

République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة

Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا

Faculté de Technologie

قسم الإلكترونيك

Département d'Électronique



Mémoire de Master

En Télécommunication

Spécialité : Réseaux & Télécommunications

Présenté par

Mezram Rania

&

Hallouz Abd ELouadoud

**Modulation et codage adaptatif AMC par
apprentissage profond pour les systèmes 5G et 6G**

Proposé par : Dr. Anou Abderrahmane

Réf : RT 12

Année Universitaire 2024-2025

Dédicace

Je dédie ce travail et cette réussite avec toute ma gratitude, mon affection et mon amour à :

Mes chers parents,

Ce travail est avant tout le fruit de votre confiance, vos encouragements et vos prières.

Votre amour inconditionnel et votre soutien sans faille, m'ont permis de croire en moi et de poursuivre mes rêves. C'est grâce à vous que j'en suis là aujourd'hui.

Mes adorables sœurs,

HIND et HADIL AMANI,

Votre présence constante et vos encouragements infinis... toujours à mes côtés pour me motiver et inspirer par vos sourires. Vous êtes mes bras, ma force et ma fierté.

Ma tante,

Tu as toujours été une seconde mère pour moi.

Ma famille,

Qui m'a toujours soutenu dans mes choix tout au long de mon parcours.

Ma meilleure amie FAIROUZ,

Notre amitié été toujours mon pilier. Merci pour vos supports dans les moments difficiles, de doute, d'hésitation et de partager les souvenirs inoubliables.

A tous ceux qu'on aime et qui nous aiment,

Merci infiniment.

MEZRAM Rania.

Dédicace

En ce moment marquant et décisif de ma vie, mes pensées et mes premiers élans de cœur vont vers les plus belles créatures que Dieu ait créées sur cette terre, vers les sources d'amour, de patience et de générosité.

*À **mes chers parents**, sans le soutien et les sacrifices desquels je n'aurais jamais atteint ce stade de mon parcours académique.*

*À **mes sœurs** bien-aimées, mon pilier dans l'adversité et ma force dans les moments difficiles.*

*À la mémoire de **mon frère** bien-aimé **Abdelmoumen**, que dieu l'ait en sa sainte miséricorde et lui accorde le paradis pour demeure.*

*À mes chers amis **Ayoub** et **Walid**, qui ont toujours été pour moi une source inépuisable d'inspiration et d'encouragement.*

Et à tous ceux qui, par un mot, un geste ou un regard, m'ont donné la force de continuer...

HALLOUZ Abd ELouadoud.

Remerciement

*Nous souhaitons remercier en premier lieu, notre directeur de mémoire, Dr « **Anou Abderrahmane** » pour son encadrement exceptionnel, son soutien moral, sa confiance et ses conseils éclairés.*

Nous tenons également à adresser nos profonds remerciements aux membres des jurys, pour avoir pris le temps d'évaluer notre mémoire.

Vos remarques constructives et vos orientations ont enrichi notre réflexion et amélioré la qualité de ce mémoire. Nous sommes honorés d'avoir eu l'opportunité de présenter nos recherches devant des jurys aussi compétent.

*Nos remerciements vont aussi au personnel pédagogique de l'université « **SAAD DAHLAB-BLIDA 1** » en particulier tous les professeurs du département « **d'Electronique** » pour leur enseignement, disponibilité et leurs conseils enrichissants.*

Enfin un grand merci à nos proches, qu'ont su nous soutenir moralement de près ou de loin durant cette aventure académique.

ملخص

تعد تقنية التعديل والترميز التكيفي AMC من التقنيات الأساسية لتحسين الكفاءة الطيفية حيث تقوم بضبط مخطط التعديل ومعدل الترميز ديناميكيا وفقا لتغيرات القناة. يهدف هذا العمل الى دمج التعلم العميق لتحسين تقنية AMC في شبكات الجيل الخامس والسادس، وبالتالي تحسين اداء الرابط اللاسلكي من خلال التكيف الديناميكي مع ظروف القناة. تم تدريب نماذج من الشبكات العصبية لتوقع المخططات المثلى ل AMC استنادا الى معلومات من بيئة الراديو. أظهرت النتائج تحسنا ملحوظا في معدل النقل والموثوقية مقارنة بالطرق التقليدية. تفتح هذه المقاربة المجال نحو أنظمة اتصالات ذكية وقابلة للتكيف ذاتيا لأجيال الشبكات اللاسلكية المستقبلية.

كلمات مفتاحية: التعلم العميق، تصنيف التعديل، التشكيل والترميز التكيفي، الشبكات العصبية، الجيل الخامس، الجيل السادس، أنظمة الاتصالات اللاسلكية، الذكاء الاصطناعي.

Résumé

La modulation et le codage adaptatif (AMC) est une technique essentielle dans les transmissions sans fil, car elle ajuste dynamiquement le schéma de modulation et le taux de codage en fonction des fluctuations du canal. L'objectif de ce travail est d'intégrer l'apprentissage profond pour optimiser la technique AMC dans les réseaux 5G et 6G, ainsi d'améliorer la performance du lien radio en s'adaptant dynamiquement aux conditions du canal. Des modèles des réseaux de neurones ont été entraînés pour prédire les schémas AMC optimaux à partir d'informations de l'environnement radio. Les résultats montrent une amélioration notable du débit et de la fiabilité par rapports aux méthodes traditionnelles. Cette approche ouvre la voie à des systèmes de communication intelligents et auto-adaptatifs pour les futures générations des réseaux sans fils.

Mots clés : apprentissage profond, classification des modulations, modulation et codage adaptatif AMC, réseaux de neurones, 5G, 6G, systèmes de communication sans fil, intelligence artificielle.

Abstract:

Adaptive modulation and coding (AMC) is a key technique in wireless transmissions, as it dynamically adjusts the modulation scheme and coding rate based on channel fluctuations. This work aims to integrate deep learning to optimize the AMC technique in 5G and 6G networks, thereby improving radio link performance by dynamically adapting to channel conditions. Neural network models were trained to predict optimal AMC schemes based on information from the radio environment. The results show a significant improvement in throughput and reliability compared to traditional methods. This approach paves the way for intelligent and self-adaptive communication systems for future wireless network generations.

Keywords: deep learning, modulation classification, Adaptive modulation and Coding (AMC), neural networks, 5G, 6G, wireless communication systems, artificial intelligence.

Table des matières

Introduction générale.....	1
Chapitre 01 : Introduction et contextes généraux	
1.1. Introduction.....	4
1.2. Les communications sans Fil.....	4
1.2.1. Evolution des systèmes de communication mobile.....	4
1.2.2. Enjeux actuels et futurs (débit, latence, fiabilité).....	5
1.3. AMC (Adaptive Modulation and Coding).....	6
1.3.1. Définition et concepts de base.....	6
1.3.2. Principe de fonctionnement du système typique d'AMC.....	7
1.4. AMC traditionnel.....	9
1.4.1. Schémas de modulation.....	9
1.4.2. Codes correcteurs d'erreurs.....	11
1.4.3. Revue des algorithmes classiques d'AMC.....	13
1.4.4. Techniques d'Estimation de Canal pour l'AMC.....	15
1.4.5. Importance et Avantages.....	16
1.4.6. Limites des approches traditionnelles.....	16
1.5. Apprentissage profond.....	17
1.5.1. Introduction et concepts fondamentaux.....	17
1.5.2. Fonctionnement du Deep Learning (DL).....	18
1.5.3. Les Réseaux de Neurones.....	18
1.6. Apprentissage profond en télécommunications.....	19
1.6.1. Architectures de réseaux de neurones pertinents.....	19

1.6.2.	Applications de l'apprentissage profond dans l'AMC.....	21
1.7.	Conclusion	22

Chapitre 02 : Modélisation du système et conception de l'architecture

2.1.	Introduction.....	24
2.2.	Modèles de canal de communication sans fil	24
2.3.	Analyse des conditions de propagation affectant les systèmes AMC	28
2.4.	Conception et implémentation de l'agent.....	30
2.4.1.	Configuration matérielle de simulation	30
2.4.2.	Base de données	30
2.4.3.	Prétraitement et nettoyage des données.....	31
2.4.4.	Plateforme de développement	32
2.4.5.	Environnements de simulation	33
2.5.	Modèle d'apprentissage profond	34
2.5.1.	Choix de l'architecture.....	34
2.5.2.	Entrées du réseau (features).....	35
2.5.3.	Sortie du réseau	36
2.5.4.	Fonction de perte	36
2.5.5.	Algorithmes d'optimisation.....	37
2.5.6.	Hyperparamètres.....	37
2.5.7.	Techniques de régularisation.....	38
2.6.	Évaluation des performances	38
2.7.	Conclusion	40

Chapitre 03 : Tests et travaux expérimentaux

3.1.	Introduction.....	42
3.2.	Méthodologie expérimentale générale.....	42

3.3.	Bilan des tests successifs et limitations	43
3.3.1.	Expérimentation 1 – Modèle LSTM.....	43
3.3.2.	Expérimentation 2 – Modèle CNN.....	45
3.3.3.	Expérimentation 3 – Modèle CNN-LSTM (augmentation des classes de modulation)	47
3.3.4.	Expérimentation 4 – Modèle DNN.....	50
3.4.	Transition vers l’optimisation finale.....	52
3.5.	Conclusion	53

Chapitre 04 : Simulation et discussion des résultats finaux.

4.1.	Introduction.....	55
4.2.	Processus d’implémentation du modèle final	55
4.3.	Paramètres du modèle d’apprentissage profond	58
4.3.1.	Bibliothèques Python.....	58
4.3.2.	Paramètres des données et prétraitement.....	59
4.3.3.	Paramètres d’architecture du modèle DNN final	59
4.3.4.	Paramètres d’entraînement	60
4.3.5.	Paramètres des conditions du canal	60
4.3.6.	Les hyperparamètres.....	61
4.3.7.	Métriques d’évaluation	61
4.4.	Analyse et Interprétation des résultats du modèle final	62
4.5.	Evaluation expérimentale de l’agent AMC intelligent via l’interface de test.....	65
4.5.1.	La phase d’entraînement du modèle sur l’interface.....	66
4.5.2.	Test 1 – conditions de canal faibles.....	67
4.5.3.	Test 2 – conditions de canal intermédiaire	69
4.5.4.	Test 3 – conditions de canal élevées.....	70

4.5.5. Analyse du modèle avec de nouvelles données.....	71
4.6. Comparaison entre les méthodes traditionnelles et l'approche proposée pour l'AMC.....	73
4.7. Conclusion.....	74
Conclusion générale et perspectives.....	75
Bibliographie.....	77

Liste des figures

Figure I. 1: Développement des systèmes de communication mobile au fil des années [5].	5
Figure I. 2: Schéma synoptique de la technique AMC.	7
Figure I. 3: Organigramme de la technique AMC.	8
Figure I. 4: Constellation de la modulation BPSK [9].	10
Figure I. 5: Constellation de la modulation QPSK [9].	10
Figure I. 6: Constellations des modulations 16-QAM et 64-QAM [11].	11
Figure I. 7: Exemple d'un codeur convolutif [13].	12
Figure I. 8: Diagramme typique d'un mot de code Reed-Solomon typique [16].	13
Figure I. 9: Évolution du (BER) en fonction du (SNR) pour différents (MCS) [23].	14
Figure I. 10: Les sous-domaines de l'intelligence artificielle [22].	18
Figure I. 11: Architectures typiques des réseaux neuronaux simple et profond [30].	19
Figure II. 1: modèle du canal AWGN [36].	26
Figure II. 2: Propagation multi-trajet NLOS [37].	26
Figure II. 3: modèle du canal Rayleigh [36].	27
Figure II. 4: Les avantages de python.	33
Figure II. 5: Environnement de simulation "Kaggle".	34
Figure II. 6: Architecture des réseaux de neurone profond [39].	35
Figure III. 1: Méthodologie d'entraînement d'ensemble d'expérimentations.	42
Figure III. 2: Courbes d'apprentissage LSTM.	44
Figure III. 3: Matrice de confusion du modèle LSTM.	44
Figure III. 4: Précision du modèle LSTM.	45
Figure III. 5: Matrice de confusion du modèle CNN.	46
Figure III. 6: Accuracy + rapport de classification du modèle CNN.	46
Figure III. 7: Courbes d'apprentissage du modèle CNN.	47
Figure III. 8: Courbes d'apprentissage du Modèle CNN-LSTM (modulations multiples).	48
Figure III. 9: Matrice de confusion du Modèle CNN-LSTM (modulations multiples).	49
Figure III. 10: Rapport de classification du Modèle CNN-LSTM (modulations multiples).	49
Figure III. 11: Précision globale du Modèle CNN-LSTM (modulations multiples).	49
Figure III. 12: Courbes d'apprentissage du Modèle DNN.	51

Figure III. 13: Matrice de confusion du Modèle DNN.....	51
Figure III. 14: Rapport de classification du Modèle DNN.....	51
Figure III. 15: Précision globale du Modèle DNN.....	51
Figure IV. 1: Organigramme méthodologique du modèle AMC basé sur DL.....	56
Figure IV. 2: Courbes d'apprentissage du Modèle DNN final.....	63
Figure IV. 3: Matrice de confusion du Modèle DNN final.....	63
Figure IV. 4: Rapport de classification du modèle DNN final.....	63
Figure IV. 5: Précision globale du Modèle DNN final.....	64
Figure IV. 6: Présentation de la page d'accueil de l'interface.....	65
Figure IV. 7: les principales fonctionnalités disponibles sur l'interface de test.....	66
Figure IV. 8: Entraînement du modèle sous l'interface.....	67
Figure IV. 9: Scénario avec faible qualité de canal.....	68
Figure IV. 10: Distribution des probabilités de modulations - canal faible.....	68
Figure IV. 11: Scénario de canal intermédiaire (conditions de canal modérées).....	69
Figure IV. 12: Distribution des probabilités de modulations - canal intermédiaire.....	70
Figure IV. 13: Scénario avec canal de haute qualité.....	70
Figure IV. 14: Distribution des probabilités de modulations - canal favorable.....	71
Figure IV. 15: Test du modèle sur des nouvelles données inédites.....	72

Liste des tableaux

Tableau I. 1: Comparaisons entre les performances des générations des systèmes mobiles.....	6
Tableau I. 2: Comparaison entre les différentes formes de QAM et PSK [11].	11
Tableau II. 1: Effets des Conditions du canal sur l'AMC.....	28
Tableau II. 2: Classification qualitative des conditions du canal radio selon les principaux paramètres physiques.....	29
Tableau II. 3: Règles de sélection adaptative de la modulation selon les conditions du canal..	29
Tableau III. 1: paramètres d'entrainement du modèle LSTM.....	43
Tableau III. 2: Observations des résultats du 1er essai modèle LSTM.	43
Tableau III. 3: Paramètres d'entrainement du modèle CNN.....	45
Tableau III. 4: Observations des résultats du 2ème essai CNN.....	47
Tableau III. 5: Paramètres d'entrainement du modèle CNN-LSTM (modulations multiples)..	48
Tableau III. 6: Observations des résultats du 3ème expérimentation CNN-LSTM (modulations multiples).....	50
Tableau III. 7: Paramètres d'entrainement du modèle DNN.....	50
Tableau III. 8: Observations des résultats du 4ème expérimentation DNN.	52
Tableau IV. 1: Les principales bibliothèques python utilisés dans le modèle du DL.	59
Tableau IV. 2: Paramètres des données et prétraitement du modèle final.	59
Tableau IV. 3: Paramètres d'architecture du modèle final.	60
Tableau IV. 4: Paramètres d'entrainement du modèle final.....	60
Tableau IV. 5: Paramètres des conditions de canal de transmission.....	60
Tableau IV. 6: les hyperparamètres du modèle final.....	61
Tableau IV. 7: Paramètres d'évaluation du modèle final.	62
Tableau IV. 8: Observations des résultats du modèle DNN final.	62
Tableau IV. 9: Test du modèle DNN sur un ensemble d'échantillon des conditions de canal.	64
Tableau IV. 10: Critères de comparaisons entre méthodes classiques et avancés pour l'AMC.	73

Liste des abréviations

1G	First Generation
2G	Second Generation
3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation
6G	Sixth Generation
AMC	Adaptive Modulation and Coding
AWGN	Additive White Gaussian Noise
BER	Binary Error Rate
BPSK	Binary Phase Shift Keying
CC	Convolutional Coding
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSI	Channel State Information
DL	Deep learning
DNN	Deep Neural Network
FEC	Forward Error Correction
GAN	Generative Adversarial Network
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
GSM	Global System for Mobile Communication
HSPA	High Speed Packet Access
IA	Intelligence Artificielle
IoT	Internet of Things
ISI	Inter Symbol Interference
LDCP	Low-Density Parity-Check
LMMSE	Linear Minimum Mean Square Error
LMS	Least Mean Squares
LS	Least Squares
LSTM	Long Short-Term Memory
LTE	Long Term Evolution
MCS	Modulation and Coding Scheme

ML	Machine Learning
NLMS	Normalized Least Mean Squares
NLOS	Non-Line of Sight
OEM	Ondes Electromagnetique
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RNN	Recurrent Neural Network
RS	Reed-Solomon
SMS	Short Message Service
SNR	Signal to Noise Ratio
SNRinst	Instantaneous Signal to Noise Ratio
UMTS	Universal Mobile Telecommunications System
ZF	Zero Forcing

Introduction générale

À l'heure actuelle, les systèmes de communication sans fil sont déposés comme le pilier de notre société numérique. L'avènement des réseaux mobiles avancés (5G/6G), exige de nouveaux besoins renforcés en termes de débit, de latence, d'efficacité spectrale et d'adaptabilité.

Au cœur de cette évolution se situe la technique « d'Adaptive Modulation and Coding (AMC) », un mécanisme fondamental basé sur l'adaptation des paramètres de modulation et de codage en fonction des variations du canal en temps réel, pour le but de maintenir un bon compromis entre débit et fiabilité et d'assurer une qualité de service (QoS) acceptable [1].

Cependant, les méthodes classiques d'AMC s'appuient sur des algorithmes fondés sur des modèles statistiques et des indicateurs de qualité du canal, notamment le (SNR) et le BER, ce qui peut entraîner des performances de transmission sous-optimales, surtout dans des situations où les délais sont importants. Par conséquent, ces approches peuvent manquer souvent de la flexibilité nécessaire pour s'adapter à la variabilité des environnements de propagation complexes et dynamiques ainsi qu'aux exigences progressives des réseaux modernes [1].

Face à ces limites, L'apprentissage profond (Deep Learning), une sous-branche avancée de L'IA apparu comme une solution automatisée prometteuse pour améliorer les performances de l'AMC. Grâce à sa capacité à analyser de grandes quantités de données complexes, il permet une adaptation plus précise et rapide des paramètres AMC tout en améliorant l'efficacité du spectre et la fiabilité des communications [2-3].

Les défis principaux abordés dans ce projet sont :

- Limites des méthodes traditionnelles d'AMC qui ne suivent pas le dynamisme et la variabilité rapide des canaux dans les réseaux modernes.
- Adaptation sous-optimale à cause des retards (forte latence), ce qui dégrade la qualité de service.
- Nécessité d'une méthode plus autonome et prédictive pour mieux gérer les conditions changeantes.
- Le potentiel de l'apprentissage profond pour une adaptation en temps réel, plus précis et plus rapide face aux conditions du canal.

Les objectifs de ce projet s'articulent autour des axes suivants :

- Intégrer des modèles d'intelligence artificielle dans les réseaux modernes 5G/6G afin de permettre une adaptation en temps réel aux conditions du canal.
- Développer un système basé sur l'apprentissage profond capable de remplacer le mécanisme traditionnel d'AMC.
- Permettre à un agent intelligent d'apprendre à prendre des décisions optimales concernant le choix de la modulation et du codage, en fonction de l'état dynamique du canal radio.
- Optimiser les performances de l'AMC en termes de débit, de latence et de qualité de service (QoS).
- Évaluer et comparer les performances du système proposé avec celles des approches classiques à travers des simulations et des expérimentations.

Dans ce cadre, la question centrale que nous cherchons à résoudre est la suivante :

Comment concevoir un agent intelligent, reposant sur l'apprentissage profond, capable de se substituer de manière dynamique au fonctionnement classique de l'AMC, tout en s'adaptant aux variations du canal radio et en maximisant les performances en matière de débit, de latence et de qualité de service ?

Notre mémoire est structurée en 3 chapitres :

Le premier chapitre présente le contexte théorique des systèmes de communications sans fils, la technique AMC ainsi que les limites de ses méthodes classiques. Ensuite, il introduit les fondements du Deep Learning, son potentiel et ses applications en télécommunications.

Le chapitre 2, présente l'environnement de propagation, puis les étapes et les moyens utilisés pour concevoir l'architecture et implémenter notre agent intelligent.

Le chapitre 3, expose les différents travaux expérimentaux réalisés avant d'aboutir au modèle final, en montrant le processus suivis au fil de ces essais.

Le dernier chapitre, décrit le développement pratique du modèle final choisis pour notre agent intelligent, les méthodologies de simulation et d'évaluations ainsi que la discussion des résultats et la comparaison avec les méthodes traditionnelles.

Chapitre 01 :
Introduction et contextes
généraux

1.1. Introduction

Les systèmes de communication sans fil ont considérablement évolué, passant des premières générations analogiques aux réseaux 5G et bientôt 6G, offrant des débits élevés, une faible latence et une fiabilité accrue, grâce notamment aux avancées en modulation et codage adaptatif (AMC).

Ce chapitre introductif présente le contexte général et les concepts clés du projet. Il explore les principes fondamentaux des communications sans fil, les défis des réseaux mobiles et l'AMC traditionnel, en mettant en avant ses limites. Il introduit ensuite l'apprentissage profond et son potentiel dans les télécommunications, avant d'aborder les architectures neuronales et les algorithmes appliqués à l'AMC.

L'objectif est de montrer comment l'intelligence artificielle, et en particulier l'apprentissage profond, peut révolutionner l'AMC, rendant les réseaux 5G et 6G plus performants, adaptatifs et intelligents.

1.2. Communications sans Fil

Les systèmes de communication sans fil assurent la transmission d'informations via des ondes électromagnétiques, permettant l'échange de données sur de longues distances sans recours à des connexions physiques. Au fil des années, ces technologies ont évolué pour s'adapter aux exigences croissantes en matière de débit, de mobilité et de connectivité mondiale. Aujourd'hui, elles constituent un élément clé des technologies modernes, facilitant de nombreuses applications telles que la téléphonie mobile, l'Internet des objets (IoT), les réseaux de capteurs et les communications par satellite [1-4].

