

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



ⵜⴰⵎⴰⵎⴰⵔⵜ ⵏ ⵉⵎⵎⴰⵏⵏ ⵏ ⵉⵔⵉⵎⴰⵏⵏ
جامعة بجاية
Université de Béjaïa

Université de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

MÉMOIRE DE MASTER

En Informatique

Option : *Administration et sécurité des réseaux*

Thème

Contrôle des pannes dans les réseaux
cellulaires avec l'intelligence artificielle

Présenté par :

OUBELAID Yasmine
MESSAOUDI Noura

Soutenu le 29/06/2025 devant le jury composé de :

Présidente
Promotrice
Examinatrice
Examinateur
Examinateur

ZIDANI Ferroudja
OUYAHIA Samira
HOUCINI Kenza
KHENNOUS Lachemi
TOUAZI Djoudi

Université de Béjaia
Université de Béjaia
Université de Béjaia
Université de Béjaia
Université de Béjaia

2024-2025

✧ Remerciements ✧

Au terme de cette aventure académique, nous souhaitons exprimer notre profonde reconnaissance envers toutes les personnes qui ont généreusement apporté leur soutien et leur contribution à la réalisation de ce travail.

En premier lieu, nous exprimons notre gratitude la plus sincère à l'égard de la Divinité suprême pour avoir béni notre parcours avec santé, détermination et persévérance, qui ont été les fondements de notre réussite.

*Nous tenons tout d'abord à adresser nos remerciements les plus chaleureux à notre encadrante, **Mme OUYAHIA Samira**, dont la sollicitude et les conseils avisés ont grandement enrichi notre travail. Sa disponibilité infailible et son écoute attentive ont été d'une valeur inestimable, éclairant nos réflexions et nous guidant dans la bonne direction.*

Nous tenons à exprimer notre profonde gratitude envers les membres du jury, qui ont accepté avec honneur d'évaluer notre travail. Leurs compétences, leur expertise ainsi que leurs évaluations objectives ont grandement contribué à la rigueur et à la qualité de ce projet.

Nous tenons également à remercier nos parents pour leur soutien indéfectible, leur amour et leur confiance en nous. Leur présence a été un véritable moteur tout au long de notre parcours universitaire.

Nous souhaitons également adresser nos remerciements à nos frères et sœurs pour leur soutien moral et leur présence dans les moments difficiles, ainsi qu'à l'ensemble de notre famille pour leur soutien inconditionnel.

✧ *Dédicaces* ✧

*Je dédie ce travail avec tout mon cœur,
A mes parents bien-aimés Abdenour et Noria,
Il n'existe pas de mots assez forts pour exprimer ma gratitude
envers vous. Vous êtes les piliers sur lesquels j'ai toujours pu
m'appuyer, ma source inépuisable d'amour et de courage. Merci
infiniment pour vos sacrifices silencieux, pour vos nuits
d'inquiétude, vos paroles tendres en or. Votre amour ce qui m'a le
plus porté dans ce long parcours. Je vous dois plus que ce travail,
je vous dois tout.*

*A mes frères Rabah et Yani,
Vous avez été mes premiers amis, mes protecteurs. Vos rires ont
été ma thérapie, vos silences un soutien discret mais profond.
Vous avez su apaiser mes moments de fatigue et me rappeler que
je ne suis jamais seule. Merci pour ces petits gestes qui, sans
bruit, veulent dire beaucoup.*

*A ma binôme Noura,
Merci au fond du cœur pour ton sérieux et ta disponibilité tout au
long de cette aventure. Dans cette aventure semée de doutes, de
longues heures de travail et parfois de découragement, tu as été un
véritable partenaire, motivé et toujours à l'écoute.*

*A toutes les personnes qui occupent une place précieuse dans ma
vie, qui ont su, par leur douceur et leur écoute, rendre les jours
plus légers et les nuits moins longues.
Merci énormément.*

✧ *Dédicaces* ✧

À tous ceux qui ont traversé les méandres de ma vie, Je désire consacrer ce projet de fin d'études à toutes les âmes ayant joué un rôle remarquable dans ma trajectoire académique et personnelle.

Votre appui indéfectible, vos encouragements sincères et votre amour inconditionnel ont été les leviers fondamentaux de ma réussite.

À mes géniteurs, vous êtes les piliers inébranlables, les éclaireurs avisés et mes plus fervents partisans. Je vous dédie ces mots sincères. À toi, cher père Boualem, incarnation de la force incomparable, et à toi, tendre mère Taous, fontaine de douceur inépuisable. À mes sœurs Karima, Wassila et Hana, je vous adresse mes pensées les plus affectueuses.

À ma binôme Yasmine, je tiens à exprimer toute ma reconnaissance pour son implication, sa collaboration sincère et son soutien tout au long de ce projet. Ce travail n'aurait pas eu la même richesse sans sa présence à mes côtés.

Sans oublier mes grands-mères, mes grands-pères, ainsi que mes tantes et mes oncles, du côté paternel comme maternel, dont l'affection et le soutien ont été précieux tout au long de mon parcours.

À mes amis, je vous rends un hommage tout particulier. Votre présence dans ma vie a été une véritable bénédiction.

MESSAOUDI Noura

Table des matières

Table des matières	i
Liste des figures	vi
Liste des tableaux	viii
Liste des abréviations	ix
Introduction Générale	1
I Fondamentaux des réseaux cellulaires	3
I.1 Introduction	3
I.2 Architecture générale des réseaux cellulaires	3
I.2.1 La partie radio (RAN)	4
I.2.2 Le cœur du réseau (Core Network)	4
I.2.3 Les équipements utilisateurs ou mobiles (ME)	5
I.3 Types de réseaux cellulaires (1G, 2G, 3G, 4G, 5G)	5
I.3.1 1G – AMPS	5
I.3.2 2G – GSM	6
I.3.3 3G – UMTS	7
I.3.4 4G – LTE	8
I.3.5 5G – NR	8
I.4 Fonctionnement des éléments du réseau cellulaire	9
I.5 Types de services offerts par le réseau cellulaire	11
I.6 Types de pannes dans les réseaux cellulaires	12
I.6.1 Pannes matérielles (hardware failures)	12
I.6.2 Pannes logicielles (software faults)	13
I.6.3 Anomalies radio (radio anomalies)	13
I.6.4 Surcharge du réseau (network congestion)	13
I.6.5 Problèmes de mobilité (handover issues)	13

I.6.6	Pannes dans le cœur de réseau (core failures)	14
I.6.7	Pannes dues à la sécurité (security-related failures)	14
I.6.8	Anomalies liées aux utilisateurs	14
I.7	Les indicateurs de performane (KPI)	14
I.7.1	Taux de succès d'établissement d'appel (CSSR)	15
I.7.2	Taux de coupure d'appel (CDR)	15
I.7.3	Taux de réussite de la transmission de données (Data Success Rate)	15
I.7.4	Débit (Throughput)	15
I.7.5	Latence	15
I.7.6	Qualité de signal (RSRP, SINR, etc.)	16
I.8	Importance de la prédiction des pannes	16
I.9	Conclusion	17
 II Introduction à l'Intelligence Artificielle		18
II.1	Introduction	18
II.2	Historique de l'intelligence artificielle	18
II.3	Définition de l'intelligence artificielle	19
II.4	L'objectif de l'intelligence artificielle	19
II.5	Les principales branches de l'intelligence artificielle	20
II.5.1	L'apprentissage automatique (Machine Learning (ML))	20
II.5.1.1	Définition de l'apprentissage automatique	20
II.5.1.2	Les types de modèles ML	21
II.5.2	L'apprentissage profond (Deep Learning (DL))	24
II.5.2.1	Définition de l'apprentissage profond	24
II.5.2.2	Les types de Modèles DL	24
II.5.2.2.1	L'apprentissage supervisé :	25
II.5.2.2.2	L'apprentissage non supervisé :	29
II.5.2.2.3	L'apprentissage hybride	31
II.6	Conclusion	32
 III Revue des travaux existants		33
III.1	Introduction	33
III.2	Les méthodes classiques de prédiction	33
III.2.1	Modèles statistiques traditionnels	34
III.2.2	Exemples de travaux utilisant les méthodes classiques	34
III.2.3	Limites	35
III.3	L'émergence de l'intelligence artificielle dans la détection des pannes	35
III.4	L'impact de l'IA sur les réseaux cellulaires	35

III.5 Travaux utilisant des techniques de l'IA	36
III.5.1 Machine Learning	36
III.5.2 Deep Learning	37
III.5.3 Méthodes hybrides	37
III.6 Travaux connexes	37
III.6.1 Article 01	38
III.6.2 Article 02	38
III.6.3 Article 03	39
III.6.4 Article 04	39
III.6.5 Article 05	40
III.6.6 Article 06	40
III.6.7 Article 07	41
III.6.8 Article 08	41
III.6.9 Article 09	42
III.7 Tableau comparatif des articles	42
III.8 Limites	44
III.9 Conclusion	45
IV Conception du modèle prédictif	46
IV.1 Introduction	46
IV.2 Types de données utilisées pour la détection et la prédiction des pannes . . .	46
IV.2.1 Données de performance réseau (KPIs)	46
IV.2.2 Logs système	47
IV.2.3 Données d'alarme	47
IV.2.4 Données historiques des pannes	48
IV.3 Étapes de conception du modèle	48
IV.3.1 Chargement des données	48
IV.3.2 Prétraitement des données	49
IV.3.2.1 Nettoyage des données	49
IV.3.2.2 Transformation des données	49
IV.3.2.3 Mise à l'échelle (standardisation ou normalisation)	50
IV.3.2.4 Réduction de dimensions	50
IV.3.2.5 Détection d'anomalies	50
IV.3.3 Choix de l'approche	50
IV.3.4 Évaluation du modèle	52
IV.3.4.1 Matrice de confusion (Confusion matrix)	52
IV.3.4.2 Rappel (Recall)	53
IV.3.4.3 Précision	53

IV.3.4.4	F-measure	53
IV.3.4.5	Le taux d'erreur moyen (AER)	53
IV.3.4.6	Les courbes ROC	54
IV.4	Conclusion	55
V	Implémentation et expérimentation	56
V.1	Introduction	56
V.2	Description des datasets utilisés	56
V.2.1	4G LTE Speed Dataset and Bandwidth	57
V.2.2	5G-NWDAF-Data	58
V.3	Choix des algorithmes utilisés	58
V.3.1	4G LTE data set	58
V.3.2	5G-NWDAF-Data	60
V.4	Résultats d'entraînement et interprétation	61
V.4.1	4G LTE	61
V.4.1.1	Les graphes (non-supervisé)	61
V.4.1.2	Les matrices de confusion	62
V.4.1.3	Les courbes Receiver Operating Characteristic (ROC)	64
V.4.1.4	Mesures de performance	66
V.4.2	5G nwdaf_data	67
V.4.2.1	Matrices de confusion	67
V.4.2.2	Les courbes ROC	70
V.4.2.3	Mesures de performance	72
V.5	Limites	74
V.6	Conclusion	75
	Conclusion générale	76
	Annexes	85
A	Environnement de développement	85
A.1	Jupyter	85
A.2	Google Colaboratory	85
A.3	Langage utilisé : Python	85
B	Bibliothèques utilisées	86
B.1	pandas	86
B.2	Matplotlib	87
B.3	Seaborn	87
B.4	NumPy	88

B.5	PyTorch	88
B.6	TensorFlow	89
B.7	Keras	89
B.8	Scikit-Learn	90
Bibliographie		91

Table des figures

I.1	Organisation cellulaire	3
I.2	Schéma représentatif de l'architecture du réseau cellulaire [61]	4
I.3	Schéma représentatif de l'évolution des réseaux cellulaires [62]	5
I.4	Schéma représentatif du réseau GSM	6
I.5	Schéma représentatif du réseau Universal Mobile Telecommunications (UMTS) [63]	7
I.6	Schéma représentatif du réseau Long Term Evolution (LTE) [64]	8
I.7	Schéma représentatif du réseau New Radio (NR) [65]	9
II.1	Diagramme de la relation entre Intelligence Artificielle (IA), Machine Learning et Deep Learning	20
II.2	Différents types de techniques d'apprentissage automatique.	21
II.3	Représentation d'un apprentissage supervisé	22
II.4	Représentation d'un apprentissage non supervisé	22
II.5	Représentation d'un apprentissage semi-supervisé	23
II.6	Représentation d'un apprentissage par renforcement	24
II.7	Les techniques d'apprentissage profond.	25
II.8	Architecture du Multi-layer perceptron (MLP)	26
II.9	Un exemple de réseau de neurones conversationnel CNN	27
II.10	La structure de Recurrent Neural Networks (RNN)	27
II.11	La structure interne d'un Long Short Term Memory (LSTM)	28
II.12	Structure de base d'une cellule Gated Recurrent Unit (GRU)	29
II.13	Structure schématique d'un réseau antagoniste génératif (GAN standard)	30
II.14	Structure d'un auto-encodeur (AE)	31
IV.1	Schéma représentatif du processus de conception d'un modèle rédictif	54
V.1	Schéma représentatif du déséquilibre de "Panne"	59
V.2	Schéma représentatif du déséquilibre de "has_anomaly"	60
V.3	OneClass SVM	62

V.4	Isolation Forest	62
V.5	Matrices de confusion obtenues sur le dataset 4G LTE	63
V.6	Courbe ROC des modèles entraînés sur le dataset 4G LTE	65
V.7	Matrices de confusion obtenues sur le dataset nwdaf_data	69
V.8	Courbe ROC des modèles entraînés sur le dataset NWDAF_Data	71
A.1	Logo de Jupyter [58].	85
A.2	Logo de Google Colab [58].	85
B.1	logo de Pandas [58].	86
B.2	logo de Matplotlib [58].	87
B.3	logo de SeaBorn [58].	87
B.4	logo de NumPy [58].	88
B.5	logo de PyTorch [58].	88
B.6	logo de Keras [58].	89
B.7	logo de Keras [58].	89
B.8	logo de Scikit-Learn [58].	90

Liste des tableaux

III.1 Métriques de performance dans les réseaux cellulaires	43
IV.1 Matrice de confusion pour une classification binaire	52
V.1 Résultats d'entraînement des modèles supervisés	66
V.2 Résultats d'entraînement des modèles supervisés	73

Liste des abréviations

1G Première Génération.

2G Deuxième Génération.

3G Troisième Génération.

4G Quatrième Génération.

5G Cinquième Génération.

ACP Analyse en Composantes Principales.

AE Auto Encoder.

AER Average Error Rate.

AF Application Function.

AlexNeT Alex Krizhevsky Neural Network.

AMF Access and Mobility Management Fuction.

ANN Artificial Neural Network.

AR Augmented Reality.

ARIMA Auto Regressive Integrated Moving Average.

AUC Area Under the Curve.

AuC Authentification Center.

AuSF Authentification Server Function.

BERT Bidirectional Encoder Representations from Transformers.

BiLSTM Bidirectional Long Short-Term Memory.

BSC Base Station Controller.

BSS Base Station Subsystem.

BTS Base Station Tranceiver.

CatBoost Categorical Boosting.

CDR Call Drop Rate.

CNN Convolutional Neural Networks.

CPU Central Processing Unit.

CQI Channel Quality Indicator.

CSSR CSSR : Call Setup Success Rate.

CSV Comma-Separated Values.

CTMC Continuous-Time Markov Chain.

DBN Deep Belief Network.

DL Deep Learning.

DNN Deep Neural Network.

DNS Domaine Name Server.

Dos Deni of Service.

DT Decision Tree.

e-UTRAN Evolved Universal Terrestrial Radio Access Network.

EIR Equipment Identity Register.

eMBB enhanced Mobile Broadband.

eNodeB Enhanced Node B.

EPC Evolved Packet Core.

f1 f1-score.

FN False Negatives.

FP False Positives.

FPR False Positive Rate.

GAN Generative Adversarial Network.

GBM Gradient Boosting Machine.

GCN Graph Convolutional Network.

GGSN Gateway GPRS Support Node.

GMSC Gateway MSC.

gNodeB next Generation Node B.

GPT Generative Pre-trained Transformer.

- GPU** Graphics Processing Unit.
- GRU** Gated Recurrent Unit.
- GSM** Global System for Mobile communications.
- HLR** Home Location Register.
- HSS** Home Subscriber Server.
- IA** Intelligence Artificielle.
- IMS** IP Multimedia Subsystem.
- Inria** Institut national de recherche en informatique et en automatique.
- IoT** Internet of Things.
- IP** Internet Protocol.
- KNN** K-Nearest Neighbors.
- KPI** Key Performance Indicators.
- LBS** Location Based Services.
- LightGBM** Light Gradient Boosting Machine.
- LR** Logistic Regression.
- LSTM** Long Short Term Memory.
- LTE** Long Term Evolution.
- Mbps** mégabits par seconde.
- MDT** Minimization of Drive Tests.
- ML** Machine Learning.
- MLP** Multi-layer perceptron.
- MME** Mobility Management Entity.
- MMS** Multimedia Messaging Service.
- mMTC** massive Machine-Type Communications.
- MONROE** Measuring Mobile Broadband Networks in Europe.
- MSC** Mobile Switching Center.
- NB** Naive Bayes.
- NBM** Naive Bayes Model.
- NFV** Network Functions Virtualization.

NLP Natural Language Processing.

NR New Radio.

NRxRSRP Neighbor Radio Received Signal Received Quality.

NSS Network SubSystem.

NSSF Network Slice Selection Function.

NWDAF NetWork Data Analytics Function.

PCA Principal Component Analysis.

PCF Policy Control Function.

PCRF Policy and Charging Rules Function.

PDN Packet Data Network.

PGW Packet GateWay.

PSTN Public Switched Telephone Network.

QoS Quality of Service.

RAM Random Access Memory.

RAN Radio Access Network.

ResNet Residual Network.

RF Random Forest.

RNC Radio Network Controller.

RNN Recurrent Neural Networks.

ROC Receiver Operating Characteristic.

RSRP Reference Signal Received Power.

RSRQ Reference Signal Received Quality.

RSSI Received Signal Strength Indicator.

SARSA State-Action-Reward-State-Action.

SBA Service-Based Architecture.

SDN Software Defined Networking.

SGSN Serving GPRS Support Node.

SGW Serving GateWay.

SHAP SHapley Additive exPlanations.

SIM Subscriber Identity Module.

SINR Signal to Interference plus Noise Ratio.

SMF Session Management Function.

SMOTE Synthetic Minority Over-sampling Technique.

SMS Short Message Service.

SNR Signal-to-Noise Ratio.

SS7 Signaling System No.7.

st-GCN Spatio-Temporal Graph Convolutional Network.

SVM Support Vector Machine.

SVR Support Vector Regression.

TN True Negatives.

TP True Positives).

TPR True Positive Rate.

TPU Tensor Processing Unit.

UDM Unified Data Management.

UE User Equipment.

UMTS Universal Mobile Telecommunications.

UPF User Plane Function.

URLLC Ultra-Reliable Low Latency Communications.

USSD Unstructured Supplementary Service Data.

UTRAN UMTS Terrestrial Radio Access Network.

VGG Visual Geometry Group.

VLR Visitor Location Register.

VoIP Voice over IP.

VoLTE Voice over LTE.

VR Virtual Reality.

Xception eXtreme inception.

XGBoost eXtreme Gradient Boosting.

Introduction Générale

Dans un temps où rester connecté est devenu omniprésent, les réseaux cellulaires occupent une place importante dans notre quotidien. Que ce soit pour passer un appel, envoyer un message ou naviguer sur Internet, ces architectures assurent un accès constant aux services numériques pour des milliards d'utilisateurs à travers le monde. L'expansion rapide des technologies sans fil, et particulièrement l'arrivée de la Cinquième Génération (5G), a considérablement augmenté les besoins en matière de bande passante, de fiabilité et de performance. Dans ce contexte, assurer une qualité de service optimale représente un enjeu stratégique majeur pour les opérateurs télécoms.

Néanmoins, malgré les progrès techniques, les réseaux cellulaires restent sensibles à différents types de pannes et d'anomalies : pannes matérielles, anomalies logicielles, erreurs de configuration ou encore congestion du trafic. Même lorsqu'elles sont brèves, ces interruptions peuvent avoir des conséquences importantes : pertes économiques, insatisfaction des utilisateurs, voire risques pour la sécurité dans le cadre de communications critiques. D'où l'importance de développer des mécanismes permettant non seulement de détecter ces incidents, mais surtout de les anticiper.

L'IA offre aujourd'hui des perspectives particulièrement prometteuses dans ce domaine. Grâce à sa capacité à traiter d'énormes quantités de données, à identifier des modèles invisibles à l'œil humain et à apprendre de l'expérience passée, l'IA permet de passer d'une approche réactive à une approche préventive. Des techniques comme le machine learning ou le deep learning sont déjà utilisées dans plusieurs secteurs pour détecter des anomalies, prévoir des événements ou améliorer les performances de systèmes complexes. Ces méthodes appliquées aux réseaux cellulaires permettent d'anticiper les pannes à partir de paramètres comme les Key Performance Indicators (KPI), les journaux d'événements ou les historiques d'alarme.

Ce mémoire a pour but d'explorer comment les outils d'intelligence artificielle peuvent être exploités pour prédire les défaillances dans les réseaux cellulaires. Pour cela, la démarche suivie comporte plusieurs phases, en allant de la collecte et la préparation des données jusqu'à la conception, l'entraînement et l'évaluation de modèles prédictifs. L'accent sera mis sur la

qualité des données, la sélection des variables pertinentes et la performance des algorithmes testés, en termes de précision.

La méthodologie adoptée repose sur plusieurs étapes : une étude approfondie des réseaux cellulaires et de leurs vulnérabilités, une revue des approches existantes en matière de maintenance prédictive par IA, une phase expérimentale de modélisation et d'évaluation. L'objectif est de proposer une solution technique réaliste et d'en mesurer la pertinence dans un contexte opérationnel.

Ce mémoire est structuré de la manière suivante : le premier chapitre présente les fondements des réseaux cellulaires, en mettant l'accent sur leurs points de fragilité. Le second introduit les concepts d'intelligence artificielle ainsi que les approches méthodologiques mobilisées. Le troisième chapitre fait un état de l'art des travaux de recherche dans ce domaine. Les chapitres suivants détaillent la démarche expérimentale, de la préparation des données à la mise en œuvre des modèles, jusqu'à l'évaluation des performances. Enfin, une discussion critique sur les résultats vient conclure cette étude.

En définitive, cette recherche vise à démontrer que l'intelligence artificielle, loin d'être une simple tendance technologique, constitue un levier concret et puissant pour améliorer la fiabilité et la résilience des réseaux mobiles, dans un monde où la connectivité ne tolère plus l'imprévu.

Chapitre I

Fondamentaux des réseaux cellulaires

I.1 Introduction

Dans un monde de plus en plus connecté, les réseaux cellulaires sont devenus des infrastructures critiques. En effet, ils permettent de transmettre la voix, les messages, les données, et de connecter des milliards d'utilisateurs à travers le monde. Ces réseaux doivent fonctionner de manière fiable, stable et rapide, même en situation de forte demande.

Avant de parler de prédiction de pannes grâce à l'IA, il est évident de comprendre comment ces réseaux sont construits, comment ils fonctionnent, quels types de services ils offrent, et quelles sont les causes les plus fréquentes de défaillances. Ce chapitre pose donc les bases nécessaires à la compréhension du système global, en détaillant les composants essentiels, les évolutions technologiques, et les défis liés à leur maintenance.

I.2 Architecture générale des réseaux cellulaires

Un réseau cellulaire, comme son nom l'indique, repose sur une structure en **cellules** comme le montre la figure I.1. Une cellule est une petite zone géographique desservie par une **station de base** (antenne). Ensemble, ces cellules couvrent des villes, des régions et parfois même des pays entiers [1].

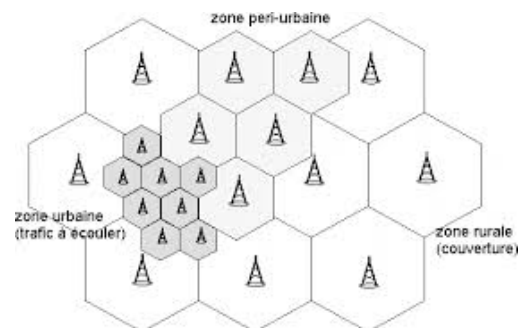


FIGURE I.1 – Organisation cellulaire

Selon [2], l'architecture du réseau cellulaire se divise en plusieurs couches principales. La figure I.2 représente l'architecture du réseau GSM, qui constitue la base historique sur laquelle se sont appuyées les générations suivantes.

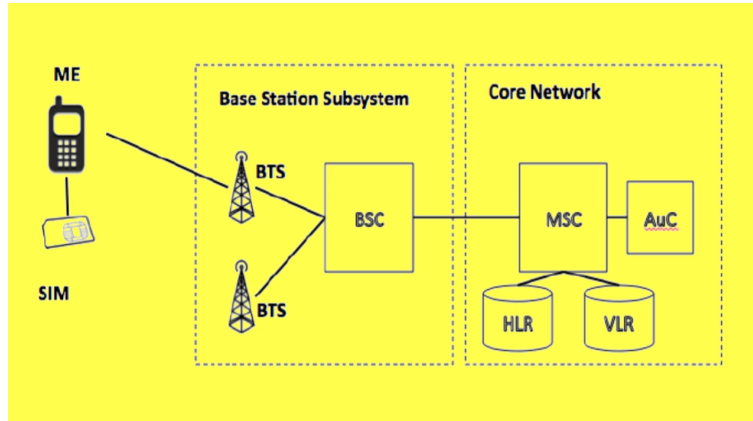


FIGURE I.2 – Schéma représentatif de l'architecture du réseau cellulaire [61]

I.2.1 La partie radio (RAN)

La partie radio, aussi appelée sous-système radio ou Radio Access Network (RAN), et sous-système des stations de base ou Base Station Subsystem (BSS) particulièrement en Deuxième Génération (2G), est l'ensemble des équipements qui assurent la communication sans fil entre les terminaux mobiles et le réseau. Elle comprend principalement les stations de base ; Base Station Transceiver (BTS) dans le Global System for Mobile communications (GSM), NodeB dans l'UMTS, Enhanced Node B (eNodeB) dans le LTE, qui gèrent la transmission et la réception des signaux radio, ainsi que les contrôleurs associés (Base Station Controller (BSC) dans le GSM, Radio Network Controller (RNC) dans l'UMTS) qui orchestrent les ressources radio et les transferts entre cellules. Cette partie permet aux utilisateurs de se connecter au réseau, de bénéficier d'une couverture optimale et de passer d'une cellule à une autre sans coupure, grâce à la gestion des handovers.

I.2.2 Le cœur du réseau (Core Network)

La partie core, ou cœur de réseau, quant à elle, regroupe l'ensemble des éléments qui assurent la gestion des appels, le routage, la mobilité et la sécurité. Dans le GSM, il s'agit du sous-système réseau (Network SubSystem (NSS)) avec le Mobile Switching Center (MSC), le Home Location Register (HLR), Visitor Location Register (VLR), le centre d'authentification (Authentication Center (AuC)) et le registre des équipements (Equipment Identity Register (EIR)). Le cœur du réseau gère l'identification et la localisation des abonnés, l'au-

thentification, la commutation des appels et l'interconnexion avec d'autres réseaux (fixes ou mobiles). Dans les réseaux plus récents comme la Quatrième Génération (4G) ou la 5G, le cœur de réseau évolue vers une architecture « full Internet Protocol (IP) » (tout IP), où la signalisation, la voix et les données transitent via des protocoles Internet, offrant ainsi plus de flexibilité et de nouvelles possibilités de services.

I.2.3 Les équipements utilisateurs ou mobiles (ME)

Ce sont les téléphones, tablettes, objets connectés qui se connectent au réseau via une carte Subscriber Identity Module (SIM).

