfin, les résultats seront visualisés sur le tableau de bord lors de la dernière étape, qui consiste à afficher les équipements prévus comme étant congestionnés.

3.3 Vue détaillée du bloc de prédiction

Suite à la présentation de l'architecture globale de notre système, nous allons désormais explorer en détail le contenu du bloc de prédiction pour permettre une meilleure compréhension de notre approche.

3.3.1 Extraction des données

L'extraction des données est généralement la première étape dans l'élaboration d'un système de science des données, et cela s'applique également à la prédiction des congestions. Cette phase implique la collecte des données de trafic réseau disponibles sur diverses plateformes et stockées dans une base de données, puis à les regrouper dans un fichier CSV afin qu'elles soient exploitables pour les étapes suivantes. Dans mon cas, cette tâche a été réalisée par mon promoteur, car l'accès aux différentes plateformes était restreint pour des raisons de confidentialité et de sécurité.

3.3.2 Préparation des données

Une fois les données extraites, la phase cruciale de préparation commence. Cette étape est essentielle pour transformer les données brutes en un format adéquat, permettant leur utilisation efficace lors de l'entraînement du modèle. La qualité des résultats dépend largement de cette préparation minutieuse. Pour garantir que nos données soient prêtes à l'emploi, nous avons suivi plusieurs sous-étapes de préparation, aboutissant à la création d'un fichier contenant des données soigneusement préparées et prêtes à être intégrées dans le processus d'apprentissage automatique.

Voici les principales tâches réalisées au cours de cette étape de préparation des données :

Suppression des doublons

Une autre étape essentielle dans la préparation des données pour l'apprentissage automatique est la suppression des doublons. Ces derniers peuvent non seulement fausser les prédictions des modèles de ML, mais aussi consommer inutilement de l'espace de stockage et allonger les temps de traitement, notamment dans les grands ensembles de données. L'identification des doublons peut se faire de différentes manières, qu'il s'agisse de comparer des enregistrements pour trouver des correspondances exactes ou d'utiliser des techniques plus avancées pour détecter des similarités approximatives. Une fois ces doublons repérés, ils peuvent être supprimés ou fusionnés, en fonction des besoins spécifiques de l'analyse.

Traitement des données manquantes

Les valeurs manquantes sont un défi fréquent en apprentissage automatique. Elles peuvent être traitées par différentes méthodes : l'imputation, qui consiste à remplacer les valeurs manquantes par des estimations ou des prédictions ; l'interpolation, où les valeurs manquantes sont dérivées à partir des points de données adjacents ; ou encore la suppression, qui implique de retirer les lignes ou colonnes contenant des valeurs manquantes de l'ensemble de données.

Transformation des données

La transformation des données vise à restructurer et à améliorer les données pour qu'elles soient plus adaptées aux besoins spécifiques de l'analyse, qu'il s'agisse de réduire la complexité, d'améliorer la qualité ou de préparer les données pour un modèle particulier.

3.3.3 Processus de prédiction

L'objectif de cette section est d'identifier, au moins une journée à l'avance, les équipements susceptibles de connaître une congestion, en s'appuyant sur l'analyse du trafic réseau. Pour ce faire, nous avons utilisé l'algorithme CatBoost pour prédire le trafic réseau de chaque équipement sur une période de quatre jours.

3.3.3.1 Présentation du modèle CatBoost

CatBoost, développé par Yandex, a gagné en popularité ces dernières années grâce à ses performances supérieures, sa capacité à gérer les variables catégorielles, et ses techniques de régularisation robustes [32]. CatBoost est une méthode d'apprentissage d'ensemble qui combine plusieurs arbres de décision pour créer un modèle prédictif puissant. L'algorithme construit une série d'arbres de manière séquentielle, où chaque arbre suivant est entraîné pour corriger les erreurs commises par les arbres précédents [33]. CatBoost utilise une structure d'arbre symétrique, ce qui garantit que le modèle n'est pas sensible à l'ordre des caractéristiques d'entrée, ce qui conduit à des prédictions plus stables et précises [34].

L'un des principaux avantages de CatBoost est sa prise en charge native des caractéristiques catégorielles. Contrairement à d'autres algorithmes de gradient boosting qui nécessitent un prétraitement étendu des variables catégorielles, comme le one-hot encoding ou le label encoding, CatBoost peut gérer directement les caractéristiques catégorielles. Il utilise une technique appelée "ordered target statistics", qui calcule une statistique cible pour chaque catégorie en se basant sur la valeur moyenne de l'étiquette des échantillons d'entraînement au sein de cette catégorie [35]. Cette approche permet à CatBoost de capturer efficacement les relations entre les caractéristiques catégorielles et la variable cible.

CatBoost se distingue aussi par son mécanisme de boosting ordonné, qui est l'une de ses principales innovations. Contrairement aux méthodes traditionnelles de gradient boosting, qui peuvent être sujettes à des biais de prédiction en raison de l'utilisation des mêmes données à la fois pour l'entraînement des modèles de base et pour l'ajustement des erreurs, CatBoost introduit une permutation aléatoire du jeu de données à chaque itération. Lors de l'entraînement, seules les données précédant chaque exemple dans cette permutation sont utilisées. Cette méthode permet de limiter le surapprentissage et renforce la robustesse du modèle [36].

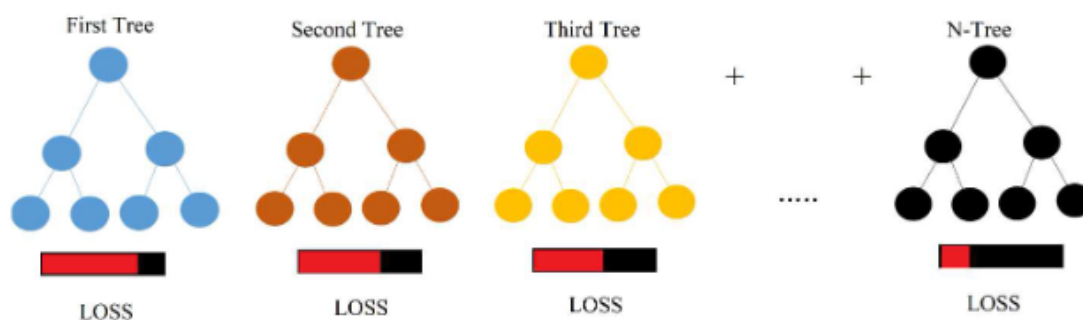


FIGURE 3.4 – Schéma de fonctionnement du modèle CatBoost

Les arbres de décision boostés par le gradient apprennent de manière itérative en construisant une collection de fonctions $F^0, F^1, F^2, \dots, F^k, \dots, F^n$ accompagnée d'un taux d'apprentissage spécifié (α). Considérons les valeurs de sortie $\{y_1, y_2, y_3, \dots, y_m\}$ correspondant à un ensemble de variables prédictives $\{X_1, X_2, X_3, \dots, X_m\}$. La fonction de perte $J(y_i, F^k)$ peut être améliorée en trouvant une autre fonction $F^{k+1} = F^k + t^{k+1}(X)$, telle que t^{k+1} minimise la valeur anticipée de la fonction de perte. Ici, t provient de l'ensemble des arbres de décision qui sont candidats à l'ajout à l'ensemble [36].