1.2.1. Evolution des systèmes de communication mobile

Les communications sans fil ont connu une évolution rapide, passant des premières transmissions radio aux réseaux mobiles modernes, marqués par des avancées majeures en débit, latence et couverture (Figure 1) [3-5-6-7].

- 1G (années 1980) : Communications analogiques limitées, conçues pour les appels vocaux.
- 2G (années 1990) : Numérisation avec le GSM, améliorant la qualité des appels et introduisant les SMS.

- 3G (années 2000) : Accès Internet mobile avec l'UMTS et l'HSPA, offrant des vitesses accrues.
- 4G (années 2010) : Haut débit mobile avec la LTE, facilitant le streaming et les services multimédias avancés.
- 5G (2020 et au-delà) : Débit ultra-élevé, latence réduite (<1 ms), ouvrant la voie à l'IoT et aux véhicules autonomes.
- 6G (vers 2030) : Réseau basé sur l'IA, intégrant des technologies comme la communication holographique et une couverture mondiale.

Chaque génération a marqué une avancée clé, transformant les communications mobiles et ouvrant la voie aux technologies du futur.

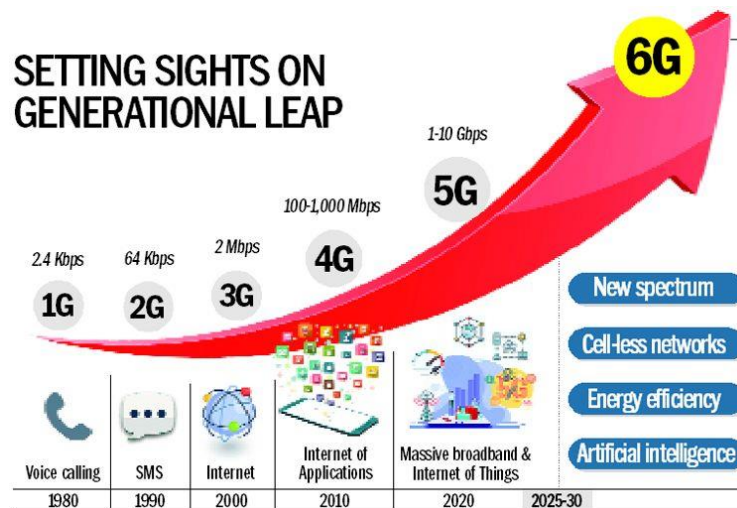


Figure I. 1: Développement des systèmes de communication mobile au fil des années [5].

1.2.2. Enjeux actuels et futurs (débit, latence, fiabilité)

Les réseaux 5G et futurs 6G visent à répondre aux exigences croissantes en débit, latence et fiabilité [6-8] :

- Débit : Augmentation des vitesses pour le streaming, les jeux en ligne et l'IoT.
- Latence : Transmission ultrarapide pour les applications en temps réel.
- Fiabilité : Communications stables et sécurisées, même dans des environnements complexes.

Ces avancées permettront d'améliorer l'expérience utilisateur et de soutenir le développement de nouvelles technologies.

Génération	Débit Max	Latence	Caractéristiques
3G	2 Mbps	100 ms	Internet mobile, voix et données
4G	1 Gbps	10 ms	LTE, haut débit mobile
5G	10 Gbps	< 1 ms	IoT massif, IA
6G (prévision)	> 100 Gbps	< 100 μ s	THz, IA avancée, ultra-fiabilité

Tableau I. 1: Comparaisons entre les performances des générations des systèmes mobiles.

1.3. AMC (Adaptive Modulation and Coding)

1.3.1. Définition et concepts de base

L'AMC est une technique fondamentale des communications sans fil modernes. Elle adapte en temps réel les schémas de modulation et les taux de codage en fonction des fluctuations de l'état du canal (CSI). Cette méthode optimise l'équilibre entre robustesse et débit, améliore l'efficacité spectrale, renforce la fiabilité et assure une qualité de service (QoS) satisfaisante [9-11].

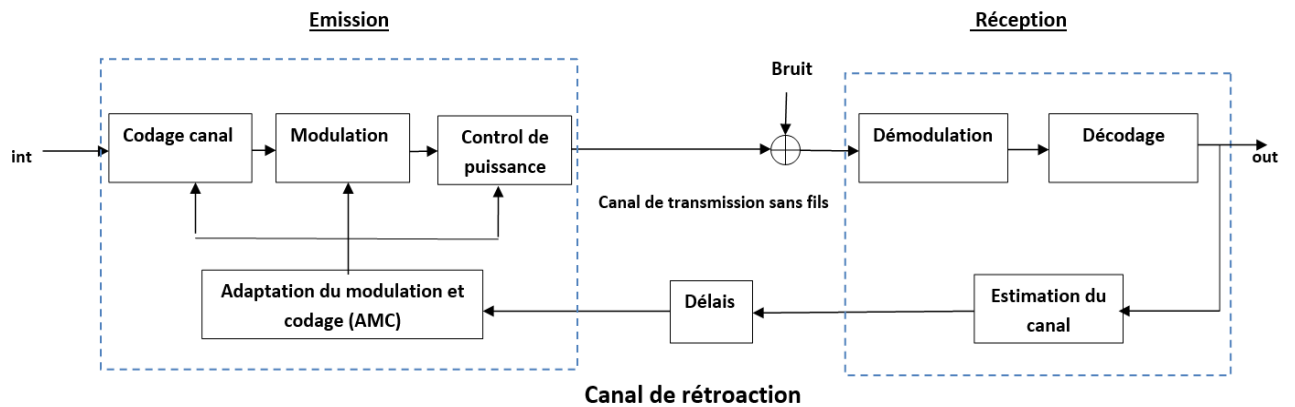


Figure I. 2: Schéma synoptique de la technique AMC.

1.3.2. Principe de fonctionnement du système typique d'AMC

L'AMC ajuste dynamiquement le schéma de modulation et le codage correcteur d'erreurs (FEC - Forward Error Correction) en fonction de la qualité du canal, généralement évaluée par le SNR ou le BER.

La figure 3 illustre l'organigramme de l'AMC, un processus décisionnel structuré reposant sur une logique séquentielle et conditionnelle. L'adaptation des MCS s'effectue en fonction du rapport signal sur bruit instantané (SNR_{inst}), permettant ainsi d'optimiser la transmission des données en s'adaptant aux fluctuations du canal.

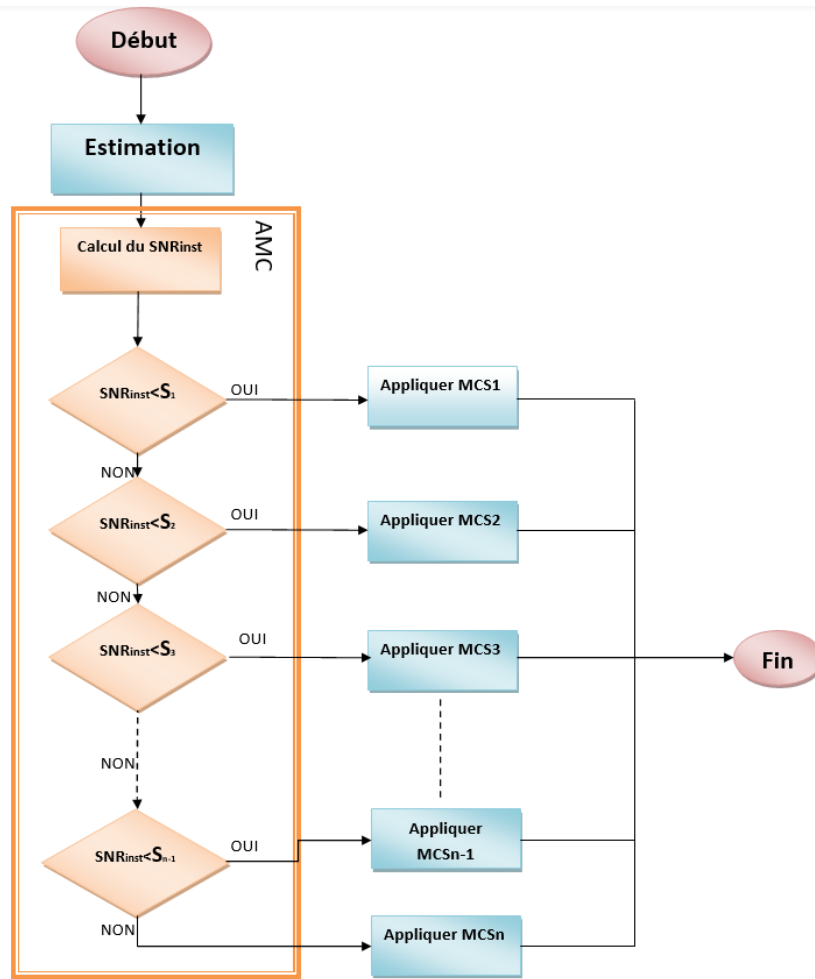


Figure I. 3: Organigramme de la technique AMC.

Lors de la réception, le signal est d'abord démodulé pour restituer les données numériques d'origine, puis décodé afin de corriger les erreurs introduites pendant la transmission. Toutes les étapes se déroulent comme suit :

1. Estimation du canal

Le récepteur évalue les caractéristiques du canal (atténuation, bruit, etc.) afin d'estimer la qualité du signal reçu en mesurant le SNR instantané (SNRinst). Une estimation précise du CSI est essentielle pour adapter efficacement la modulation et le codage.

2. Calcul du SNRinst

Le SNRinst représente le rapport entre la puissance du signal utile et celle du bruit dans le canal. Il est calculé à partir des données issues de l'estimation du canal et constitue un indicateur clé de la qualité du lien de communication.

3. Comparaison avec les seuils

Le SNR_{inst} est comparé à une série de seuils prédéfinis (S_1, S_2, \dots, S_{n-1}), représentant différents niveaux de qualité du canal.

4. Sélection du MCS optimal

En fonction du SNR_{inst} , un schéma de modulation et de codage (MCS) est sélectionné :

- Si $SNR_{inst} \leq S_1 \rightarrow$ MCS1 est appliqué : faible débit, mais forte robustesse, adapté aux canaux très dégradés.
- Si $S_1 < SNR_{inst} \leq S_2 \rightarrow$ MCS2 est choisi : compromis entre débit et robustesse.
- Si SNR_{inst} dépasse tous les seuils, le MCS $_n$, le plus performant, est utilisé.

Un MCS élevé offre un débit supérieur, mais plus sensible aux erreurs, tandis qu'un MCS faible privilégie la robustesse au détriment du débit. Le choix des seuils (S_1, S_2, S_3 , etc.) est donc crucial pour optimiser l'équilibre entre débit et qualité du lien.

5. Application du MCS et fin du processus

Une fois le MCS sélectionné, les paramètres de modulation (nombre de bits par symbole) et de codage (taux de correction d'erreur) sont ajustés pour optimiser la transmission.

1.4. AMC traditionnel

1.4.1. Schémas de modulation

La modulation est un processus qui convertit un signal d'origine en une forme adaptée au canal de transmission en modifiant certains paramètres d'une onde sinusoïdale porteuse, tels que l'amplitude, la fréquence ou la phase [9].

Dans le cadre de l'AMC, la modulation joue un rôle clé en facilitant la transmission des signaux sur de longues distances tout en optimisant le transfert des données en fonction des conditions du canal de communication [10].

Différents schémas de modulation sont utilisés en AMC (voir tableau 1), notamment :

- **BPSK (Binary Phase Shift Keying)** : Cette modulation modifie la phase de l'onde porteuse selon le signal d'information. Également appelée 2-PSK, elle utilise deux phases distinctes (0° et 180°) pour représenter les bits. La BPSK est très robuste, mais offre un faible débit (1 bit par symbole). La figure 1-4 illustre la constellation de la modulation BPSK [9].

- **QPSK (Quadrature Phase Shift Keying)** : connue aussi sous le nom de 4-PSK, elle constitue un compromis entre débit et robustesse. Son diagramme de constellation comprend quatre symboles, où chaque symbole code 2 bits, améliorant ainsi le débit tout en maintenant une bonne résistance aux interférences (voir figure 1-5) [9].

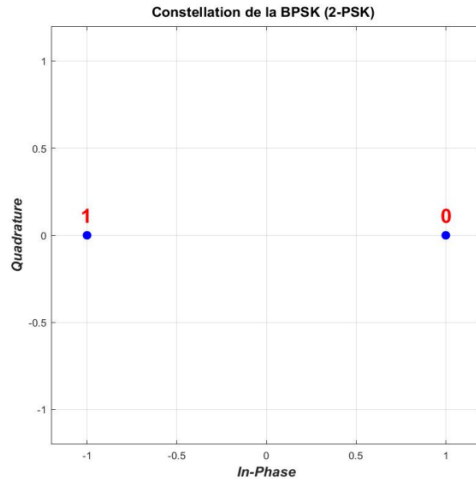


Figure I. 4: Constellation de la modulation BPSK [9].

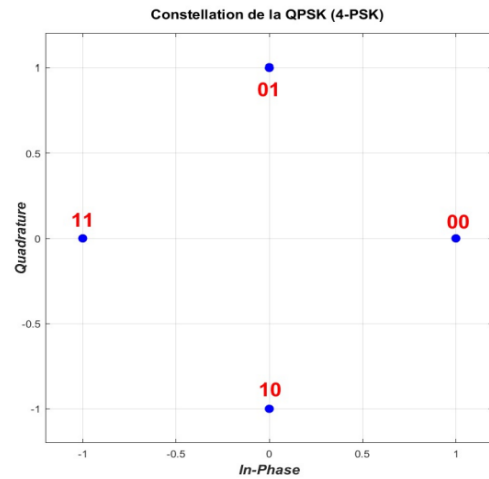


Figure I. 5: Constellation de la modulation QPSK [9].

- **La M-QAM (Quadrature Amplitude Modulation)** : est une modulation bidimensionnelle qui combine à la fois l'amplitude et la phase du signal [9]. Elle repose sur deux porteuses AM pour encoder plusieurs bits en une seule transmission [10].

La 16-QAM permet de transmettre 4 bits par symbole et la 64-QAM transmet 6 bits par symbole, offrant un débit plus élevé. La figure 1-5 illustre les constellations des modulations 16-QAM et 64-QAM. Bien qu'elle permette un débit binaire élevé, la M-QAM est plus sensible au bruit et aux interférences, ce qui en fait un compromis entre performance et robustesse.

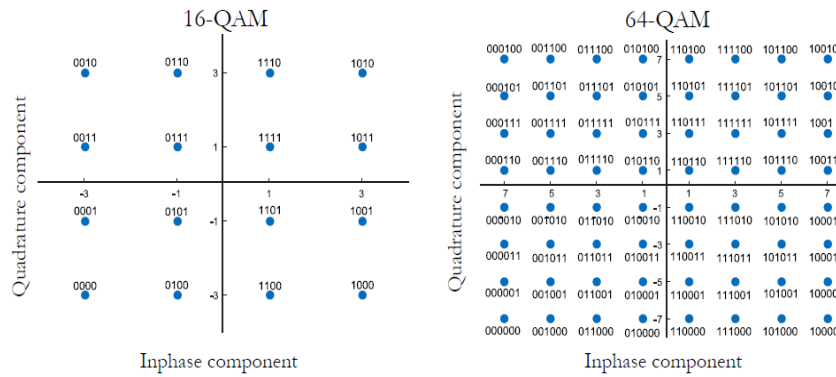


Figure I. 6: Constellations des modulations 16-QAM et 64-QAM [11].

Modulation	Bits par symbole	Symbol Rate
BPSK	1	1 débit binaire
QPSK	2	1/2 débit binaire
8-PSK	3	1/3 débit binaire
16-QAM	4	1/4 débit binaire
32-QAM	5	1/5 débit binaire
64-QAM	6	1/6 débit binaire
256-QAM	8	1/8 débit binaire

Tableau I. 2: Comparaison entre les différentes formes de QAM et PSK [11].

1.4.2. Codes correcteurs d'erreurs

Le codeur correcteur d'erreurs, aussi appelé codeur de canal, est une technique qui ajoute des bits de redondance au message original afin de détecter et/ou corriger les erreurs de transmission à la réception, sans nécessiter de retransmission des données [12].

Le FEC (Forward Error Correction), ou autocorrection, est une méthode qui utilise une redondance accrue pour détecter et corriger les erreurs directement au niveau du décodeur.

Dans le cadre de l'AMC, ces codes correcteurs jouent un rôle clé en améliorant la fiabilité de transmission en sélectionnant dynamiquement le code correcteur d'erreurs le plus adapté aux conditions du canal.

➤ Code Convolutif

Les codes convolutifs sont des codes linéaires où chaque bit de sortie dépend à la fois du bit d'entrée actuel et des bits précédents, introduisant ainsi un effet de mémoire.

Un exemple de codeur convolutif est illustré dans la figure 1-6, où chaque bit de sortie est généré en fonction du bit d'entrée actuel et des bits stockés dans des registres à décalage [12]. Ces codes sont obtenus par convolution linéaire des messages binaires à l'aide de polynômes générateurs.

Un codeur convolutif est composé des éléments suivants :

- Registres à décalage : Stockent les bits précédemment codés pour assurer une mémoire du système.
- Additionneurs modulo 2 : Combinent les bits d'entrée avec ceux des registres pour produire les bits de sortie.
- Polynômes générateurs : Définissent les règles de transformation des bits d'entrée en bits de sortie.

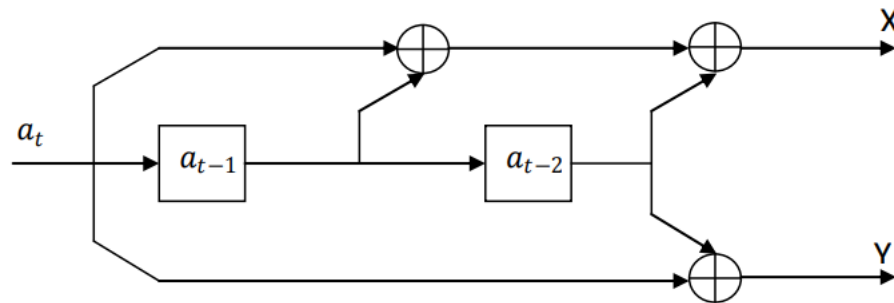


Figure I. 7: Exemple d'un codeur convolutif [13].

➤ Code Reed-Solomon :

Les codes Reed-Solomon sont des codes correcteurs d'erreurs puissants, capables de détecter et corriger plusieurs erreurs symboliques, c'est-à-dire qu'ils opèrent sur des symboles plutôt que sur des bits [14]. Ils appartiennent à la famille des codes cycliques non binaires et reposent sur des corps finis (ou corps de Galois), exploitant des polynômes pour le processus de codage et de décodage [12]. Le codeur Reed-Solomon ajoute des symboles de redondance aux blocs de données numériques, permettant au décodeur Reed-Solomon d'analyser chaque bloc reçu, de corriger les erreurs et de récupérer les données d'origine. La figure 7 illustre la structure typique d'un mot de code Reed-Solomon. Un code Reed-Solomon est défini sous la forme $RS(n, k)$, où:

n : Nombre total de symboles dans un mot de code.

k : Nombre de symboles de données.

s : Nombre de bits par symbole.

Cela signifie que le codeur prend k symboles de données de s bits chacun et génère un mot de code contenant n symboles, en ajoutant $(n - k)$ symboles de parité.

Un décodeur Reed-Solomon peut corriger jusqu'à t symboles erronés dans un mot de code, où $2t = n - k$ [15].

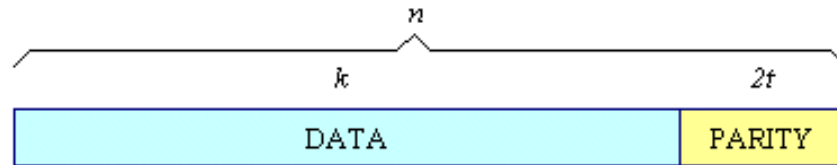


Figure I. 8: Diagramme typique d'un mot de code Reed-Solomon typique [16].

1.4.3. Revue des algorithmes classiques d'AMC

La figure 9 illustre l'évolution du taux d'erreur binaire (BER) en fonction du rapport signal sur bruit (SNR) pour différents schémas de modulation et de codage (MCS).

➤ Approche basée sur le SNR

Les premières méthodes d'AMC utilisent le SNR comme critère principal pour sélectionner le MCS optimal. Un algorithme classique consiste à comparer la valeur estimée du SNR avec des seuils prédéfinis afin d'ajuster dynamiquement la modulation et le codage. Bien que cette approche soit simple et efficace, elle peut être sous-optimale face aux fluctuations rapides du canal.

SNR faible → Modulation robuste (BPSK, QPSK) avec un codage plus protecteur.

SNR élevé → Modulation à haut débit (16-QAM, 64-QAM) avec un codage plus léger.

➤ Approche basée sur le BER

Cette méthode optimise le choix du MCS en visant un taux d'erreur binaire (BER) cible, garantissant ainsi un compromis entre fiabilité et débit.

Lorsque le BER dépasse un seuil critique, on réduit l'ordre de modulation ou on renforce le codage correcteur. Cette approche offre une meilleure robustesse, mais nécessite une estimation précise du BER en fonction des conditions du canal.

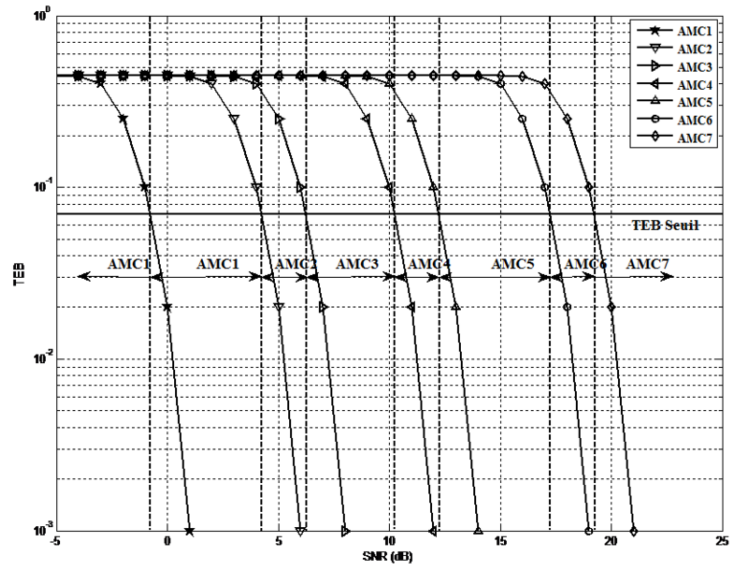


Figure I. 9: Évolution du (BER) en fonction du (SNR) pour différents (MCS) [23].