I.3 Types de réseaux cellulaires (1G, 2G, 3G, 4G, 5G)

L'évolution des réseaux cellulaires a marqué des avancées technologiques énormes, améliorant la connectivité et les services mobiles à chaque génération. Selon [3] les générations de réseaux mobiles ont continuellement évolué pour offrir de meilleures performances et répondre aux nouvelles exigences des utilisateurs. Ces générations sont représentées dans la figure I.3. Voici les standards principaux de chaque génération :

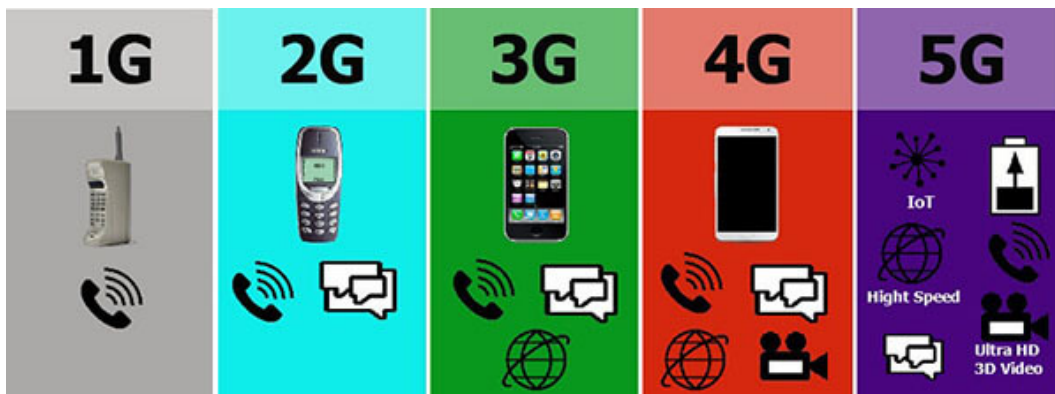


FIGURE I.3 – Schéma représentatif de l'évolution des réseaux cellulaires [62]

I.3.1 1G – AMPS

La Première Génération (1G), ou première génération de téléphonie mobile, correspond aux réseaux analogiques apparus dans les années 1980. Elle permettait uniquement la transmission de la voix via des signaux radio analogiques. Bien que cette technologie offrait une mobilité sans fil, elle présentait des limites importantes, notamment une qualité sonore faible

et une faible sécurité des communications. La 1G a néanmoins posé les bases essentielles pour les générations suivantes, qui ont introduit des services numériques plus avancés.

I.3.2 2G – GSM

La deuxième génération des réseaux cellulaires a marqué la transition des communications analogiques vers le numérique. L'architecture 2G est centrée autour de trois composantes principales : la station de base (BTS), le contrôleur de station de base (BSC) et le centre de commutation mobile (MSC). Le BTS assure la communication radio avec les mobiles, tandis que le BSC coordonne plusieurs BTS et gère le handover et l'allocation des ressources radio. Le MSC, quant à lui, s'occupe du routage des appels vocaux et de la mobilité. Des bases de données telles que le HLR et le VLR permettent la gestion des abonnés et leur localisation. L'ensemble des BTS et les BSC constituent la partie radio ou BSS, quant à la partie core ou NSS est constituée des éléments MSC, HLR, VLR, EIR et AuC.

Ce réseau était optimisé pour la voix, avec une commutation de circuits, et permettait un débit très limité pour les données. L'architecture de cette génération est représentée dans la figure I.4.

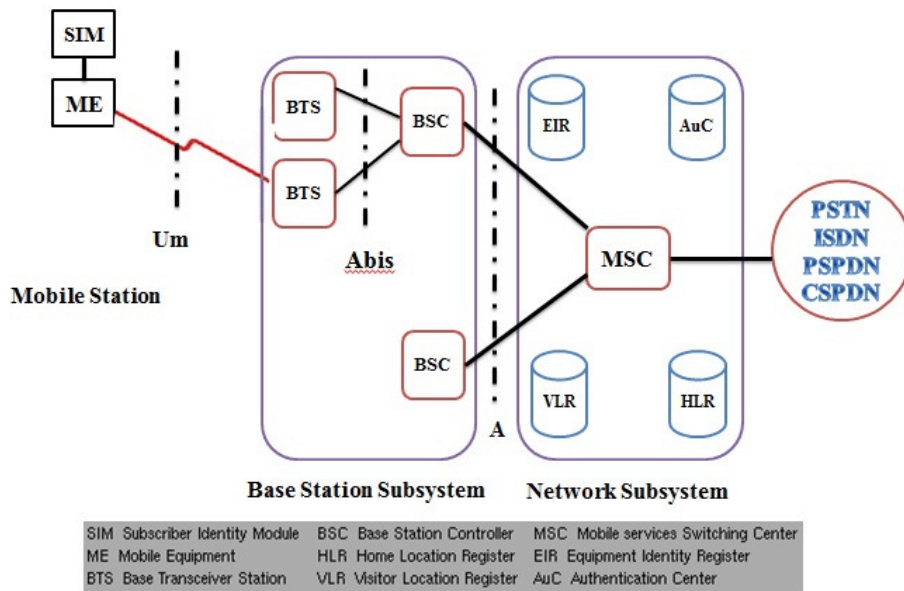


FIGURE I.4 – Schéma représentatif du réseau GSM

I.3.3 3G – UMTS

L'architecture de la Troisième Génération (3G) se compose de trois grandes parties : le réseau d'accès radio UMTS Terrestrial Radio Access Network (UTRAN), le cœur de réseau circuit (Circuit Core) et le cœur de réseau paquet (Packet Core).

L'**UTRAN** comprend l'équipement utilisateur (User Equipment (UE)), les stations NodeB (équivalentes aux antennes-relais) et les contrôleurs RNC. Les NodeB assurent la communication radio avec les terminaux, tandis que les RNC gèrent la mobilité, le contrôle de ressources radio, et la liaison avec le cœur de réseau.

Le cœur circuit **Circuit Core**, dédié aux communications vocales, comprend plusieurs éléments : le MSC, le VLR, le HLR, le Gateway MSC (GMSC) pour l'interconnexion avec d'autres réseaux (comme le Public Switched Telephone Network (PSTN) ou d'autres), ainsi que l'AuC et l'EIR pour l'authentification et la vérification des équipements.

En parallèle, le cœur paquet **Packet Core** gère le transport des données IP via le Serving GPRS Support Node (SGSN), qui s'occupe de la gestion de session et de mobilité, et le Gateway GPRS Support Node (GGSN), qui assure l'interconnexion avec les réseaux de données externes (Internet, intranet). L'ensemble de cette architecture permet au réseau UMTS de fournir à la fois des services voix et données de manière intégrée.

L'architecture est représentée dans la figure I.5.

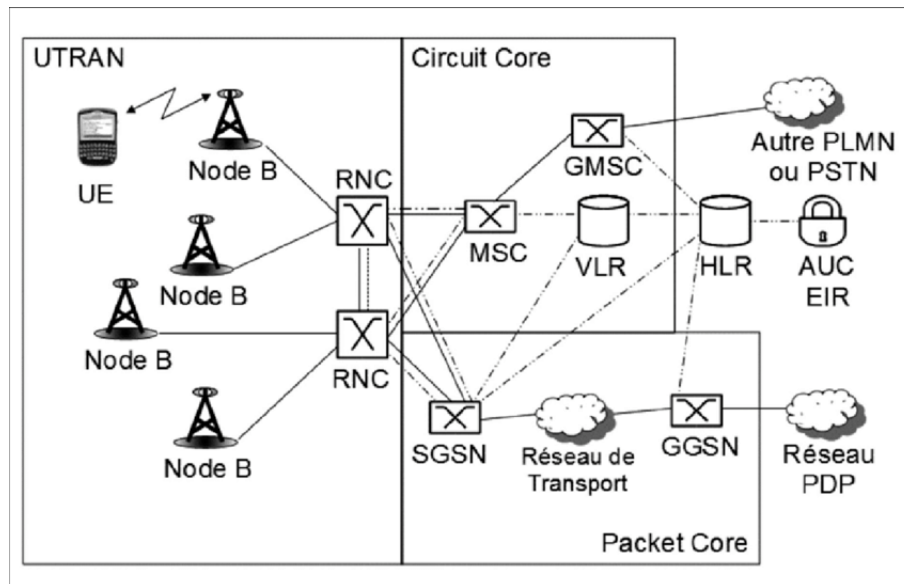


FIGURE I.5 – Schéma représentatif du réseau UMTS [63]

I.3.4 4G – LTE

L'architecture de la 4G repose principalement sur deux grands sous-systèmes : l'**Evolved Universal Terrestrial Radio Access Network (e-UTRAN)** et l'**Evolved Packet Core (EPC)**. Les équipements utilisateurs (UE) communiquent avec le réseau via les stations de base évoluées appelées eNodeB (eNB), qui assurent la connexion radio. Le cœur de réseau EPC comprend plusieurs entités : le Mobility Management Entity (MME) qui gère la mobilité et l'authentification des utilisateurs, le Serving GateWay (SGW) qui assure la gestion du trafic entre le réseau radio et le cœur de réseau, et le Packet GateWay (PGW) qui connecte le réseau 4G à l'internet et aux autres réseaux de données externes (Packet Data Network (PDN)). Le Home Subscriber Server (HSS) stocke les informations des abonnés, tandis que le Policy and Charging Rules Function (PCRF) définit les règles de gestion de la qualité de service et de facturation. Cette architecture tout-IP permet une transmission efficace des données, une meilleure gestion de la mobilité, et supporte des services à haut débit comme la vidéo en streaming, la voix sur IP (Voice over LTE (VoLTE)) et l'Internet mobile. Cette architecture est représentée dans la figure I.6.

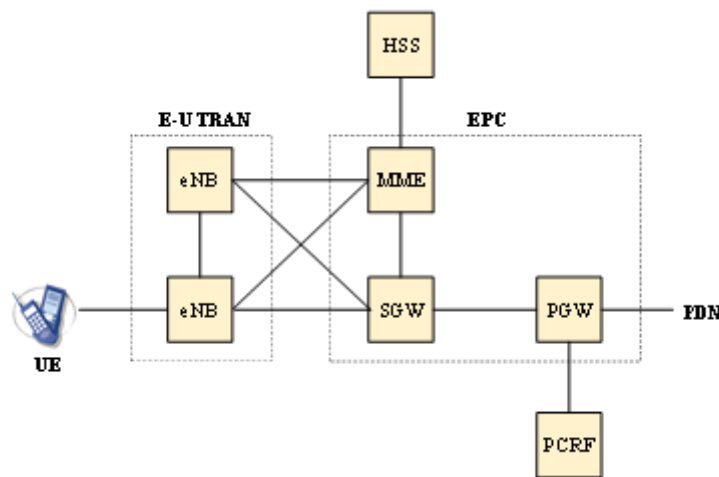


FIGURE I.6 – Schéma représentatif du réseau LTE [64]

I.3.5 5G – NR

L'architecture 5G est organisée autour de plusieurs fonctions réseau spécialisées qui communiquent entre elles pour gérer la connexion, la sécurité et le transfert de données. Le mobile 5G se connecte à une antenne, puis les fonctions comme l'Access and Mobility Management Function (AMF) pour la gestion de la mobilité, le Session Management Function (SMF) pour la gestion des sessions, et le User Plane Function (UPF) passerelle de données, prennent en charge le trafic et la gestion des utilisateurs. D'autres fonctions comme l'Authentication

Server Function (AuSF) pour l'authentification, l'Unified Data Management (UDM) pour la gestion des abonnés et la Network Slice Selection Function (NSSF) pour la sélection des tranches réseau, assurent la sécurité, l'identification et l'optimisation des ressources. Le Policy Control Function (PCF) applique les règles de contrôle, tandis que l'Application Function (AF) gère les applications. Cette architecture modulaire permet à la 5G d'être très flexible, rapide et adaptée à de nombreux usages, du haut débit à l'Internet des objets. La figure I.7 représente l'architecture de la 5G.

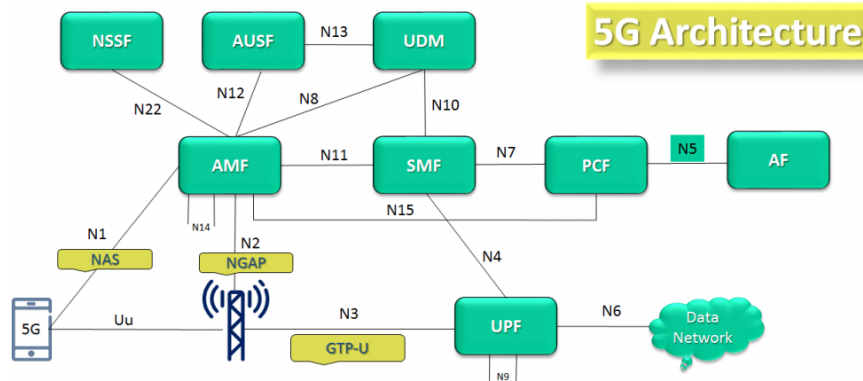


FIGURE I.7 – Schéma représentatif du réseau NR [65]

I.4 Fonctionnement des éléments du réseau cellulaire

Voici une description du fonctionnement des éléments du réseau cellulaire, en mettant en lumière les rôles des principales composantes communes aux différentes générations (2G à 5G) :

1. Station de base (BTS / Node B / eNodeB / next Generation Node B (gNodeB))

La station de base est l'élément radio du réseau. Elle établit la liaison sans fil entre le téléphone mobile et le réseau.

- ★ En **2G**, on parle de **BTS**.
- ★ En **3G**, elle devient **Node B**.
- ★ En **4G**, c'est le **eNodeB**, qui intègre aussi les fonctions de contrôle.
- ★ En **5G**, c'est le **gNodeB**, qui gère les communications radio et certaines fonctions avancées comme le beamforming (orienter le signal vers l'utilisateur).

Elle émet et reçoit les signaux radio, assure le codage/décodage, la modulation/démodulation, et gère les connexions avec les mobiles dans sa cellule [3].

2. Contrôleur radio (BSC / RNC / intégré dans eNodeB/gNodeB)

Ce composant coordonne les stations de base :

- ★ En 2G, le **BSC** gère plusieurs BTS : handover, allocation de fréquences, gestion de puissance.
- ★ En 3G, le **RNC** a un rôle similaire mais avec plus de fonctionnalités Quality of Service (QoS).
- ★ En 4G et 5G, cette logique est **intégrée directement** dans l'eNodeB ou gNodeB.

Il assure également la mobilité de l'utilisateur (changement de cellule), et optimise l'utilisation des ressources radio [3].

3. Cœur de réseau (Core Network)

C'est le centre nerveux du réseau, responsable de la **commutation**, de la **mobilité**, de la **gestion des abonnés** et du **routage des communications** vers Internet ou d'autres réseaux.

- ★ **MSC** en 2G : gère les appels vocaux.
- ★ **SGSN / GGSN** en 2G/3G (pour les données).
- ★ **EPC** en 4G : inclut **MME, SGW, PGW, PCRF**.
- ★ en 5G : virtualisé, orienté services (Service-Based Architecture (SBA)), avec de nombreuses fonctions logiques (AMF, SMF, UPF, etc.).

Fonctionnement :

- **MME / AMF** : gère la signalisation, l'enregistrement et la mobilité.
- **SGW / UPF** : transporte les données utilisateur (voix/données).
- **PGW / UPF** : connecte le réseau mobile à Internet.
- **HSS / UDM** : base de données centrale pour authentification, profils d'abonnés [3].

4. HLR/VLR – HSS/UDM

Ces bases de données gèrent l'identité des abonnés, leur localisation, et les paramètres de service.

- ★ **HLR** en 2G/3G stocke les informations d'abonné.
- ★ **VLR** gère les abonnés en itinérance dans une zone.
- ★ En 4G/5G, ces fonctions sont fusionnées dans le **HSS** ou **UDM** [3].

5. Interfaces (liaisons entre éléments) Chaque composant est relié par des interfaces normalisées :

- ★ **Um** : entre mobile et station de base.
- ★ **Abis** : entre BTS et BSC.
- ★ **Iu** (3G), **S1** (4G), **NG** (5G) : entre radio et cœur de réseau.
- ★ **Gi / N6** : sortie vers Internet [3].

6. Fonctions avancées dans les réseaux modernes (4G/5G)

- ★ **Commutation par paquets** : tout est traité comme des données, même la voix.
- ★ **QoS** : priorisation des flux (ex : vidéo, appels).
- ★ **Mobility Management** : assurer la continuité lors du déplacement de l'utilisateur.
- ★ **Sécurité** : chiffrement, authentification mutuelle (carte SIM - réseau) [3].

I.5 Types de services offerts par le réseau cellulaire

Voici une présentation des types de services assurés par les réseaux mobiles, depuis les premières générations (2G) jusqu'aux plus récentes (5G). Ces services évoluent selon les capacités techniques du réseau, la bande passante, la latence, et la densité de connexions supportées.

1. Services de base (2G - voix et messagerie)

Selon [4] les réseaux mobiles ont d'abord été conçus pour les communications vocales :

- ★ **Appels vocaux classiques** : commutation de circuits.
- ★ **Short Message Service (SMS)** : messages texte courts.
- ★ **Unstructured Supplementary Service Data (USSD)** : codes pour interagir avec les services de l'opérateur (ex : *123 pour voir son solde).

2. Services multimédias (3G)

Avec l'introduction de la commutation par paquets et des débits plus élevés, la 3G permet selon [4] :

- ★ Appels vidéo : via le protocole 3G-324M ou plus tard par Internet (Voice over IP (VoIP)).
- ★ Accès à Internet mobile : navigation web, e-mails, réseaux sociaux.
- ★ Multimedia Messaging Service (MMS) : envoi de photos, audio et vidéos.
- ★ Services de localisation Location Based Services (LBS) : géolocalisation, cartes et navigation.

3. Services IP et haut débit (4G)

La 4G, entièrement basée sur IP, permet des services plus riches [4] :

- ★ VoIP / VoLTE : appels vocaux haute qualité sur IP.
- ★ Streaming vidéo en haute définition : YouTube, Netflix, etc.
- ★ Téléchargements rapides et jeux en ligne.
- ★ Applications mobiles connectées : messagerie instantanée, cloud, réseaux sociaux.
- ★ Partage de connexion (mode modem / hotspot).

4. Services avancés (5G)

La 5G étend le champ des possibles avec des services spécialisés par type d'usage, les voici selon [5] :

- ★ **enhanced Mobile Broadband (eMBB)**
 - Réalité virtuelle (Virtual Reality (VR)) / augmentée (Augmented Reality (AR)).
 - Streaming 4K/8K, cloud gaming.
 - Expériences immersives (concerts, événements sportifs en direct).
- ★ **Ultra-Reliable Low Latency Communications (URLLC)**
 - **Applications critiques** en temps réel :
 - Médecine à distance (télésurveillance, téléchirurgie).
 - Véhicules autonomes.
 - Contrôle industriel (robotique, automatisation).
- ★ **massive Machine-Type Communications (mMTC)**
 - Internet des objets (Internet of Things (IoT)) massif : capteurs, compteurs, smart cities.
 - Domotique, agriculture connectée, surveillance environnementale.
 - Jusqu'à 1 million d'appareils/km².

I.6 Types de pannes dans les réseaux cellulaires

Voici une description structurée des types de pannes et anomalies que l'on peut rencontrer dans les réseaux cellulaires, classées selon leur nature (matérielle, logicielle, radio, ou liée au cœur de réseau) :

I.6.1 Pannes matérielles (hardware failures)

Ces anomalies concernent les **équipements physiques** du réseau, souvent dans la station de base ou le cœur de réseau :

- ★ **Panne d'alimentation** : coupure de courant ou défaillance de batteries de secours dans une station[6].
- ★ **Défaillance d'antenne** : connecteurs endommagés, mauvaise orientation, ou panne de l'émetteur/récepteur[6].
- ★ **Défaillance d'un eNodeB/gNodeB** : indisponibilité locale du service dans une cellule[7].
- ★ **Surchauffe d'un équipement** : peut entraîner un arrêt automatique ou des baisses de performance[6].

I.6.2 Pannes logicielles (software faults)

Ces pannes concernent le **logiciel** de gestion du réseau :

- ★ **Crash d'un module logiciel** : par exemple, une défaillance dans la gestion de la mobilité ou de la base de données HSS peut entraîner une interruption du service[7].
- ★ **Mauvaise configuration** : des erreurs dans les paramètres de fréquence ou d'allocation de ressources peuvent perturber le fonctionnement optimal du réseau[7].
- ★ **Bogue dans un algorithme de handover** : des erreurs dans les algorithmes de transfert entre cellules peuvent empêcher une transition fluide, entraînant des coupures de communication[7].
- ★ **Mise à jour défectueuse du firmware** : une mise à jour incorrecte du firmware d'un eNodeB/gNodeB ou d'un composant du cœur de réseau peut provoquer des dysfonctionnements[7].

I.6.3 Anomalies radio (radio anomalies)

Ces problèmes sont dus à l'environnement sans fil :

- ★ **Atténuation du signal (path loss)** : distance trop grande, murs, météo[8].
- ★ **Interférences radio** : entre cellules voisines ou d'autres équipements (Wi-Fi, micro-ondes)[9].
- ★ **Effet de fading** : affaiblissement temporaire du signal dû aux réflexions et obstacles[9].
- ★ **Pertes de synchronisation** entre mobile et réseau[7].
- ★ **Mauvaise qualité du signal (Reference Signal Received Power (RSRP), Signal to Interference plus Noise Ratio (SINR))** entraînant des coupures, lenteur ou appels interrompus[9].

I.6.4 Surcharge du réseau (network congestion)

Le réseau peut devenir temporairement indisponible même sans panne matérielle [10] :

- ★ **Trop d'utilisateurs simultanés** : dans un stade, une gare ou un concert.
- ★ **Manque de ressources radio** : le scheduler ne peut plus allouer de bande passante.
- ★ **Saturation des liens backhaul** : entre les stations de base et le cœur de réseau.

I.6.5 Problèmes de mobilité (handover issues)

Les réseaux mobiles doivent assurer une transition fluide entre cellules [11] :

- ★ **Handover raté ou trop lent** : l'appel ou la session de données est coupée.

- ★ **Ping-pong effect** : le mobile bascule rapidement entre deux cellules voisines, perturbant la qualité de service.
- ★ **Zone grise** : zone à faible couverture située entre deux cellules.

I.6.6 Pannes dans le cœur de réseau (core failures)

Problèmes touchant la gestion centrale [12] :

- ★ **Défaillance de l'HSS** : les abonnés ne peuvent pas s'authentifier.
- ★ **Panne du MME/AMF** : la gestion de la mobilité et des sessions est bloquée
- ★ **Défaillance du PGW/UPF** : perte de connectivité Internet.
- ★ **Panne de Domaine Name Server (DNS) ou de serveur IP Multimedia Subsystem (IMS)** : impact sur les services VoLTE ou services IP.

I.6.7 Pannes dues à la sécurité (security-related failures)

Selon [13], voici les plus populaires :

- ★ **Attaque par déni de service (Deni of Service (Dos))** : surcharge volontaire du réseau.
- ★ **Clonage de carte SIM** ou accès frauduleux aux ressources du réseau.
- ★ **Intrusion dans les interfaces de signalisation** (ex. Signaling System No.7 (SS7), Diameter).

I.6.8 Anomalies liées aux utilisateurs

Parfois, la cause ne vient pas du réseau mais du terminal [13] :

- ★ Téléphone mal configuré ou en mode avion.
- ★ SIM défectueuse ou expirée.
- ★ Logiciel obsolète ou instable sur le mobile.

I.7 Les indicateurs de performane (KPI)

Pour évaluer la qualité d'un réseau cellulaire, on ne se contente pas d'observer s'il fonctionne ou non. On utilise ce qu'on appelle des indicateurs de performance (ou KPI. Ces indicateurs permettent aux opérateurs et ingénieurs de mesurer **l'efficacité, la fiabilité et la qualité du service offert aux utilisateurs**. Selon [15], voici les principaux :

I.7.1 Taux de succès d'établissement d'appel (CSSR)

Le CSSR mesure la capacité du réseau à **établir un appel téléphonique avec succès**. En d'autres termes, c'est le pourcentage d'appels qui réussissent à se connecter, sans être rejetés ou coupés dès la demande. Un bon réseau doit afficher un taux proche de 100 %. Si ce chiffre chute, cela signifie que les utilisateurs ont des difficultés à passer des appels.

I.7.2 Taux de coupure d'appel (CDR)

Une fois qu'un appel est établi, il est important qu'il reste actif jusqu'à la fin. Le **taux de coupure** mesure la proportion d'appels qui sont interrompus de manière involontaire. Un taux élevé est un signe de mauvaise qualité : il peut être causé par une faible couverture, une surcharge du réseau, ou un problème technique. L'objectif des opérateurs est de maintenir ce taux aussi bas que possible.

I.7.3 Taux de réussite de la transmission de données (Data Success Rate)

Avec l'usage massif d'Internet mobile, cet indicateur est devenu crucial. Il mesure la **capacité du réseau à permettre une connexion Internet stable et rapide** pour l'utilisateur. Cela inclut l'ouverture d'une page web, le téléchargement d'un fichier, ou le lancement d'une vidéo. Un faible taux peut indiquer une surcharge ou une défaillance technique.

I.7.4 Débit (Throughput)

Le débit représente la **quantité de données** que le réseau peut transmettre à l'utilisateur sur une période donnée. Il se mesure en **mégabits par seconde (Mbps)** (mégabits par seconde). On distingue souvent :

- ★ Le **débit descendant (download)** : réception des données (vidéos, pages web),
- ★ Le **débit montant (upload)** : envoi des données (photos, vidéos, appels vidéo).

Un bon débit permet une expérience fluide pour les utilisateurs, sans lenteurs ni interruptions.

I.7.5 Latence

La latence est le **délai entre l'envoi d'une donnée et sa réception**. Elle est particulièrement importante pour les applications en temps réel comme les jeux en ligne, les appels vidéo ou les voitures connectées. Une faible latence (idéalement < 50 ms) est essentielle pour

une communication instantanée. Les réseaux 5G, par exemple, visent une latence inférieure à 1 milliseconde dans certains cas.

I.7.6 Qualité de signal (RSRP, SINR, etc.)

Ces indicateurs techniques mesurent la **force et la qualité du signal** radio entre le téléphone et l'antenne.

- ★ Le **RSRP** donne une idée de la puissance du signal reçu.
 - ★ Le **SINR** mesure le rapport entre le signal utile et les interférences ou le bruit.
- Plus ces valeurs sont bonnes, plus la connexion est stable et rapide.

I.8 Importance de la prédiction des pannes

Les réseaux cellulaires sont devenus la colonne vertébrale de nos communications quotidiennes. Qu'il s'agisse de passer des appels, d'envoyer des messages, de surfer sur les réseaux sociaux ou d'accéder à des sites web en ligne, nous dépendons quotidiennement d'un réseau mobile stable, rapide et fiable. Dans ce contexte, une panne, même minimale et courte, peut avoir des conséquences importantes : **interruption des services, pertes économiques, insatisfaction des clients**, voire même des **situations critiques** dans les secteurs de la santé, de la sécurité ou du transport.

Traditionnellement, les opérateurs interviennent **après** qu'un problème ait été détecté, c'est ce qu'on appelle une approche **réactive**. Cependant, cette méthode n'est plus suffisante à l'ère de la 5G et des objets connectés, où les attentes en matière de disponibilité sont très élevées. C'est là que la prédiction des pannes entre en jeu.

La prédiction consiste à **anticiper une défaillance avant qu'elle ne se produise**, grâce à l'analyse des données générées par le réseau : qualité du signal, surcharge de trafic, erreurs de systèmes, comportements anormaux, etc. En exploitant ces données avec des outils d'intelligence artificielle (comme le machine learning ou les réseaux de neurones), il devient possible de **repérer les signes avant-coureurs ou annonceurs d'une panne**, parfois plusieurs heures ou jours à l'avance.