$$t^{k+1} = \arg \min J(y, F_k) \quad (3.1)$$

En termes de F^{k+1} et F^k l'expression peut être écrite comme suit :

$$J(y, F^{k+1}) = J(y, F^k + t^{k+1}) \quad (3.2)$$

3.3.3.2 Principaux hyperparamètres de CatBoost

CatBoost, comme beaucoup d'autres algorithmes d'apprentissage automatique, permet à l'utilisateur de modifier de nombreux hyperparamètres pour optimiser la performance du modèle selon des besoins spécifiques. Ces paramètres peuvent inclure :

- **iterations** : Cet hyperparamètre détermine le nombre total d'arbres à construire lors de l'entraînement du modèle. Chaque arbre est ajouté séquentiellement dans un processus de boosting, où chaque nouvel arbre corrige les erreurs des prédictions des arbres précédents. Un nombre d'itérations trop faible peut conduire à un sous-ajustement, tandis qu'un nombre trop élevé peut entraîner un sur-ajustement, surtout si le taux d'apprentissage est élevé [37].
- **learning_rate** : Le taux d'apprentissage contrôle la contribution de chaque arbre ajouté au modèle final. Il réduit l'impact de chaque nouvel arbre, en ralentissant le processus d'apprentissage pour obtenir une meilleure généralisation. Un taux d'apprentissage plus bas nécessite plus d'itérations pour atteindre une bonne performance, mais peut améliorer la stabilité et la capacité du modèle à généraliser. Un taux trop élevé peut faire converger le modèle trop rapidement et risquer de manquer un minimum global [37].
- **depth** : La profondeur détermine la complexité de chaque arbre en contrôlant le nombre de niveaux qu'un arbre de décision peut avoir [37]. Plus la profondeur est élevée, plus l'arbre est capable de capturer des relations complexes dans les données. Une profondeur trop faible peut conduire à un sous-ajustement, où le modèle n'est pas assez flexible pour capturer les patterns des données. Une profondeur trop élevée peut conduire à un sur-ajustement, où le modèle s'adapte trop étroitement aux données d'entraînement [38].
- **l2_leaf_reg** : Cet hyperparamètre est un terme de régularisation L2 appliqué aux feuilles des arbres. La régularisation L2 pénalise les coefficients des arbres pour éviter qu'ils ne deviennent trop grands, ce qui aide à prévenir le sur-apprentissage [38]. Un coefficient de régularisation élevé peut rendre le modèle trop simple (sous-ajustement), tandis qu'un coefficient trop bas peut conduire à un modèle trop complexe (sur-ajustement). Cela permet de contrôler l'équilibre entre la variance et le biais.
- **loss_function** : La fonction de perte détermine comment l'erreur du modèle est mesurée par rapport aux cibles réelles. Elle guide le processus d'optimisation en indiquant comment ajuster les prédictions du modèle pour minimiser cette perte. Le choix de la fonction de perte dépend de la nature du problème à résoudre (classification, régression, etc.) et affecte directement la manière dont le modèle est optimisé [38].
- **boosting_type** : Cet hyperparamètre permet de choisir le schéma de boosting, qui peut être configuré en mode "plain" pour le schéma classique du gradient boosting, ou en mode "ordered", offrant une meilleure qualité sur des ensembles de données plus petits [39].

3.4 Conclusion

Dans ce chapitre, nous avons commencé par présenter notre approche ainsi que la manière dont elle résout le problème de la prédiction de congestion dans les équipements réseaux en utilisant CatBoost. Ensuite, nous avons examiné l'architecture globale de notre pipeline et le rôle de chaque composant. Enfin, nous avons décrit l'algorithme CatBoost que nous avons adopté dans notre approche, en mettant en avant les principaux hyperparamètres qui permettent d'optimiser les performances du modèle.

Dans le chapitre suivant, nous détaillerons tous les aspects liés à l'implémentation de notre approche.

Chapitre 4

Expérimentation et évaluation

4.1 Introduction

Ce chapitre est consacré à la phase d'expérimentation et d'évaluation des modèles développés dans le cadre de notre projet de prédiction du trafic réseau et de détection des congestions. Nous y décrivons les principales étapes suivies pour préparer les données, développer les modèles de prédiction, et tester leur performance. Pour conclure ce chapitre, nous présenterons un tableau de bord interactif permettant de visualiser les équipements identifiés comme congestionnés.

4.2 Description du dataset

4.2.1 Description

Un dataset (ou ensemble de données en français) est une collection structurée de données utilisée pour des analyses statistiques, des recherches ou des applications d'apprentissage automatique. Il peut être constitué de données de différentes natures (numériques, textuelles, images, etc.) et est généralement organisé sous forme de tableau, où chaque ligne représente une observation et chaque colonne une variable ou caractéristique. Dans l'apprentissage automatique, un dataset est utilisé pour entraîner un modèle qui apprend à prédire des résultats en se basant sur des données d'entrée.

Le dataset utilisé dans ce projet contient l'historique du trafic réseau de 39 équipements d'antennes de Djazzy, avec un total de 93 186 observations. Chaque enregistrement reflète le pourcentage d'utilisation maximale d'un équipement dans les deux directions, "réception" et "transmission" sur un intervalle de 15 minutes. Les données, collectées sur une période de 20 jours, couvrent l'intervalle du 1er au 20 janvier 2024.

Ci-dessous une description des colonnes de ce jeu de données :

- **ResourceName** : Nom d'un équipement d'antenne, une antenne peut être composée d'un ou plusieurs équipements. Le nom d'un équipement donné commence par le nom de l'antenne à laquelle il appartient, exemple : nomAntenne-nomEquipement.
- **CollectionTime** : La date et l'heure d'enregistrement de la ligne de trafic.
- **PORT_RX_BW_UTILIZATION_MAX** : Le pourcentage d'utilisation maximal (%) lors de la réception (RX) de l'équipement/liens portant le nom spécifié dans la colonne « ResourceName ».
- **PORT_TX_BW_UTILIZATION_MAX** : Le pourcentage d'utilisation maximal (%) lors de la transmission (TX) de l'équipement/liens portant le nom spécifié dans la colonne "ResourceName".

	CollectionTime	ResourceName	PORT_RX_BW_UTILIZATION_MAX	PORT_TX_BW_UTILIZATION_MAX
38601	2024-01-01 00:00:00	C40S247_IDU1-5(PLA-5)	17.26	100.00
32121	2024-01-01 00:00:00	O27T104 IDU1-1(PLA-1)	41.31	6.98
72989	2024-01-01 00:00:00	O31S341_MSC24_IDU1(BOR4H)_O31X386-22-SLB2EG4-2...	70.94	10.96
64567	2024-01-01 00:00:00	A44X138 IDU-3-ISV3-1(To A44X018)	22.98	4.54
36895	2024-01-01 00:00:00	A06S287_IDU1(BJA3H)-17-EG2D-1(To ALU TSS MAIN)	25.05	6.70
...
2404	2024-01-20 23:45:00	A09X017_V_ZAOUIA-1(PLA-1)	69.54	8.11
79110	2024-01-20 23:45:00	O31S341_MSC24_IDU1(BOR4H)_O31X386-22-SLB2EG4-2...	68.59	8.77
79567	2024-01-20 23:45:00	O02X254_IDU2-2(PLA-2)	48.49	4.06
83928	2024-01-20 23:45:00	A44X021_NE_2-12-ISV3-1(RTNIF-1)	46.67	7.28
67800	2024-01-20 23:45:00	A16X387 IDU 2-17-EG4-3(cascad 3G IDU1)	9.44	72.24

93186 rows × 4 columns

FIGURE 4.1 – Aperçu du dataset

4.2.2 Préparation de données

Comme expliqué dans le chapitre précédent, il était essentiel de préparer les données avant de les utiliser. Pour ce faire, nous avons supprimé les doublons, remplacé les valeurs manquantes et effectué une transformation des données.

