République algérienne démocratique et populaire
وزارة التـعليـم العالـي والبحـث العلمـــي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département d'électronique et télécommunication

Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en
Domaine : Science et de Technologie
Filière : Télécommunication
Spécialité : Réseaux et Télécommunication
Thème

Détection d'activité spectrale à l'aide de RTL-SDR et de l'IA

Présenté Par :
1) Melle Belhocine Chaimaa
2) Melle Benbachir Nada Chaimaa

Devant le jury composé de :

Dr BENOSMAN Morad	MCA	UAT.B.B (Ain Temouchent)	Président
Dr HAMLILI Heyem	MCB	UAT.B.B (Ain Temouchent)	Examinatrice
Dr YAGOUB Reda	MCB	UAT.B.B (Ain Temouchent)	Encadrant

Année Universitaire 2025/2026

REMERCIEMENTS

Au terme de ce travail de recherche, nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à toutes les personnes ayant contribué, de près ou de loin, à la réalisation de ce mémoire de fin d'études.

Nous remercions avant tout Allah, le Tout-Puissant, de nous avoir accordé la santé, la force, la patience et la persévérance nécessaires pour mener à bien ce travail.

Nous adressons notre plus profonde reconnaissance à notre encadrant, Mr. Yagoub Reda, pour la qualité de son accompagnement, sa disponibilité constante, ses orientations pertinentes ainsi que pour ses précieux conseils scientifiques. Son encadrement rigoureux, son soutien permanent et la confiance qu'il nous a accordée ont constitué un apport essentiel à l'élaboration et à l'aboutissement de cette étude.

Nous exprimons également nos sincères remerciements aux honorables membres du jury pour l'intérêt qu'ils ont porté à ce travail en acceptant de l'évaluer. Nous leur sommes reconnaissants pour leurs remarques constructives, leurs suggestions pertinentes ainsi que pour le temps précieux qu'ils ont consacré à l'examen de ce mémoire.

Nos remerciements s'adressent aussi à l'ensemble des enseignants du département, dont le professionnalisme, le savoir et le dévouement ont largement contribué à notre formation universitaire et à l'acquisition des connaissances nécessaires à la réalisation de ce travail.

Nous tenons à exprimer notre profonde gratitude à nos parents et à nos proches pour leur soutien moral indéfectible, leurs encouragements constants, leur patience et les sacrifices consentis tout au long de notre parcours académique. Leur confiance et leur présence ont été pour nous une source permanente de motivation et de détermination.

Enfin, nous adressons nos remerciements les plus sincères à toutes les personnes ayant participé, directement ou indirectement, à la réalisation de ce mémoire.

DEDICACE

Je dédie ce modeste travail :

À celui qui a toujours été mon exemple de courage, de sacrifice et de dignité, à ***mon cher père***, qui n'a jamais cessé de me soutenir et de croire en moi. Aucun mot ne saurait exprimer toute ma reconnaissance pour les efforts qu'il a consentis afin de me voir réussir. Que Dieu le protège et lui accorde santé, bonheur et longue vie.

À la lumière de ma vie,
à ma très chère mère, symbole d'amour, de tendresse et de patience. Merci pour tes prières, ton affection infinie et ton soutien inconditionnel qui m'ont donné la force d'avancer malgré les difficultés. Que Dieu te garde auprès de nous et t'accorde une vie remplie de joie et de sérénité.

À mes chers frères :
Yacine, Younes et Raziko,
pour leur présence, leur affection, leurs encouragements et tous les moments de bonheur partagés ensemble. Puisse notre lien rester éternel et rempli d'amour.

À mon honorable encadreur,
Monsieur Reda Yagoub,
pour sa patience, sa disponibilité, ses précieux conseils et son accompagnement tout au long de ce travail. Son sérieux, son professionnalisme et son soutien scientifique ont été pour moi une véritable source d'inspiration et de motivation. Je lui exprime ici ma profonde gratitude, mon grand respect et ma sincère reconnaissance.

À ma chère binôme,
Benbachir Nada Chaimaa,
avec qui j'ai partagé les efforts, les moments de doute, de fatigue mais aussi de réussite et de joie. Cette expérience restera à jamais gravée dans nos mémoires comme une belle aventure académique et humaine.

À mes chères amies,
qui ont toujours été présents pour me soutenir, m'encourager et illuminer mon parcours par leur gentillesse et leur sincérité.