Cette approche présente de nombreux avantages selon [16], les voici :

- ★ Elle **réduit les interruptions** de service, voire les évite totalement.
- ★ Elle **optimise la maintenance**, en permettant des interventions préventives plutôt que correctives.
- ★ Elle **améliore l'expérience utilisateur**, car les clients ne subissent pas ou peu d'interruptions.

- ★ Elle **diminue les coûts d'exploitation**, en ciblant mieux les zones à surveiller ou à réparer.

La prédiction des pannes transforme la manière dont les réseaux cellulaires sont gérés. Elle permet de passer d'un fonctionnement passif à un réseau proactif et intelligent, capable de s'auto-surveiller et de s'adapter en temps réel. Dans un monde où la connectivité est devenue essentielle, cette capacité d'anticipation est non seulement un avantage technologique, mais aussi une nécessité stratégique.

I.9 Conclusion

Ce premier chapitre a permis de comprendre le fonctionnement des réseaux cellulaires, leurs composants, les types de services offerts, ainsi que les principales causes d'une panne. On a également vu que ces réseaux, bien qu'essentiels à notre quotidien, restent sensibles et vulnérables à divers incidents techniques.

Face à ces défis, la prédiction des pannes apparaît comme une solution stratégique. En anticipant les problèmes avant qu'ils ne se produisent, elle permet de garantir une meilleure qualité de service (QoS). L'IA joue à ce stade un rôle clé, en analysant les données du réseau pour détecter les anomalies et alerter les ingénieurs à prendre les mesures correctives nécessaires à temps.

Chapitre II

Introduction à l'Intelligence Artificielle

II.1 Introduction

Après avoir découvert le fonctionnement des réseaux cellulaires et les différents types de pannes auxquelles ils peuvent être confrontés, il est important de comprendre comment ces problèmes peuvent être anticipés grâce aux outils modernes. Parmi eux, l'intelligence artificielle (IA) occupe une place de plus en plus centrale.

L'intelligence artificielle permet aujourd'hui d'aller bien au-delà des méthodes classiques de surveillance réseau. En analysant d'énormes volumes de données techniques, elle est capable de détecter automatiquement des anomalies, d'identifier des tendances invisibles à l'œil humain, et surtout, de prédire les pannes avant qu'elles n'impactent les utilisateurs.

Ce chapitre a pour objectif de présenter les principes de base de l'intelligence artificielle, les algorithmes les plus utilisés dans la prédiction, ainsi que la manière dont ils sont appliqués concrètement dans le contexte des réseaux cellulaires. L'idée est de mieux comprendre comment ces technologies transforment la gestion des réseaux en les rendant plus intelligents, proactifs et fiables.

II.2 Historique de l'intelligence artificielle

L'intelligence artificielle a été proposée pour la première fois par John McCarthy en 1956 lors de sa première conférence universitaire sur le sujet. L'idée de machines fonctionnant comme des êtres humains a commencé à être le centre de l'esprit des scientifiques et le mathématicien Alan Turing a introduit l'idée de savoir s'il est possible de faire en sorte que les machines aient la même capacité de penser et d'apprendre par elles-mêmes. Alan Turing a pu concrétiser ses hypothèses et ses questions en testant si «les machines peuvent penser» ?, Après une série de tests (plus tard appelé Test de Turing), il s'avère qu'il est possible de

permettre aux machines de penser et d'apprendre comme les humains. Le test de Turing utilise l'approche pragmatique pour être en mesure d'identifier si les machines peuvent réagir en tant qu'humains [17].

II.3 Définition de l'intelligence artificielle

L'intelligence artificielle (IA) est la capacité des machines à effectuer des tâches typiquement associées à l'intelligence humaine, comme l'apprentissage, le raisonnement, la résolution de problème, la perception ou la prise de décision. L'intelligence artificielle désigne également le champ de recherche visant à développer de telles machines, ainsi que les systèmes informatiques qui en résultent [17].

Contrairement aux programmes classiques, qui suivent des instructions figées, les systèmes d'IA peuvent s'adapter : ils apprennent à partir des expériences passées (données historiques), s'améliorent avec le temps et peuvent ajuster leur comportement face à de nouvelles situations.

II.4 L'objectif de l'intelligence artificielle

Les objectifs principaux de l'intelligence artificielle sont :

- **Automatiser des tâches complexes** : que ce soit dans l'industrie, les télécommunications, la médecine ou les transports, l'IA aide à réaliser des actions autrefois réservées à des experts humains.
- **Analyser de grandes quantités de données** : l'IA peut traiter des volumes d'informations bien plus vastes que ce qu'un humain pourrait analyser seul.
- **Prendre des décisions intelligentes** : l'IA aide à prédire des événements (comme une panne), proposer des solutions optimales ou détecter des anomalies.
- **Créer des systèmes autonomes** : dans certains cas, l'objectif est de permettre aux machines d'agir sans intervention humaine, tout en restant fiables et efficaces.

Dans le contexte des réseaux cellulaires, l'IA est utilisée pour surveiller automatiquement le comportement du réseau, repérer les signes de dysfonctionnement et prévoir à l'avance les pannes possibles, ce qui représente un véritable progrès pour la qualité et la continuité du service.

II.5 Les principales branches de l'intelligence artificielle

L'intelligence artificielle est un domaine vaste, qui regroupe plusieurs techniques différentes. Parmi elles, certaines sont particulièrement utilisées dans la prédiction des pannes, notamment dans les réseaux cellulaires. La figure II.1 présente les principales branches de l'IA, on trouve le machine learning (apprentissage automatique), qui permet aux systèmes d'apprendre à partir de données sans être explicitement programmés. Une sous-catégorie du machine learning est le deep learning (apprentissage profond), qui repose sur des réseaux neuronaux artificiels pour traiter des données complexes. Ces réseaux neuronaux, inspirés du fonctionnement du cerveau humain, sont au cœur des modèles les plus puissants utilisés aujourd'hui [17].



FIGURE II.1 – Diagramme de la relation entre IA, Machine Learning et Deep Learning [59].

II.5.1 L'apprentissage automatique (ML)

II.5.1.1 Définition de l'apprentissage automatique

L'apprentissage automatique ML, également connu sous le nom de Machine Learning, est un domaine de l'intelligence artificielle qui se concentre sur le développement de techniques et d'algorithmes permettant aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances avec l'expérience, sans être explicitement programmés.

Deux définitions classiques illustrent ce concept : Arthur Samuel (1956) décrit l'apprentissage automatique comme « la capacité des ordinateurs permettant d'apprendre sans être précisément programmés », tandis que Tom Mitchell (1997) le définit comme « le processus

pour un ordinateur d'apprendre à effectuer une tâche en améliorant sa performance grâce à l'expérience » [18].

II.5.1.2 Les types de modèles ML

L'apprentissage automatique est généralement subdivisé en quatre composantes en fonction de l'approche utilisée pour résoudre le problème d'apprentissage, comme le montre la figure II.2. Les quatre catégories prédominantes d'apprentissage sont l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé et l'apprentissage par renforcement [19].

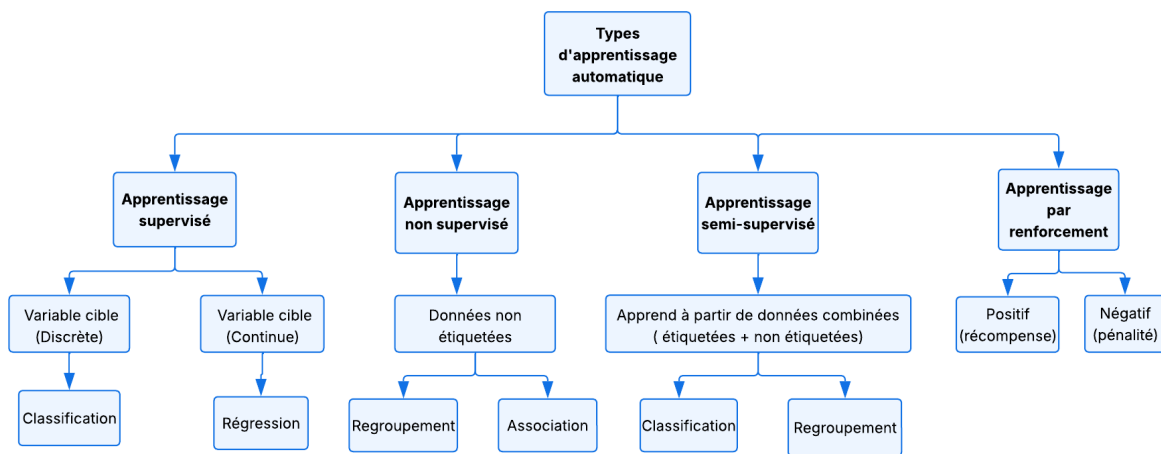


FIGURE II.2 – Différents types de techniques d'apprentissage automatique.

★ L'apprentissage supervisé (Supervised learning)

L'apprentissage supervisé est une approche du ML dans laquelle un modèle est entraîné à partir d'exemples étiquetés. L'objectif de l'apprentissage supervisé est de prédire des valeurs ou des étiquettes correctes à partir de nouvelles données non étiquetées, en se basant sur les connaissances acquises à partir des exemples d'entraînement. [19]

Dans le cadre de l'apprentissage supervisé, le modèle est présenté à un ensemble de données d'entraînement qui comprend à la fois les échantillons de données et les étiquettes correspondantes. Le modèle utilise ces exemples étiquetés pour apprendre les relations et les schémas entre leurs caractéristiques et les étiquettes, afin de généraliser et de faire des prédictions sur de nouvelles données non étiquetées. La figure II.3 représente un exemple d'apprentissage supervisé [19].

L'apprentissage supervisé peut être utilisé dans une variété de tâches, telles que la classification, la régression et la prédiction. Dans la classification, le modèle apprend à assigner des

classes ou des catégories à de nouvelles instances en fonction des exemples d'entraînement [19].

Les principaux algorithmes du machine learning supervisé sont les suivants : forêts aléatoires, arbres de décision, algorithme K-Nearest Neighbors (KNN), régression linéaire, algorithme de Naïve Bayes, (Support Vector Machine (SVM)), régression logistique et boosting de gradient. [19]

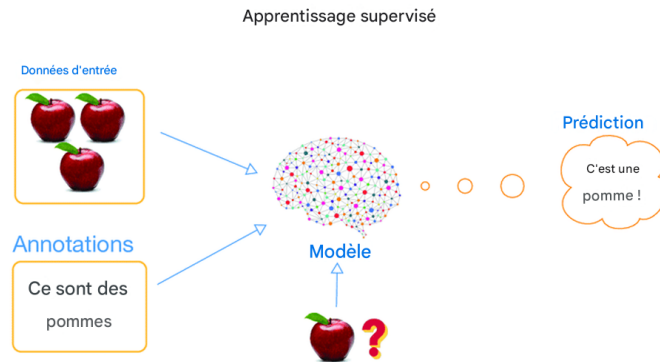


FIGURE II.3 – Représentation d'un apprentissage supervisé [60].

★ L'apprentissage non supervisé (Unsupervised learning)

L'apprentissage non supervisé est une approche ML dans laquelle un modèle est entraîné sur un ensemble de données d'entrées non étiquetées. L'objectif est de découvrir des structures ou des modèles intrinsèques dans les données sans connaître les sorties attendues. [18]

Les algorithmes d'apprentissage non supervisé sont couramment utilisés pour effectuer des tâches telles que le regroupement (clustering) des données en groupes similaires ou la réduction de dimension pour représenter les données de manière plus concise. La figure II.4 représente un exemple d'apprentissage non supervisé.

Les principaux algorithmes de la machine learning non supervisé sont les suivants : K-Means, clustering/regroupement hiérarchique et réduction de la dimensionnalité.[19]

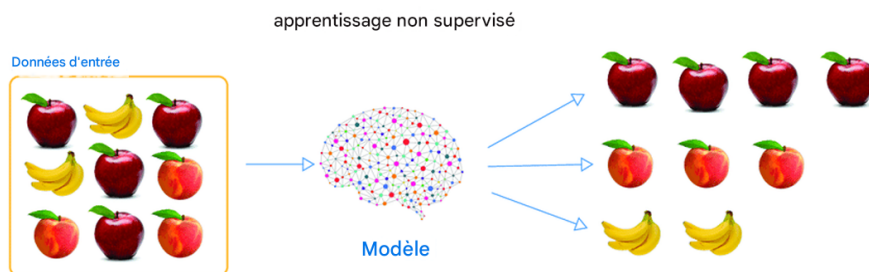


FIGURE II.4 – Représentation d'un apprentissage non supervisé [60].

★ L'apprentissage semi-supervisé (Semi-supervised learning)

L'apprentissage semi-supervisé est une approche ML qui utilise à la fois des données étiquetées et non étiquetées pour entraîner un modèle. Cela permet d'exploiter les informations des données non étiquetées pour améliorer les performances du modèle tout en réduisant la nécessité d'étiqueter manuellement de grandes quantités de données. La figure II.5 représente un exemple d'apprentissage non supervisé. Les algorithmes d'apprentissage semi-supervisé combinent des techniques d'apprentissage supervisé et non supervisé pour tirer parti de ces deux types de données [19].

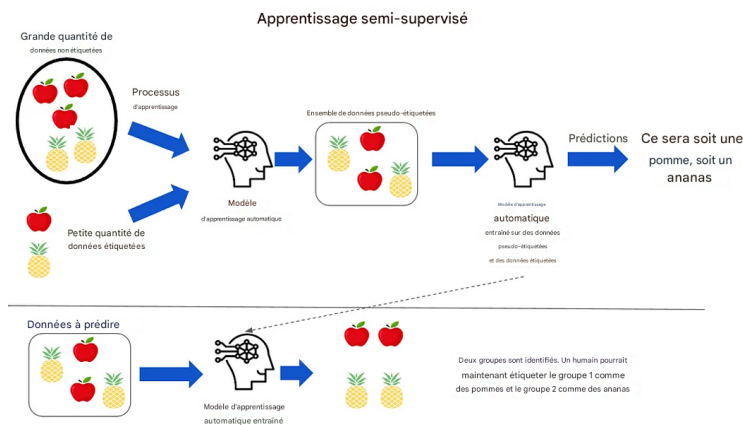


FIGURE II.5 – Représentation d'un apprentissage semi-supervisé [60].

★ L'apprentissage par renforcement (Reinforcement learning)

L'apprentissage par renforcement est une méthode ML où un agent apprend à prendre des décisions séquentielles en interagissant avec un environnement. Comme le montre la figure II.6, l'agent reçoit des récompenses ou des pénalités en fonction de ses actions, ce qui lui permet d'ajuster son comportement pour maximiser les récompenses sur le long terme. Les algorithmes d'apprentissage par renforcement sont utilisés dans des domaines tels que les jeux, la robotique et la planification automatique.

Les principaux algorithmes de l'apprentissage par renforcement sont les suivants : Q-learning, et State-Action-Reward-State-Action (SARSA) [19].

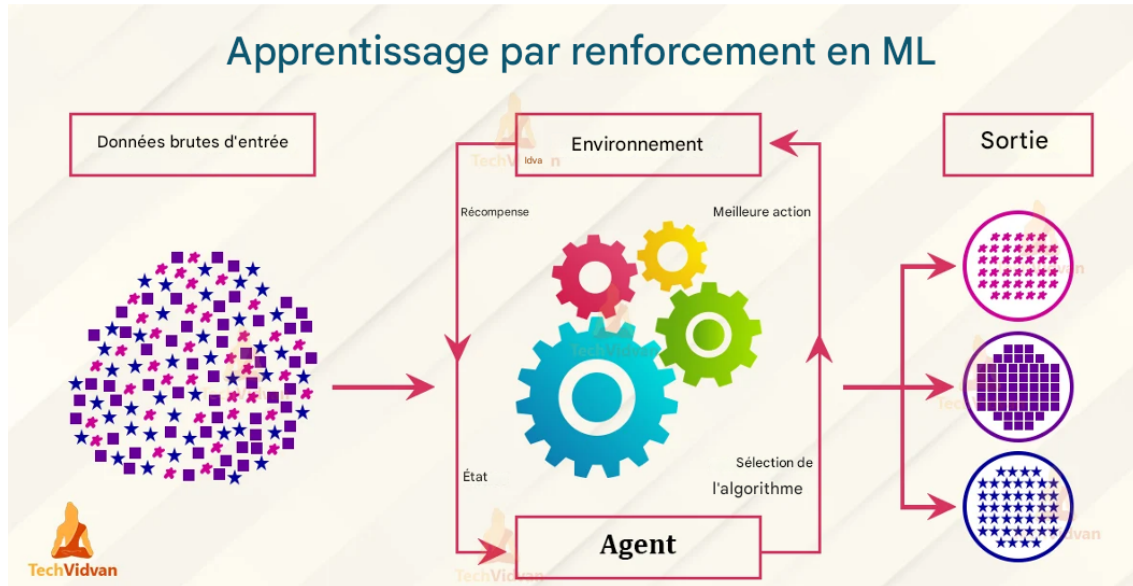


FIGURE II.6 – Représentation d'un apprentissage par renforcement [60].

II.5.2 L'apprentissage profond (DL)

Le DL est une branche de ML spécialisée dans les réseaux de neurones profonds, on l'a présenté comme étant une branche de l'IA ;

II.5.2.1 Définition de l'apprentissage profond

L'apprentissage profond est une classe d'algorithmes d'apprentissage automatique appelés réseaux neuronaux. Les réseaux de neurones sont des modèles mathématiques inspirés de la structure du cerveau. L'apprentissage profond permet à l'algorithme de réseau neuronal de très bien fonctionner dans la construction de modèles de prédiction autour de problèmes complexes tels que la vision par ordinateur et la modélisation du langage [20].

II.5.2.2 Les types de Modèles DL

Les techniques de DL sont divisées en trois grandes catégories, La figure II.7 les illustre, telles que ; les réseaux profonds pour l'apprentissage supervisé ou discriminatif, les réseaux profonds pour l'apprentissage non supervisé ou génératif et les réseaux profonds pour l'apprentissage hybride combinant les deux et les autres pertinents, comme le montre la figure suivante. Dans ce qui suit, nous abordons brièvement chacune de ces techniques qui peuvent être utilisées pour résoudre des problèmes du monde réel dans divers domaines d'application en fonction de leurs capacités d'apprentissage [21].

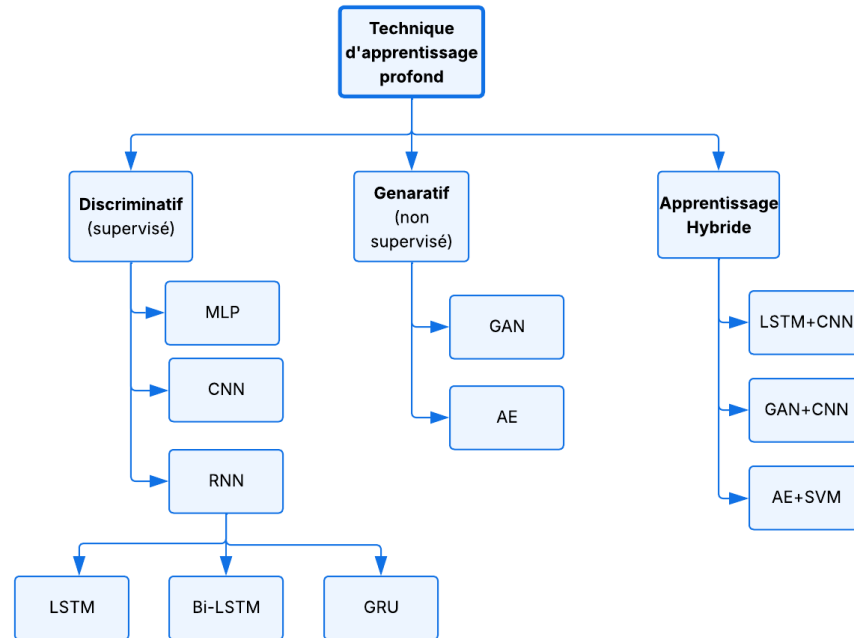


FIGURE II.7 – Les techniques d'apprentissage profond.

II.5.2.2.1 L'apprentissage supervisé : cette catégorie de techniques d'apprentissage à distance est utilisée pour fournir une fonction discriminante dans les applications supervisées ou de classification. Les architectures profondes discriminatives sont généralement conçues pour donner un pouvoir discriminant pour la classification des motifs en décrivant les distributions a posteriori des classes conditionnées par des données visibles. Les architectures discriminatives comprennent principalement le Perceptron multicouche (MLP), les réseaux neuronaux convolutifs (Convolutional Neural Networks (CNN)), les réseaux neuronaux récurrents (RNN), ainsi que leurs variantes [21].

– **Perceptron multicouche (MLP)**

Une approche d'apprentissage supervisé est un type de réseau neuronal artificiel (Artificial Neural Network (ANN)). Elle est également connue sous le nom d'architecture de base des réseaux de neurones profonds (Deep Neural Network (DNN)) ou d'apprentissage profond.

La figure II.8 illustre les différentes composantes d'une architecture de perceptron multicouche (MLP), telles que la couche d'entrée qui reçoit les données d'entrée, la couche de sortie qui prend une décision ou une prédiction sur le signal d'entrée, ainsi qu'une ou plusieurs couches cachées entre les deux qui sont considérées comme le moteur de calcul du réseau. La sortie d'un réseau MLP est déterminée à l'aide d'une variété de fonctions d'activation, également appelées fonctions de transfert. Pour s'entraîner, MLP utilise l'algorithme le plus largement utilisé, la «rétropropagation» ou «Backpropagation», une technique d'apprentis-

sage supervisé, également connue sous le nom de bloc de construction le plus élémentaire d'un réseau neuronal [21].

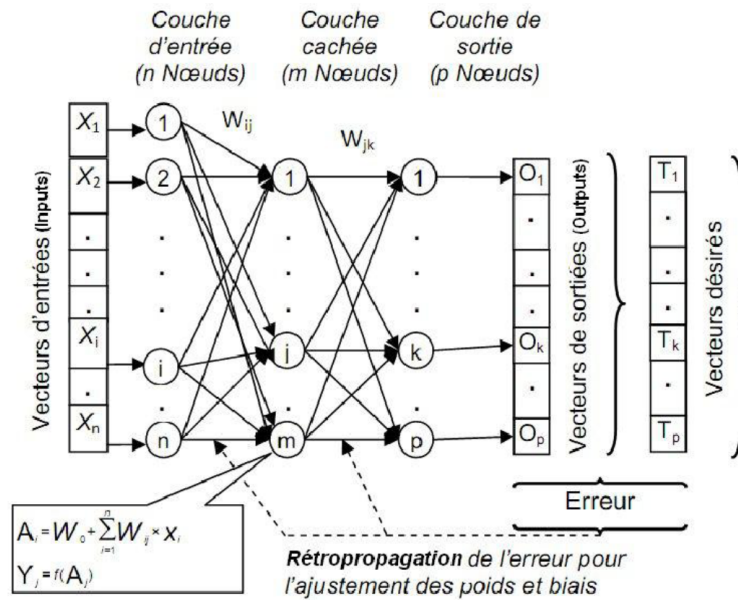


FIGURE II.8 – Architecture du MLP [21].

– Réseaux de neurones à convolution (CNN)

Les réseaux convolutifs, également appelés réseaux neuronaux convolutifs CNN, est une architecture d'apprentissage profond discriminative populaire qui apprend directement à partir de l'entrée sans avoir besoin d'extraire des caractéristiques humaines [21]. La figure V.5(c) montre un exemple de réseau de neurones convolutifs (CNN) comprenant plusieurs couches de convolution et de pooling. Le CNN est un type spécifique de réseau neuronal spécialisé dans le traitement de données de type grille. Des exemples de ce type de données sont les séries chronologiques et les images, qui peuvent être considérées comme une grille à 1 dimension et une grille à 2 dimensions de pixels, respectivement [20].

En conséquence, le CNN améliore la conception des réseaux MLP traditionnels comme les réseaux MLP régularisés. Chaque couche dans CNN prend en compte des paramètres optimaux pour une sortie significative et réduit la complexité du modèle. CNN utilise également un « dropout » qui peut résoudre le problème du surapprentissage (overfitting), qui peut survenir dans un réseau traditionnel. Les réseaux convolutifs ont été largement utilisés dans divers problèmes du monde réel, tels que le traitement du langage naturel (Natural Language Processing (NLP)), la vision par ordinateur, la reconnaissance vocale, etc. [20]

Il existe plusieurs variantes de CNN, notamment le groupe de géométrie visuelle (Visual Geometry Group (VGG)), Alex Krizhevsky Neural Network (AlexNet), eXtreme inception

(Xception), Inception, Residual Network (ResNet), etc., qui peuvent être utilisés dans divers domaines d'application en fonction de leurs capacités d'apprentissage [21].

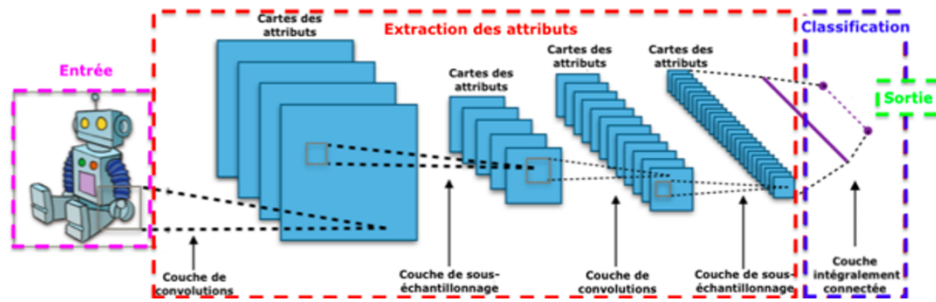


FIGURE II.9 – Un exemple de réseau de neurones conversationnel CNN [21].

– Réseaux de neurones récurrents (RNN)

Les réseaux neuronaux récurrents sont une gamme de réseaux neuronaux artificiels appropriés pour l'analyse de données séquentielles telles que des séquences de texte ou de données temporelles [20]. Les réseaux récurrents apprennent des entrées d'entraînement, mais se distinguent par leur « mémoire », qui leur permet d'influencer l'entrée et la sortie actuelles en utilisant les informations des entrées précédentes, comme illustré dans la figure II.10. Contrairement aux DNN typiques, qui supposent que les entrées et les sorties sont indépendantes les unes des autres, la sortie des RNN dépend d'éléments antérieurs dans la séquence. Cependant, les réseaux récurrents standards ont le problème de la disparition des gradients, ce qui rend difficile l'apprentissage de longues séquences de données [21]. Les RNN sont couramment utilisés pour des tâches telles que la traduction automatique, la génération de texte et l'analyse de sentiments [19].

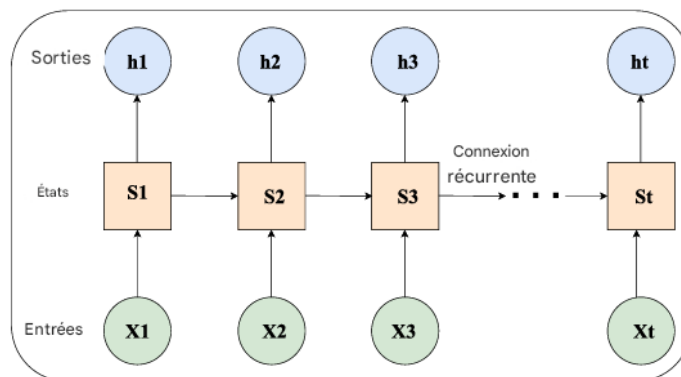


FIGURE II.10 – La structure de RNN [19].

Il existe plusieurs variantes de RNN, qui minimisent les problèmes et fonctionnent bien

dans de nombreux domaines d'application du monde réel, notamment LSTM, Bidirectional RNN/LSTM, GRU.