Suppression des doublons

Après avoir identifié la nécessité de supprimer les doublons dans les données, une approche en deux étapes a été mise en œuvre. Tout d'abord, une suppression globale des doublons a été effectuée pour éliminer toutes les lignes identiques dans le dataset. Cette étape a permis de garantir que chaque enregistrement était unique et qu'aucune observation redondante ne subsistait.

Ensuite, pour les cas où plusieurs enregistrements correspondaient au même équipement (ResourceName) et à la même période (CollectionTime), une stratégie d'agrégation a été appliquée. Plutôt que de simplement supprimer ces doublons potentiels, nous avons choisi de regrouper ces enregistrements et de conserver les valeurs maximales de chacune des colonnes PORT_TX_BW_UTILIZATION_MAX et PORT_RX_BW_UTILIZATION_MAX.

Nombre d'enregistrements	Nombre d'enregistrements
Etape 1: 85551	Etape2: 73635

FIGURE 4.2 – Nombre d'observations après la première et la deuxième étape de suppression des doublons

Après l'application des étapes de suppression des doublons, la taille du dataset a été considérablement réduite. Initialement, l'ensemble comptait 93 186 enregistrements. Après la première étape, ce nombre est passé à 85 551, puis à 73 635 à l'issue de la deuxième étape.

Traitement des valeurs manquantes

Compte tenu des valeurs manquantes présentes dans notre jeu de données, sous forme d'enregistrements manquants à certains intervalles de temps, il était crucial de procéder à une réindexation des données avant d'appliquer une méthode d'imputation appropriée. Pour ce faire, nous avons d'abord réindexé les enregistrements de chaque équipement selon une plage de temps continue, assurant ainsi que chaque intervalle de 15 minutes soit représenté, même si des enregistrements étaient absents.

Une fois cette réindexation effectuée, nous avons opté pour l'imputation des valeurs manquantes en utilisant une méthode basée sur des moyennes de temps similaires. Concrètement, pour chaque enregistrement manquant, nous avons identifié les jours de la semaine et les heures similaires, puis calculé la moyenne des valeurs correspondantes dans le même créneau horaire sur ces jours-là. Ces moyennes ont été utilisées pour combler les valeurs manquantes, ce qui nous a permis de maintenir la continuité des données sans perdre d'informations critiques.

ResourceName	
16x928 IDU2-23-SLB2EX1-1(16X830R EB H)	11
A06S287-IDU4-3-ISV3-1(A06X159 main)	9
A06S287_IDU1(BJA3H)-17-EG2D-1(To ALU TSS MAIN)	2
A06S287_IDU1(BJA3H)-7-EM6T-1(Cascade to A06S287...	2
A06X030_IDU-2(PLA-2)	8
A06X183_IDU1-12-ISV3-1(To A06X025)	6
A09T003_IDU1(BLI04H)-17-SLB2EG4-2(TO CX MAIN)	2
A09X017_V_ZAOUIA-1(PLA-1)	11
A16M831-IDU2-22-SLB2EG4-2(TO-CX-SL1SUB1P14 STBDY)	11
A16T022_BARAKI_IDU1(BA07H)-11-ISV3-1(TO A16X492)	3
A16T028_EBAND-1-MXUG4-5(10GE FO)	4
A16T122_IDU2_B6-17-SLB2EG4-1(TO CX 0/2/3)	4
A16X051-14-ISV3-1(RTNIF-1)	8
A16X151--10(PLA-10)	10
A16X181-EBAND-1-MXUF4-1(IF)	11
A16X1999_NE_1-1-EG4-2(To eNode A16X1999)	7
A16X387 IDU 2-17-EG4-3(cascad 3G IDU1)	7
A44S336_IDU2-13-ISV3-1(RTNIF-1)	7
A44X018_IDU1_Djelidha-10-ISV3-1(To A44X138 H STB)	2
A44X021_NE_2-12-ISV3-1(RTNIF-1)	11
A44X051-3-ISV3-1(To A44X014)	6
A44X138 IDU-11-ISV3-1(To A44X030 MAIN V)	2
A44X138 IDU-3-ISV3-1(To A44X018)	2
A44X138 IDU-5-ISV3-1(To A44X018)	2
C40S247_IDU1-5(PLA-5)	12
O02X151_IDU2-18-SLB2EX1-1(Eband O02X174)	10
O02X195-IDU1-4-ISV3-1(TO O02X054 V)	4

FIGURE 4.3 – Nombre de valeurs manquantes par équipement

ResourceName	
16x928 IDU2-23-SLB2EX1-1(16X830R EB H)	0
A06S287-IDU4-3-ISV3-1(A06X159 main)	0
A06S287_IDU1(BJA3H)-17-EG2D-1(To ALU TSS MAIN)	0
A06S287_IDU1(BJA3H)-7-EM6T-1(Cascade to A06S287...	0
A06X030_IDU-2(PLA-2)	0
A06X183_IDU1-12-ISV3-1(To A06X025)	0
A09T003_IDU1(BLI04H)-17-SLB2EG4-2(TO CX MAIN)	0
A09X017_V_ZAOUIA-1(PLA-1)	0
A16M831-IDU2-22-SLB2EG4-2(TO-CX-SL1SUB1P14 STBDY)	0
A16T022_BARAKI_IDU1(BA07H)-11-ISV3-1(TO A16X492)	0
A16T028_EBAND-1-MXUG4-5(10GE FO)	0
A16T122_IDU2_B6-17-SLB2EG4-1(TO CX 0/2/3)	0
A16X051-14-ISV3-1(RTNIF-1)	0
A16X151--10(PLA-10)	0
A16X181-EBAND-1-MXUF4-1(IF)	0
A16X1999_NE_1-1-EG4-2(To eNode A16X1999)	0
A16X387 IDU 2-17-EG4-3(cascad 3G IDU1)	0
A44S336_IDU2-13-ISV3-1(RTNIF-1)	0
A44X018_IDU1_Djelidha-10-ISV3-1(To A44X138 H STB)	0
A44X021_NE_2-12-ISV3-1(RTNIF-1)	0
A44X051-3-ISV3-1(To A44X014)	0
A44X138 IDU-11-ISV3-1(To A44X030 MAIN V)	0
A44X138 IDU-3-ISV3-1(To A44X018)	0
A44X138 IDU-5-ISV3-1(To A44X018)	0
C40S247_IDU1-5(PLA-5)	0
002X151_IDU2-18-SLB2EX1-1(Eband 002X174)	0
002X195-IDU1-4-ISV3-1(TO 002X054 V)	0

FIGURE 4.4 – Nombre de valeurs manquantes par équipement après l'imputation

Transformation des données

Dans le but d’optimiser le format des données et réduire la consommation en mémoire tout en améliorant l’efficacité de l’apprentissage, nous avons réduit la dimensionnalité des données. Au lieu de conserver 96 valeurs de trafic par jour, enregistrées toutes les 15 minutes, nous avons opté pour une seule observation par heure. Après avoir exploré plusieurs méthodes, nous avons retenu la valeur maximale par heure, car elle préservait au mieux les informations du trafic.