Interprétation de la Figure I. 9 :

- Axe des abscisses (SNR) : Représente la qualité du canal. Un SNR élevé indique une transmission plus fiable et moins d'erreurs.
- Axe des ordonnées (BER) : Indique la probabilité d'erreur sur les bits transmis. Un BER faible est souhaitable pour assurer une meilleure qualité de communication.
- Seuil de BER : Une ligne horizontale marque un BER seuil en dessous duquel la transmission est considérée comme acceptable.
- Sélection du MCS : En fonction du SNR instantané mesuré sur le canal, le système adapte dynamiquement le MCS :
 - SNR faible → Sélection d'un MCS robuste, mais avec un débit réduit (AMC1, AMC2).
 - SNR élevé → Sélection d'un MCS performant offrant un débit élevé (AMC6, AMC7).

1.4.4. Techniques d'Estimation de Canal pour l'AMC

L'estimation de canal est essentielle dans les communications sans fil pour adapter dynamiquement la transmission aux conditions du canal et assurer une communication fiable.

Plusieurs techniques existent, chacune présentant un compromis entre complexité et performance.

❖ Méthodes classiques :

- Moindres Carrés (LS) : simple à implémenter, mais sensible au bruit et inefficace à faible SNR.
- LMS (Least Mean Squares) : Algorithme adaptatif robuste, mais à convergence lente.
- LMMSE (Linear Minimum Mean Square Error) : Offre des estimations plus précises en tenant compte des statistiques du canal et du bruit, mais au prix d'une complexité accrue.
- NLMS (Normalized LMS) ** : Version améliorée du LMS, avec une meilleure stabilité face aux variations du signal.

❖ Méthodes basées sur des Signaux Pilotes :

- Estimation par pilotes : Insère des symboles connus dans le signal transmis pour évaluer la réponse du canal.
- Méthodes aveugles/semi-aveugles : Exploitent les statistiques du signal sans nécessiter de pilotes, économisant ainsi de la bande passante, mais avec une complexité accrue.

❖ Techniques avancées :

- Filtre de Kalman : Suit l'évolution temporelle du canal en temps réel, idéal pour les environnements mobiles, mais exige une modélisation précise.
- Estimateur Zero Forcing (ZF) : élimine l'interférence intersymboles (ISI), mais amplifie le bruit, dégradant ainsi la performance en faible SNR.
- Transformée de Fourier : Exploite la transformée de Fourier pour analyser la réponse fréquentielle du canal, efficace, mais sensible au bruit.
- Compressed Sensing : utilise la parcimonie du canal pour réduire le nombre d'échantillons requis, efficace en large bande, mais requiert des algorithmes de reconstruction complexes.

❖ Approches basées sur l'Intelligence Artificielle :

- Maximum de Vraisemblance (ML) : Estimation statistiquement optimale, mais très coûteuse en calculs.
- Apprentissage automatique : Exploite des réseaux de neurones pour apprendre les caractéristiques du canal et améliorer les performances, surtout en conditions bruitées et dynamiques.

Ces différentes méthodes permettent d'optimiser l'estimation du canal en fonction des contraintes système et des conditions du canal.

1.4.5. Importance et Avantages

L'estimation de canal et l'AMC jouent un rôle clé dans l'optimisation des performances des systèmes de communication sans fil, offrant plusieurs avantages :

- Optimisation du débit en sélectionnant dynamiquement le schéma de modulation et de codage adapté.
- Amélioration de la qualité de service (QoS) en réduisant les pertes de données et en garantissant une transmission fiable.
- Diminution des erreurs de transmission grâce à une adaptation efficace aux variations du canal.
- Flexibilité et adaptabilité aux conditions changeantes du canal, garantissant une meilleure résilience du réseau.
- Amélioration de l'efficacité spectrale en exploitant au mieux les ressources disponibles.
- Utilisation optimale du spectre radio en ajustant la modulation et le codage en fonction des conditions de propagation.
- Renforcement de la robustesse du signal, assurant une communication stable même en présence d'interférences ou de bruit.

1.4.6. Limites des approches traditionnelles

Malgré leur efficacité, les méthodes classiques d'AMC et d'estimation de canal présentent plusieurs limites :

- Réactivité limitée aux variations rapides du canal, ce qui peut affecter les performances dans des environnements dynamiques.
- Incapacité à exploiter pleinement les avancées modernes**, notamment l'apprentissage automatique et l'intelligence artificielle.

- Complexité accrue** dans les réseaux à haute densité, rendant leur mise en œuvre plus difficile.
- Difficulté d'adaptation** aux environnements complexes et dynamiques, notamment dans les réseaux 5G et 6G.
- Dépendance aux modèles statistiques** du canal, nécessitant des hypothèses précises souvent difficiles à obtenir.
- Délais d'ajustement trop longs**, inadaptés aux scénarios nécessitant des décisions en temps réel.
- Complexité algorithmique élevée**, ce qui peut limiter l'efficacité dans les systèmes contraints en ressources.

1.5. Apprentissage profond

1.5.1. Introduction et concepts fondamentaux

L'apprentissage profond (Deep Learning - DL) est une branche de l'intelligence artificielle (IA) et plus précisément de l'apprentissage automatique (Machine Learning - ML), comme illustré dans la figure I. 10. Il repose sur l'utilisation de réseaux de neurones artificiels profonds, composés de plusieurs couches, permettant d'analyser de vastes volumes de données et d'extraire des modèles complexes. Grâce à sa capacité à prendre des décisions basées sur les données, le DL s'est imposé comme un outil puissant dans divers domaines, y compris les télécommunications, où il offre des solutions innovantes pour optimiser les performances des systèmes de communication [17-18-20-21].

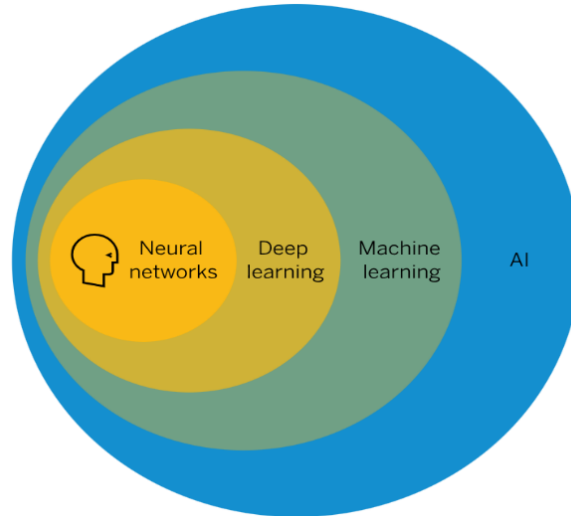


Figure I. 10: Les sous-domaines de l'intelligence artificielle [22].

1.5.2. Fonctionnement du Deep Learning (DL)

L'apprentissage profond repose sur l'entraînement d'un modèle à partir d'un large ensemble de données. Ce processus permet au modèle d'identifier et de classifier des motifs complexes, facilitant ainsi les prédictions et la prise de décisions sur de nouvelles données. L'ajustement des poids internes du réseau s'effectue à l'aide d'algorithmes d'optimisation, tels que la rétropropagation, qui permet d'améliorer progressivement la précision du modèle [24-25-26].

1.5.3. Réseaux de Neurones

Les réseaux neuronaux artificiels constituent la base du Deep Learning. Ils se composent de plusieurs couches de traitement qui fonctionnent en parallèle, s'inspirant du fonctionnement du système nerveux humain. Ces réseaux sont constitués de nœuds interconnectés (neurones), comme illustrés dans la figure I. 11 [27-28].

Une architecture typique comprend :

- Une couche d'entrée, qui reçoit les données brutes.
- Une ou plusieurs couches cachées, chargées de transformer et d'extraire des caractéristiques complexes à partir des données.
- Une couche de sortie, qui génère le résultat final en fonction des informations traitées par le réseau [29].

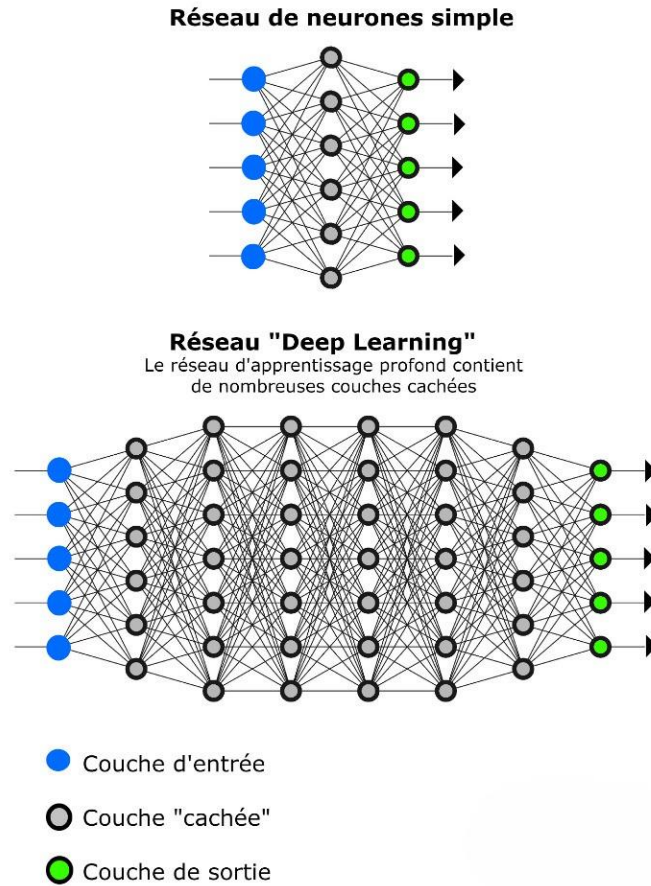


Figure I. 11: Architectures typiques des réseaux neuronaux simple et profond [30].

1.6. Apprentissage profond en télécommunications

1.6.1. Architectures de réseaux de neurones pertinents

➤ Réseaux de Neurones Convolutifs (CNN)

- Conçus pour traiter des données structurées sous forme de grille, comme les images et les spectrogrammes de signaux.
- Principalement utilisés en vision par ordinateur pour la classification et la détection d'images.
- En télécommunications, ils sont efficaces pour la classification automatique des schémas de modulation et l'identification des signaux. Ils utilisent des opérations de convolution pour extraire les caractéristiques pertinentes [30-31].

- Les CNN sont composées de plusieurs couches notamment les couches convolutionnelles, couches de pooling, couches entièrement connectées.
 - Les CNN possèdent des architectures avancées telles que AlexNet, VGGNet, ResNet et Inception.
- **Réseaux de Neurones Récurrents (RNN)**
- Adaptés au traitement des données séquentielles, comme le texte et la parole, grâce à une mémoire interne qui prend en compte les informations passées pour prédire les résultats futurs [19-32].
 - Inclues des variantes comme les LSTM (Long Short-Term Memory) et GRU (Gated Recurrent Units), essentielles pour la modélisation des variations temporelles des canaux de communication.
- **Réseaux de Neurones Profonds (DNN)**
- Modèles neuronaux multicouches capables de traiter des tâches complexes de classification et de régression.
 - Utilisés pour modéliser les fonctions de décision en prenant en compte des paramètres du canal (ex : SNR, fading...).
- Applications en AMC :
- Classification des états du canal.
 - Prédiction des paramètres optimaux de modulation et codage.
- **Réseaux antagonistes génératifs (GAN)**
- Utilisent deux réseaux de neurones en compétition pour générer des données synthétiques réalistes du canal.
 - Améliorent l'estimation du canal en fournissant des échantillons d'entraînement plus représentatifs [33].
- **Réseaux de Neurones hybrides (CNN + RNN)**
- Combinaison des CNN pour l'extraction de caractéristiques et des RNN (LSTM/GRU) pour la modélisation temporelle des données.
 - Permettent une analyse plus précise des signaux variant dans le temps.
- **Réseaux de Transformateurs**

- Basés sur des couches de ****self-attention**** suivies de couches entièrement connectées.
- Initialement développés pour le traitement du langage naturel, mais montrent un fort potentiel dans la prédiction des états du canal grâce à leur capacité à capturer des dépendances à long terme dans les données [34].

1.6.2. Applications de l'apprentissage profond dans l'AMC

Bien que l'apprentissage profond et l'Adaptative Modulation and Coding (AMC) soient deux domaines distincts, l'intelligence artificielle (IA) joue un rôle clé dans les communications sans fil, en particulier dans le cadre de l'AMC [35].

- **Classification automatique des schémas de modulation et de codage** : Les réseaux neuronaux, notamment RNN et CNN, sont utilisés pour identifier automatiquement le type de modulation et de codage d'un signal reçu (ex. reconnaissance de modulations comme QPSK, 16-QAM, 64-QAM ou de codages tels que les codes convolutifs et LDPC).
- **Prédiction des conditions du canal** : L'apprentissage profond permet d'anticiper l'état du canal de communication ainsi que ses variations en temps réel, facilitant ainsi une adaptation plus efficace de la modulation et du codage.
- **Détection et correction d'erreurs** : Les modèles de réseaux neuronaux peuvent être entraînés à identifier et corriger les erreurs de transmission, améliorant ainsi la fiabilité des communications.
- **Amélioration de l'estimation de la qualité du canal** : Une estimation précise de la qualité du canal est essentielle pour l'AMC. L'apprentissage profond contribue à affiner cette estimation, en particulier dans des environnements complexes avec des variations rapides du canal.
- **Réduction de la consommation énergétique** : L'intégration de schémas d'AMC optimisés par IA permet d'économiser de l'énergie et de prolonger l'autonomie des appareils mobiles.
- **AMC basé sur l'apprentissage par renforcement (RL)** : Le Reinforcement Learning (RL) est utilisé pour ajuster dynamiquement les stratégies d'AMC en réponse aux variations du canal.

- **Réduction des interférences** : Les modèles d'apprentissage profond permettent d'atténuer les interférences co-canal et inter-symboles, améliorant ainsi la qualité du signal transmis.
- **Optimisation de la qualité de service (QoS)** : Le développement d'algorithmes d'AMC adaptatifs garantit une qualité de service élevée pour diverses applications, notamment la voix, la vidéo et les données.

1.7. Conclusion

Ce chapitre a posé les bases théoriques et contextuelles nécessaires à la compréhension de l'AMC et de son rôle dans l'optimisation des transmissions sans fil. Bien que les méthodes traditionnelles d'AMC restent efficaces, elles montrent des limites face aux environnements de communication complexes des réseaux 5G et 6G. L'apprentissage profond apparaît comme une solution prometteuse pour surmonter ces défis en permettant le développement d'agents intelligents capables d'optimiser l'AMC. Les chapitres suivants approfondiront les méthodologies et outils pour concevoir et évaluer un tel système.

Chapitre 02 :
Modélisation du système
et conception de
l'architecture

2.1. Introduction

L'intelligence artificielle, en particulier le Deep Learning, repose comme une évolution des réseaux de neurones qui permet d'accroître le nombre de données à traiter. L'application de l'apprentissage profond dans le contexte d'AMC ouvre la voie à des systèmes intelligents capables d'optimiser la transmission en temps réel et de s'adapter aux environnements complexes des futurs réseaux, en rendant le processus de l'AMC plus intelligent, flexible et performant.

Ce chapitre est consacré à la présentation de la modélisation du modèle et la conception de l'architecture d'apprentissage utilisée pour la modulation et codage adaptatif (AMC), basée sur l'apprentissage profond dans les systèmes de communication 5G/6G.

Tout d'abord, le chapitre expose les modèles des canaux de propagation sans fil (AWGN, Rayleigh) qui influencent directement les performances de la transmission. Ensuite, Il introduit la base de données utilisée avec un ensemble d'étapes liées aux préparations des données (prétraitement et nettoyage) en décrivant la configuration matérielle utilisée pour la simulation, ainsi que les outils de développement adopté. La deuxième partie du chapitre décrit l'architecture neuronale proposée pour le modèle du Deep Learning, incluant les types des entrées et sorties du réseau, les hyperparamètres, ainsi que les techniques de régularisation appliquées. Enfin, une évaluation complète des performances du modèle est réalisée à l'aide de plusieurs métriques afin de tester son efficacité.

Notre objectif est de créer un agent intelligent, basé sur l'IA, qui puisse améliorer les performances de l'AMC de façon adaptative, même dans des environnements complexes et changeants.

2.2. Modèles de canal de communication sans fil

Pour modéliser le système et concevoir l'architecture, il est obligatoire de comprendre les différents modèles de canaux sans fil et de prédire le comportement des signaux dans des environnements réels, en particulier dans la 5G et 6G, ainsi que leur impact sur les performances des systèmes de communication.

Ces canaux reposent sur des OEM pour transmettre les données, ils sont caractérisés par des variations rapides dues à des phénomènes complexes tels que l'atténuation, la réflexion, la diffraction et la diffusion. La quantité d'informations qui peuvent être transmises est déterminée selon la largeur de bande de ces canaux.

Pour notre approche, les deux types de canaux suivants sont appropriés :

- ❖ **Canal avec bruit blanc gaussien additif** : désigné par l'acronyme AWGN (Additive White Gaussian Noises), c'est le modèle le plus utilisé en théorie, car il représente un canal parfait où le signal reçu est affecté seulement par le bruit. Ce modèle est considéré comme une référence basique pour comparer les performances des systèmes de communication, sans prendre en charge d'autres effets (évanouissements, interférences...).

Le signal reçu est le résultat d'une addition du signal modulé avec un bruit représenté par une variable aléatoire ; la figure 2-1 le montre.

$$r(t) = x(t) + n(t) \quad (2-1)$$

Où :

- $r(t)$ est le signal reçu,
- $x(t)$ est le signal transmis,
- $n(t)$ est le bruit blanc gaussien, qui suit une distribution normale de moyenne nulle et variance « σ^2 ».

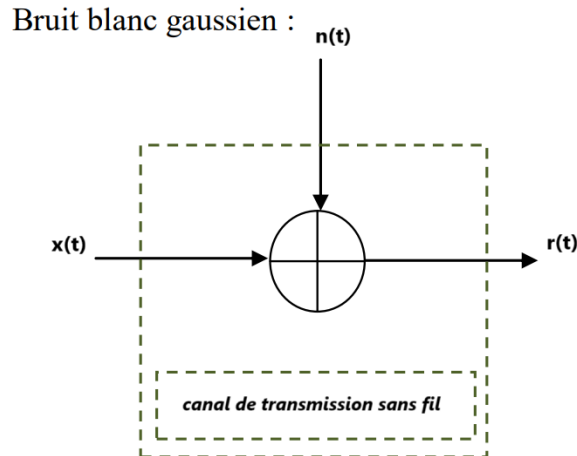


Figure II. 1: modèle du canal AWGN [36].

- ❖ **Canal Rayleigh** : contrairement au canal AWGN, ce modèle prend en considération l'effet d'évanouissement (fading) car le canal subit différentes variations rapides. Rayleigh est fréquemment utilisé dans le modèle multi-trajet avec la non-visibilité du trajet multiple, c'est-à-dire dans des environnements où il n'y a pas une ligne de vue directe (NLOS) entre l'émetteur et le récepteur. La figure II.1 illustre ce phénomène, où le signal transmis subit des réflexions, diffractions et diffusions avant d'atteindre le récepteur.

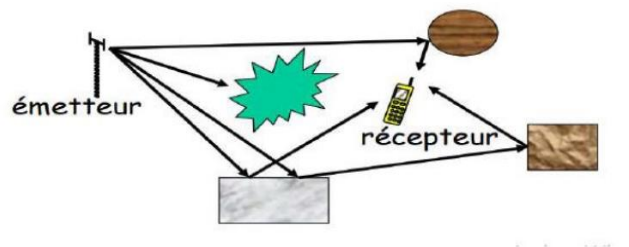


Figure II. 2: Propagation multi-trajet NLOS [37].

L'équation de la sortie du canal $r(t)$ est donnée par :

$$r(t) = \alpha(t)x(t) + n(t) \quad (2-2)$$

Où :

- $r(t)$: le signal reçu,
- $x(t)$: le signal transmis,

- $\alpha(t)$: le coefficient de fading de Rayleigh,
- $n(t)$ est le bruit AWGN.

Pour obtenir l'enveloppe du signal résultant, il faut générer une variable aléatoire complexe représentant le coefficient complexe du canal comme $\alpha = \beta + j\gamma$, où « β, γ » sont deux variables aléatoires. La figure 14 expose un schéma synoptique d'un canal Rayleigh.

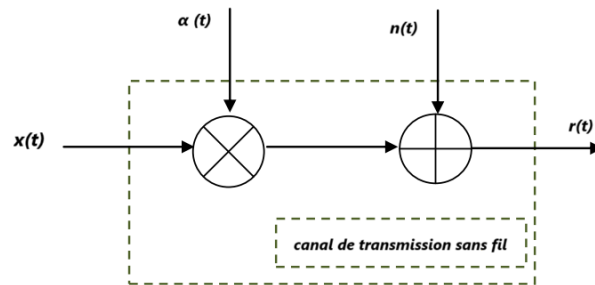


Figure II. 3: modèle du canal Rayleigh [36]

La distribution de Rayleigh est définie à partir du variable aléatoire « α ». La fonction de densité de probabilité est donnée par :

$$P(\alpha) = \frac{\alpha}{\sigma^2} e^{-\frac{\alpha}{2\sigma^2}} \quad \alpha \geq 0 \quad (2-3)$$

Où :

« σ^2 » : la puissance moyenne du signal reçu.

« α » : le paramètre de Rayleigh représentant la valeur la plus probable :

$$E(\alpha) = \sigma \sqrt{\frac{\pi}{2}} \quad (2-4)$$

Et une variance de :

$$\text{Var}(\alpha) = \left(\frac{4 - \pi}{2}\right) \sigma^2 \quad (2-5)$$

2.3. Analyse des conditions de propagation affectant les systèmes AMC

L'efficacité de l'adaptation de l'AMC repose sur la capacité du système à ajuster dynamiquement les paramètres de transmissions, en fonctions de l'état du canal radio. Dans les systèmes sans fil de nouvelles générations, le canal de transmission est fortement variable, influencé par divers phénomènes qui ont un impact direct sur la qualité du signal reçu, ce qui rend l'adaptation en temps réel des paramètres AMC essentiels pour maintenir un bon compromis entre débit et fiabilité. La section suivante décrit ces facteurs en mettant en évidence leur impact sur le choix des paramètres de transmission « tableau 3 ».