- ★ **Mémoire longue à court terme (LSTM)** il s'agit d'une architecture populaire de réseau de neurones récurrents (RNN) qui utilise des unités spéciales pour traiter le problème du gradient qui disparaît. Une cellule de mémoire dans une unité LSTM peut stocker des informations pendant de longues périodes, et le flux d'informations à l'intérieur et à l'extérieur de la cellule est régulé par trois portes. Une illustration graphique de la structure d'un LSTM est présentée dans la figure V.5(g). Dans cette structure, la porte d'oubli « Forget Gate » détermine quelles informations de la cellule d'état précédente seront mémorisées et quelles informations qui ne sont plus utiles seront supprimées, tandis que la porte d'entrée « Input Gate » détermine quelles informations doivent entrer dans l'état de la cellule et le Porte de sortie « output Gate » gère les informations à transmettre en sortie. Le LSTM est considéré comme l'un des RNN les plus performants, car il permet de résoudre les problèmes liés à l'apprentissage des réseaux récurrents traditionnels. Il a été appliqué avec succès à de nombreux problèmes, telles que la reconnaissance vocale et la classification de texte [21].

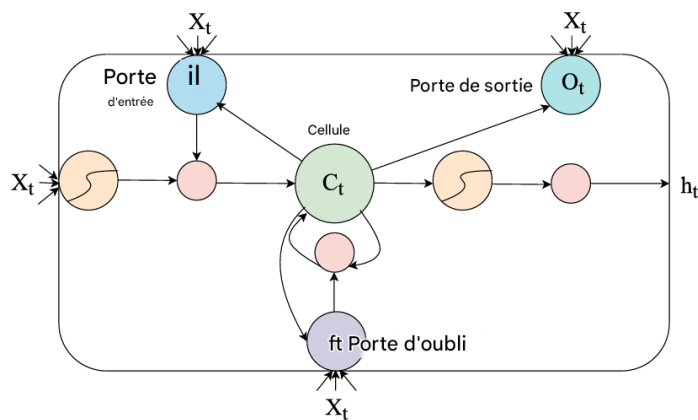


FIGURE II.11 – La structure interne d'un LSTM [19].

- ★ **Bidirectional RNN/LSTM** Les RNN bidirectionnels connectent deux couches cachées qui s'exécutent dans des directions opposées à une seule sortie, ce qui leur permet d'accepter des données du passé et du futur. Les RNN bidirectionnels, contrairement aux réseaux récurrents traditionnels, sont entraînés à prédire les directions temporelles positives et négatives en même temps. Un LSTM bidirectionnel, souvent connu sous le nom de Bidirectional Long Short-Term Memory (BiLSTM), est une extension du LSTM standard qui peut augmenter les performances du modèle sur les problèmes de classification de séquences. Il s'agit d'un modèle de traitement séquentiel composé

de deux LSTM : l'un fait avancer l'entrée et l'autre la fait reculer. Le LSTM bidirectionnel en particulier est un choix populaire dans les tâches de traitement du langage naturel [21].

- ★ **GRU** Une unité récurrente à grille (GRU) est une variante populaire des réseaux de neurones récurrents (RNN) qui repose sur des mécanismes de contrôle pour réguler le flux d'informations entre les cellules du réseau. Le GRU est similaire au LSTM, mais il comporte moins de paramètres, car il ne dispose que de deux portes : une porte de réinitialisation et une porte de mise à jour, contrairement au LSTM qui inclut également une porte de sortie, comme le montre la figure II.12. Cette architecture simplifiée permet au GRU de capturer efficacement les dépendances à long terme dans des séquences, sans négliger les informations issues des étapes précédentes. Grâce à sa structure plus légère, le GRU offre souvent des performances comparables à celles du LSTM, tout en étant moins coûteux en calcul. Il a d'ailleurs été montré que le GRU peut surpasser le LSTM dans certains cas, notamment sur des ensembles de données de petite taille ou contenant des événements peu fréquents. En résumé, ces deux variantes de RNN ont démontré leur efficacité dans diverses tâches de modélisation séquentielle [21].

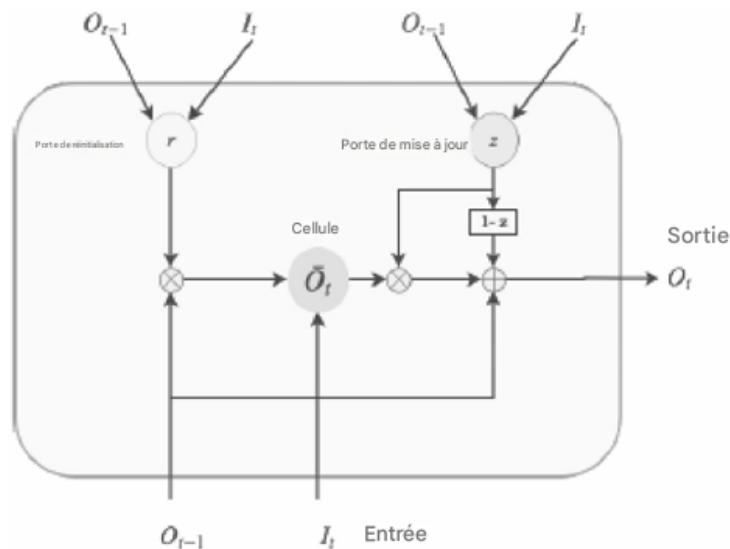


FIGURE II.12 – Structure de base d'une cellule GRU [21].

II.5.2.2.2 L'apprentissage non supervisé : cette catégorie de techniques DL est généralement utilisée pour caractériser les propriétés ou les caractéristiques de corrélation d'ordre élevé pour l'analyse ou la synthèse de motifs, ainsi que les distributions statistiques conjointes des données visibles et de leurs classes associées. Les techniques de réseaux de neurones pro-

fonds couramment utilisées pour l'apprentissage non supervisé ou génératif sont le réseau antagoniste génératif (Generative Adversarial Network (GAN)), l'auto-encodeur (Auto Encoder (AE)) [21].

– Réseau antagoniste génératif (GAN)

Un réseau antagoniste génératif (GAN), est un type d'architecture de réseau neuronal pour la modélisation générative afin de créer de nouveaux échantillons plausibles à la demande. Il s'agit de découvrir et d'apprendre automatiquement des régularités ou des modèles dans les données d'entrée afin que le modèle puisse être utilisé pour générer ou sortir de nouveaux exemples à partir de l'ensemble de données d'origine. Comme le montre la figure II.13, les GAN sont composés de deux réseaux neuronaux, un générateur G qui crée de nouvelles données ayant des propriétés similaires aux données d'origine, et un discriminateur D qui prédit la probabilité qu'un échantillon ultérieur soit tiré de données réelles plutôt que de données fournies par le générateur. Ainsi, dans la modélisation GAN, le générateur et le discriminateur sont entraînés à se faire concurrence. Alors que le générateur essaie de tromper et de confondre le discriminateur en créant des données plus réalistes, le discriminateur essaie de distinguer les données authentiques des fausses données générées par G [21].

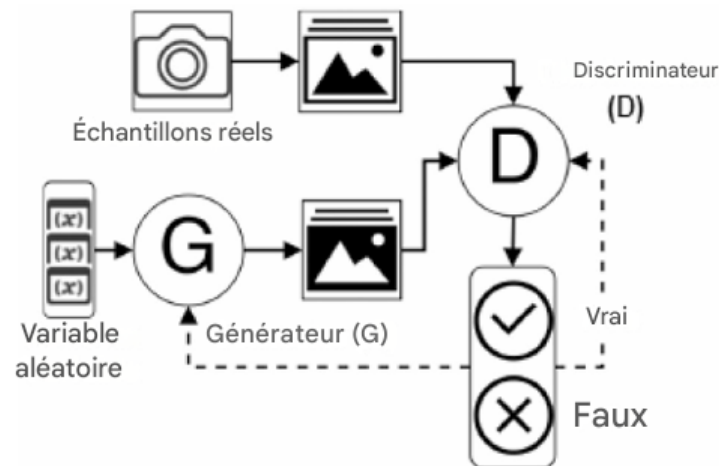


FIGURE II.13 – Structure schématique d'un réseau antagoniste génératif (GAN standard) [21].

– Auto-encodeur (AE)

Un auto-encodeur (AE) est une technique d'apprentissage non supervisé populaire dans laquelle les réseaux neuronaux sont utilisés pour apprendre des représentations. En règle générale, les encodeurs automatiques sont utilisés pour travailler avec des données de grande dimension, et la réduction de la dimensionnalité explique comment un ensemble de données

est représenté. L'encodeur, le code et le décodeur sont les trois parties d'un auto-encodeur. L'encodeur compresse l'entrée et génère le code, que le décodeur utilise ensuite pour reconstruire l'entrée, comme le montre la figure II.14. Les AE ont récemment été utilisés pour apprendre des modèles de données génératives. L'auto-encodeur est largement utilisé dans de nombreuses tâches d'apprentissage non supervisées par exemple la réduction de dimensionnalité, l'extraction de caractéristiques, le codage efficace, la modélisation générative, la détection d'anomalies ou de valeurs aberrantes [21].

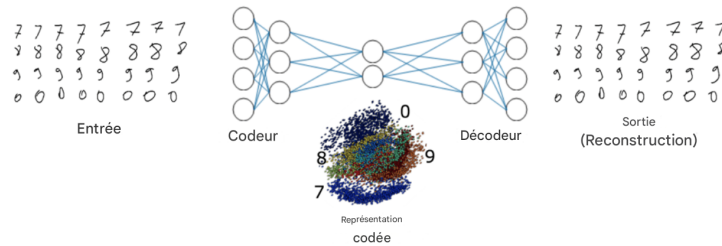


FIGURE II.14 – Structure d'un auto-encodeur (AE) [21].

II.5.2.2.3 L'apprentissage hybride Les modèles génératifs sont adaptables, avec la capacité d'apprendre à partir de données étiquetées et non étiquetées. Les modèles discriminants, en revanche, sont incapables d'apprendre à partir de données non étiquetées, mais surpassent leurs homologues génératifs dans les tâches supervisées. Un cadre permettant d'entraîner simultanément des modèles génératifs profonds et discriminatifs peut profiter des avantages des deux modèles, ce qui motive les réseaux hybrides.

Les modèles hybrides d'apprentissage profond sont généralement composés de plusieurs (deux ou plus) modèles d'apprentissage profond de base, où le modèle de base est un modèle d'apprentissage profond discriminatif ou génératif dont nous avons parlé précédemment. Sur la base de l'intégration de différents modèles génératifs ou discriminatifs de base, les trois catégories ci-dessous de modèles hybrides d'apprentissage profond pourraient être utiles pour résoudre des problèmes du monde réel [21].

- **modèle 1** : une intégration de différents modèles génératifs ou discriminatifs pour extraire des caractéristiques plus significatives et robustes. Des exemples pourraient être CNN+LSTM, AE+GAN, etc.
- **modèle 2** : une intégration d'un modèle génératif suivi d'un modèle discriminatif. Il peut s'agir, par exemple, de Deep Belief Network (DBN)+MLP, de GAN+CNN, d'AE+CNN, etc.
- **modèle 3** : intégration d'un modèle génératif ou discriminatif suivi d'un classificateur non deep learning. Il peut s'agir, par exemple, de AE+SVM, CNN+SVM, etc. [21].

II.6 Conclusion

Le Machine Learning ML consiste à enseigner à un système informatique à apprendre des modèles à partir de données. L'apprentissage profond DL est en réalité une application croissante de l'apprentissage automatique. Avec ses architectures complexes et ses avancées technologiques, il a transformé de nombreux domaines, notamment la vision par ordinateur, la reconnaissance vocale et la traduction automatique. La principale valeur de l'apprentissage profond repose sur l'amélioration des applications actuelles de l'apprentissage automatique, en raison de sa capacité innovante à traiter des couches hiérarchiques. Nous notons également que les pratiques actuelles d'apprentissage profond dans les réseaux mobiles utilisent largement l'apprentissage supervisé et l'apprentissage par renforcement.

Cette étude offre une base solide pour comprendre comment l'intelligence artificielle et ses algorithmes peuvent être exploités dans des domaines aussi variés que la reconnaissance de motifs, la prévision, la détection d'anomalies ou encore l'optimisation de systèmes complexes. Les chapitres suivants s'appuieront sur ces concepts pour les appliquer à des problématiques concrètes liées aux réseaux cellulaires.

Chapitre III

Revue des travaux existants

III.1 Introduction

La prédiction des pannes dans les réseaux cellulaires est devenue un sujet très important face à l'évolution des technologies mobiles et à l'augmentation de la demande en connectivité fiable. Pour garantir un service de qualité, les opérateurs doivent être capables d'anticiper les différentes défaillances techniques avant qu'elles n'arrivent aux utilisateurs finaux. C'est dans ce contexte que de nombreux chercheurs et ingénieurs ont développé, au fil des années, différentes méthodes pour prévenir les pannes dans les réseaux cellulaires.

Ce chapitre propose une revue ou un aperçu des principaux travaux réalisés dans le domaine de la prédiction de pannes, en mettant en lumière les approches utilisées, les types de données analysés ainsi que les performances obtenues. Il s'agit de mieux comprendre comment la recherche a évolué, en partant des méthodes statistiques classiques, simples mais limitées à cette ère, jusqu'aux techniques modernes d'intelligence artificielle, capables d'apprendre et de s'adapter à la complexité croissante des réseaux.

III.2 Les méthodes classiques de prédiction

Avant l'apparition des méthodes d'intelligence artificielle, la prédiction des pannes dans les réseaux cellulaires reposait principalement sur des méthodes statistiques classiques. Ces approches utilisaient des modèles mathématiques simples pour identifier les comportements anormaux ou ce qu'on appelle les tendances, détecter des anomalies ou prévoir des défaillances sur la base de séries temporelles de données réseau. Bien qu'elles aient posé les bases de la surveillance proactive, elles montraient rapidement leurs limites face à la complexité croissante des réseaux mobiles modernes.

III.2.1 Modèles statistiques traditionnels

Parmi les plus utilisés, on retrouve :

- **Auto Regressive Integrated Moving Average (ARIMA)** : un modèle répandu pour l'analyse de séries temporelles. Il est souvent utilisé pour prévoir les indicateurs de performance du réseau (comme le taux de perte de paquets ou le délai de transmission).
- **Méthodes à base de seuils fixes** : si un indicateur réseau dépasse un seuil prédéfini (par exemple, le taux d'occupation Central Processing Unit (CPU) $> 90\%$), une alerte est générée.
- **Régressions linéaires ou logistiques** : pour modéliser la relation entre différents facteurs et la probabilité d'occurrence d'une panne.

Ces modèles ont l'avantage d'être simples à mettre en œuvre et faciles à interpréter. Toutefois, ils supposent souvent que les données sont stationnaires, linéaires et qu'elles suivent une distribution connue, ce qui est rarement le cas dans les environnements cellulaires complexes [22].

III.2.2 Exemples de travaux utilisant les méthodes classiques

Dans cet élément, nous allons exposer certains travaux qui ont utilisé les méthodes classiques qui sont déjà définies dans l'élément précédant ;

- **Kumar et al. (2010)**

Dans cette étude sur les réseaux cellulaires, **Kumar et al.** ont utilisé une approche basée sur l'analyse de séries temporelles classiques pour détecter les anomalies dans les performances des stations de base. Ils ont mis en œuvre des techniques comme la régression polynomiale et le suivi de tendance pour identifier les périodes à risque. Bien que cette méthode ait permis de repérer certaines baisses de performance, elle restait peu sensible aux événements imprévus ou aux défaillances multifactorielles [23].

- **Santos et al. (2015)**

Une autre étude, menée par **Santos et al.**, a appliqué des modèles ARIMA pour prévoir les KPI d'un réseau 3G. Le but était d'anticiper les congestions avant qu'elles n'aient un impact sur les utilisateurs. Les résultats étaient encourageants à court terme, mais l'approche ne s'adaptait pas bien aux comportements non linéaires ou aux changements soudains dans le trafic [24].

III.2.3 Limites

Les méthodes classiques, bien qu'efficaces pour des réseaux de faible complexité, montrent leurs limites dans les contextes modernes :

- Faible capacité à modéliser des relations complexes ou non linéaires entre variables.
- Sensibilité aux bruits et aux données manquantes.
- Peu de capacité d'adaptation aux environnements dynamiques, comme les réseaux mobiles de dernière génération (4G, 5G).

Ces contraintes ont permis l'apparition des approches plus avancées, comme celles basées sur l'intelligence artificielle, qui seront détaillées dans la suite du chapitre [25].

III.3 L'émergence de l'intelligence artificielle dans la détection des pannes

Avec l'évolution rapide des réseaux cellulaires du 3G au 5G, les systèmes sont devenus de plus en plus complexes, dynamiques et riches en données. Cette complexité rend les approches classiques de prédiction (comme les modèles statistiques simples) souvent insuffisantes, voire inadaptées. C'est dans ce contexte que l'intelligence artificielle (IA) a commencé à jouer un rôle central dans la détection et la prédiction des pannes réseau.

L'IA offre des outils capables de traiter d'énormes volumes de données, de reconnaître des schémas complexes, et surtout, de s'adapter automatiquement à des environnements changeants. Cela correspond exactement aux besoins des réseaux cellulaires modernes, où les pannes peuvent dépendre de nombreux facteurs : qualité du signal, surcharge, erreurs logicielles, comportement utilisateur, météo, etc.

Contrairement aux méthodes classiques traditionnelles, qui se basent souvent sur des règles fixes, les modèles d'IA apprennent directement à partir des données pour détecter des comportements anormaux, même imprévisibles [26].

III.4 L'impact de l'IA sur les réseaux cellulaires

L'introduction de l'IA a permis plusieurs avancées importantes :

1. **Détection proactive** : l'IA peut détecter des signes qui introduisent une panne, bien avant qu'elle n'apparaisse réellement [27].
2. **Réduction des faux positifs** : les algorithmes peuvent mieux distinguer une vraie panne d'une simple modification du réseau, ce qui limite les fausses alertes [28].

3. **Apprentissage continu** : certains systèmes IA peuvent s'adapter de manière continue à l'évolution du réseau et à de nouveaux types de défaillances [29].
4. **Automatisation du diagnostic** : en plus de prévoir les pannes, l'IA peut aussi proposer ou effectuer des mesures correctives automatiquement [30].

III.5 Travaux utilisant des techniques de l'IA

Ces dernières années, plusieurs études ont exposé l'efficacité de l'IA pour détecter ou prédire les pannes dans les réseaux cellulaires. Ces travaux utilisent différents types de modèles, selon les objectifs (détection d'anomalies, classification, prédiction de séries temporelles, etc.) et le type de données disponibles (KPI, logs d'erreurs, trafic, etc.).

Dans ce présent élément, certains travaux sont cités, qui exploitent les techniques modernes de l'IA ;

III.5.1 Machine Learning

Certains modèles d'apprentissage automatique ML sont identifiées dans cet élément, avec des exemples d'articles qui les ont utilisés ;

1. SVM

Les SVM sont largement utilisés pour la classification binaire (dans notre cas "normal vs panne"). Ils sont efficaces même sur des jeux de données de petite taille, à condition de bien choisir les paramètres.

Mahboob et al. (2016) ont utilisé un modèle SVM pour prédire les dégradations de performance dans un réseau LTE, en analysant les KPI tels que le taux de retransmission et le débit.

Leur modèle a atteint une précision de 91%, montrant une amélioration par rapport aux méthodes basées sur des seuils fixes [31].

2. Random Forest (Forêts aléatoires)

Les forêts aléatoires sont connues pour leur robustesse, leur précision, et leur capacité à gérer les données bruitées ou incomplètes.

Ahsan et al. (2019) ont utilisé Random Forest pour prédire les pannes dans un réseau mobile en combinant plusieurs KPI (latence, taux d'erreur, débit).

Le modèle a atteint plus de 94% de précision, dépassant les performances du SVM et du KNN [31].

3. KNN

Le KNN est une méthode puissante quand les classes sont bien séparées. Elle a été utilisée dans des scénarios où les KPI sont directement comparés à des situations passées similaires.

Mendoza et al. (2017) ont testé plusieurs algorithmes, dont le KNN pour classifier les états de santé d'un réseau.

Le KNN a obtenu 88% de précision ; sa limite, c'est qu'il demandait beaucoup de mémoire, ce qui borne son usage en temps réel [33].

III.5.2 Deep Learning

Les réseaux de neurones sont utiles lorsqu'on dispose des données temporelles, comme le "timestamp".

Certains modèles d'apprentissage profond DL sont identifiées dans cet élément, avec des exemples d'articles qui les ont utilisés ;

1. **LSTM**

Zhang et al. (2020) ont utilisé LSTM pour prédire les KPI (ex : taux de blocage) 5 minutes à l'avance dans un réseau 4G. Résultat : meilleure précision que les réseaux classiques (ANN) et meilleure adaptation aux variations du trafic [34].

2. **CNN**

Lee et al. (2018) ont utilisé un CNN pour analyser visuellement des "cartes de chaleur" générées à partir des KPI afin de localiser les zones à risque. Très utile pour des outils de supervision visuelle automatisée [35].

III.5.3 Méthodes hybrides

Des approches récentes combinent plusieurs modèles ou utilisent des techniques d'explicabilité pour comprendre pourquoi une panne est prédite.

Al-Masri et al. (2022) ont combiné Random Forest avec des techniques d'explicabilité (SHapley Additive exPlanations (SHAP)) pour non seulement détecter les pannes, mais aussi expliquer les causes possibles. Cela facilite la prise de décision par les ingénieurs réseau [36].

III.6 Travaux connexes

Nous présenterons dans cette section certains travaux qui ont utilisé certaines méthodes du ML et DL :

III.6.1 Article 01

A Fault Diagnosis Method for 5G Cellular Networks Based on Knowledge and Data Fusion

Zhao et al. (2024) proposent une méthode innovante de diagnostic des pannes dans les réseaux cellulaires 5G, combinant des techniques d'intelligence artificielle et des connaissances expertes. Leur approche repose sur l'intégration de trois composantes principales :

1. Génération de données synthétiques : utilisation de réseaux antagonistes génératifs (GAN) pour équilibrer les classes de pannes dans le jeu de données, améliorant ainsi la représentativité des différentes catégories de défauts.
2. Pré-diagnostic basé sur les connaissances : application d'un modèle bayésien naïf (Naive Bayes Model (NBM)) combiné à l'expertise humaine pour effectuer une première analyse des données, aboutissant à la création d'un graphe d'association topologique reflétant les relations entre les paramètres du réseau.
3. Diagnostic final via apprentissage profond : intégration du graphe d'association et des données enrichies dans un réseau de neurones convolutifs sur graphes (Graph Convolutional Network (GCN)), permettant une classification précise des types de pannes.

Le jeu de données sur lequel sont effectués ces tests est réel, collecté en août 2019 à Lu'an, en Chine, via la technologie (Minimization of Drive Tests (MDT)).

Les résultats expérimentaux indiquent une précision de 90,56 % et un score f1-score (f1) macro de 88,41 %, surpassant les méthodes traditionnelles de diagnostic [37].

III.6.2 Article 02

Automatic Fault Detection and Diagnosis in Cellular Networks and Beyond 5G : Intelligent Network Management

Rezaei et ses collègues (2022) proposent une méthode intelligente pour détecter automatiquement les pannes dans les réseaux cellulaires, y compris ceux de la 5G et au-delà. L'étude repose sur 2 ensembles de données : les données de support de performance (Performance Support System) et les données issues des tests de conduite (Drive tests). La méthodologie est fondée sur la collecte des données en premier, qui est effectuée dans 3 scénarios :

- ★ un appel court, pour identifier les pannes lors de l'établissement d'un appel ;
- ★ un appel long détecte les échecs des handovers et les interruptions d'appel ;
- ★ le mode veille qui analyse le comportement du signal radio.

Ils utilisent d'abord des techniques de regroupement (clustering) pour identifier des comportements similaires dans les données, puis appliquent des algorithmes d'apprentissage su-

pervisé, notamment un réseau neuronal dynamique, pour diagnostiquer précisément les types de pannes. Cette méthode permet d'identifier efficacement des problèmes comme la congestion ou les échecs de transfert d'appel, tout en suggérant des actions correctives. Le modèle a atteint une précision de 96,86% grâce aux réseaux neuronaux dynamiques [38].

III.6.3 Article 03

Cell Coverage Degradation Detection Using Deep Learning Techniques

Les auteurs de cet article explorent l'utilisation des réseaux neuronaux récurrents (RNN) pour détecter la dégradation des performances radio d'une cellule. L'étude se concentre plus particulièrement sur la détection des dégradations de couverture cellulaire, notamment les cellules dormantes et la détérioration du signal radio.

Le jeu de données utilisé pour cette approche est constitué de données simulées de puissance du signal RSRP, dans le but de détecter les anomalies plus efficacement que les méthodes classiques. Pour l'évaluation, les auteurs comparent plusieurs modèles d'intelligence artificielle : le Support Vector Machine (SVM) est utilisé comme modèle de référence, tandis que les RNN sont testés avec différentes architectures (jusqu'à trois couches cachées). Après les tests, il a été constaté que le RNN avec une seule couche cachée offre la meilleure précision, tout en optimisant le temps de calcul.

Finalement, l'étude montre que DL permet une meilleure anticipation des pannes de signal, facilitant la maintenance proactive des réseaux [39].

III.6.4 Article 04

Cell Fault Management Using Machine Learning Techniques

L'article de David Mulvey et al. (2019) examine comment les techniques de machine learning (ML) peuvent améliorer la gestion des pannes dans les réseaux cellulaires, en automatisant leur détection et leur correction. Les auteurs identifient plusieurs types de pannes récurrentes, notamment les interférences, les échecs de handover et la dégradation du signal.

L'étude compare trois approches : les systèmes à base de règles, les méthodes algorithmiques classiques, et les algorithmes de machine learning. Parmi les modèles testés figurent les machines à vecteurs de support (SVM), les forêts aléatoires (Random Forest (RF)) et les réseaux neuronaux profonds.

Les résultats montrent que ces derniers offrent la meilleure performance en termes de précision du diagnostic [40].

III.6.5 Article 05

Cellular Fault Detection Using Spatio-Temporal Graph Convolutional Networks

L'article de Qian et Xie (2023) propose une méthode avancée pour la détection des pannes dans les réseaux cellulaires en utilisant des réseaux de neurones convolutifs graphiques spatio-temporels (Spatio-Temporal Graph Convolutional Network (st-GCN)). L'approche vise à améliorer les techniques classiques en prenant en compte à la fois les dépendances spatiales et temporelles des paramètres du réseau, ce qui permet une modélisation plus précise de la dynamique du réseau.

Les résultats expérimentaux montrent que cette approche offre des performances supérieures par rapport aux méthodes traditionnelles. En particulier, la méthode st-GCN atteint une précision de 95 %, surpassant les autres techniques de détection des pannes qui utilisent des modèles classiques ou basés sur des réseaux de neurones simples. De plus, les auteurs rapportent une réduction significative du taux de faux positifs, ce qui est essentiel pour éviter les interventions inutiles dans un réseau opérationnel. Enfin, l'approche a également démontré sa capacité à prédire des pannes de manière proactive en temps réel, ce qui peut améliorer la gestion des pannes dans les réseaux 5G en anticipant les problèmes avant qu'ils ne surviennent [41].

III.6.6 Article 06

System failure prediction within software 5G core networks using time series forecasting

L'article de Pousali Chakraborty et al. (2021) présente une méthode proactive pour prédire les défaillances dans les réseaux cœurs 5G virtualisés. Face à la complexité croissante des architectures 5G, notamment en raison de la virtualisation des fonctions réseau (Network Functions Virtualization (NFV)) et de l'utilisation de réseaux définis par logiciel (Software Defined Networking (SDN)), les auteurs proposent d'intégrer des analyses basées sur les séries temporelles afin d'anticiper les pannes et d'améliorer la fiabilité du réseau.