Enfin, à toute ma famille ainsi qu'à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire.

Je vous dédie ce travail avec toute mon affection, mon respect et ma gratitude,
en espérant qu'il soit le reflet de tous les efforts, du soutien et de l'amour que vous m'avez offerts.

Chaimaa

DEDICACE

Tout d'abord, je remercie Dieu le tout puissant de m'avoir accordé la force, la patience et l'inspiration nécessaires à la réalisation de ce projet.

Je dédie ce modeste travail, fruit de plusieurs années d'efforts, de persévérance et de sacrifices, à toutes les personnes qui ont contribué, de près ou de loin pour ce parcours académique et personnel.

À moi-même,

À celle qui n'a jamais abandonné, qui a cru en ses rêves malgré les nuits de fatigue et les moments de doute ; Je dédie ce succès à ma propre persévérance, en hommage à chaque effort fourni pour atteindre ce but.

À ma très chère mère Soraya

Source infinie d'amour, de tendresse et de soutien, merci pour tes prières, tes sacrifices et ta présence constante tout au long de mon parcours. Ton affection, tes conseils et ta confiance ont toujours été ma plus grande force et ma source permanente de motivation. Ce modeste travail est le fruit de tous les efforts et sacrifices que t'as consentis pour ma réussite. Puisse Dieu ;le tout puissant t'accorder santé, bonheur et longue vie.

À mon cher père Mohamed

Modèle de force, de sagesse et de bienveillance, merci pour ton soutien indéfectible, tes précieux conseils, ta confiance et tous les sacrifices consentis tout au long de mon parcours. Ta présence, ton affection m'ont toujours été une source de motivation et de courage. Aucune parole ne saurait exprimer pleinement mon profond respect, ma gratitude infinie et tout l'amour que je te porte ; cher père. Puisse Dieu t'accorder santé, bonheur et longue vie.

À mes chers frères Issam, Anes et Fouad

Je vous exprime ma profonde gratitude pour votre soutien constant, votre amour et votre présence tout au long de mon parcours. Vos encouragements et votre solidarité ont été une véritable source de force et de motivation. Que Dieu vous accorde bonheur, santé et prospérité. Vos présences ont toujours été précieuses et essentielles dans ma vie.

À toute ma famille

Pour leur soutien, leur affection et leurs prières qui ont toujours été une source de motivation et de réconfort.

À mon encadreur, Mr Reda Yagoub

Je tiens à vous remercier vivement pour votre soutient et accompagnement et les si précieux conseils que vous m'avez faits tout au long de mon labeur ;ainsi que votre disponibilité morale et intellectuelle tout au long de la réalisation de ce mémoire.

À ma binôme Belhocine Chaimaa, avec qui j'ai partagé les efforts et les défis de cette aventure académique, en lui souhaitant beaucoup de réussite.

Enfin, j'adresse ma sincère gratitude à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Benbachir Nada Chaimaa

RESUME

Les systèmes modernes de traitement de l'information ont connu une évolution importante grâce à la transformation numérique et à l'intégration des technologies intelligentes dans l'analyse des données. Les signaux physiques sont désormais convertis en données numériques afin d'être traités à l'aide d'algorithmes mathématiques avancés permettant d'en extraire des informations pertinentes. L'intelligence artificielle joue également un rôle central dans l'amélioration des performances de classification, d'analyse et de prise de décision automatique. Ces technologies sont largement utilisées dans des domaines variés tels que les télécommunications, la surveillance et l'analyse de données complexes, offrant ainsi une meilleure compréhension des environnements dynamiques et une optimisation des performances des systèmes

Dans le cadre de ce travail, nous avons développé une approche expérimentale basée sur l'utilisation du récepteur RTL-SDR pour l'acquisition de signaux radiofréquences réels, suivie d'un prétraitement et d'une modélisation par des algorithmes d'apprentissage supervisé tels que Random Forest, K-Nearest Neighbors (KNN) et XGBoost. Les résultats expérimentaux ont montré une capacité de détection d'activité spectrale robuste, validée par des matrices de confusion et une comparaison des performances entre les modèles.