	CollectionTime	ResourceName	PORT_TX_BW_UTILIZATION_MAX	PORT_RX_BW_UTILIZATION_MAX
2079	2024-01-07 15:00:00	A06X030_IDU-2(PLA-2)	64.64	7.27
2130	2024-01-09 18:00:00	A06X030_IDU-2(PLA-2)	62.13	9.15
15896	2024-01-03 08:00:00	O31X160_IDU4-17-SLB2EG4-2(To CX 1/016)	7.86	34.82
16810	2024-01-01 10:00:00	O31X233_TORRO-5-ISV3-1(O31S341 STBY V)	7.75	40.53

FIGURE 4.5 – Aperçu des données Agrégées par Heure

Dans le même processus de transformation des données, nous avons également créé deux nouvelles colonnes basées sur CollectionTime, à savoir hour pour indiquer l’heure de la journée, et weekday pour identifier le jour de la semaine. Ces caractéristiques temporelles ont été introduites dans le but de mieux capturer les variations temporelles dans les données et ainsi améliorer la précision des prédictions.

	CollectionTime	ResourceName	PORT_TX_BW_UTILIZATION_MAX	PORT_RX_BW_UTILIZATION_MAX	hour	weekday
2079	2024-01-07 15:00:00	A06X030_IDU-2(PLA-2)	64.64	7.27	15	6
2130	2024-01-09 18:00:00	A06X030_IDU-2(PLA-2)	62.13	9.15	18	1
15896	2024-01-03 08:00:00	O31X160_IDU4-17-SLB2EG4-2(To CX 1/016)	7.86	34.82	8	2
16810	2024-01-01 10:00:00	O31X233_TORRO-5-ISV3-1(O31S341 STBY V)	7.75	40.53	10	0

FIGURE 4.6 – Extrait des données après l’ajout des caractéristiques temporelles

4.3 Environnement de développement

Dans cette section, nous présentons l’environnement logiciel et matériel utilisé pour réaliser notre travail.

4.3.1 Environnement logiciel

- **Jupyter Notebook** : Jupyter Notebook est devenu un outil incontournable pour les chercheurs, les ingénieurs et les professionnels de la science des données. Son interface

conviviale et son support multi-langage en font un choix idéal pour la création de rapports interactifs, l’exploration de données et le développement de modèles d’apprentissage automatique. Les fonctionnalités de collaboration de Jupyter Notebook permettent également aux équipes de travailler efficacement ensemble sur des projets en partageant facilement des notebooks via des plateformes en ligne telles que GitHub. Avec sa grande communauté d’utilisateurs et de développeurs, Jupyter Notebook continue d’évoluer et de s’adapter aux besoins changeants du domaine de la science informatique [40].

- **Python** : Python est le langage privilégié en matière de Data Science en raison de sa simplicité, sa lisibilité, sa propreté et sa flexibilité. Il est compatible avec de nombreuses plateformes et ses nombreuses bibliothèques offrent une large gamme de fonctionnalités pour effectuer différentes tâches dans ce domaine [41].

4.3.2 Bibliothèques Utilisées

- **NumPy** : est une bibliothèque fondamentale pour le calcul scientifique en Python. Elle offre un puissant objet tableau multidimensionnel, ainsi que des outils pour effectuer des opérations mathématiques et logiques sur ces tableaux de manière efficace. NumPy est largement utilisé pour le traitement de données, l’algèbre linéaire, la transformation de Fourier, et plus encore, ce qui en fait un pilier essentiel pour de nombreuses applications de Data Science et d’apprentissage automatique [42].
- **Pandas** : Pandas est une bibliothèque Python largement employée pour la manipulation et l’analyse de données. Elle fournit des structures de données et des outils puissants pour gérer des données tabulaires et temporelles, ce qui en fait un choix privilégié pour le nettoyage, la transformation et l’analyse des données avant leur utilisation dans des modèles d’apprentissage automatique [42].
- **Matplotlib** : Matplotlib est la bibliothèque Python la plus couramment utilisée pour créer des graphiques et des visualisations bidimensionnelles. Développée à l’origine par John D. Hunter, elle bénéficie aujourd’hui du soutien d’une large communauté de développeurs. Matplotlib est spécialement conçue pour produire des graphiques de qualité professionnelle, prêts pour la publication. Bien que d’autres bibliothèques de visualisation soient disponibles, Matplotlib est la plus répandue et s’intègre parfaitement à l’écosystème Python. Elle constitue donc un choix fiable pour la visualisation de données [42].
- **Scikit-learn** : Souvent abrégé en ‘sklearn’, est une bibliothèque Python largement utilisée pour l’apprentissage automatique. Elle offre une gamme complète d’outils efficaces pour la modélisation prédictive, incluant des algorithmes de classification, de régression, de clustering, et de réduction de dimensionnalité. Scikit-learn est par-

ticulièrement apprécié pour sa simplicité d'utilisation, son intégration fluide avec d'autres bibliothèques comme NumPy et Pandas, et sa documentation claire. Elle est conçue pour être accessible aux débutants tout en étant suffisamment puissante pour répondre aux besoins des experts en machine learning [42].

- **CatBoost** : CatBoost est spécifiquement conçue pour implémenter l'algorithme de boosting par gradient appelé CatBoost. Cette bibliothèque a été développée par Yandex pour simplifier l'application de cet algorithme dans diverses tâches de machine learning, en offrant une API simple et efficace pour entraîner des modèles de classification, de régression, et de ranking. CatBoost est réputé pour sa facilité d'utilisation, notamment grâce à ses interfaces Python qui s'intègrent de manière fluide avec scikit-learn. Cette compatibilité permet d'exploiter pleinement les fonctionnalités avancées de scikit-learn, telles que la validation croisée, l'optimisation des hyperparamètres, et la gestion des ensembles de données, tout en profitant de la puissance de l'algorithme CatBoost [43].
- **os** : La bibliothèque os en Python est une bibliothèque standard qui permet d'interagir avec le système d'exploitation. Elle fournit un large éventail de fonctions pour manipuler les fichiers, les répertoires, les processus, et d'autres éléments du système [44].

4.3.3 Environnement Matériel

La configuration de la machine utilisée est comme suit :

- Processeur : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
- RAM : 12.00 GO
- Disque Dur : 1.00 TO
- Type du système : 64 bits, processeur x64
- Système d'exploitation : Windows 10 Professionnel

4.4 Construction et évaluation des modèles de prédiction

4.4.1 Construction des modèles

Il convient de rappeler que nos données représentent l'évolution de la quantité du trafic pour chaque équipement réseau considéré dans notre ensemble. Chaque équipement est associé à deux variables clés, `PORT_RX_BW_UTILIZATION_MAX` et `PORT_TX_BW_UTILIZATION`.

mesurant respectivement le trafic en réception et en transmission. Ainsi, chaque équipement génère une série temporelle multivariée. L'ensemble de nos données se compose donc de plusieurs séries temporelles qui sont à la fois multiples et multivariées.

Compte tenu de la nature multivariée et multiple de nos séries temporelles, chaque équipement étant associé à deux directions de trafic distinctes (réception et transmission), il a été jugé nécessaire de construire deux modèles par équipement, chacun dédié à une direction spécifique.