La méthodologie repose sur la collecte de métriques clés du réseau, telles que l'utilisation du CPU et de la mémoire des composants du cœur 5G, la durée moyenne des opérations et le taux d'échec des procédures. Ces données sont extraites à l'aide de l'outil de benchmarking de la plateforme Open5GCore de Fraunhofer FOKUS, puis stockées dans une base de données InfluxDB via Prometheus. Pour la prédiction, le modèle de séries temporelles Facebook Prophet est utilisé, choisi pour sa capacité à gérer des ensembles de données limités tout en capturant les tendances et les saisonnalités.

Les résultats expérimentaux indiquent que cette approche permet de détecter efficacement les anomalies susceptibles de conduire à des défaillances système, avec une précision de prédiction atteignant 75%. Bien que les auteurs reconnaissent certaines limitations, notamment la taille restreinte des ensembles de données et les défis liés à la collecte de données dans des environnements instables, ils soulignent le potentiel de cette méthode pour améliorer la disponibilité et la performance des réseaux 5G. Ils suggèrent également que l'intégration continue de telles solutions analytiques pourrait offrir une surveillance en temps réel, renforçant ainsi la résilience des infrastructures réseau [42].

III.6.7 Article 07

Towards zero downtime : using machine learning to predict network failure in 5G and beyond

L'objectif clé de l'étude est de garantir une disponibilité continue du réseau en anticipant les défaillances avant qu'elles n'affectent les utilisateurs finaux.

Les auteurs de l'article proposent une approche de prédiction des pannes basée sur l'apprentissage automatique, appliquée à un réseau 5G en production. Plusieurs modèles supervisés sont comparés, notamment le Support Vector Regression (Support Vector Regression (SVR)), les forêts aléatoires et le Gradient Boosting Regressor. Les résultats montrent que le SVR amélioré est le plus efficace, atteignant un score f1 supérieur à 0,9. Ce modèle permet de prédire, avec une avance de 10 minutes, la probabilité qu'une panne soit susceptible de survenir [43].

III.6.8 Article 08

Fault Prediction and Reliability Analysis in a Real Cellular Network

L'article de Mulvey et al. (2017) explore l'application de techniques d'apprentissage automatique pour prédire les pannes et analyser la fiabilité dans un réseau cellulaire réel. L'objectif principal est de développer une approche proactive permettant d'anticiper les défaillances du réseau, réduisant ainsi les interruptions de service et améliorant la qualité globale du réseau.

Les auteurs ont utilisé des données opérationnelles collectées sur un réseau cellulaire en fonctionnement, incluant des indicateurs clés de performance (KPI) tels que le taux de réussite des appels, la qualité du signal et les taux de transfert de données. En appliquant des modèles d'apprentissage profond, notamment des réseaux de neurones, ils ont pu identifier des schémas précurseurs de pannes. L'étude révèle que ces approches prédictives sont particulièrement

efficaces pour détecter les anomalies avant qu'elles ne conduisent à des défaillances majeures, permettant ainsi une intervention préventive.

Les résultats expérimentaux démontrent que les modèles développés offrent une précision élevée dans la prédiction des pannes, surpassant les méthodes traditionnelles de surveillance réactive. Cette précision accrue permet aux opérateurs de réseau de mettre en place des stratégies de maintenance préventive plus efficaces, réduisant les coûts opérationnels et améliorant la satisfaction des utilisateurs [44].

III.6.9 Article 09

" MonTrees : Automated Detection and Classification of Networking Anomalies in Cellular Networks "

L'article de Mohamed Moulay et al. (2021) propose une méthode innovante pour automatiser la détection et la classification des anomalies dans les réseaux cellulaires. Face à la croissance rapide et à la complexité accrue des réseaux mobiles, les auteurs combinent des techniques d'apprentissage supervisé et non supervisé pour identifier et catégoriser les anomalies réseau.

La méthodologie repose sur l'analyse de données réelles collectées à partir de tests de conduite (drive-tests) et de la plateforme Measuring Mobile Broadband Networks in Europe (MONROE). Les auteurs utilisent des algorithmes de clustering non supervisés pour regrouper les comportements similaires, suivis de modèles d'apprentissage supervisé pour classifier précisément les types d'anomalies identifiées. Cette approche hybride permet de détecter automatiquement des problèmes tels que les dégradations de performance ou les défaillances de configuration, facilitant ainsi une intervention rapide et ciblée.

Les résultats expérimentaux démontrent l'efficacité de la méthode MonTrees, avec une capacité à identifier et classer les anomalies réseau de manière précise et en temps opportun. Cette approche améliore significativement les processus de dépannage dans les réseaux cellulaires, contribuant à une gestion plus autonome et efficace des infrastructures mobiles [45].

III.7 Tableau comparatif des articles

Le tableau suivant récapitule les articles déjà représentés avec leurs résultats ;

TABLEAU III.1 – Métriques de performance dans les réseaux cellulaires

Titre de l'article	Objectif principal	Méthodologie	Résultats clés
A Fault Diagnosis Method for 5G Cellular Networks Based on Knowledge and Data Fusion	Diagnostic des pannes dans les réseaux 5G via fusion de données et connaissances.	Utilisation d'un GAN pour équilibrer les classes, un modèle bayésien naïf (NBM) pour le pré-diagnostic, et un GCN pour affiner le diagnostic.	Précision : 90.56%, Score f1 : 88.41%.
Automatic Fault Detection and Diagnosis in Cellular Network and Beyond 5G	Détection automatique des pannes dans les réseaux 5G et au-delà.	Collecte des données (3 scénarios), clustering non supervisé, et réseaux neuronaux dynamiques supervisés.	Précision : 96.86%.
Cell Coverage Degradation Detection Using DL Techniques	Détection des dégradations de couverture cellulaire (cellules dormantes, signal détérioré).	Comparaison entre SVM et RNN avec différentes architectures (jusqu'à 3 couches cachées).	RNN avec une couche cachée offre la meilleure précision.
Cell Fault Management Using ML Techniques	Amélioration de la gestion des pannes par techniques ML (détection et correction).	Comparaison entre systèmes basés sur règles, SVM, Random Forest, et réseaux neuronaux profonds.	Les réseaux neuronaux profonds offrent la meilleure performance pour le diagnostic.
Cellular Fault Detection Using Spatio-Temporal Graph Convolutional Networks	Détection des pannes cellulaires via réseaux convolutionnels spatio-temporels sur graphes (GCN).	Modélisation graphique pour identifier causalités et prédire les défaillances futures.	Amélioration significative de la précision par rapport aux méthodes classiques.
System Failure Prediction Within Software 5G	Prédiction des défaillances dans le cœur du réseau 5G via analyse de séries temporelles.	Modèles ARIMA et LSTM appliqués aux logs réseau collectés via Open5GCore.	Identification des tendances anormales avant les pannes.

Titre de l'article	Objectif principal	Méthodologie	Résultats clés
Towards Zero Downtime	Prédiction proactive des pannes pour garantir disponibilité continue du réseau 5G.	Comparaison entre SVR, Random Forest, et Gradient Boosting Regressor.	SVR amélioré atteint un score $f1 > 0.9$, prédiction jusqu'à 10 minutes avant la panne.
Fault Prediction and Reliability	Approche proactive pour la prédiction des pannes et l'optimisation de la fiabilité réseau.	Modèles SVM, réseaux neuronaux profonds avec auto-encodeurs, chaîne de Markov en temps continu (Continuous-Time Markov Chain (CTMC)).	Les auto-encodeurs offrent la meilleure performance pour détecter les tendances cachées et prédire les pannes.
MonTrees	Méthodologie combinée pour détection et classification des anomalies réseau dans la transition vers la 5G.	Apprentissage supervisé (arbres de décision) et non supervisé (k-means).	Identification précise des causes sous-jacentes des problèmes réseau.

III.8 Limites

Bien que les technologies d'apprentissage automatique facilitent le développement de méthodes d'auto-réparation pour les réseaux cellulaires, plusieurs défis majeurs existent qui peuvent affecter la performance et les implémentations pratiques. Nous classifions ces défis en cinq catégories suivantes :

- ★ Déséquilibre des données : dans les réseaux cellulaires, en raison de la rareté de certains événements (par exemple, les pannes de réseau), les ensembles de données collectés sont généralement déséquilibrés.
- ★ Insuffisance de données : le manque de données de haute qualité peut entraîner un sur-apprentissage (overfitting) important des modèles d'apprentissage automatique (par exemple, des classificateurs).
- ★ Insensibilité au coût : une certaine proportion d'erreurs de classification est inévitable, mais les modèles ne prennent pas toujours en compte l'impact réel (ou le coût) de ces erreurs.
- ★ Réponse non temps réel : la plupart des mécanismes d'auto-réparation existants ne répondent pas aux exigences de réactivité en temps réel, en raison de leur nature réactive [46].

III.9 Conclusion

Une analyse détaillée des travaux précédents liés à l'utilisation de l'intelligence artificielle pour la détection et la prédiction des pannes de réseaux cellulaires révèle que les approches classiques sont remplacées par des méthodes plus avancées et intelligentes. En particulier, les méthodes supervisées, telles que SVM, Random Forest ou KNN, se sont avérées efficaces dans la classification des anomalies et des pannes à partir des KPI du réseau, même pour les cas de LTE. Néanmoins, ces méthodes ont des limites de scalabilité et de capacité d'adaptation dans un contexte dynamique.

Les réseaux de neurones profonds, y compris les LSTM, les CNN et les DNN, ont permis d'utiliser au mieux les variations temporelles et spatiales dans les indicateurs. Tous ces développements ont donné de bons résultats dans la prévision proactive des défaillances. Enfin, les modèles hybrides qui combinent la détection de défaillance et la prédiction, la classification, le clustering et la régression et offrent des justifications de leur prévision marquent une étape séduisante vers de tels systèmes.

Chapitre IV

Conception du modèle prédictif

IV.1 Introduction

Pour assurer une bonne qualité de service dans les réseaux cellulaires, il est essentiel de pouvoir détecter rapidement les pannes et, si possible, les prévoir avant qu'elles ne surviennent. Ce chapitre présente les différentes étapes de conception d'un modèle de prédiction des pannes réseau.

Nous commençons par décrire les types de données utilisés pour alimenter le modèle, comme les indicateurs de performance du réseau, les journaux système, les alarmes ou encore les historiques de pannes. Ensuite, nous détaillons les étapes nécessaires à la construction du modèle : chargement et préparation des données, choix de l'approche et évaluation des résultats obtenus.

L'objectif de ce chapitre est de montrer comment transformer des données brutes en un outil de prédiction fiable, capable d'aider les opérateurs à mieux gérer leur réseau, pour assurer une qualité de service (QoS) fiable et optimale.

IV.2 Types de données utilisées pour la détection et la prédiction des pannes

Les sources de jeux de données utilisés pour l'entraînement des modèles de l'intelligence artificielle sont nombreuses. Selon l'article [16], voici les plus utilisées :

IV.2.1 Données de performance réseau (KPIs)

Comme on les a déjà présentés en haut, dans le premier chapitre, les indicateurs de performance, ou KPIs, constituent une source essentielle d'information pour évaluer la qua-

lité de service (QoS) offerte par un réseau cellulaire. Ils permettent de mesurer des aspects clés tels que le taux de réussite des appels (CSSR : Call Setup Success Rate (CSSR)), le taux de coupure des appels (Call Drop Rate (CDR)), le débit (throughput) et la latence. Ces indicateurs sont collectés de manière continue et servent à détecter toute dégradation de performance. Par exemple, une hausse soudaine du taux de coupure peut indiquer un problème de couverture ou une surcharge du réseau. Les KPIs sont fréquemment utilisés dans des modèles prédictifs pour anticiper les anomalies avant qu'elles n'impactent l'utilisateur final.

IV.2.2 Logs système

Des journaux qui enregistrent des informations détaillées sur les événements et les états des composants du réseau, tels que les routeurs, les commutateurs et les liens de communication. Ils sont essentiels pour détecter les anomalies, diagnostiquer les pannes et prévoir les défaillances potentielles.

IV.2.3 Données d'alarme

Des données générées par divers équipements du réseau (comme les routeurs, commutateurs et stations de base), signalent des événements inhabituels ou des anomalies pouvant indiquer des défaillances imminentes ou en cours.

Dans TelOps, les données d'alarme passent par plusieurs étapes pour être utiles ;

- Couche de données : on commence par collecter et nettoyer les alarmes venant de différents équipements. Comme ces équipements parlent des "langages" différents (car ils viennent de plusieurs fabricants), cette étape sert à harmoniser les données pour qu'elles soient compréhensibles et comparables.
- Couche de connaissances : ensuite, on utilise l'expérience des techniciens et les habitudes connues pour interpréter ces alarmes. Cela permet de voir si certains types d'alertes se répètent ou sont liés entre eux.
- Couche d'apprentissage automatique : puis, des modèles d'IA (comme les réseaux de neurones) sont utilisés pour analyser ces alarmes automatiquement. Ils peuvent repérer des anomalies, prévoir des pannes et même suggérer quoi faire pour éviter ou résoudre les problèmes.
- Couche d'application : enfin, les résultats sont utilisés pour agir concrètement : repérer les pannes, trouver ce qui ne va pas et organiser les réparations.

Ces données jouent un rôle central dans la surveillance et la maintenance des réseaux de télécommunications.

IV.2.4 Données historiques des pannes

Selon TelOps, ces données sont très précieuses puisqu'elles :

- Elles permettent de comprendre les anciens problèmes, les causes fréquentes de pannes et les réactions apportées. Cela aide à repérer des tendances ou des erreurs qui reviennent souvent.
- Elles permettent d'identifier les signes d'une panne.
- Enfin, grâce à ces données, on peut prédire les pannes futures et même suggérer des actions préventives.

IV.3 Étapes de conception du modèle

La conception des modèles de prédiction avec l'intelligence artificielle (IA) repose sur toute une architecture selon [47] ; en partant de la phase de chargement jusqu'à la phase de l'évaluation du modèle. Ces étapes seront détaillées dans ce présent élément.

IV.3.1 Chargement des données

La première étape de la démarche de conception d'un modèle de prédiction, c'est le chargement des données à exploiter pour alimenter le modèle de l'intelligence artificielle (Deep Learning ou Machine Learning) ou ce qu'on appelle un **Dataset**. Donc, le dataset est un ensemble de données organisées souvent sous forme d'un tableau, où chaque ligne représente un élément et chaque colonne une caractéristique de cet élément. Dans ce contexte de prédiction de pannes, les lignes représentent une analyse de trafic réseau cellulaire qui est caractérisée par le temps, par exemple, l'identité de la cellule, la perte de paquets, la latence, ... et qui représentent les colonnes de ce tableau. A vrai dire, le dataset est le résultat de la collecte de données présentée précédemment.

Charger un dataset consiste en lire ce tableau dans un programme informatique. La lecture de dataset dépend de son extension ; c-à-d le format dans lequel les données sont enregistrées. On trouve donc plusieurs façons, .csv (Comma-Separated Values : valeurs séparées par des virgules), .txt (séparées par tabulations, espaces, ...), .xlsx (Excel), .json (JavaScript Object Notation), .xml (eXtensible Markup Language) ou d'autres extensions.

Donc, chaque chargement dépend de l'extension du dataset à exploiter.

IV.3.2 Prétraitement des données

La phase qui suit celle du chargement du dataset est le prétraitement des données, qui sera développée dans ce qui suit.

Selon l'article [48], les algorithmes d'apprentissage sont sensibles aux valeurs d'entrées, ce qui rend l'étape de prétraitement des données cruciale, puisque :

1. Les données brutes sont souvent imparfaites :
les données telles qu'on les collecte sont souvent incomplètes, ou mal formatées. Si on les utilise telles quelles, les modèles risquent de mal apprendre, ou d'apprendre des choses fausses.
2. Les modèles sont sensibles aux données :
Par exemple, si une variable prend des valeurs très grandes et une autre très petites, certains algorithmes auront du mal à bien fonctionner. Il faut donc normaliser les échelles pour qu'aucune variable ne domine les autres.
3. Quand les données sont bien préparées (nettoyées, normalisées, transformées si besoin), les algorithmes apprennent plus rapidement et de manière plus fiable.

Le prétraitement des données constitue une phase cruciale du processus d'apprentissage automatique. Il vise à améliorer la qualité des données et à les adapter aux exigences des algorithmes utilisés. Les principales étapes mentionnées sont les suivantes :

IV.3.2.1 Nettoyage des données

Cette étape consiste à détecter et corriger les erreurs, valeurs manquantes ou aberrantes dans les données. Le nettoyage peut inclure la suppression d'observations incomplètes, le remplacement des valeurs manquantes par des moyennes ou des médianes, ou encore l'identification des données extrêmes (outliers).

IV.3.2.2 Transformation des données

Le document souligne l'importance de transformer les variables lorsque cela est nécessaire. Cela peut inclure des transformations logarithmiques pour réduire la variabilité, ou des transformations de variables catégorielles en représentations numériques (comme l'encodage one-hot).

IV.3.2.3 Mise à l'échelle (standardisation ou normalisation)

Il est essentiel d'ajuster l'échelle des variables, surtout si elles présentent des ordres de grandeur différents. La normalisation (valeurs entre 0 et 1) ou la standardisation (centrée réduite) permet de garantir que chaque variable contribue équitablement à l'analyse.

IV.3.2.4 Réduction de dimensions

Le document évoque l'utilisation de techniques comme l'Analyse en Composantes Principales (Analyse en Composantes Principales (ACP)/Principal Component Analysis (PCA)) pour réduire le nombre de variables tout en conservant l'essentiel de l'information. Cette étape est utile pour faciliter la visualisation ou diminuer le bruit dans les données.

IV.3.2.5 Détection d'anomalies

Le prétraitement peut inclure l'identification d'observations atypiques qui peuvent fausser l'analyse. Ces anomalies peuvent être traitées ou retirées selon le contexte.

IV.3.3 Choix de l'approche

Pour pouvoir effectuer un choix de l'approche de l'IA à exploiter, il faut passer par une série d'étapes selon [48], qui consistent en :

1. La nature du problème

Cette étape consiste à bien identifier l'objectif de la prédiction et à formaliser le problème qu'on cherche à résoudre, et qui peut être :

- **Classification** : de prédire à quelle catégorie ou classe appartient une donnée (par exemple panne ou non).
- **Régression** : de prédire une valeur numérique continue à partir de données d'entrée (par exemple, prédire la température demain).
- **Clustering (regroupement)** : de regrouper automatiquement des données similaires sans connaître les catégories à l'avance (par exemple, regrouper des clients selon leurs comportements).
- **Détection d'anomalies** : d'identifier des comportements inhabituels dans les données.
- **La prédiction de séries temporelles** : est de prédire l'évolution d'une valeur dans le temps.

Cette étape dépend directement du choix de l'approche (type du modèle à utiliser).

2. Les données disponibles

La qualité, la quantité et la représentation influencent fortement le choix de l'approche et affectent ses performances ;

- Types de variables : elles sont numériques, catégorielles, temporelles, textuelles, ...
- Qualité des données : données manquantes, déséquilibrées, ...
- Volume de données : petit, moyen ou grand jeu de données.
- Nombre de variables (features) : faible ou élevé.

3. Sélection des modèles candidats

Cette étape consiste à dresser une liste ou une table des options d'algorithmes par familles ;

- Méthodes classiques : régression linéaire, logistique, ...
- Arbres de décision et forêts aléatoires : RandomForest, extra-trees, ...
- Méthodes à noyaux : Support Vector Machine (SVM).
- Méthodes à voisinage : K-Nearest Neighbors (KNN)
- Boosting : eXtreme Gradient Boosting (XGBoost), CatBoost, ...
- Réseaux de neurones : convolutionnels, récurrents, ...

4. Correspondance entre le modèle et le type de données

Dans cette étape, on analyse l'adéquation de l'approche avec le dataset ou le jeu de données disponible. Voici un aperçu selon les articles [50] et [51],

- ★ Le **SVM** et le **SVR** sont utiles lorsqu'on dispose de petits jeux de données avec une forte dimension (plusieurs variables). SVM pour la classification et SVR pour la régression. Ces deux modèles sont robustes mais lents sur de gros volumes de données.
- ★ Le **XGBoost** et **Light Gradient Boosting Machine (LightGBM)** sont performants et rapides lorsqu'il s'agit des données tabulaires, ils sont même tolérants aux valeurs manquantes, mais peu adaptés pour les images et les textes bruts.
- ★ Le **Random Forest** est robuste, simple et bon pour les données tabulaires. Il est moins précis qu'XGBoost, mais plus interprétable.
- ★ Le **KNN** est utile pour les petits jeux de données, mais coûteux pour prédire et sensible au bruit.
- ★ Le **MLP** est un réseau de neurones polyvalent pour les données normalisées, performant avec un bon tuning et des données suffisantes.
- ★ Le **CNN** est excellent pour les images et les vidéos, très puissant, mais nécessite de gros volumes de données et de la GPU.
- ★ Pour **RNN**, **LSTM** et **GRU** sont adaptés aux données temporelles (temps, texte), performants pour les prédictions temporelles.

- ★ Le quant à lui, il traite les données en graphes (topologies, réseaux), il gère les relations entre les entités.
- ★ Le **st-GCN** combine à la fois le temps et les graphes, il est idéal pour les données spatio-temporelles (capteurs, trafic).
- ★ Pour les **transformers**, comme **Generative Pre-trained Transformer (GPT)** et **Bidirectional Encoder Representations from Transformers (BERT)** sont spécialisés dans le traitement du langage naturel NLP, ils sont très performants lorsqu'il s'agit de la traduction, de la classification du texte, de la génération de contenu, mais nécessitent beaucoup de données et de puissance de calcul (GPU).

IV.3.4 Évaluation du modèle

Selon [49], il existe plusieurs critères pour évaluer la performance d'un modèle prédictif, qu'on va présenter dans cet élément.

IV.3.4.1 Matrice de confusion (Confusion matrix)

La matrice de confusion est un outil d'évaluation important en apprentissage automatique supervisé, notamment dans les tâches de classification. Elle permet d'évaluer la performance d'un modèle en comparant ses prédictions aux résultats réels.

Dans le cas d'une classification binaire, la matrice de confusion est une table à deux dimensions, comme le présente la table suivante :

	Prédit	
Réel	Négatif	Positif
Négatif	TN	FP
Positif	FN	TP

TABLEAU IV.1 – Matrice de confusion pour une classification binaire

ou :

- **Vrai Positif True Positives (TP)** : le modèle prédit correctement la classe positive.
- **Faux Positif False Positives (FP)** : le modèle prédit la classe positive alors que l'instance est en réalité négative.
- **Faux Négatif False Negatives (FN)** : le modèle prédit la classe négative alors que l'instance est en réalité positive.

- **Vrai Négatif True Negatives (TN)** : le modèle prédit correctement la classe négative.

IV.3.4.2 Rappel (Recall)

On appelle rappel, ou sensibilité (sensitivity), le taux de vrais positifs, c-à-d la proportion d'exemples positifs correctement identifiés comme tels :

$$\text{Rappel} = \frac{\text{Vrai Positifs (TP)}}{\text{Vrai Positifs (TP)} + \text{Faux Négatifs (FN)}}$$

IV.3.4.3 Précision

On appelle précision, ou valeur positive prédictive (positive predictive value, PPV) la proportion de prédictions correctes parmi les prédictions positives :

$$\text{Précision (Precision)} = \frac{TP}{TP + FP}$$

Remarque

L'anglais distingue precision (la précision ci-dessus) et accuracy (exactitude), qui est la proportion d'exemples correctement étiquetés, soit le complémentaire à 1 du taux d'erreur, aussi traduit par précision en français. On utilisera donc ces termes avec précaution.

$$\text{accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

ou

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

IV.3.4.4 F-measure

On appelle F-mesure (F-score ou f1 en anglais) la moyenne harmonique de la précision et du rappel :

$$\text{F-mesure (f1)} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

IV.3.4.5 Le taux d'erreur moyen (AER)

Le taux d'erreur moyen est une mesure simple qui évalue la proportion de prédictions incorrectes par rapport au total. Il est surtout utilisé en classification.

$$\text{Taux d'erreur moyen} = \frac{\text{Nombre de prédictions incorrectes}}{\text{Nombre total de prédictions}} \times 100$$

ou : Taux d'erreur = 1 - accuracy

IV.3.4.6 Les courbes ROC

Selon [52], la courbe ROC est une représentation graphique qui permet d'évaluer les performances d'un modèle de classification binaire en analysant sa capacité à distinguer entre les classes positives et négatives.

Principe de fonctionnement

Elle trace, pour tous les seuils possibles de décision, le taux de vrais positifs (True Positive Rate (TPR)) ou recall (sensibilité) en fonction du taux de faux positifs (False Positive Rate (FPR)). Ce qui permet de visualiser le compromis entre ces deux entités (sensibilité et spécificité). Plus la courbe s'approche du coin supérieur gauche, plus le modèle est performant. En effet, l'aire sous la courbe (Area Under the Curve (AUC)) quantifie cette performance (AUC est un score entre 0 et 1 qui mesure la capacité d'un modèle à distinguer entre deux classes (positif / négatif)) : une AUC de 1 indique une classification parfaite, tandis qu'une AUC de 0,5 indique une performance équivalente à un modèle aléatoire.

La courbe ROC est utile même dans les cas où les classes sont déséquilibrées, car elle n'est pas affectée ou influencée par la distribution des classes dans le jeu de données.

Ces étapes sont récapitulées dans le schéma ci-après, comme le représente la figure IV.1.

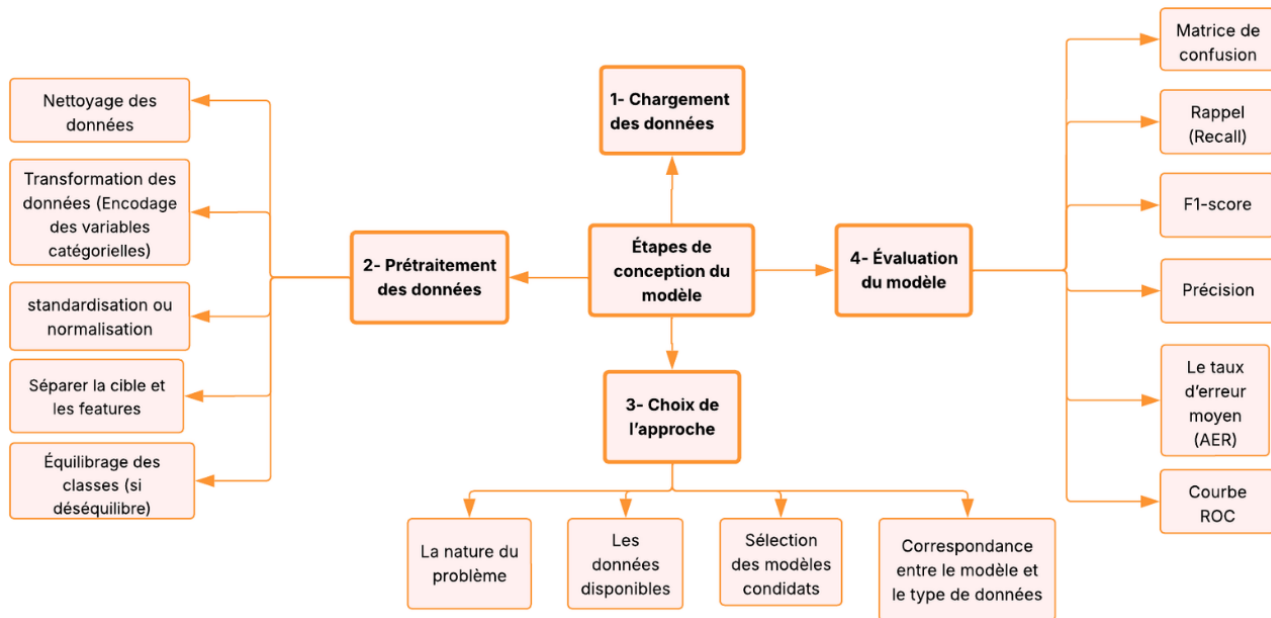


FIGURE IV.1 – Schéma représentatif du processus de conception d'un modèle prédictif

IV.4 Conclusion

La conception d'un modèle de prédiction des pannes constitue une étape clé dans la mise en place d'un système intelligent de surveillance des réseaux cellulaires. Dans ce chapitre, nous avons présenté les différentes sources de données exploitées, ainsi que les étapes nécessaires à la préparation, à la modélisation et à l'évaluation du système.