Par ailleurs, une application web a été conçue afin de rendre la détection d'activité spectrale accessible et interactive. Elle intègre une interface de capture en temps réel des signaux Radio, une fonctionnalité d'importation et d'analyse de fichiers CSV, ainsi qu'un tableau de bord permettant la visualisation des résultats d'analyse. Cette plateforme constitue une contribution pratique, offrant un outil flexible pour l'analyse et la détection spectrale dans des environnements radio complexes.

Mots-clés : RTL-SDR, apprentissage supervisé, Random Forest, KNN, XGBoost, détection d'activité spectrale, radio cognitive, application web.

ABSTRACT

Modern information processing systems have undergone significant evolution driven by digital transformation and the integration of intelligent technologies into data analysis. Physical signals are now converted into digital data to be processed using advanced mathematical algorithms, enabling the extraction of relevant information. Artificial intelligence plays a central role in improving classification, analysis, and automated decision-making performance. These technologies are widely applied across diverse fields such as telecommunications, surveillance, and complex data analysis, offering a deeper understanding of dynamic environments and enhanced system performance optimization.

In this work, we developed an experimental approach based on the use of the RTL-SDR receiver for the acquisition of real radio frequency signals, followed by preprocessing and modeling using supervised learning algorithms including Random Forest, K-Nearest Neighbors (KNN), and XGBoost. The experimental results demonstrated a robust spectral activity detection capability, validated through confusion matrices and a comparative performance analysis of the models.

Furthermore, a web application was designed to make spectral activity detection accessible and interactive. It incorporates a real-time radio signal capture interface, a CSV file import and analysis feature, and a dashboard for visualizing analysis results. This platform represents a practical contribution, providing a flexible tool for spectral analysis and detection in complex radio environments.

Keywords: RTL-SDR, supervised learning, Random Forest, KNN, XGBoost, spectral activity detection, cognitive radio, web application.

ملخص

شهدت أنظمة معالجة المعلومات الحديثة تطوراً ملحوظاً بفضل التحول الرقمي ودمج التقنيات الذكية في تحليل البيانات. باتت الإشارات الفيزيائية تُحوَّل إلى بيانات رقمية لمعالجتها باستخدام خوارزميات رياضية متقدمة، تتيح استخلاص المعلومات ذات الصلة منها. كما تؤدي الذكاء الاصطناعي دوراً محورياً في تحسين أداء التصنيف والتحليل واتخاذ القرار الآلي. وتُوظَّف هذه التقنيات على نطاق واسع في مجالات متعددة كالاتصالات والمراقبة وتحليل البيانات المعقدة، مما يُسهم في فهم أعمق للبيئات الديناميكية وتحسين أداء الأنظمة.

في إطار هذا العمل، طوّرنا مقاربةً تجريبيةً تستند إلى استخدام جهاز الاستقبال RTL-SDR لتسجيل إشارات ترددية راديوية حقيقية، يعقبها مرحلة معالجة مسبقة ونمذجة بواسطة خوارزميات التعلم الآلي المُشرَف عليه، لا سيما (Random Forest) وخوارزمية (KNN) وXGBoost وقد أظهرت النتائج التجريبية قدرةً قوية على الكشف عن النشاط الطيفي، تم التحقق منها من خلال مصفوفات الارتباك ومقارنة أداء النماذج.

علاوةً على ذلك، جرى تصميم تطبيق ويب بهدف جعل الكشف عن النشاط الطيفي في متناول المستخدمين بصورة تفاعلية وسهلة الاستخدام. يتضمن هذا التطبيق واجهةً لالتقاط الإشارات في الوقت الفعلي عبر جهازRTL-SDR ، وإمكانية استيراد ملفات CSV وتحليلها، فضلاً عن لوحة تحكم لعرض نتائج التحليل بصرياً. وتُمثّل هذه المنصة إسهاماً تطبيقياً ملموساً، إذ توفر أداةً مرنة لتحليل الطيف الترددي والكشف عن نشاطه في البيئات الراديوية المعقدة.

الكلمات المفتاحية : RTL-SDR ، التعلم المُشرَف عليه، Random Forest ، KNN، XGBoost، الكشف عن النشاط الطيفي، الراديو الإدراكي، تطبيق ويب.