La construction des modèles de prédiction pour le trafic réseau a été réalisée en suivant une approche systématique pour chaque équipement, tant pour le trafic en réception (RX) que pour le trafic en transmission (TX). Bien que les deux modèles suivent une logique de construction similaire, ils présentent quelques différences mineures dans la sélection des hyperparamètres et les colonnes de données spécifiques utilisées pour la prédiction. Voici comment ces modèles ont été construits en détail :

Sélection des Caractéristiques et séparation des Données

Pour chaque équipement, les données ont été soigneusement filtrées pour isoler les informations spécifiques à l'équipement en question. Les caractéristiques temporelles telles que l'heure de la journée et le jour de la semaine ont été intégrées pour capturer les variations temporelles inhérentes au trafic réseau. Les colonnes clés diffèrent selon la direction du trafic : **PORT_RX_BW_UTILIZATION_MAX** pour la prédiction du trafic en réception et **PORT_TX_BW_UTILIZATION_MAX** pour le trafic en transmission. Les données ont ensuite été divisées en ensembles d'entraînement et de test, en conservant 80% des données pour l'entraînement et 20% pour le test.

Création des Séquences Temporelles

Étant donné la nature séquentielle des données de trafic, il était crucial de structurer les données en séquences temporelles. Chaque séquence a été formée en utilisant une fenêtre glissante de 3 observations. Cette structuration en séquences permet aux modèles d'apprendre les tendances temporelles, ce qui est essentiel pour capturer les dynamiques du trafic réseau au fil du temps.

Modélisation avec CatBoost

Une recherche en grille a été menée pour optimiser plusieurs hyperparamètres, dont le nombre d'itérations, la profondeur des arbres, le taux d'apprentissage, et la régularisation L2, comme détaillé dans le tableau ci-dessous.

TABLE 4.1 – Valeurs des hyperparamètres possibles pour les modèles de prédiction

Hyperparamètre	Valeurs testées pour RX	Valeurs testées pour TX
iterations	100, 200, 300	200, 300, 500
depth	2, 3, 4	2, 3, 4
learning_rate	0.03, 0.04, 0.05	0.03, 0.04, 0.05
l2_leaf_reg	1, 3, 5	1, 3, 5

La fonction de perte utilisée était l’erreur quadratique moyenne (RMSE) et le type de boosting a été fixé à ‘Ordered’. Une validation croisée en cinq segments (`n_splits=5`) a été réalisée, en respectant la structure temporelle des données, afin d’optimiser les hyperparamètres du modèle, garantissant ainsi une meilleure généralisation sur des données non vues et réduisant le risque de surajustement.

Sélection du Meilleur Modèle

Après l’optimisation des hyperparamètres, le meilleur modèle pour chaque équipement a été sélectionné, que ce soit pour la prédiction du trafic en réception ou en transmission. Ces modèles seront utilisés pour prédire le trafic dans le futur.

4.4.2 Métriques d’évaluation utilisées

Une fois les modèles construits et optimisés, nous les avons évalués en mesurant l’erreur entre les valeurs réelles et les valeurs prédites à l’aide des métriques suivantes :

Racine de l’erreur quadratique moyenne (RMSE)

La RMSE (Root Mean Square Error) est la racine carrée de la moyenne des carrés des erreurs entre les valeurs réelles et les valeurs prédites par un modèle. Il s’agit d’une mesure de la précision d’un modèle, exprimée dans les mêmes unités que la variable cible, la RMSE est définie comme suit :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4.1)$$

Où :

- y_i : est la valeur réelle pour l’observation i
- \hat{y}_i : est la valeur prédite pour l’observation i
- n est le nombre d’observations

Interprétation de la RMSE :

- Bas RMSE : Indique que le modèle prédit les valeurs proches des valeurs réelles.
- Haut RMSE : Suggère que le modèle a de grandes erreurs de prédiction.

Erreur Absolue Moyenne (MAE)

La MAE (Mean Absolute Error) mesure la moyenne des erreurs absolues entre les valeurs réelles et les valeurs prédites par un modèle. C'est une mesure simple de la précision d'un modèle qui indique en moyenne de combien les prédictions s'écartent des valeurs réelles. Elle est définie par la formule suivante :

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

Interprétation de la MAE :

- MAE faible : Indique que les prédictions du modèle sont en moyenne proches des valeurs réelles.
- MAE élevé : Suggère que les prédictions s'écartent en moyenne davantage des valeurs réelles.

R² score

Le R² score, aussi appelé R-squared ou coefficient de détermination est une mesure statistique utilisée pour évaluer la qualité d'un modèle de régression. Il indique la proportion de la variance dans la variable dépendante (cible) qui est expliquée par le modèle.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4.3)$$

Où :

- \bar{y} : est la moyenne des valeurs réelles

Interprétation du R² Score :

- $R^2 = 1$: Le modèle explique parfaitement toute la variance des données (les prédictions sont exactement égales aux valeurs réelles).
- $R^2 = 0$: Le modèle n'explique aucune variance, c'est-à-dire que les prédictions ne sont pas meilleures qu'une prédiction constante égale à la moyenne des valeurs réelles.
- $R^2 < 0$: Le modèle est pire que de simplement prédire la moyenne des valeurs réelles. Cela peut se produire si le modèle ne capture pas du tout la tendance des données ou si la modélisation est inappropriée.

4.4.3 Évaluation des modèles construits

Après avoir construit les modèles de prédiction pour chaque direction (réception et transmission) de chaque équipement, nous les évaluerons sur l'ensemble des données de test. Les performances seront évaluées à l'aide des métriques précédemment mentionnées : l'erreur absolue moyenne (MAE), la racine de l'erreur quadratique moyenne (RMSE) et le coefficient de détermination (R^2), garantissant ainsi une analyse précise pour chaque modèle.

a- Analyse détaillée des performances des 39 modèles de prédiction pour le pourcentage d'utilisation en réception (PORT_RX_BW_UTILIZATION_MAX)

Dans ce qui suit, nous montrons la variation de chacune des métriques étudiées en utilisant des boîtes à moustaches :

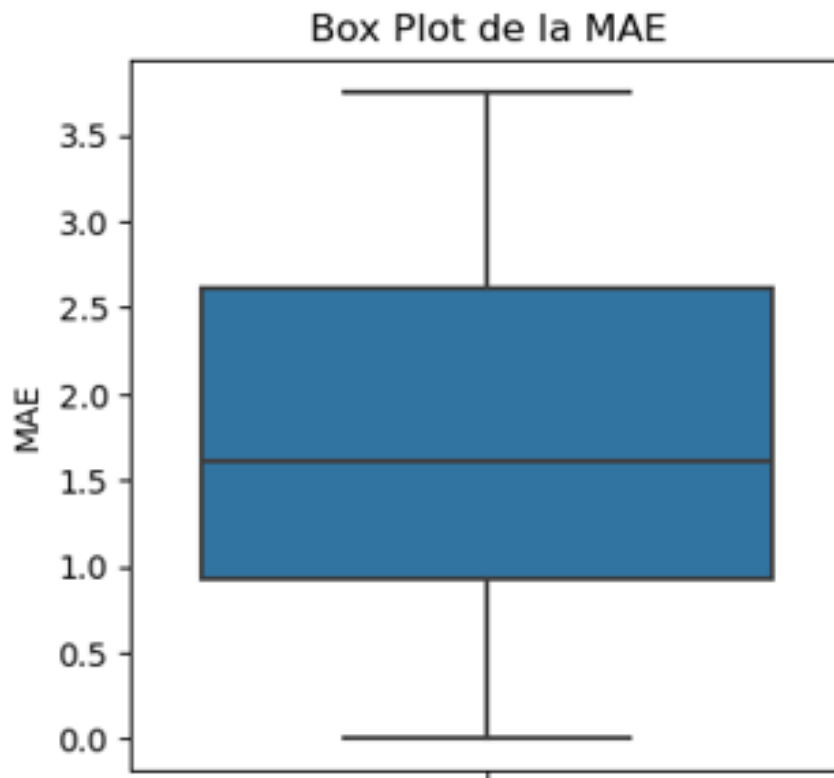


FIGURE 4.7 – Boîte à moustache de la MAE des modèles CatBoost pour la direction réception(RX)