L'analyse des données réseau, qu'il s'agisse des KPIs, des logs système, des alarmes ou des historiques de pannes, permet d'alimenter le modèle avec des informations riches et pertinentes. Grâce à un prétraitement rigoureux et au choix réfléchi des techniques de modélisation, il devient possible de détecter les défaillances de manière plus précise, voire de les anticiper.

Les résultats obtenus à l'aide d'indicateurs comme la précision, le rappel ou la F-mesure permettent d'évaluer l'efficacité du modèle et d'orienter les améliorations futures. Ainsi, ce travail de conception représente une base solide pour le développement d'un système prédictif capable de renforcer la fiabilité et la continuité des services dans les réseaux cellulaires.

Chapitre V

Implémentation et expérimentation

V.1 Introduction

Après avoir présenté les fondements théoriques, ce chapitre est consacré à la phase pratique du travail, à savoir l'implémentation et l'expérimentation. L'objectif de ce présent chapitre est d'implémenter et d'expérimenter ces fondements en code, d'entraîner les modèles choisis sur les jeux de données disponibles et d'évaluer leurs performances à l'aide de ce qu'on a déjà présenté dans le chapitre précédent.

Dans un premier temps, nous présentons dans l'annexe A l'environnement de développement utilisé, à savoir Jupyter Notebook et Google Colab, en association avec le langage de programmation Python. Dans un second temps, l'annexe B décrit les principales bibliothèques utilisées pour l'implémentation, à savoir Pandas, NumPy, Scikit-Learn, TensorFlow et PyTorch. Chacune de ces bibliothèques joue un rôle bien défini dans la manipulation des données, la modélisation et l'entraînement des modèles. Ce chapitre présente les jeux de données utilisés, les étapes de leur préparation, ainsi que les critères ayant guidé le choix des algorithmes appliqués.

Enfin, les résultats d'entraînement sont présentés, analysés et interprétés de manière à évaluer la pertinence des approches proposées.

V.2 Description des datasets utilisés

Pour pouvoir concevoir un modèle prédictif de pannes dans les réseaux cellulaires selon notre thématique, on est obligé de suivre une suite d'instructions, comme on l'a déjà mentionné dans le chapitre précédent, où la première phase consiste à charger le data set. Avant de pouvoir effectuer cette étape, on doit disposer d'un data set.

Comme les données de trafic réseau sont considérées sensibles et confidentielles, les opérateurs mobiles ou ceux qui s'occupent de la collecte des données du réseau restreignent l'accès à leurs informations, ce qui a rendu les datasets de cette catégorie souvent indisponibles.

Sur cet effet, nous avons fait le choix de deux jeux de données parmi ceux qui sont disponibles sur Internet.

V.2.1 4G LTE Speed Dataset and Bandwidth

Le data set **LTE Dataset** disponible [53] est un ensemble de données contenant 135 traces de mesures 4G LTE, chacune durant en moyenne quinze minutes, avec des débits observés allant de 0 Mbps à des valeurs plus élevées. Ces traces enregistrent des informations détaillées sur la performance du réseau LTE.

- **Timestamp** : date et heure de l'échantillon mesuré.
- **Longitude** et **Latitude** : coordonnées GPS de l'appareil mobile au moment de la mesure.
- **Velocity** : vitesse de déplacement de l'appareil mobile en kilomètres par heure (kph).
- **Operatorname** : nom de l'opérateur cellulaire.
- **CellId** : l'identifiant unique de la cellule concernée.
- **NetworkMode** : type de communication mobile (3G/4G).
- **RSRP** : puissance du signal reçu, indicateur de la qualité du signal LTE.
- **Reference Signal Received Quality (RSRQ)** : qualité du signal reçu, combinant puissance et interférences.
- **Received Signal Strength Indicator (RSSI)** : la puissance brute du signal radio reçu par un récepteur en dBm.
- **Signal-to-Noise Ratio (SNR)** : la qualité du signal en comparant la puissance du signal utile à celle du bruit de fond.
- **SINR** : rapport signal sur interférences et bruit, mesure de la qualité du signal.
- **Channel Quality Indicator (CQI)** : indicateur de la qualité du canal radio.
- **DL_bitrate (DownLink_bitrate)** : (ou débit descendant) représente la vitesse de transmission des données du réseau vers l'utilisateur.
- **UL_bitrate (UpLink_bitrate)** : (ou débit montant) représente la vitesse de transmission des données de l'utilisateur vers le réseau.
- **State** : état du processus de téléchargement. Il peut avoir deux valeurs : soit I (idle, not downloading, inactif) ou D (Downloading, en cours de téléchargement).
- et **Neighbor Radio Received Signal Received Quality (NRxRSRP)** : Valeurs RSRQ et RSRP pour la cellule voisine.
- **Cell_Longitude** et **Cell_Latitude** : coordonnées GPS de l'eNodeB desservi.

- **Distance** : distance entre la cellule de service et l'appareil mobile en mètres.

Ce dataset est utilisé pour l'analyse des performances réseau, la prédiction des débits ou la détection d'anomalies dans les réseaux.

V.2.2 5G-NWDAF-Data

Le dataset `nwdaf_data` est un fichier Comma-Separated Values (CSV) structuré contenant plusieurs colonnes correspondant à des mesures et événements liés au réseau 5G, un jeu de données synthétique développé par le groupe de recherche 5G de l'Université Bogazici, conçu spécifiquement pour la fonction NetWork Data Analytics Function (NWDAF) dans les réseaux 5G. Le dataset est disponible sur le GitHub [54].

Ce dataset contient les colonnes suivantes :

- **t** : le temps ou l'instant auquel la mesure a été prise.
- **cell_id** : l'identifiant unique de la cellule réseau concernée par la mesure, permettant de localiser la donnée dans le réseau 5G.
- **cat_id** : la catégorie d'abonné, qui peut représenter différents profils ou types d'utilisateurs (par exemple, abonnés grand public, entreprises, IoT).
- **pe_id** : l'identifiant de l'équipement utilisateur (User Equipment ID), indiquant le type ou l'unité spécifique d'appareil connecté.
- **load** : la charge réseau mesurée dans la cellule à l'instant t, reflétant l'utilisation des ressources ou le trafic réseau.
- **has_anomaly** : une valeur binaire (0 ou 1) indiquant la présence ou l'absence d'une anomalie dans la cellule à l'instant t.

Le data set est étiqueté (contient la cible à prédire, qui est "has_anomaly" dans ce cas), ce qui permet d'entraîner des modèles supervisés pour la classification et la prédiction d'anomalies.

V.3 Choix des algorithmes utilisés

V.3.1 4G LTE data set

Ce dataset n'étant pas étiqueté, des méthodes d'apprentissage non supervisé ont été appliquées, notamment les modèles **Isolation Forest** et **OneClass SVM**.

Il est possible d'entraîner des modèles d'apprentissage supervisé après avoir créé une variable cible artificielle représentant l'état du réseau à prédire. La variable **Panne** a ainsi été construite sur la base de seuils prédéfinis : lorsque certaines mesures dépassent ces valeurs

critiques, un comportement anormal est suspecté (panne = 1); dans le cas contraire, le fonctionnement du réseau est considéré comme normal (panne = 0).

Les seuils utilisés pour RSRP, RSRQ, SINR, CQI et les débits sont basés sur les spécifications du 3GPP (notamment TS 36.133, 36.213, 36.214), ainsi que sur les valeurs de référence adoptées par les principaux équipementiers (Huawei, Nokia, Ericsson) et organismes de mesure réseau (Keysight, Anritsu, ITU). Ces seuils sont largement utilisés dans le cadre de l'évaluation de la qualité de service LTE et de la détection des dégradations réseau.

Les seuils utilisés pour détecter une anomalie dans le réseau sont les suivants :

- ★ **RSRP** < **-110 dBm** : le signal reçu est très faible.
- ★ **RSRQ** < **-12 dB** : la qualité du signal est très dégradée.
- ★ **SNR** < **5 dB** : le rapport signal/bruit est bas, donc le signal est peu fiable.
- ★ **CQI** < **5** : l'indicateur de qualité du canal est mauvais.
- ★ **Débit descendant (DL_bitrate)** < **1 Mbps** : la vitesse de téléchargement est trop faible.
- ★ **Débit montant (UL_bitrate)** < **0,5 Mbps** : la vitesse d'envoi est très basse.

Si au moins 3 de ces seuils sont dépassés, la situation est considérée comme anormale (présence d'une panne ou dégradation du service).

Après avoir créé la variable cible "panne" et visualisé sa distribution à l'aide de l'histogramme présenté à la figure V.1, un déséquilibre important a été constaté. Celui-ci a été corrigé à l'aide de la technique Synthetic Minority Over-sampling Technique (SMOTE), mentionnée précédemment.

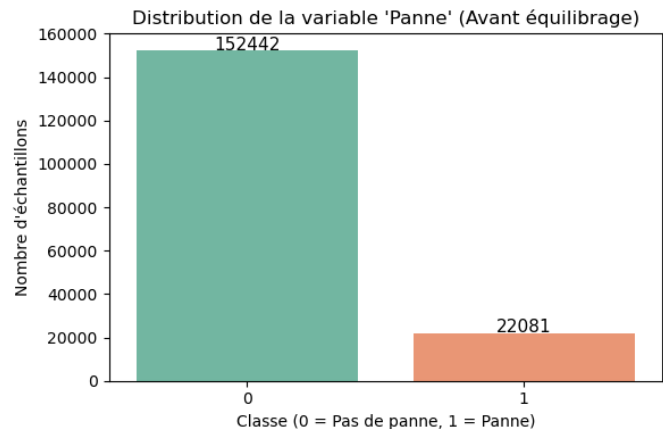


FIGURE V.1 – Schéma représentatif du déséquilibre de "**Panne**"

Les modèles entraînés sur ce jeu de données sont : SVM, GRU, LSTM, KNN, KNN, Decision Tree (DT) Naive Bayes (NB) et Logistic Regression (LR).

V.3.2 5G-NWDAF-Data

Ce dataset présente un déséquilibre important entre les classes, comme l'illustre la figure V.2. Cette situation peut affecter négativement les performances des modèles d'apprentissage, car ces derniers ont tendance à privilégier et prédire la classe majoritaire, qui est généralement associée à l'absence de panne (classe 0), tout en négligeant la classe minoritaire, représentant les pannes (classe 1).

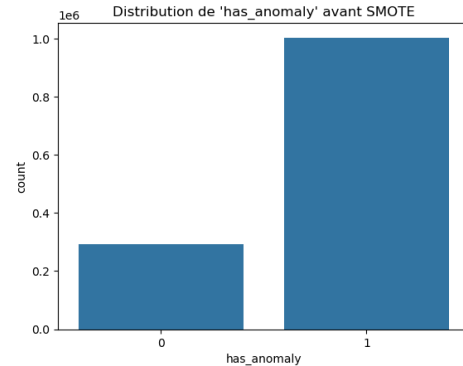


FIGURE V.2 – Schéma représentatif du déséquilibre de "**has_anomaly**"

Il existe plusieurs méthodes pour équilibrer les classes d'un dataset. Parmi les plus courantes, on trouve le **sous-échantillonnage (undersampling)**, qui consiste à réduire la taille de la classe majoritaire en supprimant certains de ses échantillons. Cette méthode est rapide et permet de diminuer le temps d'apprentissage, mais elle présente l'inconvénient potentiel de faire perdre des informations importantes en éliminant des données utiles.

À l'inverse, le **suréchantillonnage (oversampling)** augmente la taille de la classe minoritaire. Deux approches principales peuvent être utilisées : le Random OverSampling, qui duplique aléatoirement des exemples de la classe minoritaire, et **SMOTE**, qui génère de nouveaux exemples synthétiques en interpolant entre des échantillons proches de cette classe. Cette dernière méthode permet de mieux préserver la diversité et la qualité des données.

Certaines méthodes permettent de gérer le déséquilibre directement au niveau du modèle. ; XGBoost, RandomForest, SVM peuvent être ajustés avec un poids pour les classes ; c'est-à-dire donner plus de poids à la classe minoritaire lors de l'entraînement, etc.

Nous avons fait le choix de la fonction **SMOTE**, pour garder les informations du dataset et améliorer la précision du modèle. En effet, contrairement à un simple suréchantillonnage qui duplique les exemples, SMOTE crée des points synthétiques, réduit le risque d'overfitting et améliore la capacité du modèle à apprendre les caractéristiques de la classe minoritaire.

Après avoir appliqué les traitements nécessaires sur le dataset — tels que la suppression des valeurs aberrantes, le nettoyage, la normalisation et l'équilibrage des classes — nous passons à l'étape d'entraînement du modèle. Le dataset est alors divisé en deux sous-ensembles : les **données d'entraînement (train)**, qui permettent au modèle d'apprendre à distinguer les cas de pannes des situations normales, et les **données de test**, utilisées pour évaluer

la qualité de l'apprentissage et la capacité du modèle à généraliser ses prédictions sur de nouvelles données.

Dans ce cas d'étude, la variable cible à prédire est `has_anomaly`, une variable binaire indiquant l'état du réseau : une valeur de 0 correspond à un fonctionnement normal, tandis qu'une valeur de 1 signale une panne. Étant donné qu'il s'agit d'un problème de classification binaire, nous avons opté pour une approche d'apprentissage supervisé. Plusieurs modèles supervisés ont été explorés, notamment la forêt aléatoire (Random Forest), l'arbre de décision (Decision Tree), le KNN, le SVM, le , ainsi que des modèles de Gradient Boosting tels que XGBoost, Gradient Boosting Machine (GBM), Categorical Boosting (CatBoost), la régression logistique (Logistic Regression) et MLP. Par ailleurs, le dataset contenant une dimension temporelle via la variable 't', nous avons également mobilisé des réseaux de neurones supervisés adaptés aux données séquentielles, en particulier les modèles LSTM et GRU.

Remarque

La fonction SMOTE modifie les lignes de manière indépendante, elle casse l'ordre temporel des données du dataset, ce qui est dangereux pour les modèles séquentiels. Pour entraîner LSTM et GRU, une méthode spécifique est adoptée afin de préserver la nature temporelle des données.

Les modèles LSTM et GRU, appliqués à ce dataset dans le cadre de la détection d'anomalies réseau reposent sur la **transformation des données tabulaires en séquences temporelles**. Cette étape permet de capturer les dynamiques d'évolution des indicateurs réseau sur une fenêtre temporelle donnée. Afin de remédier à l'éventuel déséquilibre des classes, notamment dû à la rareté des anomalies, une technique de suréchantillonnage telle que le **Random OverSampling** est appliquée aux séquences générées. Le modèle récurrent ainsi entraîné est capable d'apprendre les patrons temporels caractéristiques des comportements normaux et anormaux, ce qui améliore significativement la capacité de détection des anomalies dans des environnements réseau dynamiques.

V.4 Résultats d'entraînement et interprétation

V.4.1 4G LTE

V.4.1.1 Les graphes (non-supervisé)

Les représentations graphiques V.3 et V.4 illustrent la variation des valeurs d'anomalie (ou score d'anomalie : Le score d'anomalie est une mesure utilisée dans le domaine de la détection d'anomalies pour évaluer à quel point un point de données diffère du compor-

tement normal observé dans un ensemble de données) qui résultent de l'entraînement des modèles **OneClass SVM** et **Isolation Forest**, mettant en évidence des pics significatifs qui pourraient indiquer des anomalies.

Dans la figure V.4, nous observons que les valeurs identifiées par le modèle **Isolation Forest** apparaissent en rouge, sur un fond de mesures qui varient autour de zéro. Ce graphique montre la capacité du modèle à isoler des instances anormales.

Le graphique V.3, quant à lui, utilise le modèle **OneClass SVM**, qui présente également des valeurs d'anomalie variables. Les anomalies apparaissent à des niveaux inférieurs à zéro, ce qui montre l'efficacité de cette méthode pour capter des comportements anormaux dans les données.

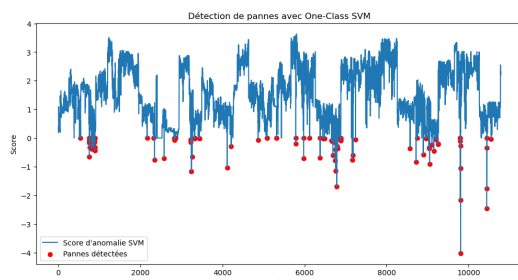


FIGURE V.3 – OneClass SVM

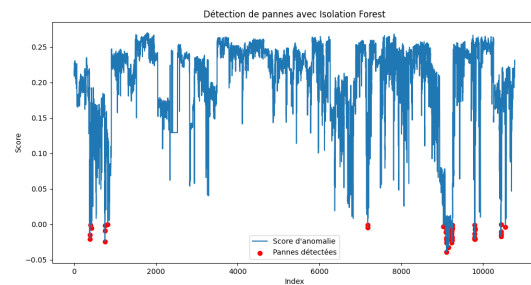


FIGURE V.4 – Isolation Forest

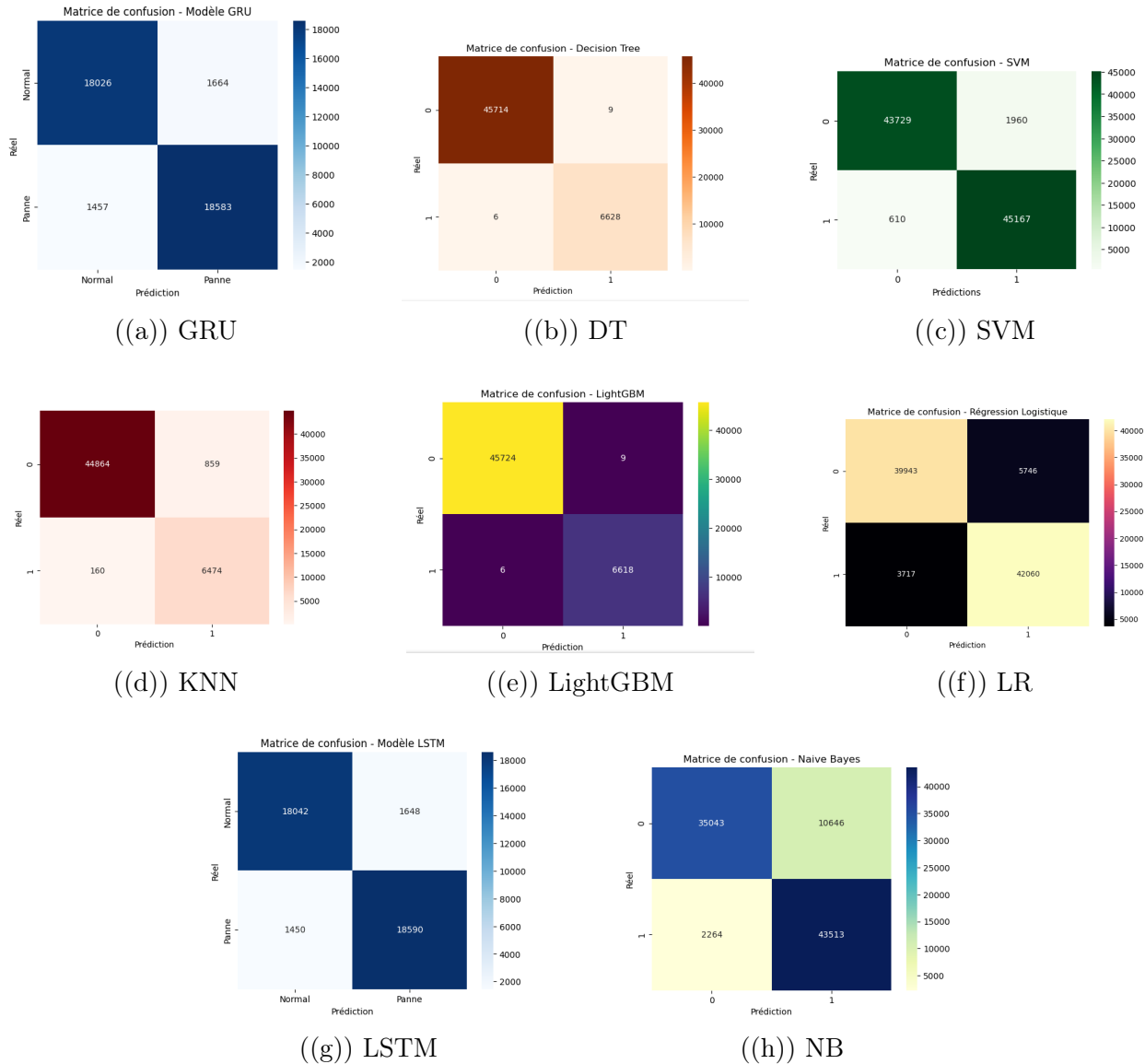
V.4.1.2 Les matrices de confusion

Les matrices illustrées dans la figure V.5 présentent les résultats de l'apprentissage supervisé appliqué au jeu de données **4G LTE**, mettant en évidence les performances des modèles de prédiction entraînés, y compris les réseaux de neurones profonds supervisés tels que LSTM et GRU.

Ces modèles ont été entraînés sur le jeu de données après l'avoir rendu étiqueté, grâce à la création de la variable cible « panne », comme décrit précédemment dans les sections antérieures.

La matrice de confusion du modèle GRU, représentée dans la figure V.5(a), montre une bonne capacité de détection, avec 18583 pannes et 18026 cas normaux correctement identifiés. Les erreurs restent modérées : 1664 fausses alertes et 1457 pannes non détectées, ce qui traduit une performance globale satisfaisante.

Selon la matrice de confusion représentée dans la figure V.5(b), DT affiche une performance remarquable avec 45714 cas normaux et 6628 pannes correctement prédites. Les erreurs sont très faibles : seulement 9 fausses alertes et 6 pannes manquées, ce qui traduit une excellente capacité de classification.


 FIGURE V.5 – Matrices de confusion obtenues sur le dataset **4G LTE**

La matrice de confusion V.5(d), montre que KNN distingue très bien les deux classes : il a correctement identifié 44864 cas négatifs et 6474 cas positifs. Les erreurs restent faibles, avec seulement 859 faux positifs et 160 faux négatifs. Cela indique une excellente capacité.

La matrice de confusion V.5(e), montre que le modèle LightGBM est très performant : il a correctement classé la quasi-totalité des exemples, avec 45724 vrais négatifs et 6618 vrais positifs. Les erreurs sont extrêmement rares, avec seulement 9 faux positifs et 6 faux négatifs, ce qui reflète une excellente précision et un très bon rappel.

La matrice de confusion V.5(f) indique que LR fonctionne globalement bien, avec 42060

pannes et 39943 cas normaux correctement identifiés. Les erreurs restent modérées : avec 5746 faux positifs et 3717 faux négatifs.

La matrice de confusion V.5(h), montre que NB a correctement identifié 35043 cas négatifs et 43513 cas positifs. Les fausses valeurs prédites sont 10646 faux positifs et 2264 faux négatifs. Cela indique une moyenne capacité de prédiction.

La matrice de confusion V.5(c), montre que SVM a correctement identifié 43729 cas négatifs et 45167 cas positifs. Les valeurs mal prédites sont 1960 faux positifs et 610 faux négatifs. Cela indique une bonne capacité de prédiction.

La matrice de confusion du modèle LSTM, représentée dans la figure V.5(g), montre une moyenne capacité de détection, avec 18590 pannes et 18042 cas normaux correctement identifiés. Les erreurs sont assez faibles : 1648 fausses alertes et 1450 pannes non détectées.

V.4.1.3 Les courbes ROC

Les courbes représentées dans la figure V.6 résultent de l'apprentissage supervisé des modèles de prédiction déjà cités, montrant ainsi leurs performances. L'axe des X représente le taux de faux positifs (FPR), tandis que l'axe des Y indique le taux de véritables positifs (TPR).

La figure V.6(a) illustre la courbe ROC pour GRU, avec un AUC de 0,98. La courbe est proche du coin supérieur gauche montre une excellente performance du modèle, suggérant qu'il identifie correctement la majorité des véritables positifs tout en maintenant un faible taux de faux positifs.

La courbes présentées dans les figures V.6(b) pour DT et V.6(e) pour LightGBM montrent que ces modèles sont parfaitement performants. En effet, le TPR atteint 1.0, tandis que le taux de FPR est révolu à 0, résultats signifiés par les lignes horizontales situées en haut du graphique. L'aire sous la courbe (AUC) étant de 1.00, cela indique que les modèles classifient parfaitement toutes les instances, ce qui les rendent très efficaces pour la tâche qu'ils sont censés accomplir.

La courbe ROC présentée dans la figure V.6(c) montre les performances de SVM en termes de classification binaire. L'AUC est de 1.00, ce qui indique une performance parfaite du modèle, car il parvient à identifier presque toutes les vraies positives sans générer de faux positifs.

La courbe ROC présentée dans la figure V.6(d) illustre les performances de KNN. La courbe, qui atteint presque le coin supérieur gauche, indique une performance excellente du modèle, se traduisant par un AUC de 0,99. Cela suggère que le modèle est particulièrement

efficace pour distinguer entre les classes, minimisant ainsi la probabilité de faux positifs tout en maximisant les vrais positifs.