Facteur de condition	Définition	Effet sur canal et AMC
SNR	Mesure la qualité du signal reçu en comparant la puissance du signal utile à celle du bruit.	Un faible SNR limite la modulation et codage possible, réduisant le débit pour assurer la fiabilité.
Facteur Fading (facteur d'atténuation)	Représente les fluctuations aléatoires d'amplitude du signal causée par des interférences ou des réflexions, souvent dues à la propagation multi-trajet.	Les variations rapides d'amplitude nécessitent un codage plus robuste et/ou des modulations de faible ordre pour éviter les erreurs.
Doppler	Phénomène lié à la vitesse du déplacement de l'émetteur ou de récepteur, entraînant un décalage de fréquence qui impacte la stabilité du signal.	Dégrade les performances du signal et limite les modulations.
Delay spread (étalement de retard)	Résulte de la réception de copies du même signal a des instants différents, à causes de chemins de propagations multiples, ce qui peut provoquer des interférences entre symboles.	Crée des ISI (Inter Symbol Interference) réduisant la qualité du signal et nécessitant des techniques de corrections.

Tableau II. 1: Effets des Conditions du canal sur l'AMC.

Paramètre	Très faible	Faible	Modéré	Bon	Excellent
SNR (dB)	< 0 dB	0–5 dB	5–10 dB	10–20 dB	> 20 dB
Facteur de fading	< 0.2	0.2–0.4	0.4–0.6	0.6–0.8	> 0.8
Doppler (Hz)	> 500 Hz	200–500 Hz	100–200 Hz	30–100 Hz	< 30 Hz
Délai (μ s)	> 10 μ s	5–10 μ s	2–5 μ s	0.5–2 μ s	< 0.5 μ s

Tableau II. 2: Classification qualitative des conditions du canal radio selon les principaux paramètres physiques.

Le tableau II.2 présente une classification des conditions de canal en fonctions de quatre paramètres essentiels : le SNR, le facteur de Fading, la fréquence de Doppler et le délai de propagation. Chaque paramètre est segmenté en cinq niveaux qualitatifs allant de « très faible » à « excellent ». Ces seuils permettent de quantifier l'état du canal pour orienter la sélection des modulation/codage optimaux dans un système AMC.

SNR	Fading	Doppler	Délai	Modulation recommandée
Faible	N'importe	N'importe	N'importe	BPSK
Moyen	Faible	Rapide	Élevé	BPSK / QPSK
Moyen	Moyen	Moyen	Moyen	QPSK / 8PSK
Moyen	Bon	Faible	Faible	16QAM / 8PSK
Élevé	Bon	Faible	Faible	64QAM / 16QAM
Élevé	Faible	Rapide	Élevé	QPSK

Tableau II. 3: Règles de sélection adaptative de la modulation selon les conditions du canal.

Le tableau II.3 ci-dessus synthétise les stratégies de modulation recommandées en fonctions des principales conditions de canal. Il met en évidence l'importance d'adapter dynamiquement le schéma de de modulation a la qualité du canal afin d'optimiser la robustesse et l'efficacité spectrale du système. Ce type de tableau guide la prise de décision de l'agent AMC pour garantir une transmission fiable et performante.

2.4. Conception et implémentation de l'agent

2.4.1. Configuration matérielle de simulation

Pour réaliser les simulations de notre modèle d'apprentissage profond, nous avons utilisé une station de travail puissante équipée d'un processeur Intel core i5-14400F cadencé à 2.5 GHz, associée à 64 Go de mémoire RAM. La puissance de calcul a été renforcée par une carte graphique NVIDIA GeForce RTX 3070 disposant de 8 Go de mémoire vidéo dédiée, permettant d'accélérer significativement l'entraînement des réseaux de neurones profonds grâce à ses 5888 cœurs CUDA et ses 184 cœurs tensor spécialisés dans les calculs matriciels. Le tout fonctionne sous Windows 11 professionnel (version 24H2) offrant un environnement stable et compatible avec les bibliothèques couramment utilisés en Deep Learning. Grâce à cette configuration, il a été possible d'implémenter et d'exécuter des modèles complexes comme DNN, ainsi de réaliser des entraînements sur des ensembles de données volumineux, tout en analysant les performances des algorithmes en temps réel.

2.4.2. Base de données

La base de données RadioML 2018.01A, développée par DeepSig Inc. Disponible sur Kaggle, et largement utilisée dans le domaine de la reconnaissance automatique de modulation. Elle sert principalement à entraîner et évaluer des algorithmes capables d'identifier le type de modulation utilisé dans un signal radio. Cette base contient une grande variété de signaux RF (radiofréquence) modulés selon différents schémas tels que QPSK, QAM, BPSK, entre autres, avec l'ajout de perturbations réalistes comme le bruit ou des décalages de fréquence [38].

L'ensemble de données combine des signaux synthétiques simulés et des enregistrements réels, couvrant 17 types de modulation numérique validés par la communauté. Il est fourni au format HDF5, un format optimisé pour stocker de grandes quantités de données. Le fichier principal, nommé « GOLD_XYZ_OSC.0001_1024.hdf5 », contient plus de 2 millions d'exemples, chacun composé de 1024 échantillons de signaux complexes représentés en virgule flottante.

Caractéristiques de l'ensemble de données :

- 17 types de modulations : 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 64PSK, 128PSK, 256PSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, 16APSK et OQPSK.
- 26 niveaux de SNR allant de -20 dB à +30 dB par pas de 2 dB.
- 4096 échantillons par combinaison modulation–SNR.
- 1024 échantillons complexes (I/Q) par trame.
- Chaque trame est étiquetée avec son type de modulation et le SNR associé.
- Les signaux sont structurés sous forme de paires I/Q (en phase et en quadrature), ce qui donne une forme matricielle de (1024, 2) par trame.
- Nombre total de trames : 1 916 928.

Structure des données :

- X : contient les signaux I/Q sous forme de séquences temporelles.
- Y : contient les étiquettes de modulation, encodées en one-hot.
- Z : contient les valeurs de SNR associées à chaque trame.

Dimensions globales :

- X : (1 916 928, 1024, 2)
- Y : (1 916 928, 24)
- Z : (1 916 928, 1)

Cette base de données constitue une ressource précieuse pour le développement de modèles d'apprentissage automatique appliqués à la reconnaissance de modulation dans des environnements proches du réel.

2.4.3. Prétraitement et nettoyage des données

Les systèmes de classifications des signaux nécessitent des données propres, normalisées et correctement formatées pour obtenir des performances optimales, particulièrement dans les tâches de détections de signaux et de classification de modulation.

Pour cela, il est obligatoire d'intégrer un pipeline de nettoyage et de prétraitement de données conçu pour traiter des données volumineuses des signaux de radiofréquences stockées au format HDF5, contenant des échantillons I/Q, des étiquettes (labels) et des valeurs de rapport signal sur bruit (SNR).

Les principales étapes clés réalisées comprennent :

- ✓ Un filtrage des échantillons selon un seuil minimal de SNR.
- ✓ Une détection et suppression des valeurs aberrantes est effectuée à l'aide d'un seuil basé sur l'écart type.
- ✓ Une élimination des valeurs manquante (NAN) pour garantir la validité des données traitées.
- ✓ La normalisation des échantillons I/Q dans une plage [0,1] pour garantir une meilleure homogénéité des données.

Ces opérations sont effectuées par traitement par lots (batch), afin de traiter efficacement de grandes quantités de données en évitant la surcharge du mémoire. Enfin, les données nettoyées sont sauvegardées dans un nouveau fichier HDF5 accompagné de métadonnées décrivant les opérations du nettoyage effectuées.

Ce processus de nettoyage et de normalisation est fondamental pour les applications d'apprentissage profond en classification de signaux, car il permet de fournir aux modèles des données fiables, sans bruit excessif ni valeurs extrêmes pouvant biaiser l'entraînement, tout en assurant la qualité et la cohérence des données en entrées du modèle.

2.4.4. Plateforme de développement

Ce projet exige l'emploi des outils pertinents en termes de rapidité et efficacité, pour le but d'effectuer les simulations complexes. L'utilisation du Python avec Jupyter est idéale pour développer, tester et optimiser efficacement les performances du modèle basé sur le Deep Learning dans les réseaux modernes.

- ❖ **Langage de programmation « Python »** : dans ce travail, nous avons choisi d'utiliser Python comme langage de programmation principale, car il s'agit d'un langage open-source, interprété et axé sur la programmation orientée objet. Il est largement utilisé pour sa polyvalence et facilité à apprendre, notamment dans le domaine d'IA et l'apprentissage profond, grâce à des bibliothèques puissantes comme Numpy, Pandas et TensorFlow. Python offre une syntaxe simple et lisible conçue pour tester rapidement différents modèles. La figure II.4 illustre pourquoi Python est un choix judicieux.

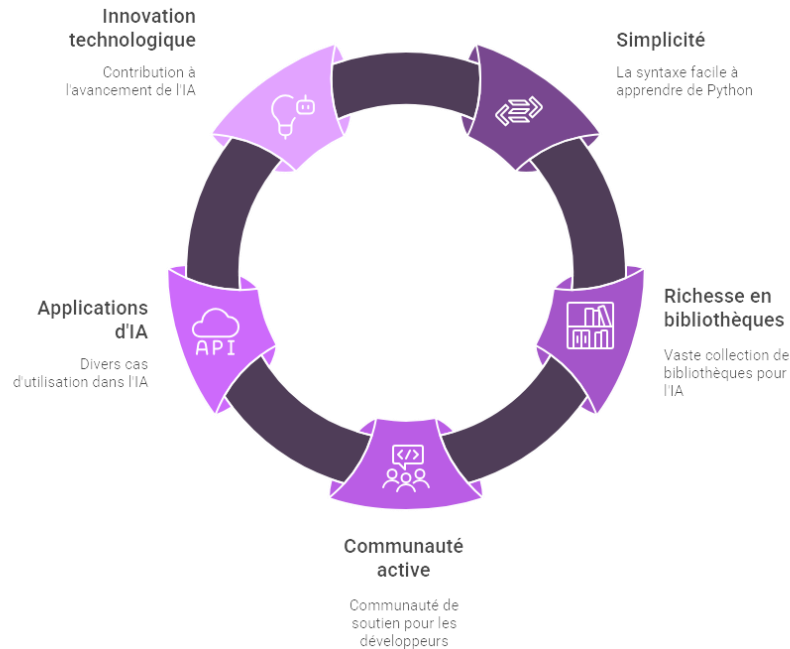


Figure II. 4: Les avantages de python.

2.4.5. Environnements de simulation

- ❖ **L'environnement de développement « Jupyter Notebook »** : est un environnement de développement interactif implémenté sous ANACONDA, permet de combiner du code Python, des exécutions et la visualisation des résultats de calculs, sous un document notebook. Jupyter, fonctionne en exécutant des codes cellule par cellule et en présentant les visualisations graphiques et tabulaires directement dans le même notebook, ce qui est crucial pour documenter le travail de manière claire et structurée.
- ❖ **Kaggle « en ligne »** : Les simulations ont été réalisées à l'aide de la plateforme Kaggle « figure 2-5 », qui offre une interface intégrée de type Jupyter Notebook, facilitant le développement, l'exécution et le partage de code en python. Cette solution présente plusieurs avantages, notamment l'accès à des ressources matérielles suffisantes (CPU, GPU) pour accélérer l'entraînement des modèles d'apprentissage profond, ainsi qu'un environnement préconfiguré avec les principales bibliothèques nécessaires telle que Tensorflow, Keras, Numpy, Matplotlib. L'utilisation de Kaggle a permis d'assurer la reproductibilité des tests et la collaboration, facilitant la visualisation des résultats et les simulations de l'intelligence artificielle appliquée aux communications numériques.

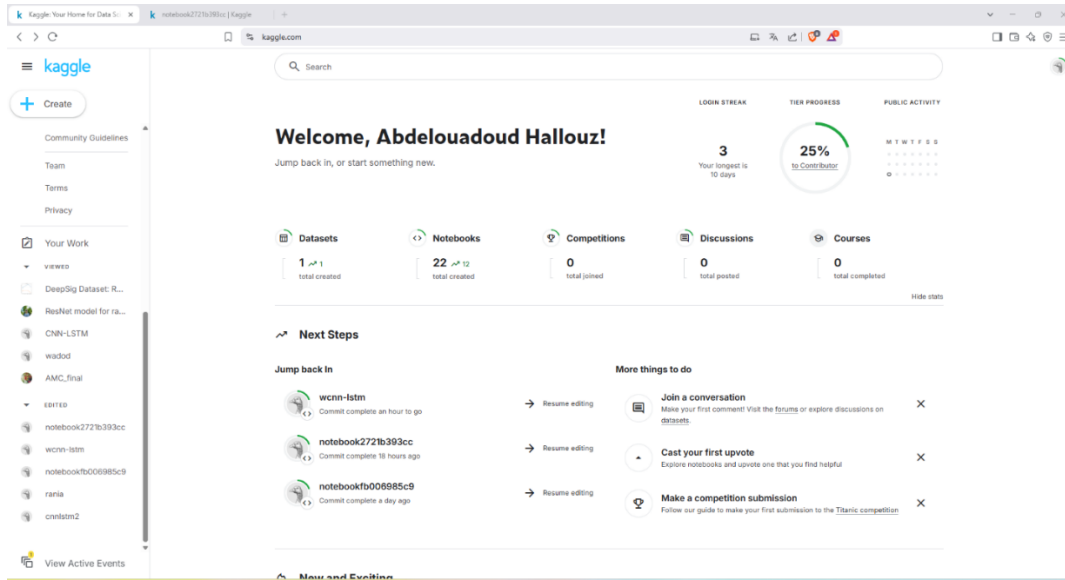


Figure II. 5: Environnement de simulation "Kaggle".

- ❖ **Streamlit** : est un Framework open-source Python qui permet de créer des applications web interactives de manière rapide et intuitive, spécialement conçu pour les projets de data science et d'apprentissage automatique. Il permet aux développeurs de transformer des scripts Python en application web (interface utilisateur) dynamique, sans nécessiter de connaissance en développement web (HTML, CSS, JavaScript). Dans le cadre de de notre projet, Streamlit a été utilisé pour tester, visualiser et afficher les résultats de prédictions du modèle AMC intelligent à partir de différents paramètres du canal.

2.5. Modèle d'apprentissage profond

2.5.1. Choix de l'architecture

Pour résoudre le problème de la prise de décision d'adaptation modulation/codage (AMC) dans des environnements de propagation dynamiques, il est essentiel de sélectionner une architecture d'apprentissage profond pertinente pour cette tâche.

Les réseaux de neurones profond DNN (Deep Neural Networks), sont des architectures avancées de modèle d'apprentissage profond inspirée du cerveau humain, composée de couches interconnectées de neurones artificielles,

Leur structure hiérarchique comprend trois composants principaux « figure II.6 » :

- Couche d'entrée : pour la réception des données brutes.
- Couches cachées : traitement progressive via transformation non linéaires, pour extraire des caractéristiques abstraites.
- Couche sortie : pour la prédiction finale.

Leur fonctionnement repose sur la propagation avant (feedforward) des données à travers les couches, ou chaque neurone effectue une opération de pondération sous forme de combinaison linéaire pondérée, suivie d'une fonction d'activation non-linéaire. Ensuite, les poids de connexion sont ajustés par une rétropropagation (backpropagation) du gradient combiné par des algorithmes d'optimisation, en minimisant l'erreur entre prédictions et résultats réelles lors de l'entraînement.

Les principales caractéristiques des DNN incluent leur profondeur architecturale, leur capacité de généralisation et leur aptitude à modéliser des relations non linéaires complexes. Dans le cadre de l'AMC, cette architecture est adaptée pour prédire les combinaisons modulation – codage optimales à partir des caractéristiques du canal.

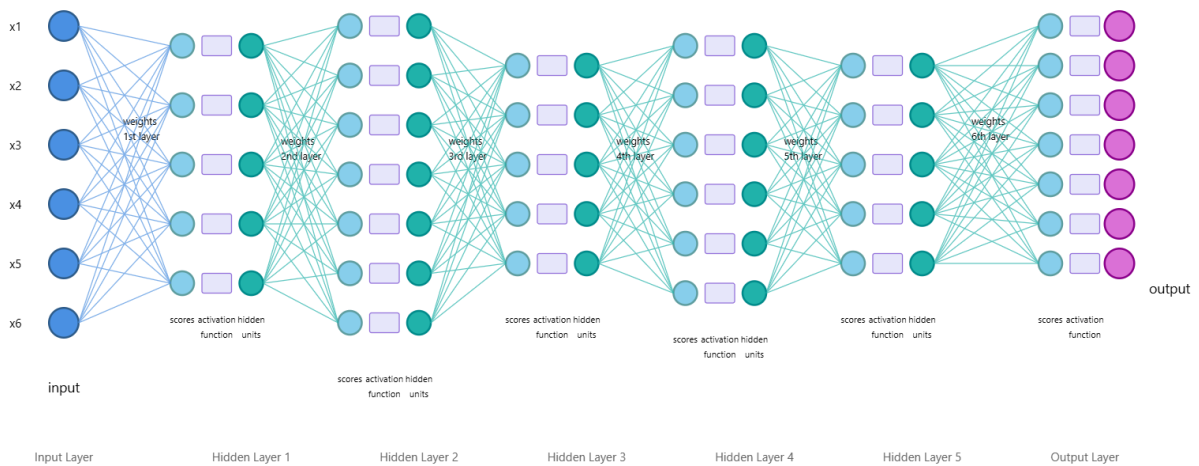


Figure II. 6: Architecture des réseaux de neurone profond [39].

2.5.2. Entrées du réseau (features)

L'entrée de notre réseau de neurones est un vecteur de dimension 6 représentant des caractéristiques extraites du canal de transmission. Ces derniers comprennent : le SNR, le

facteur d'évanouissement (fading), l'étalement Doppler, retard de propagation, ainsi que deux indicateurs dérivés : le SNR effectif et la qualité du canal (CQI). Ces paramètres ont été soigneusement choisis pour représenter l'état du canal de communication sans fil et permettent au modèle DNN d'apprendre à prédire la modulation et codage optimaux adapté à chaque condition du canal.

2.5.3. Sortie du réseau

La sortie du réseau représente la décision d'AMC à prendre, c'est-à-dire de prédire la modulation optimale en fonction du signal reçu et de la qualité estimée du canal. Cette sortie est une probabilité de classification pour chaque modulation, où le modèle génère en sortie un vecteur de probabilité de dimensions correspond aux nombres de classes de modulation. Plus précisément, ce vecteur représente la probabilité que le signal d'entrée appartienne à chaque classe, où chacune correspond à un type de modulation numérique sur l'ensemble des schémas de modulation (classification multiclasse/7 classes). Cette probabilité est produite par une couche dense avec activation softmax en sortie du réseau, permettant d'optimiser la probabilité de chaque classe. Sachant que la prédiction finale consiste à sélectionner la modulation dont la plus forte probabilité (argmax), permettant ainsi une classification précise.

En outre, la décision AMC finale n'est pas uniquement basée sur cette sortie brute, elle est couplée à une stratégie adaptative par des règles prédéfinies, où des architectures spécialisées (DNN) sont entraînées pour différentes conditions du canal.

2.5.4. Fonction de perte

Dans notre approche, le problème est formulé comme une classification multiclasse supervisée où le modèle doit identifier la classe de modulation appropriée à partir d'un signal reçu.

La fonction de perte c'est une mesure de l'erreur entre la sortie prédite et la sortie réelle. Elle guide l'optimisation du modèle. La fonction plus appropriée pour notre contexte est la categorical cross-entropy (entropie croisée catégorielle), cette dernière mesure la différence entre la distribution prédite par le modèle (sortie du softmax) et la distribution réelle (étiquette one-hot), permettant d'optimiser les paramètres du réseau pour maximiser la probabilité de la classe correcte.

Cette fonction de perte est définie lors de la compilation de chaque architecture de modèle dans le code, garantissant une convergence efficace et classification robuste des schémas de modulations.

La fonction de perte d'entropie croisée est définie par l'équation 2-6 ci-dessous :

$$CE = - \sum_{i=1}^N P_i \cdot \log(\hat{P}_i) \quad (2-7)$$

Avec :

- CE : le nombre total de classe.
- P_i : est l'étiquette réelle (encodée en one-hot).
- \hat{P}_i : la probabilité prédite par le modèle pour la classe i .
- N : le nombres de classes.

2.5.5. Algorithmes d'optimisation

L'optimiseur dans le DL représente un algorithme utilisé pour ajuster les poids du modèle afin de minimiser la fonction de perte. Notre modèle d'apprentissage profond utilise l'algorithme Adam (Adaptive Moment Estimation) pour l'optimisation des poids. Il s'appuie sur une version avancée de la descente de gradient stochastique, en adaptant automatiquement le taux d'apprentissage de chaque paramètre grâce à la combinaison des moments du premier et du second ordre. Cette approche améliore à la fois la stabilité de l'entraînement et la robustesse du modèle face aux variations des signaux radio.

2.5.6. Hyperparamètres

Le modèle d'apprentissage profond repose sur un ensemble d'hyperparamètres soigneusement choisis qui contrôlent le processus d'apprentissage d'un modèle. Ils sont fixés avant l'apprentissage et influencent directement les performances et la généralisation du modèle entraîné, en déterminant la structure du réseau, la dynamique d'apprentissage et les méthodes de régularisation.

Les principaux hyperparamètres en apprentissage profond incluent :

- Learning Rate (taux d'apprentissage) : il contrôle la vitesse des mises à jours des poids du réseaux lors de l'entraînement. Un taux trop élevé peut empêcher la convergence, tandis qu'un taux trop faible peut ralentir l'apprentissage.

- Epochs (nombres d'époques) : il s'agit du nombre de fois que l'algorithme parcourt l'ensemble du jeu de données d'entraînement.
- Batch size (taille du lot) : représente le nombre d'échantillons utilisés pour mettre à jours les poids lors d'une itération. Elle influence la stabilité et la rapidité de l'entraînement.
- Fonction d'activation : fonction non linéaires appliquées à la sortie d'un neurone. Elle permet au réseau de modéliser des relations complexes. Exemples : ReLU, Sigmoïde, Tanh, LeakyReLU.

2.5.7. Techniques de régularisation

Pour limiter le surapprentissage (overfitting) et améliorer la capacité de généralisation du modèle, plusieurs techniques de régularisation ont été mises en œuvre, notamment :

- Le Dropout : qui consiste à désactiver aléatoirement une partie des neurones d'une couche pendant l'entraînement, ainsi que leurs connexions. Cette technique est appliquée afin d'éviter que le modèle ne s'ajuste trop étroitement aux données d'apprentissage.
- Batch normalization (La normalisation par lots) : normalise les entrées de chaque couche, permet de stabiliser et d'accélérer l'apprentissage en normalisant les activations, tout en jouant un rôle de régularisation supplémentaire. Dans l'AMC, elle est particulièrement utile pour gérer les variations d'amplitude du signal reçu.
- Coefficient L1/L2 : méthodes permettant de pénaliser les poids excessifs dans le réseau pour améliorer la généralisation. L1 favorise la parcimonie et L2 réduit les grandes valeurs de poids.