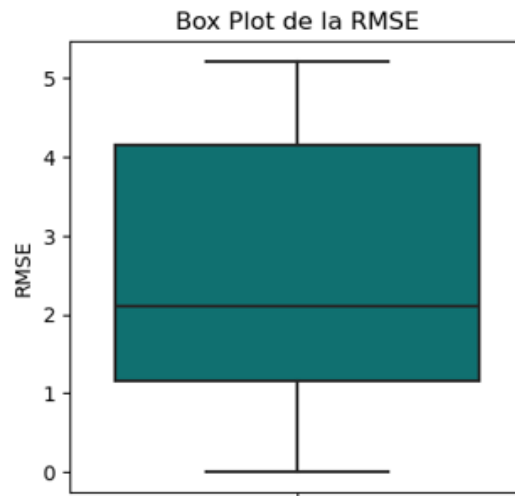


FIGURE 4.8 – Boîte à moustache de la RMSE des modèles CatBoost pour la direction réception(RX)

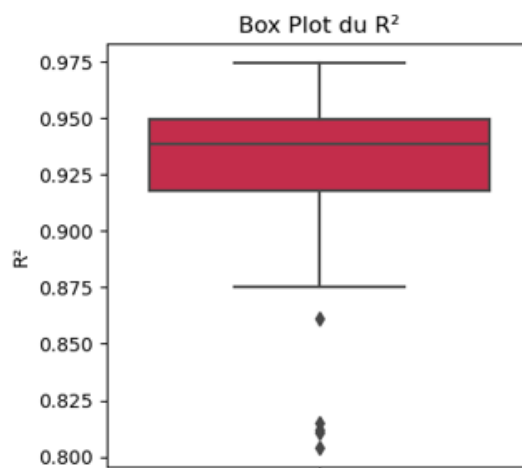


FIGURE 4.9 – Boîte à moustache du R² des modèles CatBoost pour la direction réception(RX)

Analyse : L'analyse des métriques MAE, RMSE, et R^2 souligne la robustesse et la précision de ces modèles de prédiction de l'utilisation en réception. Avec une médiane du MAE à 1.62 et des quartiles entre 0.93 et 2.61, les erreurs moyennes restent faibles et bien contenues. La médiane du RMSE à 2.10, avec des quartiles allant de 1.15 à 4.14, montre que même les grandes erreurs restent modérées. Le R^2 , avec une médiane de 0.94 et des quartiles entre 0.92 et 0.95, reflète une excellente capacité explicative des modèles.

b- Analyse détaillée des performances des 39 modèles de prédiction pour le pourcentage d'utilisation en transmission (PORT_TX_BW_UTILIZATION_MAX)

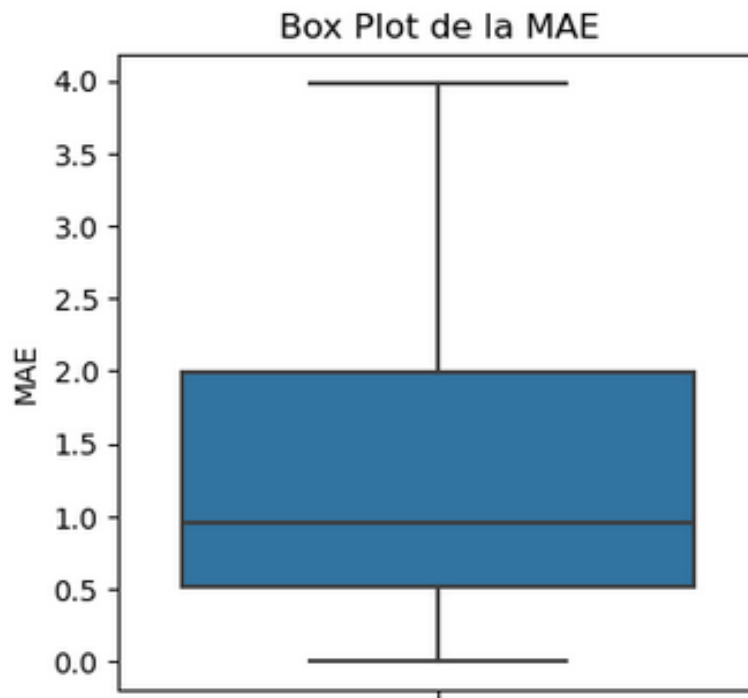


FIGURE 4.10 – Boîte à moustache de la MAE des modèles CatBoost pour la direction transmission(TX)

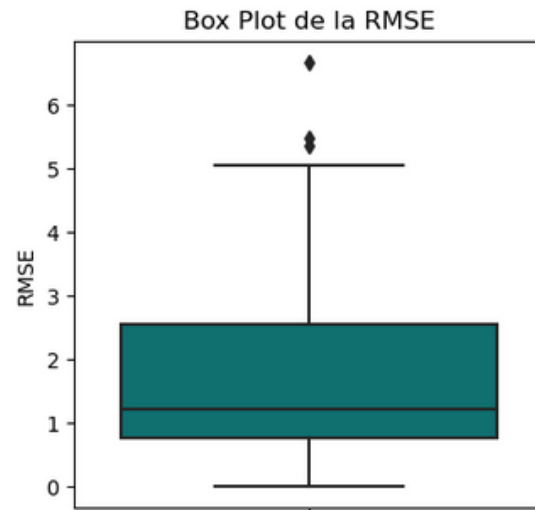


FIGURE 4.11 – Boîte à moustache de la RMSE des modèles CatBoost pour la direction transmission(TX)



FIGURE 4.12 – Boîte à moustache du R^2 des modèles CatBoost pour la direction transmission(TX)

Analyse : L'analyse des performances des modèles de prédiction pour la transmission est globalement positive. La MAE, avec une médiane de 0,9523, indique des erreurs absolues moyennes faibles, ce qui reflète une performance solide. La RMSE, légèrement plus élevée avec une médiane de 1.2132, reste dans une plage acceptable. Le R^2 , particulièrement élevé avec une médiane de 0.9138, indique que les modèles capturent efficacement la variance des données. Globalement, ces résultats témoignent de la robustesse et de l'efficacité des modèles, malgré quelques cas isolés de grande erreur.

4.5 Identification et visualisation des appareils congestionnés

L'identification des congestions est basée sur une règle métier qui prédit les cas où les équipements atteignent leur capacité maximale. Lorsque le pourcentage d'utilisation prédit dépasse 80 %, l'équipement est marqué comme congestionné. Notre solution identifie ainsi quotidiennement les équipements potentiellement congestionnés. Les résultats obtenus pour chaque journée ont été soigneusement conservés et serviront à alimenter le tableau de bord conçu à cet effet.