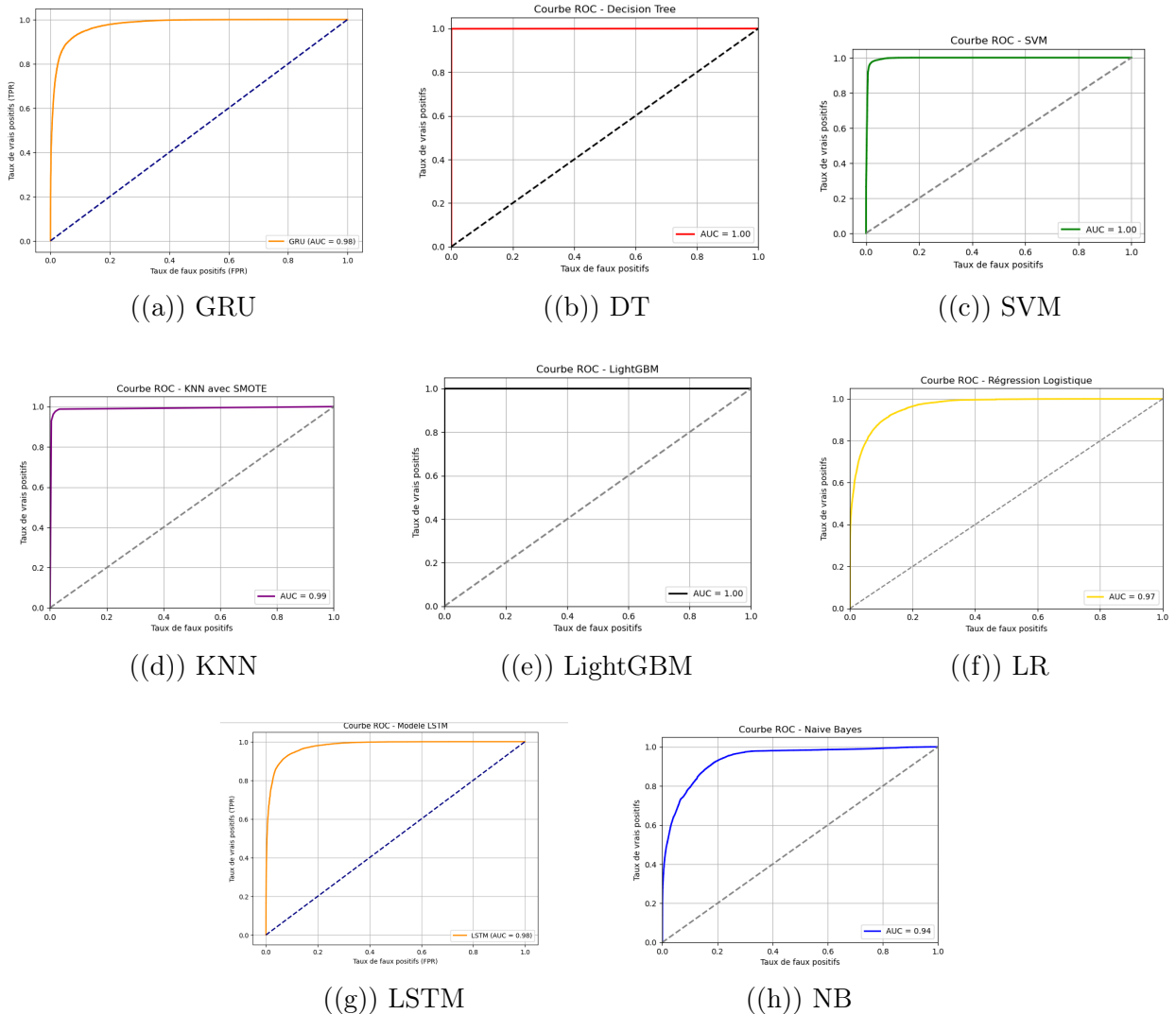


FIGURE V.6 – Courbe ROC des modèles entraînés sur le dataset **4G LTE**

La courbe ROC présentée dans la figure V.6(f) illustre la performance de LR. Avec une AUC de 0,97, le modèle affiche une excellente capacité à discriminer entre les classes positives et négatives, indiquant qu’il fait peu d’erreurs de classification. La courbe se rapproche significativement du coin supérieur gauche, ce qui montre une performance de prédiction très efficace.

La courbe ROC du modèle LSTM illustrée dans la figure V.6(g), présente un taux de vrais positifs (TPR) presque parfait, atteignant presque 1, ce qui indique une excellente

capacité du modèle à identifier correctement les cas positifs. Le score AUC de 0,98 témoigne d'une performance globale très élevée.

La courbe ROC présentée dans la figure V.6(h) montre l'efficacité de NB dans la classification. La courbe se rapproche du coin supérieur gauche, indiquant une performance excellente. Avec une AUC de 0.94, cela signale que le modèle est capable de distinguer correctement entre les classes avec une grande précision, ce qui en fait un outil efficace pour les tâches de classification.

V.4.1.4 Mesures de performance

À partir des matrices représentées dans les sections précédentes, les principales mesures de performance sont calculées, à savoir l'exactitude (accuracy), la précision (precision), le rappel (ou sensibilité / recall) et le f1-score.

TABLEAU V.1 – Résultats d'entraînement des modèles supervisés

Modèle	SVM	GRU	LSTM	DT	KNN	NB	LR	LightGBM
Accuracy	0.967761	0.9214	0.9221	0.9997	0.9805	0.8589	0.8963	0.9997
Recall	0.986675	0.9273	0.9276	0.9991	0.976	0.9505	0.9188	0.9991
Precision	0.958397	0.9180	0.9186	0.9986	0.883	0.8032	0.8798	0.9986
f1-score	0.972343	0.9226	0.9230	0.9986	0.926	0.8699	0.8997	0.9986
AER	0.032239	0.0785	0.0779	0.0002865	0.0195	0.1411	0.1037	0.0003

Selon les résultats d'expérimentation présentés dans le tableau précédent, on peut conclure par la classification suivante :

1. Modèles très performants

Les modèles arborescents tels que DT et LightGBM ont montré d'excellentes performances sur le dataset LTE, caractérisé par des données tabulaires et des seuils bien définis.

- LightGBM, en tant que modèle d'ensemble, a atteint une accuracy supérieure à 99.9% et un f1-score dépassant 0.99, grâce à sa robustesse face à la complexité des données réseau.
- DT, quant à lui, tire parti de sa capacité à générer des règles simples et interprétables, bien adaptées à la structure logique des indicateurs techniques LTE.

2. Modèles moyennement performants

SVM, LSTM, GRU et KNN avec une accuracy entre 92% et 97%, sont moins performants sur ce type de données.

- SVM est sensible au volume du dataset.
- LSTM et GRU sont adaptés aux données séquentielles temporelles, ils ne sont pas toujours optimaux sur les données tabulaires classiques. Mais aussi le jeu de données doit être volumineux.
- KNN sensible à la distance, dépend du choix du paramètre k .

3. Modèles moins performants

- l’hypothèse d’indépendance de variables supposée par NB a dégradé ces performances dans ce contexte, ou les variables sont corrélées ; l’une dépend de l’autre.
- LR reste intermédiaire, fonctionne bien mais loin des modèles modernes.

L’analyse comparative des performances des modèles supervisés sur le dataset LTE a mis en évidence la supériorité du modèle LightGBM . Ce modèle a obtenu d’excellents résultats.

Cependant, les modèles classiques comme NB et LR sont moins efficaces à causes de leurs principes de fonctionnement limités face à ce genre de données. Quand aux modèles séquentiels, comme LSTM et GRU n’ont pas su bien tirer parti de la structure des données disponibles.

Pour conclure, les modèles LightGBM et DT sont les plus adaptés et les plus efficaces pour la détection des pannes dans les réseaux LTE.

V.4.2 5G nwdaf_data

V.4.2.1 Matrices de confusion

La figure V.7 présente les matrices de confusion de différents modèles de classification appliqués à la détection des pannes.

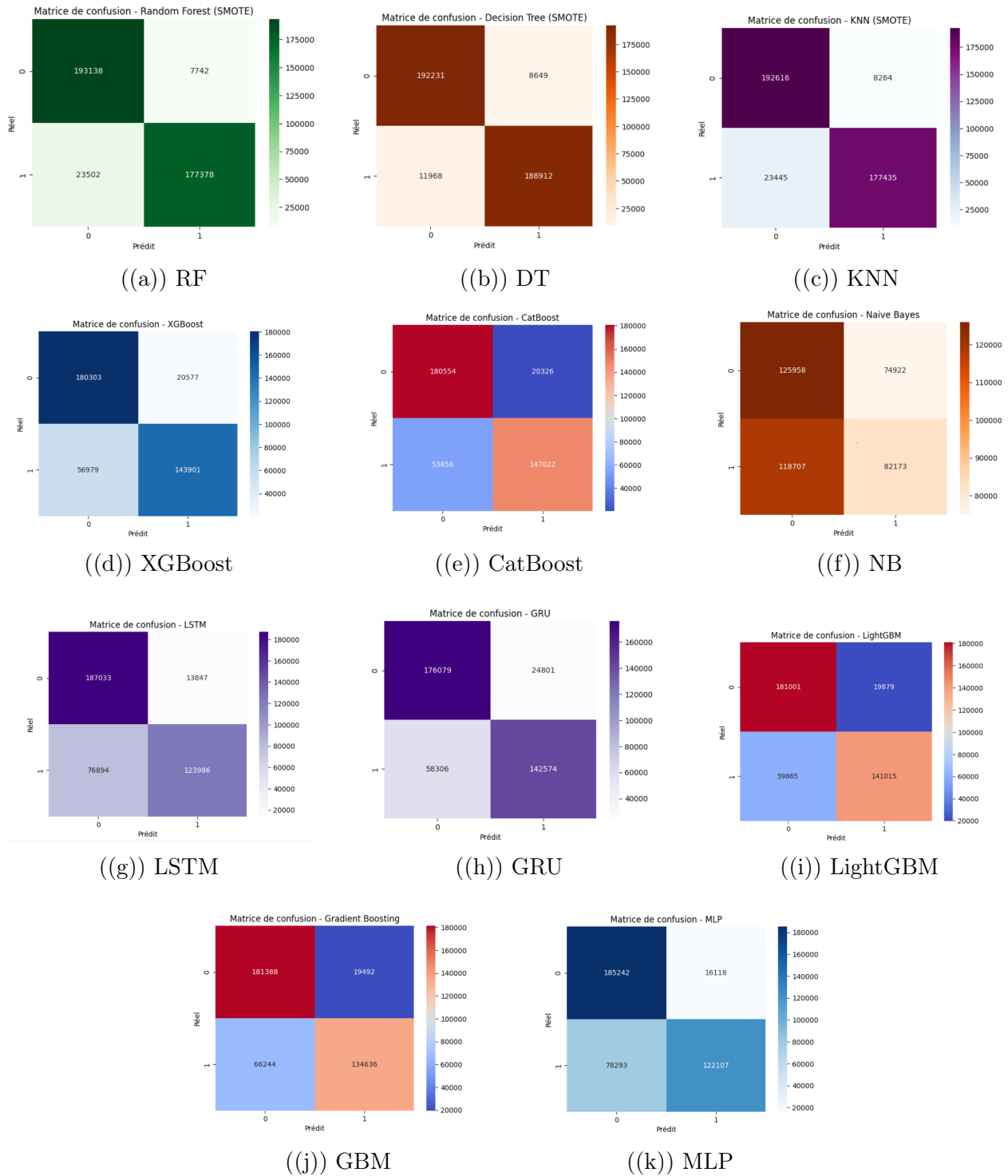
Le modèle **Random Forest** (Figure V.7(a)) montre une excellente performance avec **193 138** cas correctement classés comme sans panne (vrais négatifs) et **177 378** pannes correctement détectées (vrais positifs). Il présente peu de fausses alertes **7 742** (faux positifs) et un nombre modéré de pannes manquées **23 502** (faux négatifs), ce qui en fait un modèle très équilibré.

Le **Decision Tree** (Figure V.7(b)) affiche également des résultats solides avec **192 231** vrais négatifs et **188 912** vrais positifs, contre seulement **8 649** faux positifs et **11 968** faux négatifs, indiquant une bonne sensibilité et une précision fiable.

Le modèle **KNN** (Figure V.7(c)) est similaire, avec **191 116** cas sans panne bien détectés

et **177 435** cas de pannes bien identifiés, bien qu'il ait commis **8 244** fausses alertes et **23 456** erreurs de non-détection, ce qui le rend légèrement moins performant que DT et RF.

Le modèle **XGBoost** (Figure V.7(d)), malgré sa puissance, a produit **20 577** fausses alertes et **56 979** pannes manquées, avec **180 303** vrais négatifs et **143 901** vrais positifs. Ces résultats révèlent un rappel plus faible, ce qui peut poser problème pour la détection exhaustive des pannes.


 FIGURE V.7 – Matrices de confusion obtenues sur le dataset `nwdaf_data`

De même, **CatBoost** (Figure V.7(e)) affiche **53 858** faux négatifs et **20 206** faux positifs, contre **180 554** vrais négatifs et **147 922** vrais positifs, indiquant une tendance similaire à XGBoost avec un compromis entre précision et sensibilité.

Le modèle **Naive Bayes** (Figure V.7(f)) présente les performances les plus faibles, avec **74 032** fausses alertes et **118 707** pannes non détectées, malgré **125 958** cas normaux bien classés et seulement **83 173** pannes correctement détectées, ce qui le rend peu adapté à ce type de tâche.

Côté réseaux de neurones, le **LSTM** (figure V.7(g)) reconnaît **193 253** cas sans panne mais omet **76 094** pannes (faux négatifs), tout en détectant **123 906** pannes réelles (vrais positifs). Cela montre une faible sensibilité malgré un bon taux de vrais négatifs.

Le **GRU** (Figure V.7(h)) montre un profil plus équilibré avec **176 079** vrais négatifs et **142 574** vrais positifs, mais encore **58 306** pannes non détectées et **24 031** fausses alertes, ce qui reste perfectible.

Le **LightGBM** (Figure V.7(i)) enregistre **191 051** vrais négatifs et **146 015** vrais positifs, contre **58 645** faux négatifs et **20 879** faux positifs, le plaçant en milieu de tableau.

Le **GBM** (Figure V.7(j)) réalise **181 388** vrais négatifs et **134 945** vrais positifs, mais au prix de **64 244** faux négatifs et **19 942** fausses alertes, ce qui indique un rappel modéré.

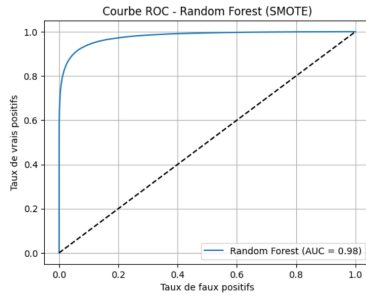
Enfin, Le **MLP** (Figure V.7(k)), quant à lui, est le moins performant parmi les modèles avancés avec **78 239** pannes non détectées et seulement **122 897** correctement identifiées, bien que **185 242** cas sans panne soient bien reconnus, accompagnés de **16 118** fausses alertes.

V.4.2.2 Les courbes ROC

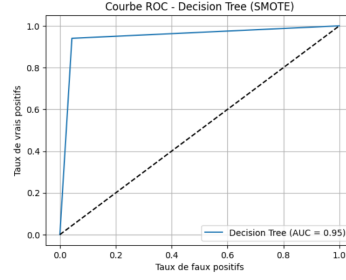
La figure V.8 illustre les courbes ROC issues de l'évaluation comparative des différents modèles de classification appliqués au dataset **nwdaf_data**. Ces courbes révèlent des écarts significatifs de performance entre les modèles. Parmi ceux-ci, le modèle **Random Forest**, dont la courbe ROC est illustrée dans la figure V.8(a), montre une très forte performance avec une AUC de 0.98. La courbe est nettement incurvée vers le coin supérieur gauche, ce qui indique une excellente capacité de discrimination entre les classes. Cela suggère que le modèle identifie correctement les classes positives tout en minimisant les faux positifs.

Dans la figure V.8(b), le modèle DT atteint une AUC de 0.95. Bien que la performance soit légèrement inférieure à celle de Random Forest, la courbe est suffisamment éloignée de la diagonale pour indiquer une bonne classification.

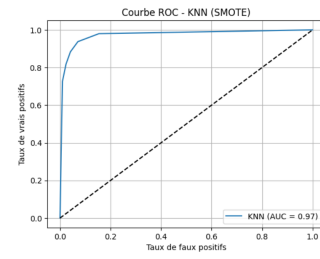
La courbe ROC du KNN, représentée dans la figure V.8(c), est très proche du coin supérieur gauche, correspondant à une AUC de 0.97. Cela témoigne d'une excellente capacité du modèle à distinguer les classes, avec un compromis optimal entre rappel et précision.



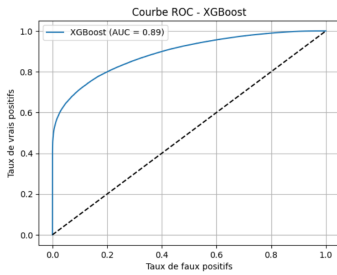
((a)) Random Forest



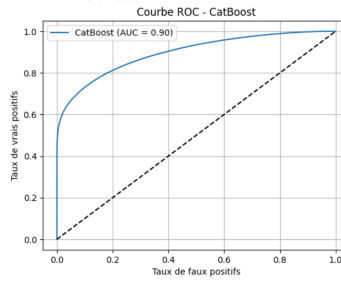
((b)) Decision Tree



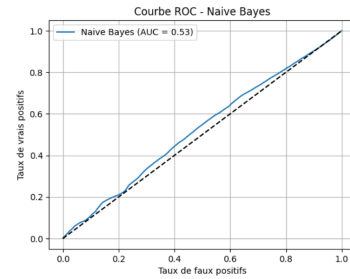
((c)) KNN



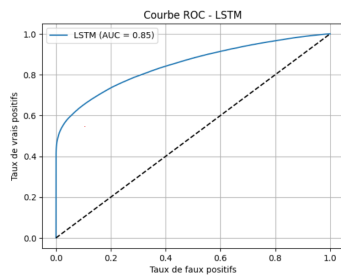
((d)) XGBoost



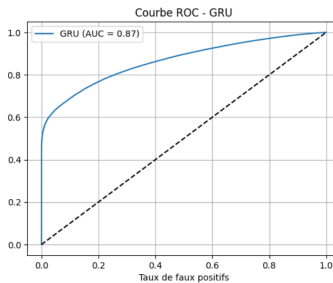
((e)) CatBoost



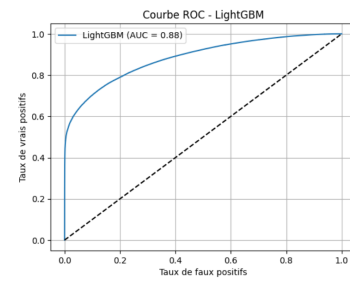
((f)) Naive Bayes



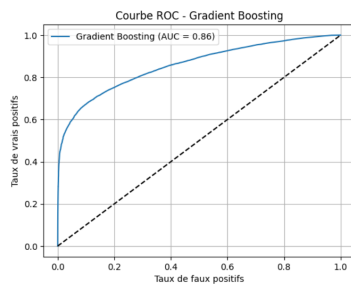
((g)) LSTM



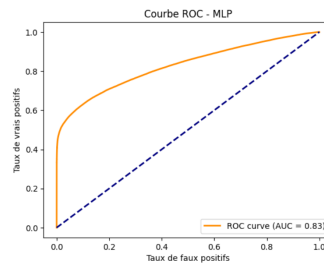
((h)) GRU



((i)) LightGBM



((j)) GBM



((k)) MLP

FIGURE V.8 – Courbe ROC des modèles entraînés sur le dataset `NWDAF_Data`

Le modèle XGBoost, dans la figure V.8(d), présente une AUC de 0.89. La courbe est bien formée, mais légèrement moins élevée que celle de KNN ou RF. Cela indique une performance correcte, mais une sensibilité potentielle aux données déséquilibrées ou bruitées.

Dans la figure V.8(e), CatBoost atteint une AUC de 0.90. Sa courbe ROC montre une bonne capacité de classification. Il se classe parmi les modèles efficaces, tout en offrant une robustesse intéressante face à des données catégorielles ou complexes.

Le modèle **Naive Bayes**, affiché dans la figure V.8(f), montre une AUC proche de **0.53**, traduisant une capacité de discrimination à peine supérieure au hasard.

Dans la figure V.8(g), le modèle LSTM atteint une AUC de 0.85. Sa courbe ROC montre une forte capacité à capter les séquences temporelles et à effectuer une classification efficace, ce qui en fait un bon choix pour les données dépendantes du temps.

Le modèle GRU, en figure V.8(h), affiche une AUC de 0.87. Bien qu'un peu moins performant que LSTM, il conserve une capacité de généralisation satisfaisante. Moins coûteux en calcul, il peut être préféré pour des environnements avec des ressources limitées.

Le modèle LightGBM, illustré dans la figure V.8(i), affiche des performances très satisfaisantes avec une AUC proche de 0.88, confirmant sa robustesse sur des données tabulaires complexes.

La courbe ROC du modèle GBM, en figure V.8(j), présente une AUC de 0.86. Ce résultat témoigne d'une performance très proche de celle de LightGBM, avec une bonne généralisation sur les données testées.

Enfin, dans la figure V.8(k), le modèle MLP atteint une AUC de 0.83. Bien que la courbe soit moins prononcée que celles des modèles basés sur les arbres ou séquentiels, elle indique une performance correcte. Cela suggère que le MLP pourrait bénéficier d'un jeu de données plus volumineux.

En conclusion, cette analyse met en évidence la supériorité des modèles de type ensemble (RF, XGBoost, LightGBM) et des réseaux récurrents (LSTM et GRU) pour la détection d'anomalies dans les données du dataset **nwdaf_data**, tandis que les approches classiques comme **Naive Bayes** présentent des résultats nettement moins performants dans ce contexte.

V.4.2.3 Mesures de performance

Les résultats d'entraînement des modèles déjà cités sur le data set **nwdaf_data** sont récapitulés dans le tableau ci-après :

TABLEAU V.2 – Résultats d’entraînement des modèles supervisés

Modèle	Accuracy	Précision	Recall	F1-score	AER
RF	0.92	0.9582	0.8819	0.9181	0.08
DT	0.95	0.9562	0.9433	0.9497	0.05
KNN	0.92	0.9554	0.8809	0.9162	0.08
XGBoost	0.81	0.8796	0.7189	0.7908	0.19
CatBoost	0.82	0.8814	0.7199	0.7920	0.18
NB	0.52	0.5230	0.4114	0.4612	0.48
LSTM	0.77	0.9129	0.6171	0.7354	0.23
GRU	0.79	0.8518	0.7138	0.7138	0.21
LightGBM	0.80	0.8646	0.7105	0.7796	0.20
GBM	0.79	0.8748	0.6767	0.7622	0.21
MLP	0.77	0.8833	0.6119	0.7230	0.23

À partir des matrices représentées dans les sections précédentes, les principales mesures de performance sont calculées, à savoir l’exactitude (*accuracy*), la précision (*precision*), le rappel (ou sensibilité / *recall*) et le *f1-score*.

1. Modèles très performants

- DT, RF et KNN se distinguent par une *accuracy* supérieure ou égale à 0.92 et un *f1* supérieur à 0.91. Leur performance remarquable s’explique par leur capacité à bien modéliser les données tabulaires issues du réseau, en capturant efficacement les relations entre les attributs. Le DT, en particulier, atteint un *f1-score* exceptionnel de 0.9497.

2. Modèles moyennement performants

- GRU, LSTM, MLP, CatBoost, LightGBM et GBM présentent une *accuracy* comprise entre 0.77 et 0.82, avec un *f1* variant entre 0.72 et 0.79. Ces résultats traduisent une performance acceptable mais inférieure aux meilleurs modèles.
- Les modèles séquentiels tels que LSTM et GRU sont conçus pour exploiter des dépendances temporelles. Or, dans le cas présent, les données tabulaires ne comportent pas de séquences temporelles, ce qui limite leur efficacité.
- Les modèles d’ensemble tels que LightGBM et XGBoost bien qu’efficaces dans de nombreux contextes, sont ici légèrement pénalisés, probablement en raison de

la taille importante du dataset, qui peut entraîner un surapprentissage ou une sensibilité accrue aux déséquilibres de classes.

- Le MLP, malgré sa capacité à capturer des relations non linéaires, présente une performance modérée en raison d'un volume de données insuffisant pour ce type de réseau.

3. Modèles les moins performants

- NB est le modèle le moins performant, avec une *accuracy* de seulement 0.52 et un Average Error Rate (AER) très élevé (0.48). Cela s'explique par l'hypothèse d'indépendance entre les variables, rarement vérifiée dans les données réseau, où les attributs sont souvent fortement corrélés.

L'analyse comparative des performances des modèles supervisés sur le dataset **nwdaf_data** révèle que les modèles basés sur les arbres de décision, tels que DT, RF et KNN, affichent les meilleures performances globales. En revanche, les modèles d'ensemble plus complexes, tels que XGBoost, GBM et LightGBM, présentent des performances inférieures. Par ailleurs, les modèles séquentiels comme LSTM et GRU n'ont pas su exploiter efficacement la structure des données.

En conclusion, les modèles DT, RF et KNN apparaissent comme les plus adaptés à la détection des pannes dans les réseaux représentés par le dataset **nwdaf_data**, surpassant les modèles séquentiels ainsi que certains modèles d'ensemble pourtant réputés performants.

V.5 Limites

Après avoir finaliser ce travail, nous avons constaté quelques limites qui ralentissent en quelque sorte l'évolution de la conception et l'implémentation du coté pratique de ce projet. Notamment :

- ★ **La qualité et la disponibilité des données**

Les données réelles des opérateurs sont souvent confidentielles, ce qui restreint l'accès. Les datasets publics peuvent ne pas refléter toutes les conditions réelles.

- ★ **Deséquilibre des classes**

Les pannes sont des évènements rares comparés au comportement normal, ce qui mène à un fort déséquilibre dans les classes. Ce déséquilibre dégrade les performances des modèles, même si des techniques comme SMOTE ou UnderSampling peuvent aider.

- ★ **Complexité des modèles restreinte**

L'utilisation des modèles avancés comme LSTM et GRU est limitée par des contraintes de ressources (Random Access Memory (RAM), Graphics Processing Unit (GPU)) ou de temps, notamment lorsqu'il s'agit d'un jeu de données volumineux.

★ **Généralisation des résultats**

Les performances obtenus dans un cadre contrôlé ne garantissent pas la même efficacité en conditions réelles.

★ **Outils et environnement limités**

Les ressources techniques comme la capacité des machines peuvent limiter les possibilités d'expérimentation.

V.6 Conclusion

Ce chapitre a présenté l'ensemble du processus d'implémentation, depuis le choix de l'environnement de développement jusqu'à l'analyse des résultats obtenus. L'utilisation d'outils puissants comme Jupyter, Google Colab et de bibliothèques telles que Scikit-Learn,... Ces moyens ont permis d'entraîner et évaluer efficacement plusieurs modèles de ML et DL. Les jeux de données LTE et NWDAF ont été exploités après une suite d'instructions, notamment la préparation et la gestion du déséquilibre grâce à des méthodes telles que SMOTE. Les résultats obtenus montrent des performances variables selon les caractéristiques de chaque modèle, avec une supériorité des modèles d'ensemble pour la détection des pannes. Cette expérimentation met en valeur l'importance de bien sélectionner les modèles et les méthodes de prétraitement pour assurer une détection fiable et automatisée des anomalies dans les réseaux mobiles.

Conclusion générale

Ce projet a permis de tester des solutions utilisant l'IA pour détecter les pannes dans les réseaux cellulaires. L'objectif était de trouver comment repérer à temps les défaillances qui peuvent nuire à la qualité de service dans les réseaux LTE, en utilisant les données techniques recueillies sur le terrain. Pour cela, nous avons utilisé des méthodes d'apprentissage supervisé capables de reconnaître les pannes à partir de données historiques.

Dans un premier temps, nous avons revisité les principes fondamentaux des réseaux cellulaires, en mettant en évidence les indicateurs clés qui peuvent identifier une anomalie ou une dégradation du service. Cette étape nous a permis de mieux comprendre le contexte technique du problème et d'orienter nos choix de traitement de données. Ensuite, nous avons réalisé une veille scientifique et un aperçu de l'existant, afin d'identifier les approches d'IA les plus adaptées dans le domaine des télécommunications.

Sur cette base, nous avons mis en place une démarche d'analyse de données : depuis le nettoyage et le prétraitement des données jusqu'à l'entraînement et la comparaison de plusieurs modèles de classification. Les algorithmes testés incluent notamment SVM, RF, XGBoost, DT et LSTM. Chaque modèle a été évalué à l'aide d'indicateurs de performance standards tels que l'accuracy, le recall, la précision, le F1-score ou encore le taux d'erreur moyen (AER). Ces métriques nous ont permis de comparer les différentes performances obtenues et la robustesse de chaque approche.

Les résultats montrent que les modèles arborescents comme DT et LightGBM, permettent d'obtenir des prédictions fiables quand il s'agit du data set **4G LTE**, quant au deuxième, **nwdaf_data**, les modèles classiques DT, RF et KNN sont les plus performants et qui permettent de prédire le plus de pannes dans le réseau 5G, illustré dans ce data set.

Toutefois, notre démarche a également mis en évidence plusieurs limites importantes. Celles-ci concernent notamment la qualité et la quantité des données disponibles, le déséquilibre entre classes, la sensibilité des modèles à certains paramètres, ou encore la complexité de certains algorithmes.

Au mépris de ces difficultés, le travail réalisé constitue une étape vers l'intégration de l'IA dans les processus de supervision et de maintenance des réseaux cellulaires. Il ouvre la

voie à de futures perspectives, que ce soit en enrichissant les données utilisées, en testant d'autres architectures de réseaux de neurones ou en développant des systèmes déployables en temps réel.