2.6. Évaluation des performances

Pour évaluer les performances du modèle AMC, plusieurs métriques complémentaires sont utilisées.

- ✓ **Le taux de précision globale (Accuracy)** : mesure la proportion de signaux correctement classés parmi l'ensemble des échantillons testés. Cette précision est suivie à chaque époque d'entraînement ainsi que durant les phases de validation et de test.
- ✓ **Les courbes d'apprentissage** : sert à présenter l'évolution de la précision et de la perte sur les ensembles d'entraînement et de validation au fil des époques.

- ✓ **Le Rapport de classification** : inclue la précision (Precision), le rappel (Recall) et le score F1 pour chaque classe de modulation. Cela permet de mieux comprendre le comportement du modèle pour chaque classe.

- **Précision** : c'est la proportion des prédictions positives (correcte) parmi toutes les prédictions faites pour une classe donnée. Elle mesure la fiabilité du modèle lorsqu'il prédit cette classe.

$$\text{Precision} = \frac{\text{Vrai Positifs (TP)}}{\text{Vrai positif(TP) +Faux Positifs (FP)}} \quad (2-8)$$

- **Rappel (Recall)** : proportion de prédiction positive correctes parmi tous les cas réellement positifs. Il mesure la capacité du modèle à détecter toutes les instances positives.

$$\text{Rappel} = \frac{\text{Vrai Positifs (TP)}}{\text{Vrai positif (TP) +Faux Négatifs (FN)}} \quad (2-9)$$

- **F1-score** : c'est la moyenne harmonique entre la précision et le rappel. Il équilibre les deux métriques, utilise pour les classes déséquilibrées.

$$\text{F1-score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \quad (2-10)$$

- ✓ **La matrice de confusion** : est un outil fondamental pour visualiser les erreurs, en particulier entre les classes proches. Elle compare les classes réelles (lignes) aux classes prédites (colonnes), permettant d'identifier non seulement le taux de bonnes classifications (valeurs sur la diagonale), mais aussi les erreurs spécifiques (hors diagonale).

Enfin, une évaluation par plages de SNR et conditions de canal est réalisée afin de juger la robustesse du modèle face aux variations du bruit, ce qui est essentiel dans les environnements dynamiques des réseaux 5G et 6G.

2.7. Conclusion

Ce chapitre a permis de définir clairement le cadre théorique et méthodologique de l'intégration de l'apprentissage profond à l'AMC, dans le contexte des réseaux 5G et 6G. Nous avons introduit les différents éléments constituant notre système intelligent, basé sur une architecture DNN et enrichi par des techniques d'apprentissage avancées. Cette architecture permet d'assurer une classification fiable et performante, même dans des environnements instables et très variables. Ces bases nous permettent désormais d'aborder, dans le chapitre suivant, la phase expérimentale, avec l'analyse des résultats et la validation des performances du système proposé.

***Chapitre 03 : Tests et travaux
expérimentaux***

3.1. Introduction

Après avoir posé les fondements théoriques et méthodologiques dans les chapitres précédents, ce chapitre présente l'ensemble des travaux expérimentaux menés dans le cadre de nos recherches de projet de fin d'étude, depuis les premières explorations architecturales jusqu'à l'aboutissement d'un modèle optimale. Cette série d'essai, organisés chronologiquement en combinant les architectures LSTM, CNN, hybride et DNN en passant par les défis d'implémentations, de simulation et les optimisations ajoutées. Cela permet de former une base précieuse, où chaque test documenté avec ses résultats et ses limites, contribue à l'amélioration progressive du modèle final.

3.2. Méthodologie expérimentale générale

L'ensemble de nos expérimentations s'appuie sur une méthodologie cohérente « figure III.1 » visant à assurer la comparabilité des résultats et la validité des conclusions. Par ailleurs, la méthodologie intègre une progression méthodique des essais, où chaque étape vise à corriger les limites identifiées précédemment. Chaque essai suit le protocole standardisé suivant :

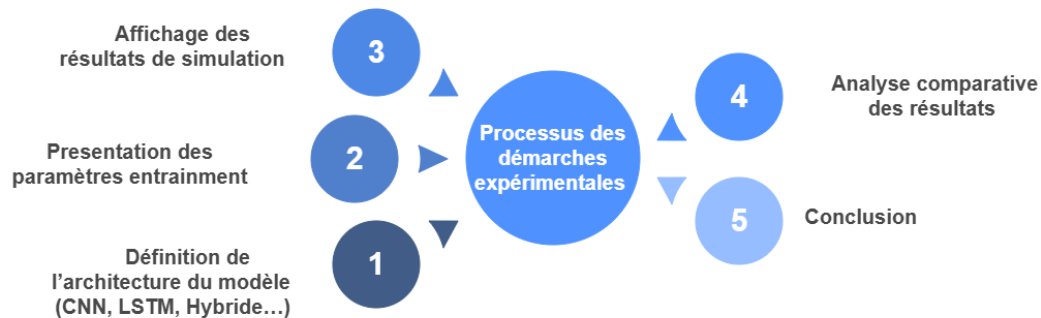


Figure III. 1: Méthodologie d'entraînement d'ensemble d'expérimentations.

3.3. Bilan des tests successifs et limitations

3.3.1. Expérimentation 1 – Modèle LSTM

Comme une première vision de construction de notre modèle, nous avons implémenté le modèle des réseaux LSTM ajustés par les paramètres présentés dans le tableau III.1 ci-dessus :

Paramètre	Valeurs
Epoques	10
Batch size	64
Couche LSTM	2
Couche dense	3
Dropout	2 couches Taux = 0.5
T'aux d'apprentissage	0.001
Modulation	7 classes
SNR	26 valeurs (-20 à 30dB)
Temps d'entraînement	72 heures

Tableau III. 1: paramètres d'entraînement du modèle LSTM.

Analyse de la 1^{ère} expérimentation et Interprétation des résultats :

Les résultats obtenus ont révélé les performances suivantes :

Métrique d'évaluation	Observations
Courbe d'apprentissage	Underfitting, stabilité médiocre, Absence de convergence.
Matrice de confusion	Problème de classification uniforme (256QAM).
Précision globale	14.29% (faible)

Tableau III. 2: Observations des résultats du 1er essai modèle LSTM.

➤ **Les courbes d'apprentissage** : la fonction de perte (Loss) reste stable (~1.946) tout au long de l'entraînement « figure III.2 », confirmant ainsi l'échec du processus d'apprentissage. De plus, les courbes de précision et de perte pour l'ensemble de validation sont instables,

suggérant que le modèle ne généralise pas bien. Une amélioration négative sur les 9 premiers époques, indiquant que le modèle ne parvient pas à apprendre.

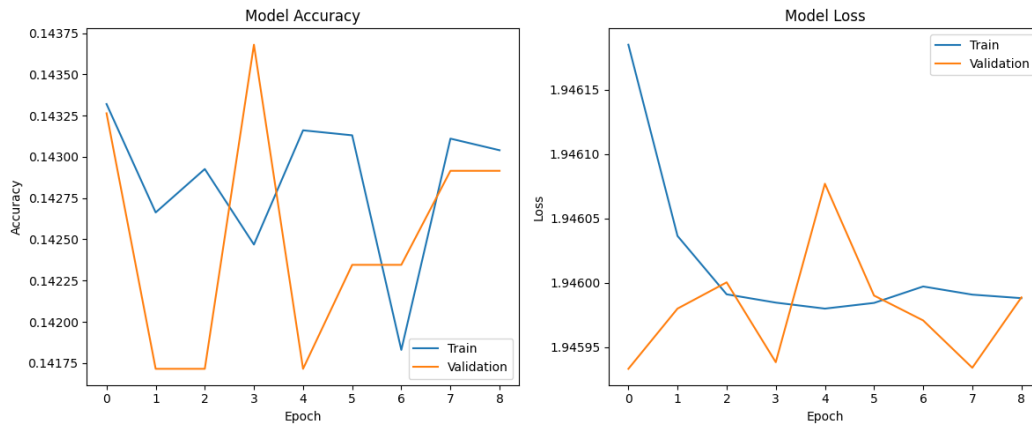


Figure III. 2: Courbes d'apprentissage LSTM.

➤ **Problème de classification uniforme :** la matrice de confusion « figure III.3 » montre que le modèle attribue systématiquement toutes les prédictions à une seule classe (256QAM), montrant qu'il n'a pas appris à extraire les caractéristiques discriminantes des signaux.

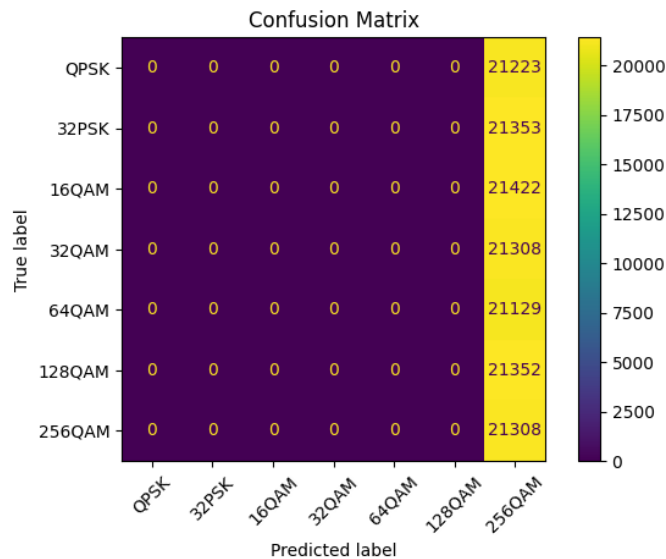


Figure III. 3: Matrice de confusion du modèle LSTM.

➤ **Accuracy** : le modèle atteint seulement 14.29% de précision globale « figure III.4», ce qui signifie que le modèle n'a pas appris à différencier les classes.

```
4660/4660 - 105s - 23ms/step - accuracy: 0.1429 - loss: 1.9460  
Model accuracy: 14.29%
```

Figure III. 4: Précision du modèle LSTM.

➤ **Temps d'entraînement excessif** : 72 heures d'entraînement pour des résultats aussi faibles, représente une utilisation inefficace des ressources.

En conclusion, Nous observons que le modèle LSTM présente un sous-apprentissage (underfitting). Ces résultats insatisfaisants et inadaptés à notre tâche, permettent d'identifier les limites du modèle LSTM seul dans ce contexte, soulignant donc la nécessité d'une réévaluation des hyperparamètres et d'une modification potentielle de l'architecture afin d'améliorer la performance de notre modèle.

3.3.2. Expérimentation 2 – Modèle CNN

Suite aux défis rencontrés avec l'architecture LSTM précédemment explorée, nous avons opté pour une approche alternative utilisant un réseau de neurones convolutif CNN. Cette transition s'inscrit dans notre démarche itérative visant à optimiser les performances du modèle.

Le tableau III.8 présente les principaux hyperparamètres sélectionnés pour cet essai. Cette architecture reflète notre volonté d'extraire des caractéristiques complexes à partir des signaux modulés.

Paramètre	Valeurs
Epoques	20
Batch size	4
Couche CNN	6
Couche dense	3
Dropout	6 couches Taux = 0.5
Taux d'apprentissage	0.001
Modulation	4 classes
SNR	2 plages [0- 10] [10 – 30]
Temps d'entraînement	15 heures 30min

Tableau III. 3: Paramètres d'entraînement du modèle CNN.

Analyse de la 2^{ème} expérimentation et interprétation des résultats :

Le deuxième test basé sur une architecture CNN, présente une amélioration significative en termes de précision globale « figure III.6 » et du temps d'entraînement, passant de 14% pour LSTM à 80% pour CNN, et de 72 heures à 15heures et 30 min respectivement.

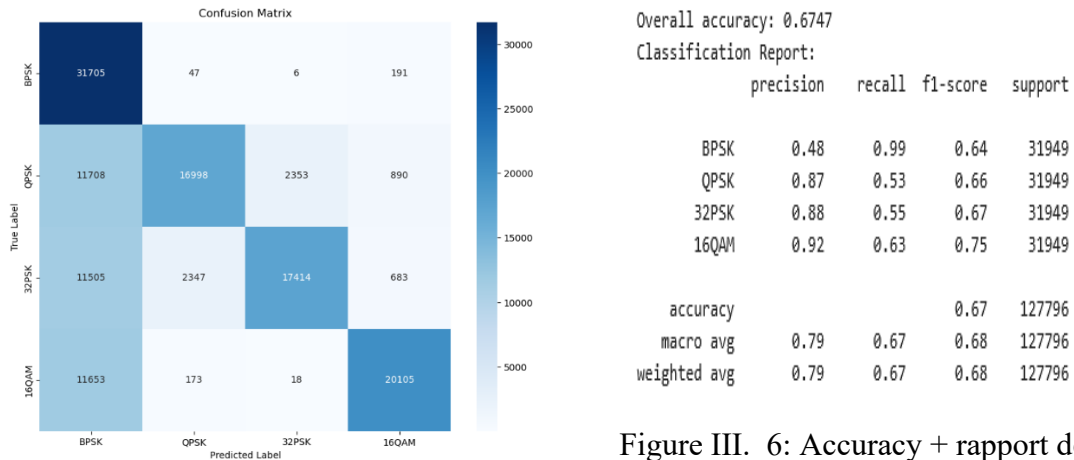


Figure III. 5: Matrice de confusion du modèle CNN.

Figure III. 6: Accuracy + rapport de classification du modèle CNN.

Cependant, les courbes d'apprentissage dans les deux plages de SNR montrent un apprentissage instable et une faible généralisation en faible SNR « figure III.7 ». D'autre part, la matrice de confusion « figure III.5 » montre un déséquilibre et confusion entre la plupart des classes de modulations.

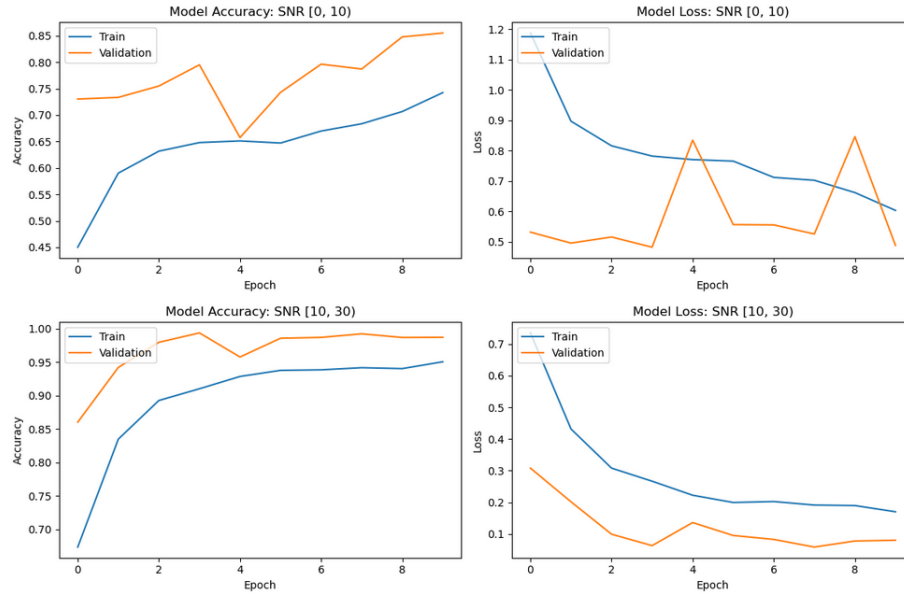


Figure III. 7: Courbes d'apprentissage du modèle CNN.

En conclusion, ce modèle a besoin d'amélioration des paramètres et évolution d'architecture en raison des limites représentées dans le tableau 9 ci-dessous.

Métrique d'évaluation	Observations
Courbe d'apprentissage	Overfitting, apprentissage inefficace/ instable.
Matrice de confusion	Déséquilibre et confusion entes classes.
Précision globale	67.47%

Tableau III. 4: Observations des résultats du 2ème essai CNN.

3.3.3. Expérimentation 3 – Modèle CNN-LSTM (augmentation des classes de modulation)

Par la suite, une approche hybride CNN-LSTM a été implémentée pour exploiter à la fois les caractéristiques spatiale et temporelles des signaux. Le tableau III.5 résume les paramètres principaux utilisés.

Cette expérimentation vise à évaluer les limites de notre architecture, par l'augmentation du nombre de schémas de modulations à classifier, en testant la robustesse du modèle et sa capacité à généraliser dans des conditions plus complexes.

Paramètre	Valeurs
Epoques	200
Batch size	64
Couche CNN	4 couches
Couche LSTM	1
Couche dense	5 couches (512- 1024)
Dropout	4 couches Taux = [0.2 – 0.5]
Taux d'apprentissage	0.0001 – 0.001
Modulation	10 classes
SNR	4 valeurs (22 à 30dB)
Temps d'entraînement	9 heures

Tableau III. 5: Paramètres d'entraînement du modèle CNN-LSTM (modulations multiples).

Analyse de la 3^{ème} expérimentation et interprétation des résultats :

- **Les courbes d'apprentissage** « figure III.8 » illustrent une convergence plus lente, en conservant sa stabilité.

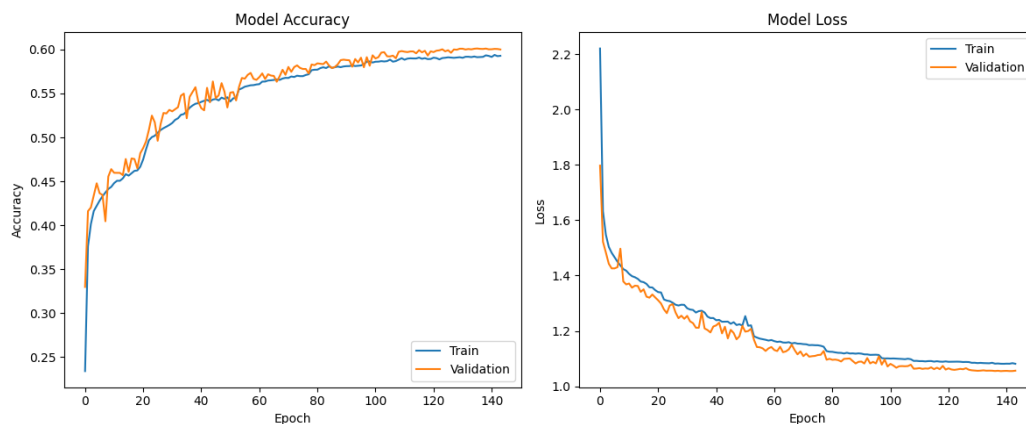


Figure III. 8: Courbes d'apprentissage du Modèle CNN-LSTM (modulations multiples).

- **La matrice de confusion** révèle plus de confusion inter-classe qu'auparavant, en particulier entre les classe proche « figure III.9 », indiquant que le modèle atteint ses limites de différenciation avec 10 classes.

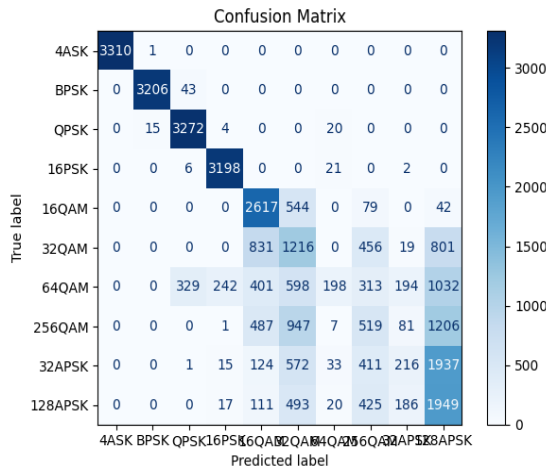


Figure III. 9: Matrice de confusion du Modèle CNN-LSTM (modulations multiples).

	precision	recall	f1-score	support
4ASK	1.0000	0.9997	0.9998	3311
BPSK	0.9950	0.9868	0.9909	3249
QPSK	0.8962	0.9882	0.9400	3311
16PSK	0.9198	0.9910	0.9541	3227
16QAM	0.5725	0.7974	0.6665	3282
32QAM	0.2783	0.3659	0.3161	3323
64QAM	0.6622	0.8599	0.1098	3307
256QAM	0.2356	0.1598	0.1904	3248
32APSK	0.3095	0.0653	0.1078	3309
128APSK	0.2797	0.6089	0.3834	3201
accuracy			0.6012	32768
macro avg	0.6149	0.6023	0.5659	32768
weighted avg	0.6152	0.6012	0.5653	32768

Figure III. 10: Rapport de classification du Modèle CNN-LSTM (modulations multiples).

- **Le rapport de classification** « figure III.10 », complète l'analyse par ses métriques où des chutes est observées dans le rappel, F1 score concernant certaines modulations.
- **Precision globale** : une diminution importante de la précision globale à 60.12% « figure 28 » par rapport à l'essai précédent (67%).

```

Restored model, accuracy: 60.12%

Summary of Improvements:
1. Added data shuffling during preprocessing and training
2. Implemented enhanced early stopping with monitoring of both accuracy and loss
3. Added learning rate reduction on plateau
4. Performed random search for hyperparameter optimization
5. Added L1 and L2 regularization to prevent overfitting
6. Best hyperparameters found: {'learning_rate': 0.0004822513868760003, 'dropout_rate': 0.3137120892351059, 'lstm_units': 236, 'dense_units': 687, 'l1_reg': 5.161322068719693e-05, 'l2_reg': 0.0004324837566608285}
    
```

Figure III. 11: Précision globale du Modèle CNN-LSTM (modulations multiples).

En conclusion, le tableau III.6 montre des performances faibles par rapport aux besoins de notre approche. Avec l'augmentation des nombres de class, l'architecture devient plus complexe en rendant le modèle instable et incapable à généraliser.

Métrique d'évaluation	Observations
Courbe d'apprentissage	Lente convergence, stabilité.
Matrice de confusion	Forte confusion, reconnaissance faible
Précision globale	60.12%

Tableau III. 6: Observations des résultats du 3ème expérimentation CNN-LSTM (modulations multiples).

3.3.4. Expérimentation 4 – Modèle DNN

Dans le cadre de l'amélioration progressive de notre système AMC intelligent, une quatrième expérimentation a été réalisée en s'appuyant cette fois-ci sur une architecture de réseau de neurone profond DNN. L'étape présente vise à évaluer la capacité du DNN à classifier et prédire les différents schémas de modulations dans diverses conditions de canal, en utilisant une configuration d'hyperparamètres spécifiques « tableau III.7».

Paramètre	Valeurs
Epoques	100
Batch size	32
Couche dense	4
Dropout	0.4
Taux d'apprentissage	0.001
Modulation	5
SNR	[-20 – 30dB]
Temps d'entraînement	3 heures

Tableau III. 7: Paramètres d'entraînement du modèle DNN.