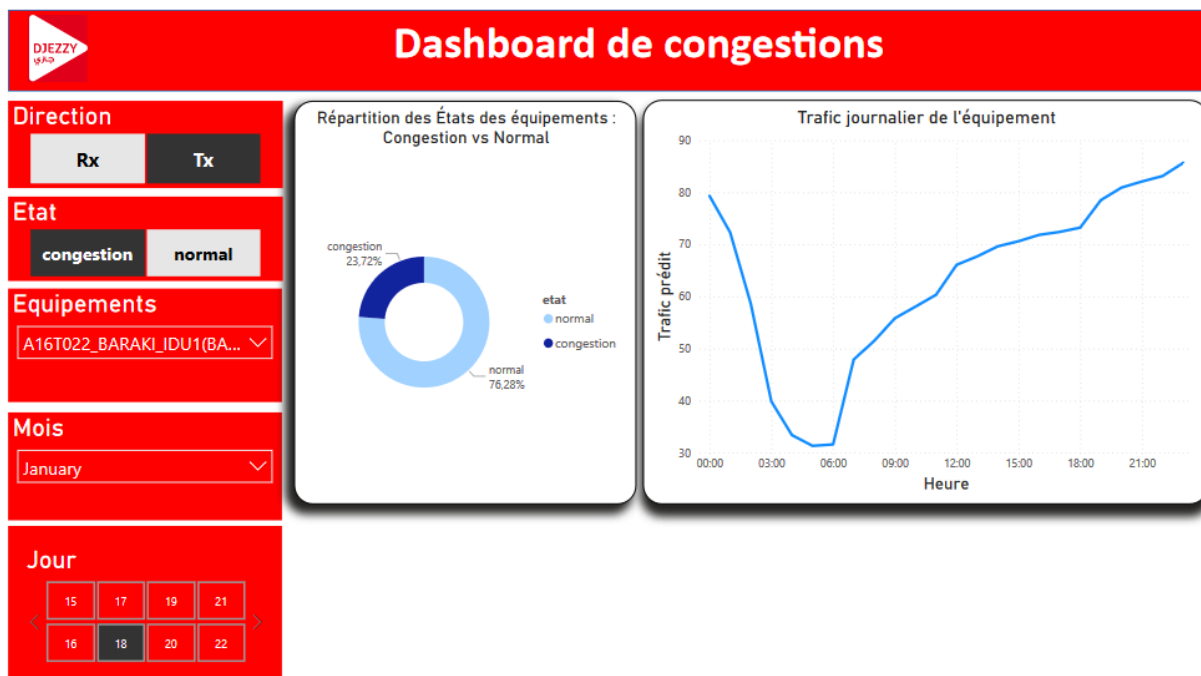


FIGURE 4.13 – Tableau de bord de prédiction des congestions

La figure ci-dessus montre le tableau de bord que nous avons conçu. Il permet de

visualiser les congestions des équipements réseau de manière interactive et intuitive. Il offre plusieurs options de filtres et des graphiques pour analyser l'état des équipements. Le filtre "Direction" permet de choisir entre la réception ("Rx") et la transmission ("Tx"), affichant ainsi le pourcentage d'utilisation maximale prédit pour l'une ou l'autre direction. Quant au filtre "État", il permet de sélectionner les équipements en fonction de leur statut, soit normaux (non sujets à la congestion), soit congestionnés. Une fois ces deux filtres appliqués, la liste des équipements se met à jour pour n'afficher que ceux correspondant aux critères choisis, comme illustré dans l'exemple où nous avons sélectionné Tx et congestion, affichant uniquement les équipements congestionnés en transmission.

Pour ce qui est du filtre "Jour", il sert à sélectionner la journée pour laquelle on souhaite afficher le trafic prédit, visible dans le graphe "Trafic journalier de l'équipement". Bien que le filtre "Mois" soit également disponible, il ne présente ici que le mois de janvier. Le diagramme en anneau fournit une représentation visuelle de la répartition des équipements entre ceux qui sont congestionnés et ceux qui fonctionnent normalement, illustrant le pourcentage d'équipements affectés par la congestion.

4.6 Conclusion

Dans ce chapitre, nous avons exposé les différentes étapes d'expérimentation mises en œuvre pour le développement et l'évaluation des modèles de prédiction du trafic réseau. Nous avons commencé par décrire le dataset utilisé, en expliquant les transformations et préparations nécessaires à son exploitation optimale. Ensuite, nous avons détaillé l'environnement logiciel et matériel qui a permis de mener à bien cette expérimentation.

La construction des modèles, basée sur une approche rigoureuse et adaptée à la nature des données, a permis de prédire le trafic réseau avec précision. Après avoir obtenu ces prédictions, nous avons appliqué une règle métier pour identifier les congestions, en sélectionnant les équipements dont le taux d'utilisation dépasse un certain seuil. Enfin, la conception d'un tableau de bord interactif marque une étape importante pour la visualisation et le suivi des équipements susceptibles d'être congestionnés.

Conclusion générale

Ce travail s'inscrit dans un contexte où la gestion proactive de la congestion des réseaux de télécommunications est devenue essentielle pour maintenir une qualité de service optimale. Les équipements réseau sont confrontés à des risques croissants de saturation, ce qui compromet les performances du réseau et la satisfaction des utilisateurs. Notre travail visait à développer une solution prédictive capable d'anticiper les congestions, en s'appuyant sur des techniques d'apprentissage automatique.

Notre approche s'est appuyée sur une analyse approfondie de la littérature scientifique pour comprendre les différentes méthodes utilisées dans la prédiction du trafic et des congestions. Bien que l'algorithme que nous avons finalement implémenté ne figurait pas parmi les techniques étudiées dans l'état de l'art, cela nous a permis de définir un cadre méthodologique solide pour la conception de notre solution.

L'algorithme CatBoost, choisi pour sa capacité à gérer efficacement les données catégorielles et à éviter le surapprentissage, a été au cœur de notre solution, garantissant une bonne précision dans les prédictions. Notre méthodologie rigoureuse a inclus la préparation des données, l'entraînement des modèles de prédiction de trafic suivi par la détection des équipements susceptibles d'être congestionnés. De plus, nous avons conçu un tableau de bord interactif pour visualiser les résultats et faciliter la prise de décision.

L'une des principales difficultés rencontrées dans la réalisation de ce travail a été le manque de références similaires pour notre travail, la majorité des recherches actuelles étant axées sur la prédiction de congestions dans le domaine de la circulation routière.

En conclusion, ce projet a été l'occasion de mettre en œuvre les compétences théoriques acquises tout au long de notre formation, tout en approfondissant nos connaissances pratiques en apprentissage automatique. Nous avons ainsi contribué à anticiper et à gérer les congestions réseau de manière proactive, en proposant un outil performant adapté aux besoins de Djezzy.

Pour l'avenir, notre travail ouvre des perspectives intéressantes :

- Une approche envisageable consisterait à regrouper les équipements selon leurs comportements afin de créer un nombre réduit de modèles d'apprentissage. Il devient possible de simplifier la gestion des prédictions tout en conservant une grande précision. Cette approche, plus facilement gérable qu'un modèle unique par équipement, permet d'optimiser les ressources tout en offrant une meilleure scalabilité pour l'application des modèles à un large ensemble d'équipements.
- Il serait également intéressant d'entraîner les modèles sur une période d'un an, étant donné que notre contexte implique des séries temporelles avec une forte saisonnalité. Cela permettrait de capturer un large éventail de caractéristiques, notamment les variations quotidiennes, hebdomadaires et mensuelles.