Bibliographie

- [1] *Académie des sciences. (2021). Rapport sur la 5G et les réseaux de communications mobiles.*, https://www.academie-sciences.fr/pdf/rapport/2021_07_12_rapport_5G.pdf
- [2] *Coupechoux, M. (2019). Cellular Architectures and Protocols. Télécom Paris.* <https://marceaucoupechoux.wp.imt.fr/files/2019/03/11-architectureandprotocolsv2.pdf>
- [3] *Commsbrief. (n.d.). What is the difference between Node B, eNodeB, ng-eNB and gNB? Commsbrief.* <https://commsbrief.com/what-is-the-difference-between-node-b-enodeb-ng-enb-and-gnb/>
- [4] *Sutton, A., Linge, N. (2015). Mobile Network Architecture Evolution : 1G to 4G. Academia.edu.* https://www.academia.edu/13885065/Mobile_Network_Architecture_Evolution_1G_to_4G
- [5] *Author(s). (Year). The Different Architectures Used in 1G, 2G, 3G, 4G, and 5G Networks. Springer.* https://link.springer.com/chapter/10.1007/978-3-031-10650-7_7
- [6] *Gougui, Y. (s.d.). Guide des Pannes Courantes et Solutions dans les Infrastructures Télécoms.* <https://fr.scribd.com/document/832707143/Guide-des-Pannes-Courantes-et-Solutions-dans-les-Infrastructures-Telecoms>
- [7] *Anritsu. (2019). LTE eNodeB Troubleshooting Guide. Armada Test.* <https://dl.cdn-anritsu.com/en-us/test-measurement/files/Manuals/Troubleshooting-Guides/11410-00566C.pdf>
- [8] *Wikipedia contributors. (2025). Atténuation du signal. Wikipedia.* https://fr.wikipedia.org/wiki/Att%C3%A9nuation_du_signal

-
- [9] *Poynting Antennas. (2021). Signal strength measure RSRP, RSRQ and SINR Reference for LTE 5G Cheat Sheet.* <https://poynting.tech/articles/antenna-faq/signal-strength-measure-rsrp-rsrq-and-sinr-reference-for-lte-cheat-sheet/>
- [10] *Boccardi, F., Heath, R. W., Lozano, A., Marzetta, T. L., Popovski, P. (2014). Five disruptive technology directions for 5G. IEEE Communications Magazine, 52(2), 74–80.* <https://doi.org/10.1109/MCOM.2014.6736746>
- [11] *Liu, K., Wang, W., Poor, H. V. (2017). Mobility management in 5G networks : A survey. IEEE Communications Surveys Tutorials, 19(4), 2466–2493.* <https://doi.org/10.1109/COMST.2017.2722921>
- [12] *Zhang, J., Cheng, N., Wang, J. (2019). Core network reliability in LTE and 5G systems : Challenges and solutions. IEEE Communications Magazine, 57(7), 32–38.* <https://doi.org/10.1109/MCOM.2019.1800627>
- [13] *Zhang, Y., Ansari, N. (2020). On harnessing the security challenges in 5G networks : A survey. IEEE Communications Surveys Tutorials, 22(1), 196–248.* <https://doi.org/10.1109/COMST.2019.2953857>
- [14] *García, F., Sánchez, L. (2018). User-related issues and troubleshooting in mobile networks. In Mobile Networks and Management (pp. 123-138). Springer.* https://doi.org/10.1007/978-3-319-94840-8_8
- [15] *Garg, V. K., Kappes, S. (2018). Fundamentals of Cellular Network Performance : Key Performance Indicators and Quality Metrics. Wiley.* <https://doi.org/10.1002/9781119389954>
- [16] *Yang, Y., Yang, S., Zhao, C., Xu, Z. (2024). TelOps : AI-driven operations and maintenance for telecommunication networks. IEEE Communications Magazine, 62(4), 104-110.* <https://arxiv.org/pdf/2412.04731>
- [17] *Alsedrah, M. K. (2017, December). Artificial intelligence. The American University of the Middle East.* <https://www.proparco.fr/fr/article/les-impacts-economiques-et-sociaux-de-la-telephonie-mobile-etude-de-cas-en-haiti>
- [18] *Bisong, E. (2019). Building machine learning and deep learning models on Google Cloud Platform : A comprehensive guide for beginners. Apress.* <https://doi.org/10.1007/978-1-4842-4470-8>

-
- [19] Sarker, I. H. (2021). *Machine learning : Algorithms, real-world applications and research directions*. *SN Computer Science*, 2(160). <https://doi.org/10.1007/s42979-021-00592-x>
- [20] Abbasi, M., Shahraki, A., Taherkordi, A. (2021). *Deep learning for network traffic monitoring and analysis (NTMA) : A survey*. *Computer Communications*, 170, 19–41. <https://doi.org/10.1016/j.comcom.2021.01.021>
- [21] Sarker, I. H. (2021). *Deep learning : A comprehensive overview on techniques, taxonomy, applications and research directions*. *SN Computer Science*, 2(420). <https://doi.org/10.1007/s42979-021-00815-1>
- [22] Labonne, M., Chatzinakis, C., Olivereau, A. (2021). *Predicting bandwidth utilization on network links using machine learning*. *arXiv*. <https://arxiv.org/abs/2112.02417>
- [23] Kumar, R., Muntz, R. R., Estrin, D. (2010). *Challenges in enabling data-centric wireless sensor networks for habitat monitoring*. *ACM Transactions on Sensor Networks (TOSN)*, 1(2), 1–33. <https://doi.org/10.1145/1105660.1105663>
- [24] Santos, L., Melo, L., Costa, R. (2015). *Using time series forecasting for proactive management in mobile networks*. *Journal of Communications and Networks*, 17(4), 365–372. <https://doi.org/10.1109/JCN.2015.000067>
- [25] Zhang, C., Patras, P., Haddadi, H. (2019). *Deep learning in mobile and wireless networking : A survey*. *IEEE Communications Surveys Tutorials*, 21(3), 2224–2287. <https://doi.org/10.1109/COMST.2019.2904897>
- [26] Moharam, M. H., Hany, O., Hany, A., Mahmoud, A., Mohamed, M., Saeed, S. (2025). *Anomaly detection using machine learning and adopted digital twin concepts in radio environments*. *Scientific Reports*, 15(1), 18352. <https://doi.org/10.1038/s41598-025-02759-5>
- [27] Edozie, E., Shuaibu, AN, Sadiq, BO, et Ukagwu, KJ (2025). *Progrès de l'intelligence artificielle dans la détection des anomalies pour les réseaux de télécommunications . Artificial Intelligence Review*, 58, article 100. <https://doi.org/10.1007/s10462-024-10345-x>
- [28] TechSpective (2019). *Using Machine Learning to Reduction False Positives* <https://www.techspective.net/using-machine-learning-to-reduce-false-positives/>
- [29] Rakuten Symphony (2025). *Network Performance Anomaly Detection*. <https://symphony.rakuten.com>

-
- [30] *5G Amériques. (2024). Intelligence artificielle et réseaux cellulaires.* <https://www.5gamericas.org/artificial-intelligence-cellular-networks>
- [31] Mahboob, W., Shakir, M. Z., Ali, M. (2016). *Fault prediction in LTE networks using support vector machines. 2016 International Conference on Communication Technologies (ComTech), 92–97.* <https://doi.org/10.1109/ComTech.2016.7508251>
- [32] Ahsan, U., Qamar, U., Khalid, S. (2019). *A Random Forest based approach for fault prediction in telecom networks. Telecommunication Systems, 71(4), 595–609.* <https://doi.org/10.1007/s11235-019-00578-2>
- [33] Mendoza, M., Villegas, R., Herrera, J. (2017). *A comparative study of machine learning models for fault detection in mobile networks. 2017 IEEE Latin-American Conference on Communications (LATINCOM), 1–6.* <https://doi.org/10.1109/LATINCOM.2017.8240213>
- [34] Zhang, L., Liu, Y., Zhao, Y. (2020). *An LSTM-based approach for network fault prediction in cellular networks. IEEE Access, 8, 88798–88808* <https://doi.org/10.1109/ACCESS.2020.2992601>
- [35] Lee, D., Kim, H., Choi, J. (2018). *Fault detection in wireless networks using CNN-based heatmap analysis. 2018 IEEE International Conference on Communications (ICC), 1–6.* <https://doi.org/10.1109/ICC.2018.8422145>
- [36] Al-Masri, A., Jabbar, S., Malik, A. (2022). *Explainable AI for fault prediction in 5G networks using Random Forest and SHAP. Journal of Network and Computer Applications, 200, 103320.* <https://doi.org/10.1016/j.jnca.2022.103320>
- [37] Zhao, L., He, C., Zhu, X. (2024). *A Fault Diagnosis Method for 5G Cellular Networks Based on Knowledge and Data Fusion. Sensors, 24(2), 401.* <https://doi.org/10.3390/s24020401>
- [38] Riaz, M.S., Qureshi, H.N., Masood, U., Rizwan, A., Abu-Dayya, A., Imran, A. (2022). *Automatic Fault Detection and Diagnosis in Cellular Networks and Beyond 5G : Intelligent Network Management. Electronics, 15(11), 432.* <https://doi.org/10.3390/electronics15110432>
- [39] Mulvey, A., Foh, J.-F. (2018). *Cell Coverage Degradation Detection Using Deep Learning Techniques. Sensors, 18(12), 4290.* <https://doi.org/10.3390/s18124290>
- [40] Mulvey, D., et al. (2019). *Cell Fault Management Using Machine Learning Techniques. Journal of Network and Computer Applications, 135, 1–13.* <https://doi.org/10.1016/j.jnca.2019.02.002>

- [41] Xie, Y., Sun, Y., He, T., Xu, C. (2020). Cellular fault detection using spatio-temporal graph convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 6498–6505. <https://doi.org/10.1609/aaai.v34i04.6116>
- [42] Chakraborty, P., Corici, M., Magedanz, T. (2017). System Failure Prediction within Software 5G Core Networks using Time Series Forecasting. In *Proceedings of the FMEC 2017 Conference* (pp. 1839–1845). <https://doi.org/10.1109/FMEC.2017.7946419>
- [43] APA : Basikolo, E., Basikolo, T. (2023). Towards zero downtime : Using machine learning to predict network failure in 5G and beyond. *ITU Journal on Future and Evolving Technologies*, 4(3), 434–446. <https://doi.org/10.52953/PYAF8065>
- [44] G. P. Hancke, R. J. Hancke, and M. S. Olivier, "Fault prediction and reliability analysis in a real cellular network," *Computers, Environment and Urban Systems*, vol. 63, pp. 1–11, 2017. <https://doi.org/10.1016/j.compenvurbsys.2017.02.001>
- [45] Moulay, M., Garcia Leiva, R., Rojo Maroni, P. J., Mancuso, V., Fernandez Anta, A., Safari Khatouni, A. (2021). MonTrees : Automated Detection and Classification of Networking Anomalies in Cellular Networks. <https://doi.org/10.48550/arXiv.2108.13156>
- [46] Zhang, T., Zhu, K., Hossain, E. (2022). Data-Driven Machine Learning Techniques for Self-Healing in Cellular Wireless Networks : Challenges and Solutions. *Intelligent Computing, 2022*, Article ID 9758169. <https://doi.org/10.34133/2022/9758169>
- [47] Herbin, S. (n.d.). *Apprentissage automatique – Introduction [Notes de cours]*. ENSTA Paris. Récupéré le 10 juin 2025, de https://perso.ensta-paris.fr/~manzaner/Cours/MI203/cours_ml_intro.pdf
- [48] Gagné, C. (2022). *Prétraitement et analyse de données [Diapositives de cours]*. Université Laval. Récupéré le 10 juin 2025, de <https://chgagne.github.io/iaa-ulaval/slides/sem12/iaa-sem12-analyse-FR.pdf>
- [49] Azencott, C.-A. (s.d.). *Introduction au machine learning [PDF]* https://cazencott.info/dotclear/public/lectures/IntroML_Azencott.pdf
- [50] Zhang, A., Lipton, Z. C., Li, M., Smola, A. J. (2020). *Dive into deep learning (Version 0.17.0)*. Retrieved from <https://arxiv.org/pdf/2106.11342>
- [51] Dou, B., Zhu, Z., Merkurjev, E., Ke, L., Chen, L., Jiang, J., ... Wei, G.-W. (2023). Machine learning methods for small data challenges in molecular science. *Chemical Reviews*, 123(13), 8736–8780. <https://doi.org/10.1021/acs.chemrev.3c00264>

- [52] Bhandari, A. (2025, mai 1). *Guide to AUC ROC curve in machine learning*. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning>
- [53] Aeryss. (n.d.). *LTE dataset [Data set]*. *Kaggle*. <https://www.kaggle.com/datasets/aeryss/lte-dataset>
- [54] Sevgican, S. (n.d.). *nwdaf_data [GitHub repository]*. *GitHub*. https://github.com/sevgicansalih/nwdaf_data
- [55] Mathivet, V. (2024). *Machine learning : Implémentation en Python avec Scikit-Learn (2 éd.)*. Éditions ENI. <https://www.editions-eni.fr/livre/machine-learning-implementation-en-python-avec-scikit-learn-2e-edition-/9782409044823>
- [56] Raschka, S., Patterson, J., Nolet, C. (2020). *Machine Learning in Python : Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence*. *Information*, 11(4), Article 193. <https://doi.org/10.3390/info11040193>
- [57] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). *PyTorch : An imperative style, high-performance deep learning library*. *Advances in Neural Information Processing Systems*, 32, 8026–8037. https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html
- [58] Amrane, L. (2023). *Diagnostic du mélanome à l'aide de l'apprentissage profond [Mémoire de master, Université Abderrahmane Mira - Béjaïa]*. Université de Béjaïa. <https://dspace.univ-bejaia.dz/>
- [59] Azencott, C.-A. (2022). *Introduction au machine learning (2e éd.)*. Dunod.
- [60] Yan, M., Liu, K., Guan, Z., Xinkai, X., Qian, X., Bao, H. (2018). *Background augmentation generative adversarial networks (BAGANs) : Effective data generation based on GAN-augmented 3D synthesizing*. *Symmetry*, 10(12), 734. <https://doi.org/10.3390/sym10120734>
- [61] Breadarose. (2023, 6 septembre). *What is the Mobile Station (MS) in GSM?* *Medium*. <https://medium.com/@Breadarose/what-is-the-mobile-station-ms-in-gsm-6528ff124200>
- [62] Wondershare. (s.d.). *Comprendre les générations de réseaux mobiles : de la 1G à la 5G*. *MobileTrans*. Récupéré le 3 juillet 2025, de <https://mobiletrans.wondershare.com/fr/5g/g-generations.html>

- [63] *Fall, D. (2018). Architecture du réseau UMTS [Figure]. Dans D. Fall, Évolution des réseaux mobiles vers la 5G (p.6). ResearchGate. https://www.researchgate.net/figure/Architecture-du-reseau-UMTS_fig3_328783680*
- [64] *Launay, F. (s.d.). Cours 4G – Réseaux mobiles [Blog]. Université de Poitiers. Récupéré le 3 juillet 2025, de <https://blogs.univ-poitiers.fr/f-launay/tag/cours-4g/>*
- [65] **CaféTele. (s.d.). 5G NR Architecture. Récupéré le 3 juillet 2025, de <https://cafetele.com/5g-nr-architecture/>**

Annexes

A Environnement de développement

A.1 Jupyter

D'après [55], **Jupyter** est un logiciel permettant de créer des « notebooks ». Chaque notebook est une page interactive composée de cellules, pouvant être de différents types :

- **Code** : exécuté avec affichage immédiat des résultats ;
- **Texte brut** : affiché sans mise en forme ;
- **Markdown** : texte enrichi, avec mise en page ou formules \LaTeX .

Jupyter est très utilisé en science des données, apprentissage automatique, et recherche scientifique. Il prend en charge plusieurs langages (Python, R, Julia) et favorise un flux de travail interactif. Le logo de Jupyter est illustré en figure A.1.



FIGURE A.1 – Logo de Jupyter [58].

A.2 Google Colaboratory

Google Colaboratory, ou Google Colab, est une plateforme cloud basée sur Jupyter Notebook, dédiée au prototypage rapide de modèles d'apprentissage automatique. Elle permet l'accès gratuit à des ressources matérielles puissantes comme les GPU et Tensor Processing Unit (TPU). Colab est particulièrement adapté à l'expérimentation et à l'exécution de scripts Python dans un environnement sans configuration. Le logo est présenté en figure A.2 [17].



FIGURE A.2 – Logo de Google Colab [58].

A.3 Langage utilisé : Python

Le langage utilisé dans les environnements Jupyter et Colab est **Python**. C'est un langage interprété de haut niveau, apprécié pour sa syntaxe claire et sa facilité d'apprentissage.

Il permet une grande lisibilité du code tout en offrant des performances via des bibliothèques optimisées.

Python est également soutenu par une vaste communauté et un écosystème riche, incluant des bibliothèques majeures comme NumPy, pandas, scikit-learn, TensorFlow, etc. Selon une enquête menée en 2019 par KDnuggets auprès de plus de 1 800 professionnels, Python s'impose comme le langage dominant en science des données, machine learning et analyse de données [56].

B Bibliothèques utilisées

Python est populaire pour ses nombreuses bibliothèques, ces collections de fonctions et de modules aident les développeurs à accomplir des tâches sans avoir à tout écrire depuis zéro. Que ce soit pour les mathématiques, le web ou la manipulation de données, il y a probablement une bibliothèque Python pour ça. Dans ce présent élément, nous allons donner un petit aperçu pour chaque bibliothèque utilisée dans la manipulation, associé du rôle de chacune.

B.1 pandas

Pandas est une bibliothèque Python essentielle pour la manipulation et l'analyse de données. Elle est particulièrement utile pour le traitement de données tabulaires et de séries temporelles. La bibliothèque offre des structures de données robustes comme le DataFrame pour stocker des données dans un format tabulaire et la série pour les données unidimensionnelles. Pandas excelle dans la manipulation de données, permettant des opérations telles que le tri, le filtrage et l'agrégation avec une grande facilité. De plus, elle offre des fonctionnalités d'importation et d'exportation flexibles, supportant divers formats de fichiers comme CSV, Excel et SQL [58]. Le logo de cette bibliothèque est représenté dans la figure B.1. Pour utiliser Pandas, commencez par importer le module Pandas :

```
import pandas as pd [18].
```



FIGURE B.1 – logo de Pandas [58].

B.2 Matplotlib

Matplotlib est la principale bibliothèque de création de graphiques en Python, largement utilisée pour créer divers types de visualisations. Elle est rapide et peut exporter vers plusieurs formats d'image, notamment PDF, SVG, JPG, PNG, BMP et GIF. Matplotlib prend en charge divers types de graphiques, notamment les graphiques linéaires, les graphiques de dispersion, les histogrammes, les graphiques à barres, les graphiques d'erreur, les graphiques circulaires et les graphiques en 3D. Elle sert de base à d'autres bibliothèques Python telles que Pandas et Seaborn, simplifiant leur accès aux fonctionnalités de Matplotlib. Fondée par John Hunter en 2002, Matplotlib a vu le jour lors d'une étude postdoctorale en neurobiologie pour la visualisation des données de l'électrocorticographie (ECoG) [58]. Le logo de cette bibliothèque est représenté dans la figure B.2. Pour utiliser matplotlib, commencez par importer le module pyplot du package matplotlib et le package seaborn :

```
import matplotlib.pyplot as plt
import seaborn as sns [18].
```

B.3 Seaborn

Seaborn est une bibliothèque de visualisation de données en Python qui est construite sur Matplotlib. Elle est conçue pour créer des graphiques statistiques à la fois attrayants et informatifs avec un minimum d'effort. Seaborn offre une variété de types de graphiques pour la visualisation de données statistiques, y compris des histogrammes, des graphiques à barres et des graphiques de chaleur. Elle fournit également des thèmes et des palettes de couleurs pour améliorer l'esthétique des graphiques. Seaborn est donc un excellent outil pour toute personne intéressée par la visualisation de données statistiques [58]. Le logo de cette bibliothèque est représenté dans la figure B.3. Pour utiliser Seaborn, commencez par importer le module Seaborn : *import seaborn as sns* [18].



FIGURE B.2 – logo de Matplotlib [58].



FIGURE B.3 – logo de SeaBorn [58].

B.4 NumPy

NumPy, qui signifie "Numerical Python", est le package fondamental pour le calcul scientifique avec Python. Il offre une gamme complète de fonctions mathématiques, de générateurs de nombres aléatoires et d'outils d'algèbre linéaire. Conçu spécifiquement pour travailler avec des tableaux, NumPy est essentiel pour effectuer des opérations sur des matrices et des tableaux multidimensionnels. En plus de sa capacité à gérer des tableaux, NumPy fournit également des fonctions pour travailler dans le domaine de l'algèbre linéaire, de la transformée de Fourier, et bien plus encore. C'est un outil open source qui est devenu la pierre angulaire de nombreuses autres bibliothèques scientifiques en Python. Grâce à sa haute performance et à sa facilité d'utilisation, NumPy est largement utilisé dans la communauté scientifique et de la science des données pour traiter et analyser des données numériques [58]. Le logo de cette bibliothèque est représenté dans la figure B.4. Pour utiliser NumPy, commencez par importer le module NumPy :

```
import numpy as np [18].
```



FIGURE B.4 –
logo de NumPy
[58].

B.5 PyTorch

Selon [56], les frameworks d'apprentissage profond privilégient souvent soit la simplicité d'utilisation, soit la performance, rarement les deux à la fois. **PyTorch** est une bibliothèque d'apprentissage automatique qui démontre la compatibilité de ces deux objectifs : **PyTorch** est une bibliothèque d'apprentissage automatique qui démontre la compatibilité de ces deux objectifs : elle a été conçue dès le départ pour prendre en charge un style de programmation impératif et pythonique, prenant en charge le code comme modèle, facilitant le débogage et compatible avec d'autres bibliothèques de calcul scientifique populaires, tout en restant performante et compatible avec les accélérateurs matériels tels que les GPU. **PyTorch** est Développée par Meta, particulièrement appréciée pour sa facilité d'utilisation, notamment dans la recherche et le prototypage de modèles de deep learning. Le logo de cette bibliothèque est représenté dans la figure B.5 [57].



FIGURE B.5 –
logo de PyTorch
[58].

B.6 TensorFlow

TensorFlow est une bibliothèque open source de Machine Learning créée par Google en 2011 sous le nom de DistBelief. Elle a été renommée TensorFlow en 2015. Elle permet de développer et d'exécuter des applications de Machine Learning et Deep Learning. La bibliothèque fonctionne avec des graphiques où les nœuds représentent des opérations mathématiques et les bordures des données appelées tensors. L'API de TensorFlow est basée sur Python, mais l'exécution se fait en C++ haute-performance. Ses avantages incluent la facilité d'implémentation et la visualisation via TensorBoard, mais elle présente aussi des inconvénients comme des variations dans les modèles entraînés sur différents systèmes [58]. Le logo de cette bibliothèque est représenté dans la figure B.6. Pour utiliser la bibliothèque TensorFlow, commencez par importer le module tensorflow :

```
import tensorflow as tf [18].
```

B.7 Keras

Keras est une bibliothèque open source écrite en Python, principalement basée sur les travaux du développeur de Google, François Chollet, dans le cadre du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). Elle a été publiée pour la première fois le 28 mars 2015. L'objectif principal de Keras est de faciliter la création rapide de réseaux neuronaux. Elle ne fonctionne pas comme un framework autonome, mais plutôt comme une interface de programmation applicative (API) pour accéder et programmer différents frameworks d'apprentissage automatique, tels que Theano, Microsoft Cognitive Toolkit (anciennement CNTK) et TensorFlow. Depuis la sortie de TensorFlow 1.4, Keras est intégré à l'API de base de TensorFlow, mais elle continue à être développée comme un logiciel indépendant. Keras est conçu pour être convivial, réduisant les actions des utilisateurs au minimum tout en offrant une grande flexibilité [58]. Le logo de cette bibliothèque est représenté dans la figure B.7. Pour utiliser la bibliothèque Keras, commencez par importer le module correspondant en Python :

```
from tensorflow import keras [18].
```



FIGURE B.6 – logo de Keras [58].



FIGURE B.7 – logo de Keras [58].

B.8 Scikit-Learn

Scikit-learn également connu sous le nom de **sklearn**, est la bibliothèque Python la plus puissante et la plus robuste pour l'apprentissage automatique. Elle offre une gamme complète d'outils pour les tâches de machine learning, telles que la classification, la régression, le clustering, ainsi que des fonctionnalités telles que l'estimation de forêts aléatoires, de régressions logistiques, d'algorithmes de classification et de machines à vecteurs de support. Cette bibliothèque est principalement écrite en Python et repose sur des bibliothèques telles que NumPy, SciPy et Matplotlib. Elle est développée par une communauté de contributeurs, y compris des instituts français de recherche tels qu'Institut national de recherche en informatique et en automatique (Inria) et Télécom ParisTech, et est conçue pour s'intégrer harmonieusement avec d'autres bibliothèques Python open source [58]. Le logo de cette bibliothèque est représenté dans la figure B.8. Pour utiliser la bibliothèque Scikit-learn, commencez par importer le module :

```
import sklearn [18].
```



FIGURE B.8 – logo de Scikit-Learn [58].

RÉSUMÉ

Ce travail s'inscrit dans un enjeu qui vise à améliorer la protection des réseaux cellulaires contre les différentes pannes qui peuvent survenir. Ces réseaux sont devenus indispensables à notre quotidien, et assurer leur bon fonctionnement représente un véritable défi. L'objectif de ce projet est d'utiliser l'IA pour anticiper les pannes et permettre aux opérateurs d'agir avant qu'elles n'impactent le service. Pour cela, nous avons utilisé des jeux de données représentant le comportement d'un réseau cellulaire dans différentes situations. Ces données ont été nettoyées et préparées afin de pouvoir être traitées efficacement par des algorithmes d'apprentissage automatique et d'apprentissage profond. Cette étape était essentielle pour garantir la fiabilité des prédictions. Plusieurs modèles ont ensuite été entraînés et testés sur ces données. Leur performance a été évaluée à l'aide d'indicateurs clés tels que la précision, le rappel, l'exactitude et le F1-score, afin de mesurer leur capacité à détecter correctement les anomalies. Les résultats ont montré que les modèles d'ensembles offraient les meilleures performances et étaient les plus adaptés à ce type de problème. Au final, ce projet montre que l'IA peut jouer un rôle clé dans le renforcement de la fiabilité des réseaux cellulaires, en permettant d'anticiper les pannes et de mieux y faire face.

Mots-clés : réseau cellulaire, intelligence artificielle, apprentissage automatique, apprentissage profond, détection d'anomalie, classification binaire, traitement des données.

ABSTRACT

This work is part of an effort to improve the protection of cellular networks against various types of failures that may occur. These networks have become essential to our daily lives, and ensuring their proper functioning is a real challenge. The main objective of this project is to use artificial intelligence (AI) to anticipate failures and enable operators to act before they impact the service. To achieve this, we used datasets representing the behavior of a cellular network under different conditions. These data were cleaned and preprocessed to be efficiently processed by machine learning and deep learning algorithms. This step was essential to ensure the reliability of the predictions. Several models were then trained and tested on these data. Their performance was evaluated using key metrics such as precision, recall, accuracy, and F1-score to measure their ability to correctly detect anomalies. The results showed that ensemble models provided the best performance and were the most suitable for this type of problem. In the end, this project demonstrates that AI can play a key role in strengthening the reliability of cellular networks by enabling failure prediction and improved response capabilities.

Keywords : cellular network, artificial intelligence, machine learning, deep learning, anomaly detection, binary classification, data preprocessing.