Analyse de la 4^{ème} expérimentation et interprétation des résultats :

- **Les courbes d'apprentissage :** la figure III.12 des courbes d'accuracy et loss, présentent une convergence progressive de la fonction de perte sur les ensembles de train et de validation. Cependant, on observe un écart notable entre ces deux courbes après certains nombres d'époques. Cette divergence suggère un début d'overfitting, où le modèle mémorise trop les exemples d'entraînement.

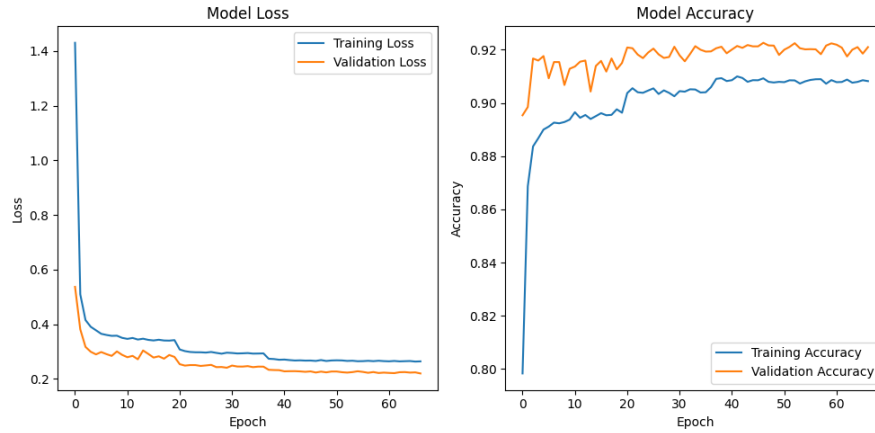


Figure III. 12: Courbes d'apprentissage du Modèle DNN.

- **La matrice de confusion** : la matrice obtenue dans la figure III.13 , met en évidence des difficultés du modèle à distinguer certaines classes de modulations, en particulier celles ayant des caractéristiques proches.

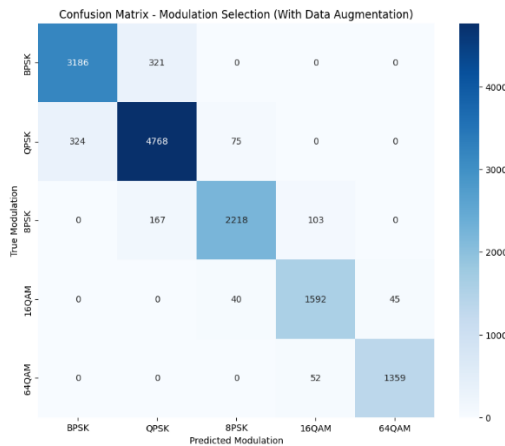


Figure III. 13: Matrice de confusion du Modèle DNN.

Classification Report:

	precision	recall	f1-score	support
BPSK	0.91	0.91	0.91	3507
QPSK	0.91	0.92	0.91	5167
8PSK	0.95	0.89	0.92	2488
16QAM	0.91	0.95	0.93	1677
64QAM	0.97	0.96	0.97	1411
accuracy			0.92	14250
macro avg	0.93	0.93	0.93	14250
weighted avg	0.92	0.92	0.92	14250

Figure III. 14: Rapport de classification du Modèle DNN.

Test Accuracy: 0.9209

Figure III. 15: Précision globale du Modèle DNN.

- **Le rapport de classification** : d'après l'analyse par classe de modulations selon des métriques (précision, Recall, F1-score), les classes de modulations les plus simples

conserver de bonnes performances, tandis que ces performances chutent pour les classes les plus complexes, comme il est montré dans le rapport de classification « figure 31 ».

- **La précision globale :** lors de ce test, la précision du modèle atteint jusqu'à 92.09% « figure 32 » indiquant un progrès notable par rapport aux tests précédents. Néanmoins, cette performance reste insuffisante et n'est pas encore pleinement adaptée aux exigences de notre approche.

Métrique d'évaluation	Observations
Courbe d'apprentissage	Overfitting.
Matrice de confusion	Confusion entre modulations proches.
Précision globale	92.09%

Tableau III. 8: Observations des résultats du 4ème expérimentation DNN.

En conclusion, cette expérimentation a permis d'identifier les limites de la configuration initiale du modèle DNN, notamment en termes de généralisation et de discrimination entre classes proches. Le tableau III.8 englobe les principales observations qui ont guidé aux ajustements ultérieurs et en conduisant à une architecture finale, plus performante et mieux adaptée à la tâche de modulation et codage adaptatif.

3.4. Transition vers l'optimisation finale

L'ensemble de ses expérimentations converge naturellement vers l'identification d'une configuration optimale, synthèse des apprentissages tirés de chaque test. Cette progression méthodologique nous a menés à développer un modèle final qui capitalise sur les forces identifiées tout en corrigeant les faiblesses observées dans les expérimentations antérieurs.

Le modèle optimal ainsi obtenu représente l'aboutissement de notre démarche de recherche, intégrant les meilleures pratiques architecturales, les stratégies d'optimisation les plus efficaces, et les techniques de régularisation les mieux adaptées à notre problématique.

3.5. Conclusion

Ce chapitre a tracé notre parcours expérimental dans le développement d'un agent AMC intelligent. Au fil des quatre essais, nous avons rencontrés diverses difficultés liées à la complexité des données, à la sensibilité aux conditions du canal, ainsi qu'aux choix d'architecture et d'hyperparamètres. Chaque étape a été l'occasion d'analyser, d'ajuster et d'optimiser les différents modèles, tout en documentant les résultats obtenus, les erreurs rencontrés et les améliorations apportées.

Encore, l'analyse et l'interprétation des résultats obtenus de chaque expérimentation nous ont permis de mettre en évidence les facteurs critiques influençant les performances de système, tout en soulignant les compromis entre précision de classification, complexité computationnelle et temps d'entraînement. Cette démarche itérative illustre la nature exploratoire de la recherche expérimentale, où l'apprentissage se fait autant par les succès que par les échecs.

Ce travail progressif ouvre la voie à une configuration d'un modèle à la fois performant, stable et adapté à diverses conditions environnementales, qui sera détaillé dans le prochain chapitre.

Chapitre 04 :

*Simulation et discussion des
résultats finaux*

4.1. Introduction

Après avoir présenté dans les chapitres précédents les fondements théoriques de la techniques AMC, et l'intégration de l'intelligence artificielle dans les systèmes de communication sans fil de nouvelle génération, ainsi que les essais successifs qui nous ont permis d'aboutir à un modèle final performant. Ce chapitre est dédié à l'évaluation pratique de ce modèle.

Nous débuterons par la description du processus suivi lors de la mise en œuvre de notre agent intelligent, illustré sous forme d'un organigramme synthétique des différentes étapes de traitement. Nous présenterons, également, les différents paramètres utilisés dans la conception de l'architecture DNN et d'entraînement du modèle. Par la suite, les résultats obtenus des simulations seront exposés et analysés, mettant en lumière les performances du modèle dans divers scénarios radio. Enfin, nous exposerons la phase de test de l'agent intelligent via une interface utilisateur développée, permettant d'évaluer la capacité de prédiction et d'adaptation du système à différentes conditions du canal.

4.2. Processus d'implémentation du modèle final

La méthodologie adoptée pour la mise en œuvre de la classification des modulations dans le cadre de l'AMC basé sur l'apprentissage profond, s'appuie sur une approche structurée en plusieurs étapes clés. Ce processus vise à exploiter la puissance des réseaux de neurones profonds, en particulier l'architecture DNN pour une classification précise des différentes classes de modulation.

Le schéma présenté dans la « figure IV.1 » illustre de manière organisée les différentes phases du développement de notre modèle.

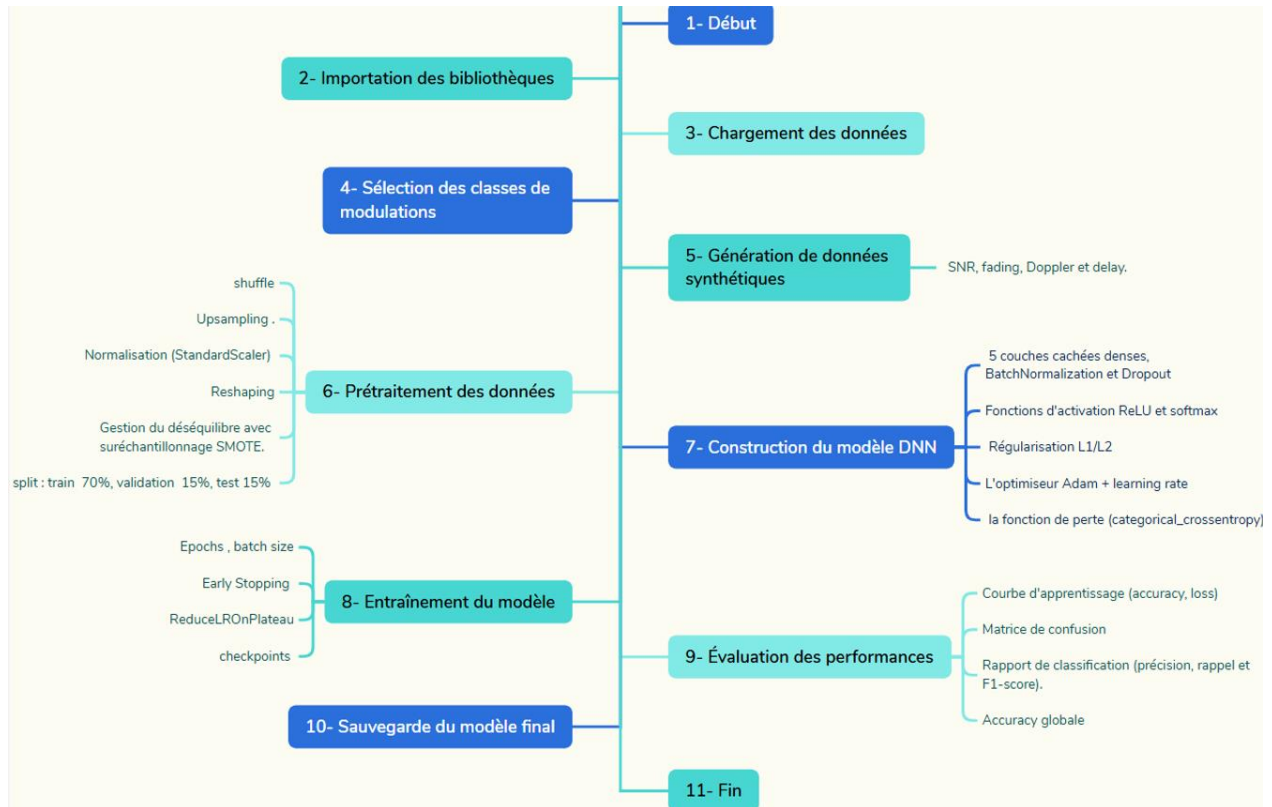


Figure IV. 1: Organigramme méthodologique du modèle AMC basé sur DL.

Les étapes principales de ce processus méthodologique sont les suivantes :

1. **Début** (initialisation du processus) : nous avons préparé le cadre d'exécution du projet, en configurant l'environnement et les outils de développement.
2. **Importation des bibliothèques** : nécessaires pour le traitement des données et la construction du modèle du DL, ainsi que l'entraînement, l'évaluation et la visualisation des résultats.
3. **Chargement des données** : les données ont été téléchargées à partir de la dataset RadioML2018, qui contient des signaux modulés en différentes classes (QAM, PSK...), simulées dans divers environnements bruités.
4. **Sélection des classes de modulations** : cette étape consiste à sélectionner un sous-ensemble des classes de modulations existées, sur lesquelles le modèle va être entraîné.
5. **Génération de données synthétiques** : création de 30 000 échantillons à partir de distributions réalistes (SNR, fading, Doppler, DeLay), calcul de deux features

supplémentaires (effective SNR et Channel Quality Index) ainsi que l'attribution d'un label de modulation optimale via un système de règles.

6. Équilibrage du dataset (Upsampling) : la technique du suréchantillonnage est utilisé pour équilibrer un jeu de données déséquilibrés, en augmentant artificiellement le nombre d'échantillons dans les classes à l'aide des méthodes telle que SMOTE (Synthetic Minority Over-sampling Technique).

7. Prétraitement des données : les données subissent plusieurs traitements essentiels incluant :

- La normalisation (StandardScaler) sur les features d'entrée.
- Le reshaping : opération qui consiste à modifier la forme ou la dimension d'un tableau de données sans changer le contenu, ce qui permet d'adapter les données d'entrée à la structure attendue par un modèle.
- Le mélange des données (Shuffle).
- Séparation en ensembles d'entraînement, validation et de test (70%, 15%, 15%).
- Data augmentation : consiste à générer des variations artificielles des signaux modulés d'origines, pour améliorer la généralisation du modèle.

8. Construction du modèle DNN :

- Architecture séquentielle avec 5 couches denses, fonction d'activation RELU (couches cachées) et softmax (sortie).
- Techniques de Régularisation : le Dropout et L1/L2, pour éviter le surapprentissage.
- Compilation avec l'optimiseur Adam.
- La fonction de perte (Categorical Crossentropy).

9. Entraînement du modèle : le modèle est entraîné sur un nombre défini d'époques et de batch size, selon un processus d'apprentissage basé sur les callbacks suivant :

- Early Stopping (en cas de stagnation de la validation).
- ReduceLRonPlateau (réduction dynamique du taux d'apprentissage).
- Checkpoint (sauvegarde des poids optimaux).

10. Evaluation des performances : l'analyse des résultats sur l'ensemble de test, à l'aide de différentes métriques :

- Courbe d'apprentissage (accuracy, loss).

- La matrice de confusion.
- Rapport de classification (précision, rappel, F1-score).
- Precision globale.

11. Sauvegarde du modèle final : Le modèle final et ses poids sont sauvegardés pour une utilisation ultérieure.

12. Fin : fin du processus de construction du modèle AMC. Le système est prêt à être testé pour prédire les classes de modulation.

4.3. Paramètres du modèle d'apprentissage profond

4.3.1. Bibliothèques Python

Le tableau IV.1 présente les principales bibliothèques Python utilisées dans notre code, accompagnée de la description de leur fonction. Chacune joue un rôle crucial dans le traitement des données, l'implémentation de l'agent, l'entraînement et l'évaluation de notre modèle AMC basé sur le DL.

Bibliothèque	Fonction principale dans le projet
Numpy	Traitement numérique, manipulation de tableaux multidimensionnels.
Pandas	Manipulation et analyse des données tabulaire (Dataframe)
h5py	Lecture des fichiers de données au format HDF5
scikit-learn	Prétraitement des données, séparation train/test, métriques d'évaluation, random search
TensorFlow	Création, entraînement et évaluation du modèle de Deep Learning.
Keras	API haut niveau de TensorFlow pour la définition des architectures de réseaux de neurones.
Matplotlib	Visualisation des résultats (graphes, courbe...).
Random	Génération de nombres aléatoires pour le mélange des données et contrôle de la reproductibilité
os	Gestion des chemins d'accès et des fichiers dans l'environnement Kaggle/Jupyter
gc	Gestion de la mémoire (libération manuelle lors de l'entraînement de gros modèles)
scipy.stats	Génération aléatoire d'échantillons pour la recherche d'hyperparamètres.
Imbalanced-learn	Extension de scikit-learn dédiée à la gestion des jeux de données déséquilibrés, notamment par des techniques de suréchantillonnage.

Tableau IV. 1: Les principales bibliothèques python utilisés dans le modèle du DL.

4.3.2. Paramètres des données et prétraitement

Le tableau IV.2 résume les caractéristiques principales du jeu de données utilisé pour configurer l'environnement de simulation et prétraitement des données. Les simulations ont été effectuée à l'aide de l'environnement Python 3.10, avec différentes bibliothèques.

Paramètre	Valeur	Description
Source	RadioML 2018 (GOLD_XYZ_OSC.0001_1024.hdf5)	Dataset utilisé pour l'apprentissage et la validation, contenant des signaux modulés variés.
Classe de modulation	BPSK, QPSK, 8PSK, 16QAM, 32QAM, 64QAM, 128QAM.	Sous-ensemble de modulations utilisées pour la classification.
Plage SNR	Train: [-20 – 30dB] Test interface: [-15 – 35dB]	Niveaux SNR sélectionnés à partir des données originales.
Split données (train/test)	70% (train), 15% (test), 15% (validation)	Répartition des données pour l'entraînement et le test avec mélange aléatoire.
Graine aléatoire (Mélange des données)	42	Assure la reproductibilité des résultats et évite le biais dans l'ordre des données.

Tableau IV. 2: Paramètres des données et prétraitement du modèle final.

4.3.3. Paramètres d'architecture du modèle DNN final

Par la suite, Le modèle proposé repose sur une architecture DNN représentée par un ensemble des paramètres utilisé pour traiter les signaux RF modulés, le tableau IV.3 ci-dessous illustre les différentes paramètres clés :

Paramètres	Valeurs	Description
Type d'architecture	DNN	Capacité d'apprendre automatiquement des relations complexes et non linéaires entre les caractéristiques du canal et modulations optimal à appliquer.
Entrées	Vecteur de dimensions 6	Vecteur de caractéristiques du canal radio. (SNR, Fading, Doppler, DeLay...).
Fonction d'activation	ReLU	Fonction d'activation non-linéaire pour éviter le problème des gradients nuls.
Couches Desnses	5 couches.	Apprentissage des représentations complexes, Traitement non linéaire profond avant la sortie.
Sortie	Softmax	Fonction d'activation utilisée en sortie du modèle pour la classification multiclasse.

Tableau IV. 3: Paramètres d'architecture du modèle final.

4.3.4. Paramètres d'entraînement

Lors de la phase d'entraînement, le modèle intègre différents callbacks pour optimiser la convergence et réduire le risque d'overfitting. Le tableau IV.4 montre les paramètres fondamentaux permettant d'assurer ces conditions.

Paramètre	Valeur	Description
Temps d'entraînement	1 heure	La durée prise par le modèle pour apprendre à partir des données d'entraînement.
Optimiseur	Adam	Optimiseur adaptatif pour la descente de gradient.
Taille du batch	256	Nombre d'échantillons traités par lot.
Nombres d'époques	150	Nombre maximal de cycles d'apprentissage.
Early Stopping	Patience = 25 époques	Arrêt anticipé pour éviter le surapprentissage.
ReduceLRonPlateau	- Facteur = 0.5 - patience = 10	Réduction automatique du Learning Rate en cas de stagnation.
Fonction de perte	Categorical Crossentropy	Utilisée pour la classification multi-classes.
Sauvegarde des poids	Oui	Conservation automatique des meilleurs poids du modèle

Tableau IV. 4: Paramètres d'entraînement du modèle final.

4.3.5. Paramètres des conditions du canal

Le tableau IV.5 présente les plages de variations des principaux paramètres du canal utilisé pour l'entraînement et le test du modèle AMC intelligent.

Condition de canal	Intervalle
SNR	[-15 – 30dB]
Facteur Fading	[0.001 – 1]
Doppler	[0 – 1000 Hz]
Delay spread	[0 - 20µs]

Tableau IV. 5: Paramètres des conditions de canal de transmission.

4.3.6. Les hyperparamètres

Dans le but d'obtenir les meilleures performances possibles tout en limitant le surapprentissage. Une série d'hyperparamètres est intégrée. Sachant que les techniques de régularisation et d'optimisation des hyperparamètres assurent une bonne généralisation du modèle. Les valeurs choisies lors de cette phase sont récapitulées dans le tableau IV.6 ci-dessous :

Paramètre	Intervalle testé	Description
Taux d'apprentissage	0.001	Pas d'ajustement, contrôle la vitesse de la mise à jour des poids, pour optimiser la convergence.
Taux de Dropout	0.4 (1 ^{er} couches) 0.2 et 0.12 (dernières couches)	Probabilité de désactiver des neurones pendant l'entraînement, pour éviter l'overfitting.
Régularisation L1	0.0001	Coefficient de régularisation testée pour réduire la complexité du modèle et améliorer sa généralisation.
Régularisation L2	0.0001	Coefficient de régularisation pour éviter le surapprentissage.

Tableau IV. 6: les hyperparamètres du modèle final.

4.3.7. Métriques d'évaluation

Une fois l'entraînement est terminé, le modèle est évalué sur les données de test comme il est présenté dans le tableau IV.7, les performances du modèle ont été évaluées à l'aide de plusieurs métriques complémentaires, permettant d'analyser finement les performances par classes de modulation.

Métriques	Description
Précision (Accuracy)	Pourcentage de bonnes prédictions sur l'ensemble de test.
Matrice de confusion	Visualisation des prédictions correctes et erronées par classe.
Rapport de classification (Précision / Recall / F1-score)	Rapport détaillé des performances par classe.
Courbe d'apprentissage (accuracy/loss)	Évolution de la précision et de la perte en fonction des époques.

Tableau IV. 7: Paramètres d'évaluation du modèle final.

4.4. Analyse et Interprétation des résultats du modèle final

La présente section illustre les résultats visuels obtenus à partir de la simulation de notre modèle de Deep Learning proposé, avec les interprétations associées. Ces visualisations permettent d'analyser le comportement du modèle au cours de l'apprentissage. Elles offrent ainsi un aperçu pertinent de la performance, la stabilité, et de la capacité de généralisation du modèle.

Le choix du modèle repose comme un équilibre optimal entre précision, stabilité, robustesse au bruit et capacité de généralisation. Le tableau IV.8 résume les performances globales des résultats de la simulation de notre modèle sur différentes conditions de canal.

Métrique d'évaluation	Observations
Courbe d'apprentissage	Stabilité, bonne convergence, absence d'overfitting.
Matrice de confusion	Absence de confusion.
Précision globale	97.07%

Tableau IV. 8: Observations des résultats du modèle DNN final.

❖ Les courbes d'apprentissage :

Les courbes du modèle DNN final présentées à la figure IV.2, mettent en évidence une convergence rapide et stable aussi bien pour la fonction de perte (loss) que pour la précision (accuracy) sur les ensembles d'entraînement et de validation. La perte diminue progressivement au fil des époques, tandis que pour la précision augmente de manière continue. L'absence d'écart notable entre les deux courbes indique une excellente capacité de généralisation du modèle et d'une absence de surapprentissage (overfitting), ce qui reflète l'efficacité des techniques de régularisation mises en œuvre ainsi que l'optimisation des hyperparamètres.