Bibliographie

- [1] Bouguen, Y., Hardouin, É., Wolff, F.-X., Pujolle, G. (2012). LTE et les réseaux 4G. Eyrolles.
- [2] Demoulin, C., Droogenbroeck, M. (2004). Principes de base du fonctionnement du réseau GSM. Revue de l'AIM, 4.
- [3] Bouchentouf, H. Boudghene Stambouli, R. (2013). Etude Des Performances Des Réseaux 4g (lte) [Mémoire de Master, Université Abou Bekr Belkaid - Tlemcen].
- [4] Flaunay. (2011, août 22). Evolution de la 1G vers la 4G | Frédéric Launay.
- [5] Salem, M. (2023, 14 novembre). LTE (4G) Network Architecture. Mobile Packet Core.
- [6] Adda Abbou, A. (2020). La Voix Sur Lte [Mémoire de Master, Université Saad Dahleb - Blida].
- [7] Team, I. R. (s. d.). A Guide To Network Congestion : Causes and Solutions I IR.
- [8] Marr, B. (2015). Key Performance Indicators for Dummies. John Wiley Sons.
- [9] Hamilton, J. D. (2020). Time Series Analysis. Princeton University Press.
- [10] Hyndman, R. J., Athanasopoulos, G. (2018). Forecasting : principles and practice. OTexts.
- [11] Borges, H., et al. (2021). Anomaly Detection in Time Series. Transactions on Large-Scale Data- and Knowledge-Centered Systems L : 46-62.
- [12] Montgomery, D. C., Jennings, C. L., Kulahci, M. (2015). Introduction to time series analysis and forecasting. John Wiley Sons.
- [13] Shumway, R. H., Stoffer, D. S. (2017). Time Series Analysis and Its Applications. Dans *Springer Texts in Statistics*.
- [14] Crespo Cuaresma, J., Hlouskova, J., Kossmeier, S., Obersteiner, M. (2004). Forecasting Electricity Spot-Prices Using Linear Univariate Time-Series Models. *Applied Energy*, 77, 87–106.

- [15] Plaud, A., Nguifo, E. M., Charreyron, J. (2019, juillet). Classification des séries temporelles multivariées par l'usage de Mgrams. *CAP 2019*.
- [16] Alsharef, A., Aggarwal, K., Sonia, Kumar, M., Mishra, A. (2022). Review of ML and AutoML Solutions to Forecast Time-Series Data. *Archives of Computational Methods in Engineering*, 29(7), 5297-5311.
- [17] Idrees, S., Alam, A., Agarwal, P. (2019). A Prediction Approach for Stock Market Volatility Based on Time Series Data. *IEEE Access*, PP, 1-1.
- [18] Box, G. E. P., Jenkins, G. M., Reinsel, G. C. (1994). *Time series analysis : Forecasting and control (Third edition)*. Prentice Hall.
- [19] Jiang, H., Li, Q., Jiang, Y., Shen, G., Sinnott, R., Tian, C., Xu, M. (2020). When Machine Learning Meets Congestion Control : A Survey and Comparison.
- [20] Andreatti, D., Troia, S., Musumeci, F., Giordano, S., Maier, G., Tornatore, M. (2019). Network Traffic Prediction based on Diffusion Convolutional Recurrent Neural Networks.
- [21] Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210 229. *IBM Journal of Research and Development*.
- [22] Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*.
- [23] « What Is Machine Learning? » s. d. Talend - A Leader in Data Integration Data Integrity.
- [24] Daoun, D., Ibnat, F., Alom, Z., Aung, Z., Azim, M. (2021). Reinforcement Learning : A Friendly Introduction. In *Proceedings of the International Conference on Machine Learning and Applications* (pp. 134–146). Springer.
- [25] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep Learning. *Nature*, 521, 436 444.
- [26] Torres, P., Marques, H., Marques, P., Rodriguez, J. (2018). Using Deep Neural Networks for Forecasting Cell Congestion on LTE Networks : A Simple Approach (p. 276 286).
- [27] Markus, E., Okereke, O., Agee, J. (2011). Predicting Telephone Traffic Congestion Using Multi Layer Feedforward Neural Networks. *Advanced Materials Research*, 367, 191 198.
- [28] Chmieliauskas, D., Gursnys, D. (2019). LTE Cell Traffic Grow and Congestion Forecasting (p. 5).
- [29] Do, Q. H., Doan, T., Nguyen, T., Duong, N., Linh, V. (2020). Prediction of Data Traffic in Telecom Networks based on Deep Neural Networks. *Journal of Computer Science*, 16, 1268 1277.

BIBLIOGRAPHIE

- [30] Jaffry, S. (2020). Cellular Traffic Prediction with Recurrent Neural Network.
- [31] Wa Nkongolo, M. (2023). Using ARIMA to Predict the Growth in the Subscriber Data Usage. *Engineering*, 4, 92.
- [32] Ostroumova, L., Gusev, G., Vorobev, A., Dorogush, A. V., Gulin, A. (2017, juin 28). CatBoost : Unbiased boosting with categorical features. *Neural Information Processing Systems*.
- [33] Dorogush, A. V., Ershov, V., Gulin, A. (2018). CatBoost : Gradient boosting with categorical features support (arXiv :1810.11363). *arXiv*.
- [34] Hancock, J. T., Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, 7(1), 28.
- [35] Anghel, A., Papandreou, N., Parnell, T., Palma, A., Pozidis, H. (2018, septembre 12). Benchmarking and Optimization of Gradient Boosting Decision Tree Algorithms.
- [36] Abdullah, M., Said, S. (2023). Performance Evaluation of Machine Learning Regression Models for Rainfall Prediction.
- [37] Kolli, A. (2024, février 13). Understanding CatBoost : The Gradient Boosting Algorithm for Categorical Data. *Medium*.
- [38] CatBoost Parameters and Hyperparameters. (2023, octobre 6). *GeeksforGeeks*.
- [39] Fast Gradient Boosting with CatBoost. (s. d.). *KDnuggets*.
- [40] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., Team, J. D. (2016). Jupyter Notebooks—A publishing format for reproducible computational workflows. *International Conference on Electronic Publishing*.
- [41] L, B. (2024, juin 24). Python : Tout savoir sur le principal langage Big Data et Machine Learning. *LEBIGDATA.FR*.
- [42] McKinney, W. (2017). *Python for Data Analysis : Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, Incorporated.
- [43] CatBoost—Yandex Technologies. (s. d.). *CatBoost — Yandex Technologies*.
- [44] os—Miscellaneous operating system interfaces. (s. d.). *Python Documentation*.

Résumé

À mesure que les réseaux de télécommunications évoluent et que la demande de services explose, les risques de congestion augmentent, menaçant la qualité du service et l'expérience utilisateur. Anticiper ces congestions est devenu un défi majeur pour les opérateurs. Ce mémoire propose une approche basée sur l'algorithme CatBoost pour prédire les congestions avant qu'elles ne surviennent.

En utilisant les données de trafic réseau d'Optimum Telecom Algérie "Djezzy", notre modèle permet d'identifier de manière proactive les équipements susceptibles de subir une surcharge. Cette solution vise à améliorer la gestion du réseau en offrant la possibilité de prendre des mesures préventives pour éviter la congestion. Les résultats sont présentés via un tableau de bord interactif, facilitant ainsi l'analyse et la prise de décision.

Mots clés : Réseau de téléphonie mobile, congestion, apprentissage automatique, prédiction, CatBoost

Abstract

As telecommunications networks evolve and the demand for services explodes, the risks of congestion increase, threatening both service quality and user experience. Anticipating these congestions has become a major challenge for operators. This thesis proposes an approach based on the CatBoost algorithm to predict congestion before it occurs.

By using network traffic data from Optimum Telecom Algeria "Djezzy", our model allows the proactive identification of equipment likely to experience overload. This solution aims to improve network management by providing the ability to take preventive measures to avoid congestion. The results are presented via an interactive dashboard, thus facilitating analysis and decision-making.

Keywords : Mobile network, congestion, machine learning, prediction, CatBoost