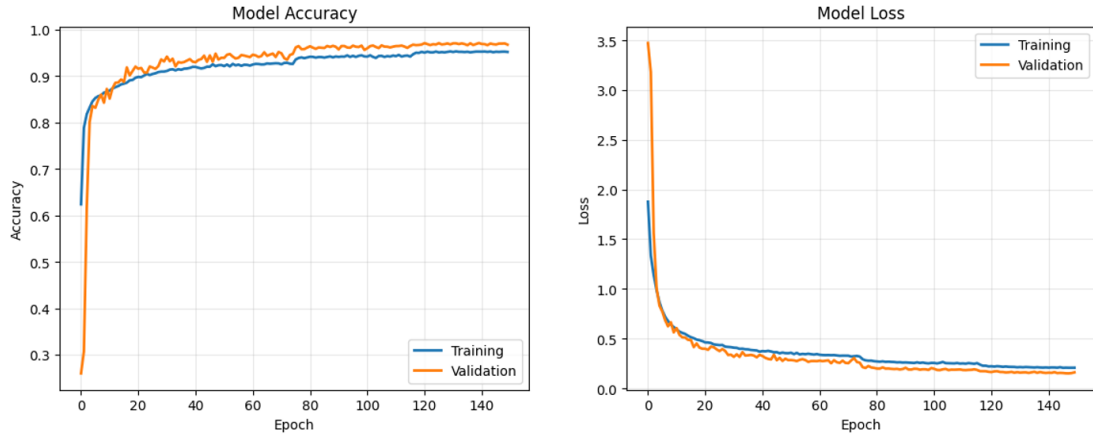


Figure IV. 2: Courbes d'apprentissage du Modèle DNN final.

❖ La matrice de confusion :

La matrice de confusion présente une distribution parfaitement diagonale illustrée dans la figure IV.3, signifie que la majorité des classes de modulations ont été correctement différenciées. Globalement le modèle parvient à minimiser les erreurs critiques, en observant très peu de classification erronée entre classes.

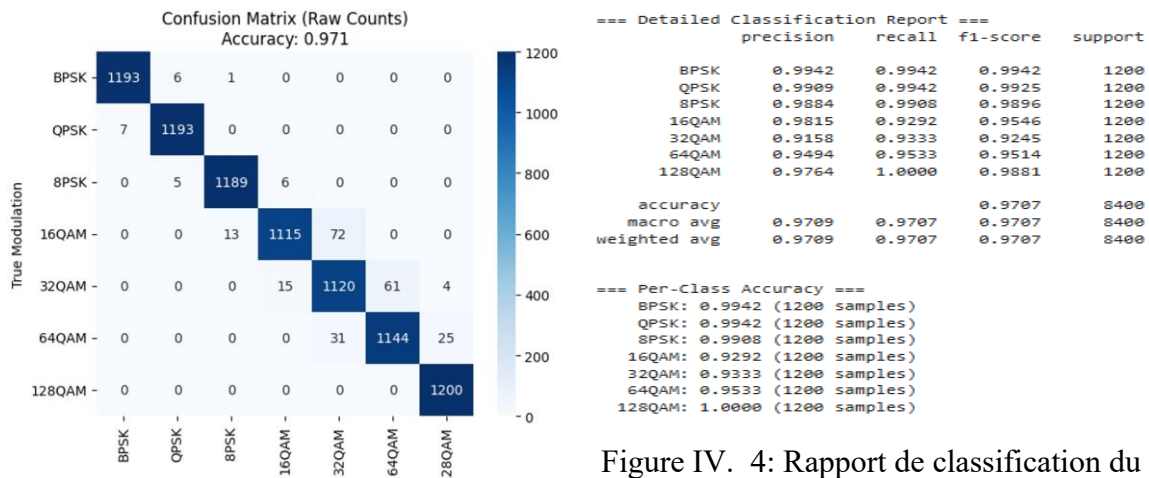


Figure IV. 3: Matrice de confusion du Modèle DNN final.

Figure IV. 4: Rapport de classification du modèle DNN final.

❖ **Le rapport de classification :**

Le rapport de classification montre des valeurs de précision, rappel et F1-score très élevées pour la plupart des classes « figure IV.4 ». L'équilibre entre ces différentes métriques montre l'absence de biais de classification.

❖ **La précision globale :**

L'accuracy globale du modèle DNN finale atteint 97.07% « figure IV.5 », un résultat optimal pour notre approche qui valide la pertinence de l'architecture DNN choisis, l'efficacité de la stratégie d'optimisation ainsi que la capacité de généralisation du modèle même avec des données bruitées.

```
=== Model Evaluation ===
Test Accuracy: 0.9707
Test Loss: 0.1507
```

Figure IV. 5: Précision globale du Modèle DNN final.

❖ **L'ensemble de test :**

Le tableau IV.9 présente les résultats de prédictions du modèle DNN sur cinq échantillons de tests, chacun caractérisé par ces conditions de canal (SNR, fading, doppler, deLay). Le modèle affiche une forte capacité de généralisation et de précision. Pour chaque échantillon testé, le modèle fournit la/les modulation prédite ainsi qu'un score de confiance élevé (tous supérieurs à 0.90), ce qui traduit une forte certitude du modèle dans ses décisions. Une incohérence est observée pour un SNR moyen (15dB), où le modèle prédit BPSK au lieu de 16QAM/32QAM, suggérant un léger biais vers les modulations robustes.

Tableau IV. 9: Test du modèle DNN sur un ensemble d'échantillon des conditions de canal.

```
=== Testing Loaded Model ===
SNR  Fading  Doppler  Delay  Predicted Confidence  Expected
-----
 25   0.90    10     0.2   128QAM    0.927  64QAM or 128QAM
  5   0.30   400     8.0    QPSK     0.909   BPSK or QPSK
 15   0.70   100     2.0     8PSK    0.907  16QAM or 32QAM
 -5   0.20   600    12.0    BPSK     1.000         BPSK
 30   0.95     5     0.1   128QAM    0.984    128QAM
Model loading and testing successful!
```

4.5. Evaluation expérimentale de l'agent AMC intelligent via l'interface de test

Afin d'évaluer le comportement, l'efficacité et la robustesse de notre agent AMC intelligent dans des conditions variées, une interface utilisateur a été développée sous Streamlit. Cette interface permet de simuler différentes conditions de canal et d'observer en temps réelle les décisions de modulation prédites dynamiquement par l'agent. Les résultats suivants illustrent le fonctionnement du système sous plusieurs scénarios représentatifs.

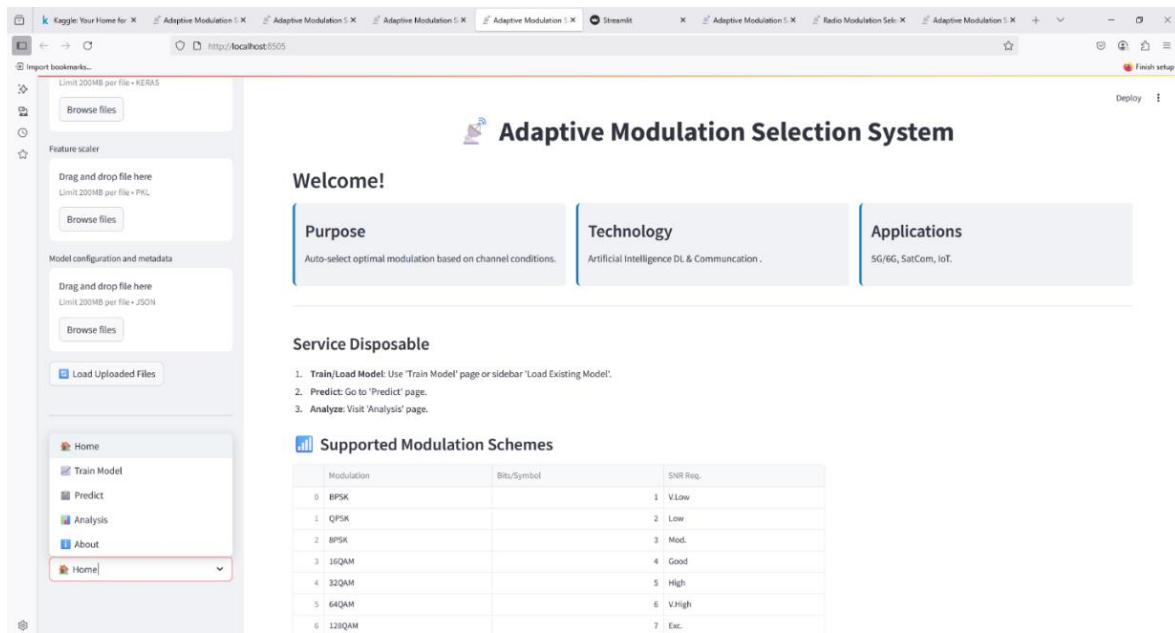


Figure IV. 6: Présentation de la page d'accueil de l'interface.

La figure IV.6 présente la page d'accueil de l'interface utilisateur, en offrant une vue d'ensemble des fonctionnalités disponibles « figure IV.7 », notamment la configuration des paramètres de canal, le lancement des tests et l'accès à la phase d'entraînement du modèle. Elle permet de faciliter la prise en main de l'outil par les utilisateurs, par l'évaluation rapide et interactive des performances de l'agent AMC dans divers contextes.

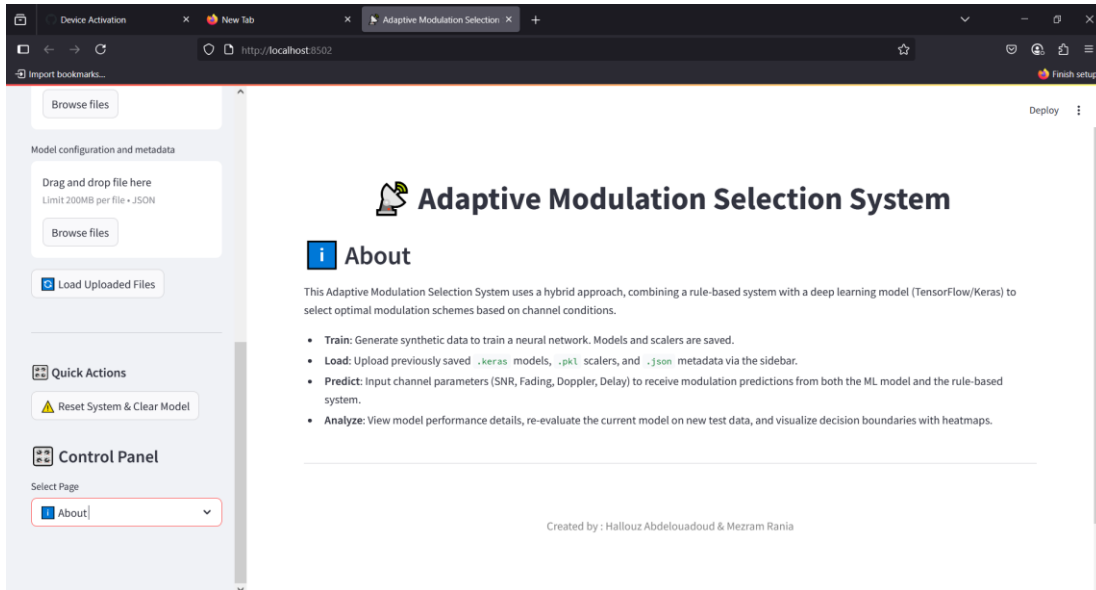


Figure IV. 7: les principales fonctionnalités disponibles sur l'interface de test.

4.5.1. La phase d'entraînement du modèle sur l'interface

La figure IV.8 illustre le module d'entraînement du modèle DNN intégré à l'interface, où le modèle ajuste ses poids interne pour apprendre la relation entre les paramètres du canal et la modulation optimale à appliquer. Plusieurs paramètres essentiels sont affichés notamment le nombre total d'échantillons, le nombres d'époques et la taille des batch. Le système affiche en temps réelle les indicateurs de performances (courbes d'apprentissage, précision, perte) permettant de suivre la progression de l'apprentissage.

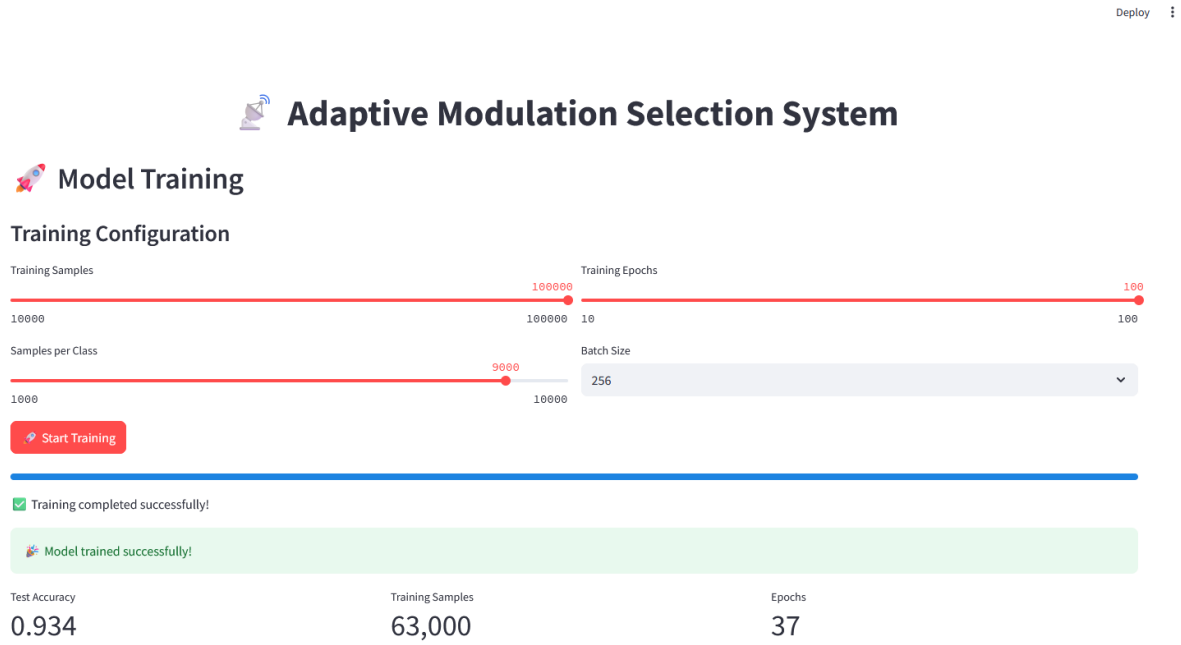


Figure IV. 8: Entraînement du modèle sous l'interface.

Cette intégration facilite le réentraînement du modèle sur de nouvelles données, ce qui est essentiel pour assurer sa robustesse et son adaptabilité dans des environnements aussi complexes.

4.5.2. Test 1 – conditions de canal faibles

Ce scénario présente un canal fortement dégradé, où les conditions sont défavorables :

- SNR : 0.5 dB (faible).
- Fading : 0.4 (fort évanouissement).
- Doppler : 170 Hz (élevé).
- Delay : 10.2 μ s (délai important).

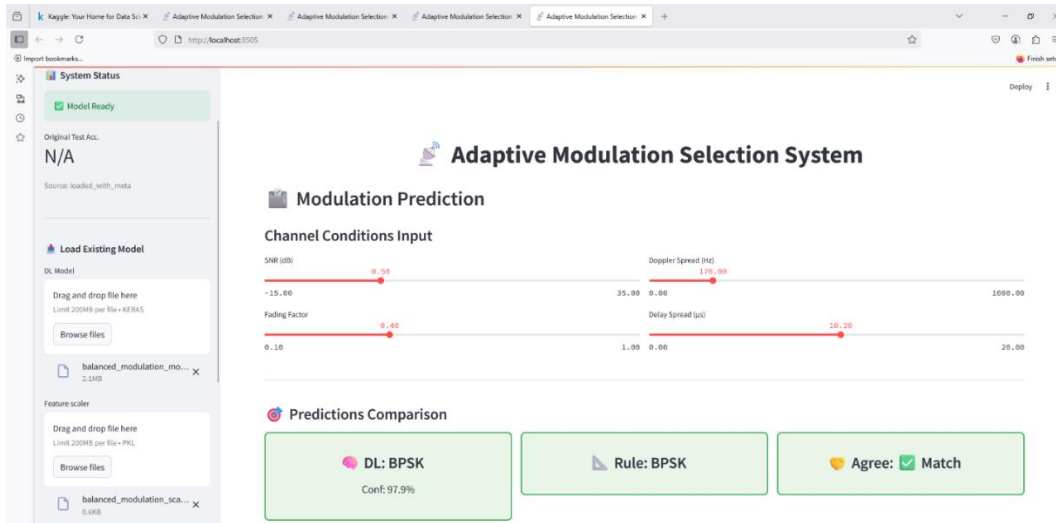


Figure IV. 9: Scénario avec faible qualité de canal.

La figure IV.9 montre que le modèle a correctement prédit une modulation robuste de type BPSK, adaptée à des conditions bruitées et instables. Cette stratégie vise à garantir la fiabilité de la transmission malgré la dégradation du canal. Le système adapte ainsi dynamiquement son choix en fonction de la situation, démontrant la capacité du modèle à préserver la qualité de service même dans des environnements complexes.

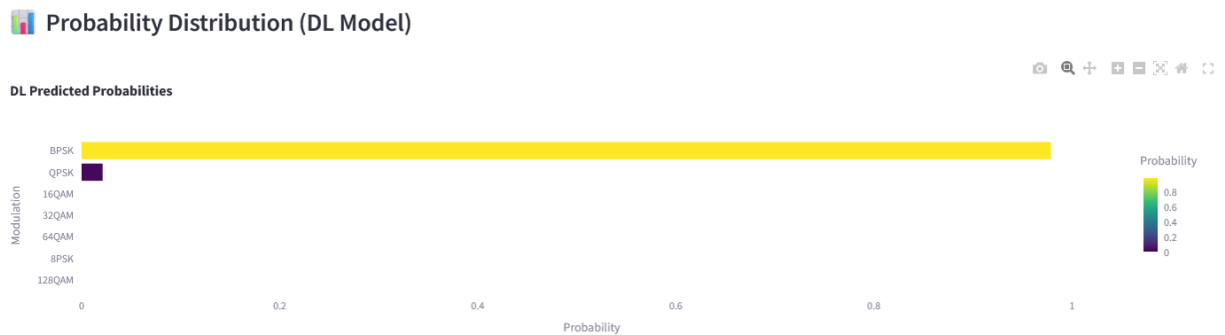


Figure IV. 10: Distribution des probabilités de modulations - canal faible.

On observe que le modèle attribue une probabilité très élevée à la modulation BPSK avec des valeurs proches de 1, tandis que les autres classes obtiennent des scores négligeables « figure IV.10 ».

4.5.3. Test 2 – conditions de canal intermédiaire

Dans le cas représenté dans « figure IV.11 », les paramètres du canal indiquent un état intermédiaire :

- SNR : 14.5 dB (modéré).
- Fading : 0.65 (évanouissement moyen).
- Doppler : 180 Hz (élevé).
- Delay : 1.9 μ s (bon).

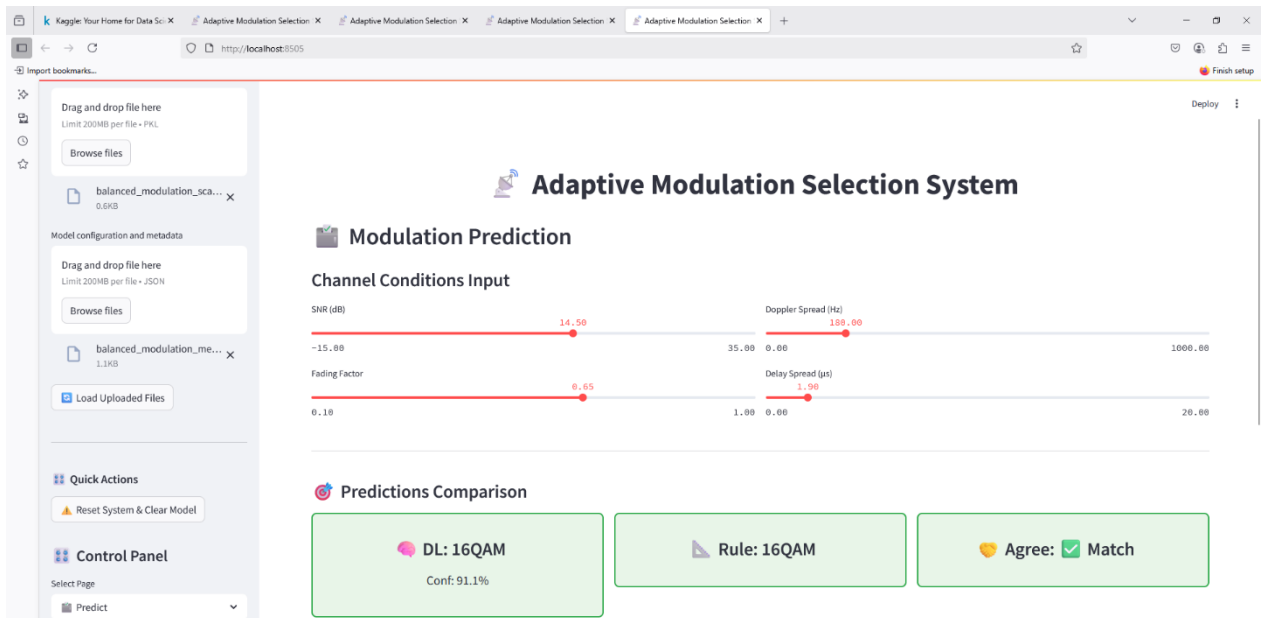


Figure IV. 11: Scénario de canal intermédiaire (conditions de canal modérées).

L'agent sélectionne des schémas de modulation de complexité moyenne (16-QAM). L'interface met en évidence l'adaptabilité du système, qui ajuste le compromis entre débit et robustesse en fonction de l'état de canal.

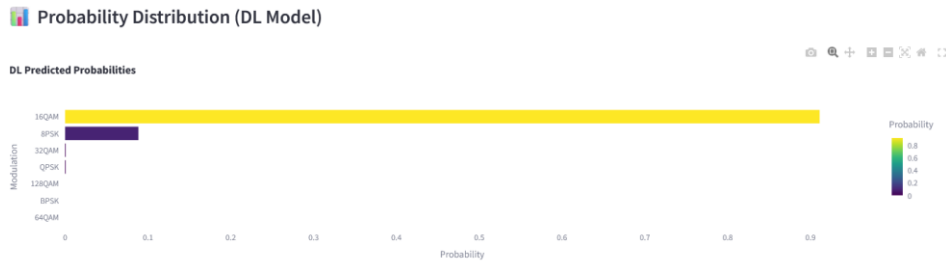


Figure IV. 12: Distribution des probabilités de modulations - canal intermédiaire.

Selon la figure IV.12, on observe une distribution centrée principalement sur la 16QAM, avec des faibles résidus pour la 8PSK, 32QAM et QPSK. Ce comportement reflète la nature ambiguë des conditions moyennes, où plusieurs configurations sont viables. La décision finale privilégiée toutefois la modulation la plus probable selon les paramètres appris.

4.5.4. Test 3 – conditions de canal élevées

L'agent AMC sélectionne un schéma de modulation d'ordre élevée (64QAM), permettant une transmission à haut débit. Ce résultat met en évidence la capacité du modèle à exploiter efficacement les ressources du canal lorsque la qualité est excellente. La prédiction est accompagnée d'un score de confiance élevé, ce qui témoigne de la certitude du modèle dans sa décision.

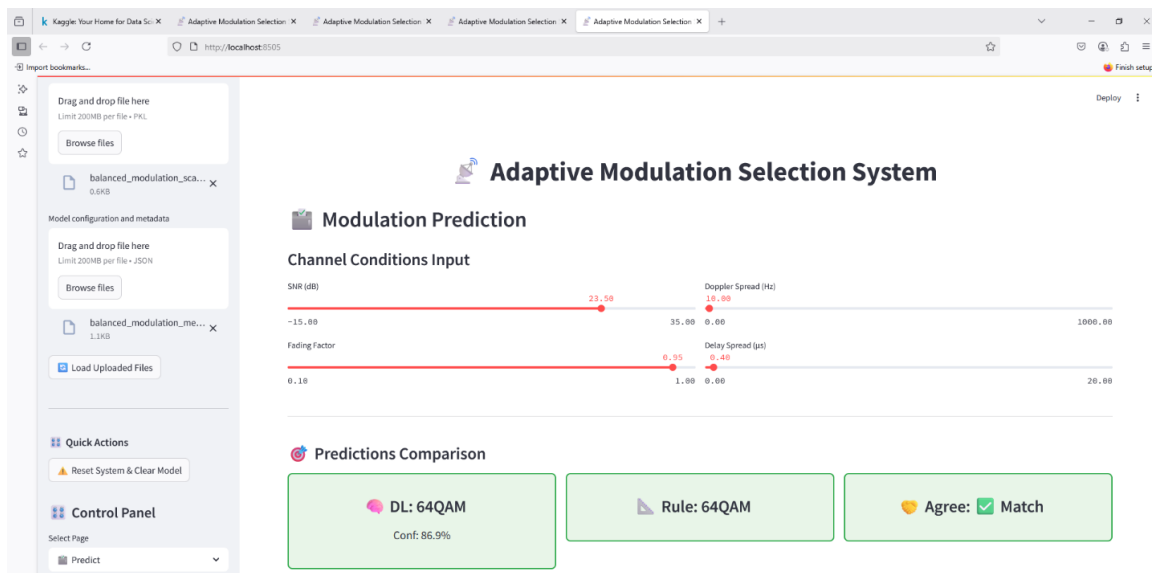


Figure IV. 13: Scénario avec canal de haute qualité.

Ce scenario présente des contions parfaite « figure IV.13 » :

- SNR : 23.5 dB (élevé).
- Fading : 0.95 (évanouissement faible).
- Doppler : 10 Hz (faible).
- Delay : 0.4 μ s (excellent).

Ce comportement est conforme aux attentes théoriques, car un canal de haute qualité permet d'exploiter pleinement les capacités du système sans risque accru d'erreurs de transmission.

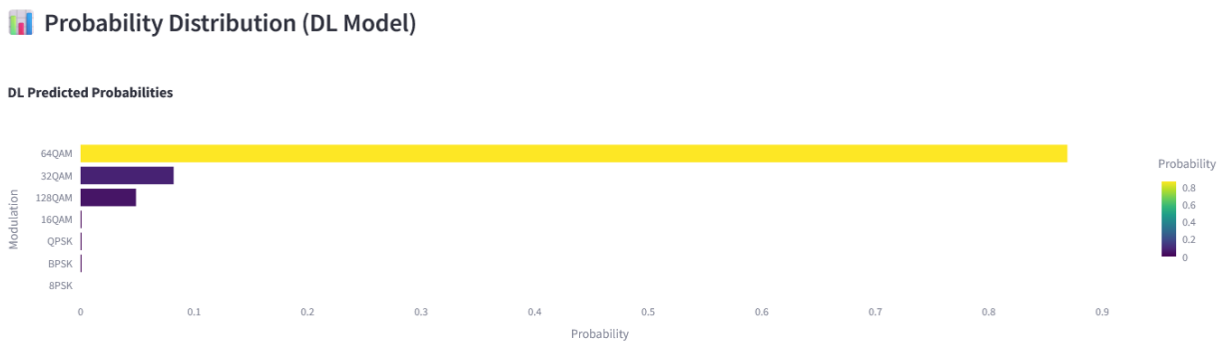


Figure IV. 14: Distribution des probabilités de modulations - canal favorable.

En présence de conditions optimales, le modèle affiche une forte probabilité pour les modulations de haut ordre « figure IV.14 » telle que 64QAM, et des probabilités quasi nulles pour les modulations robustes mais moins efficaces (BPSK, QPSK, ...). Cela montre la capacité du modèle à maximiser le débit lorsque la qualité du canal le permet.

4.5.5. Analyse du modèle avec de nouvelles données

La figure IV.15 illustre la phase du modèle DNN finale lorsqu'il traite des données inédites en dehors du jeu d'entraînement. Par ailleurs, la capacité du DNN à maintenir une précision élevée sur de nouvelles données, valide que le modèle a généralisé correctement les règles apprises durant l'entraînement.

Chapitre 04 : Simulation et discussion des résultats finaux

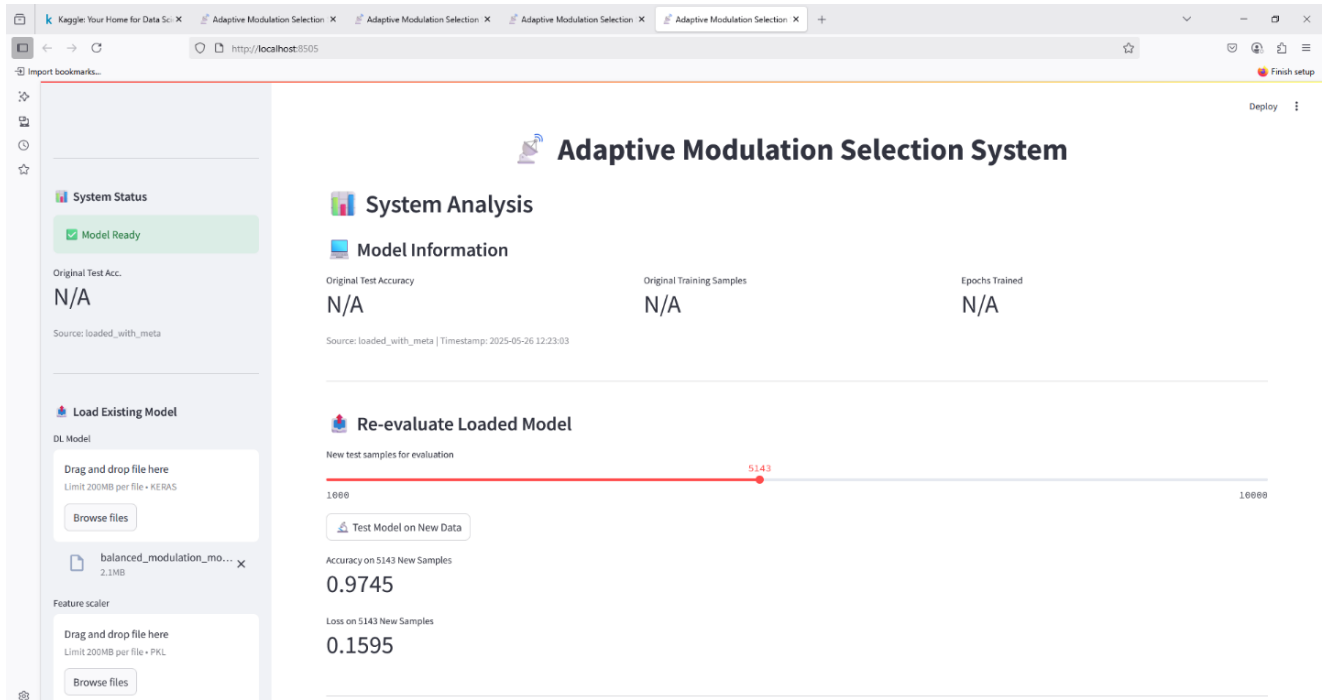


Figure IV. 15: Test du modèle sur des nouvelles données inédites.

Notre modèle ne sert pas à mémoriser des cas connus, mais s'adapte à des configurations variées, cette fonctionnalité est essentielle pour un système AMC, qui doit s'adapter en temps réel à la dynamique du canal sans nécessiter de réentraînement fréquent.

4.6. Comparaison entre les méthodes traditionnelles et l'approche proposée pour l'AMC

Le tableau 4-9 montre que notre solution proposée basée sur l'apprentissage profond surpasse les approches traditionnelles sur plusieurs aspects critiques pour les réseaux de nouvelles générations, en offrant une meilleure adaptabilité, efficacité spectrale, précision et qualité de service. Bien que plus complexes à mettre en œuvre, elle représente une solution évolutive et efficace pour répondre aux exigences des systèmes AMC intelligent en 5G/6G.

Critère de comparaison	Méthodes traditionnelles	Notre Approche DNN
Principe de fonctionnement	Basé sur des seuils fixes de SNR.	Apprentissage supervisé, extraction automatique des caractéristiques complexes.
Paramètres utilisés	SNR	SNR, Fading, Doppler, Delay.
Adaptabilité	Adaptation statique et limitée.	Adaptation dynamique en temps réel.
Precision	Sensible au bruits complexes.	Robustesse accrue.
Latence	Lent	Rapide.
Flexibilité	Limitée, difficile à adapter a de nouveau scenarios sans configuration manuelle.	Elevée (réentraînable, généralisable).
Efficacité spectrale	Sous-optimale.	Supérieure, compromis entre débit et robustesse
Qualité de service (QoS)	Dégradé et difficile à garantir dans les scenarios fortement dynamiques.	Améliorée, grâce à l'adaptation continue et la prédiction précise des schémas de modulation.
Complexité de mise en œuvre	Faible cout calculatoire	Moyenne à élevée dans l'entraînement.
Performance en conditions variables	Dégradation rapide	Robustesse et meilleure généralisation.
Nécessité d'une estimation précise du canal	Critique, dépend fortement de la qualité de l'estimation	Moins sensible.
Utilisation pratique en 5G/6G	Limitée à des scenario classique 3G/4G.	Recommandée pour des systèmes avancés.

Tableau IV. 10: Critères de comparaisons entre méthodes classiques et avancés pour l'AMC.

4.7. Conclusion

Ce dernier chapitre a présenté la mise en œuvre pratique de notre modèle finale pour un système AMC basé sur l'apprentissage profond. A travers une série de simulation, d'analyse et de tests via une interface interactive, nous avons démontré la capacité du modèle à s'adapter dynamiquement à des conditions de canal variées. La comparaison avec les méthodes traditionnelles met en évidence les gains en termes de plusieurs critères clés. Les résultats obtenus illustrent le potentiel du Deep Learning dans l'amélioration des performances des réseaux mobiles de nouvelle génération, tout en démontrant la pertinence de l'approche proposée.

Conclusion générale et perspectives

Ce mémoire a porté sur l'intégration de l'apprentissage profond dans la technique de modulation et codage adaptatif (AMC) pour les réseaux mobiles de nouvelle génération 5G/6G. Après une revue sur les méthodes traditionnelles et leurs limitations dans les environnements radio complexes, et à travers une démarche méthodologique rigoureuse, nous avons développé un agent intelligent basé sur un réseau de neurone profond (DNN), capable de prédire dynamiquement le schéma de modulation optimale en fonction des conditions du canal, optimisant ainsi le compromis entre débit, latence et qualité de service.

L'ensemble des expérimentations menées, la génération de données synthétiques, l'optimisation des hyperparamètres et l'analyse des performances via une interface interactive, ont permis de démontrer la robustesse, la précision et l'efficacité de notre solution proposée. La comparaison avec les méthodes classiques valide la pertinence de notre approche en termes d'efficacité spectrale, fiabilité et d'adaptabilité.

Il est important de souligner que ce travail représente une contribution originale, en proposant une approche intelligente et automatisée de l'AMC par les DNN, qui n'a pas été réalisée auparavant dans ce contexte précis. Cette nouveauté confère à notre projet une valeur ajoutée significative, en explorant des pistes innovantes pour l'application de l'apprentissage profond à l'AMC dans les systèmes 5G/6G.

Plusieurs axes d'amélioration et de prolongement peuvent être envisagés pour enrichir ce projet :

- L'intégration du codage d'erreur au même niveau que la modulation pour une solution AMC complète.
- Extension à des scénarios plus complexes, telle que la mobilité des utilisateurs et les interférences multi-utilisateurs.
- Exploration des architectures plus avancées pour capturer des dépendances à long terme dans les signaux.
- L'intégration de l'Apprentissage par renforcement pour la prise de décision en interagissant directement avec l'environnement.

Conclusion générale et perspectives

- Utilisation des données réelles pour un déploiement pratique par l'intégration du modèle sur des plateformes réelles (environnements de test 5G/6G réel), pour évaluer ses performances en conditions opérationnelles.

En conclusion, ce travail illustre le potentiel de l'IA dans les télécommunications. Les perspectives envisagées soulignent la nécessité de poursuivre les recherches, en ouvrant la voie à des systèmes de communications sans fil plus intelligents, adaptatifs et performants, répondant aux défis croissants des réseaux mobiles futurs.

Bibliographie

- [1] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [2] E. Dahlman, S. Parkvall, et J. Skold, *5G NR: The Next Generation Wireless Access Technology*. Elsevier Science, 2020.
- [3] J. G. Andrews et al. « What Will 5G Be? », *IEEE J. Sel. Areas Commun.*, vol. 32, n° 6, p. 1065-1082, juin 2014.
- [4] Z. Zhang et al. "6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28-41, Sept. 2019.
- [5] « Bharat 6G Alliance - PWOlyIAS ». [En ligne]. Disponible sur: <https://pwoonlyias.com/current-affairs/bharat-6g-alliance/>. Consulté le: 25 février 2025
- [6] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, et M. Zorzi, « Toward 6G Networks: Use Cases and Technologies », *IEEE Commun. Mag.*, vol. 58, n° 3, p. 55-61, mars 2020.
- [7] P. Urien, *Cours M2 : Évolution des technologies réseaux et sécurité des réseaux sans fil*, Telecom ParisTech, 2009. [En ligne]. Disponible : https://perso.telecom-paristech.fr/urien/cours_m2_2009.pdf. Consulté le: 1 février 2025.
- [8] "*Communication sans fil*", cours en ligne, Embedded Computing, HEIA-FR, 2023. [En ligne]. Disponible : <https://embedded-computing.pages.forge.hefr.ch/2023-il-embedded-computing/Lecture/Communication/wireless/#conclusion>. Consulté le: 1 février 2025.
- [9] Z. M. W. Harous, « Reconnaissance automatique de type de modulation dans les réseaux MIMO coopératifs », 2019.
- [10] TechTarget, "*Qu'est-ce que la modulation ?*", *SearchNetworking*, 2025. Disponible: <https://www.techtarget.com/searchnetworking/definition/modulation>. Consulté le: 20 février 2025.
- [11] P. Malladi, "Signal Quality Characterization of RF Communication Systems", *Broadband Library*, 2023. Disponible sur : <https://broadbandlibrary.com/signal-quality-characterization-of-rf-communication-systems/>. Consulté le: 23 février 2025.

Bibliographie

- [12] L. Mostari, *Cours & Travaux Dirigés M2 : Codage de Canal*, Université Hasiba Ben Bouali-Chlef, 2016/2017. Disponible: <https://www.univ-chlef.dz/ft/wp-content/uploads/2020/04/Latifa-MOSTARI-Cours-et-TD-M2-Codage-de-Canal.pdf>. Consulté le: 23 février 2025.
- [13] G. Gontrand, *Systèmes de communication – Première partie*, ISTE Group, 2020. Disponible: https://www.istegroup.com/wpcontent/uploads/2020/08/689_Syst%C3%A8mes-de-communication_Gontrand_Premi%C3%A8re-partie.pdf. Consulté le : 26 juin 2025.
- [14] N. Heniat et I. Ghenadra, « Les codes de Reed Solomon : étude et simulation », Mémoire de fin d'études, Département de Génie Electrique, Université Echahid Hamma Lakhdar d'El Oued, 2016. Disponible : <https://dspace.univ-eloued.dz/server/api/core/bitstreams/e8ee8b2d-f146-4c3f-92a7-c44761fca1f5/content>.
- [15] G. Baum, "*Reed-Solomon Codes*," Carnegie Mellon University, 2021. [En ligne]. Disponible: https://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html.
- [16] G. Blleloch, "Reed-Solomon Codes," *Algorithms in the Real World*, Carnegie Mellon University, 2001. Disponible sur : https://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html.
- [17] A. D. Kora, "*Estimation de canal des systèmes multi-antennes*", Éditions Universitaires Européennes, 2020. Disponible: <https://www.abebbooks.co.uk/Estimation-canal-syst%C3%A8mes-multi-antennes-Ahmed-Dooguy/30730106392/bd>. Consulté le: 10 février 2025.
- [18] H. Yu et C. Yang, "*An Improved DFT-based Channel Estimation for Mobile Communication Systems* ", 2015, arXiv:1504.07510.
- [19] C. R. Berger, Z. Wang, J. Huang, et S. Zhou, "*Application of compressive sensing to sparse channel estimation* ", IEEE Commun. Mag., vol. 48, n° 11, p. 164-174, nov. 2010.
- [20] J. Gao, C. Zhong, G. Y. Li, et Z. Zhang, "*Deep Learning based Channel Estimation for Massive MIMO with Hybrid Transceivers* ", IEEE Trans. Wirel. Commun., vol. 21, n° 7, p. 5162-5174, juill. 2022.
- [21] M.-O. Berger, "*Méthodes d'estimation*", Université de Lorraine, 2020. Disponible: https://members.loria.fr/MO Berger/Enseignement/AVR/Cours/estimation_avr_2020.pdf.

Bibliographie

- [22] VIRTU-DESK, "*Machine Learning*", Virtu-Desk – Définitions Intelligence Artificielle, 2025. Disponible sur: <https://www.virtu-desk.fr/pages/les-definitions/l-intelligence-artificielle/machine-learning.html>.
- [23] I. Goodfellow, Y. Bengio, et A. Courville, "*Deep Learning*". MIT Press, 2016.
- [24] J. Gao, C. Zhong, G. Y. Li, et Z. Zhang, "*Deep Learning based Channel Estimation for Massive MIMO with Hybrid Transceivers*", IEEE Trans. Wirel. Commun., vol. 21, n° 7, p. 5162-5174, juill. 2022, doi: 10.1109/TWC.2021.3137354.
- [25] T. O'Shea et J. Hoydis, "*An Introduction to Deep Learning for the Physical Layer*", IEEE Trans. Cogn. Commun. Netw, vol3, n°4, p563-575, déc2017.
- [26] H. He, C.-K. Wen, S. Jin, et G. Y. Li, "*Deep Learning-based Channel Estimation for Beamspace mmWave Massive MIMO Systems*", janv 2019, arXiv: arXiv:1802.01290.
- [27] « J. Lee et al, "*Deep Learning-Based Adaptive Modulation and Coding in 5G, IEEE Transactions on Wireless Communications*", vol. 19, no. 5, pp. 3485-3495, May 2021. - SciSpace Literature Review ». Consulté le: 1 février 2025. Disponible sur: <https://typeset.io/search>
- [28] G. Zhang, C. Ma, K. Chen, Y. Li, H. Li, et C. Han, "*Multiple-symbol noncoherent learning detection of coded QAM signals in IEEE 802.15.3 Wireless Multi-media Networks*", Phys Commun, vol. 55, n° C, déc 2022.
- [29] « Qu'est-ce qu'un réseau de neurones en Deep Learning ». Consulté le: 1 février 2025. [En ligne]. Disponible sur: <https://www.jedha.co/formation-ia/reseau-neurones-deep-learning>
- [30] « *Comment l'intelligence artificielle permettra-t-elle d'anticiper le phénomène El Niño ? - Surf-Report* », Surf-report.com. Consulté le: 23 février 2025. [En ligne]. Disponible sur: <https://www.surf-report.com/news/alerte-houle/el-nino-ia-intelligence-artificielle-mavericks-jaws-924197694.html>
- [31] « Apprentissage profond : la mécanique de la magie », ISO. Consulté le: 1 février 2025. [En ligne]. Disponible sur: <https://www.iso.org/fr/intelligence-artificielle/apprentissage-profond-deep-learning>
- [32] Jedha, "*Réseau de neurones & Deep Learning*", Jedha – Formation IA, Disponible sur : <https://www.jedha.co/formation-ia/reseau-neurones-deep-learning>

Bibliographie

- [33] C. Lv et Z. Luo, " *Deep Learning for Channel Estimation in Physical Layer Wireless Communications: Fundamental, Methods, and Challenges* ", Electronics, vol. 12, n° 24, Art. n° 24, janv. 2023.
- [34] O. Stenhammar, G. Fodor, et C. Fischione, " *A Comparison of Neural Networks for Wireless Channel Prediction* ", août 2023, arXiv: arXiv:2308.14020.
- [35] S. Hanna, C. Dick, et D. Cabric, " *Combining Deep Learning and Linear Processing for Modulation Classification and Symbol Decoding* ", sep 2020, arXiv: arXiv:2006.00729.
- [36] L. Djemli, " *Étude et simulation de FS-AMTuC pour le système de transmission OFDM* ", Mémoire de Master, Département d'Électronique, Université Saad Dahlab de Blida, Blida, Algérie, 2015. Disponible: <https://di.univ-blida.dz/xmlui/bitstream/handle/123456789/1945/FS-AMTuC-OFDM.pdf?sequence=1&isAllowed=y>.
- [37] A. Attatra et B. Cherifi, " *Étude et analyse de l'estimation des canaux multi-trajets dans un système OFDM* ", mémoire de Master, Département Électronique et Télécommunications, Université 8 Mai 1945 Guelma, juin 2023.
- [38] DeepSig Inc, " *DeepSig Dataset: RadioML 2018.01A* " Kaggle, 2018. Disponible : <https://www.kaggle.com/datasets/pinxau1000/radioml2018>.
- [39] G. Paaß, « *Deep Learning: How do deep neural networks work?* », Lamarr Institute, 21 avril 2021. [En ligne]. Disponible sur : <https://lamarr-institute.org/blog/deep-neural-networks/>