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

ACRONYMES ET ABREVIATIONS

ADC : Analog to Digital Converter

AI : Artificial Intelligence

BTS : Base Transceiver Station

DVB-T : Digital Video Broadcasting — Terrestrial

FFT : Fast Fourier Transform

FM : Frequency Modulation

GPS : Global Positioning System

GSM : Global System for Mobile Communications

IA : Intelligence Artificielle

IQ : In-phase / Quadrature

ISM : Industrial, Scientific and Medical band

KNN : K-Nearest Neighbors

LNA : Low Noise Amplifier

LO : Local Oscillator

ML : Machine Learning

NLP : Natural Language Processing

QoS : Quality of Service

RF : Random Forest

RTL-SDR : Récepteur SDR basé sur puce RTL2832U

SDR : Software Defined Radio

SNR : Signal-to-Noise Ratio

XGBoost : Extreme Gradient Boosting

INTRODUCTION GENERALE

Les systèmes de communication sans fil occupent aujourd'hui une place centrale dans les technologies modernes, en raison de leur rôle essentiel dans l'échange d'informations à grande échelle. L'augmentation continue du volume de données, la diversité des services numériques ainsi que la complexité des environnements radiofréquences ont mis en évidence les limites des architectures classiques de radio. Ces limitations concernent principalement le manque de flexibilité, la faible capacité de reconfiguration et les difficultés d'adaptation aux environnements dynamiques et bruités.

Dans ce contexte, l'émergence du paradigme de la Software Defined Radio (SDR) représente une avancée significative dans le domaine des communications radio. Cette approche repose sur la virtualisation des fonctions traditionnellement implémentées en matériel, en les remplaçant par des traitements logiciels. Elle permet ainsi une grande souplesse dans la conception, l'expérimentation et l'évolution des systèmes de communication. Parmi les solutions accessibles et largement utilisées dans la recherche et l'enseignement, la plateforme RTL-SDR permet l'acquisition de signaux radiofréquences réels et leur conversion en données numériques exploitables pour l'analyse et le traitement.

Les signaux captés dans les systèmes SDR sont généralement représentés sous forme complexe à travers les composantes In-phase et Quadrature (IQ), ce qui permet de conserver à la fois l'amplitude et la phase du signal. Cette représentation constitue une base fondamentale pour le traitement numérique du signal. L'analyse fréquentielle est ensuite réalisée à l'aide de techniques mathématiques telles que la Fast Fourier Transform (FFT), qui permet de transformer le signal du domaine temporel vers le domaine fréquentiel afin d'en extraire les caractéristiques spectrales essentielles. Ces caractéristiques sont indispensables pour comprendre la structure et le comportement des signaux dans des environnements radio variés.

Dans ce cadre, l'objectif principal de ce mémoire est de développer une approche intelligente basée sur la technologie RTL-SDR, les techniques de traitement numérique du signal et les méthodes d'apprentissage automatique, afin d'améliorer la détection, l'analyse des activités spectrales dans des environnements radiofréquences complexes et dynamiques.

Par ailleurs, les progrès réalisés dans le domaine de l'intelligence artificielle et du Machine Learning ont permis d'introduire de nouvelles méthodes d'analyse automatique des signaux. Les algorithmes d'apprentissage supervisé tels que Random Forest, XGBoost et K-Nearest Neighbors (KNN) sont particulièrement utilisés pour exploiter les caractéristiques extraites et réaliser des tâches de classification avec une grande précision. Ces méthodes permettent également d'améliorer la capacité des systèmes à s'adapter à des conditions variables et à des signaux fortement bruités.

Chapitre 1 : Machine Learning : Concepts, Algorithmes et Évaluation des Performances

1. Introduction

Le Machine Learning constitue aujourd'hui un domaine central de l'intelligence artificielle, permettant aux systèmes informatiques d'apprendre automatiquement à partir des données sans être explicitement programmés pour chaque tâche. Grâce à l'augmentation massive des données et à la puissance de calcul disponible, les techniques d'apprentissage automatique sont devenues essentielles dans de nombreux domaines tels que les télécommunications, la santé, la finance et l'analyse du spectre radio.

Le principe fondamental du Machine Learning repose sur la capacité des algorithmes à identifier des relations, des modèles ou des structures cachées dans les données afin de réaliser des prédictions ou des décisions intelligentes. Parmi les différentes approches existantes, on distingue notamment l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement, chacun répondant à des problématiques spécifiques.

Dans ce premier chapitre, plusieurs algorithmes d'apprentissage supervisé sont étudiés, notamment K-Nearest Neighbors (KNN), Random Forest et XGBoost, qui sont largement utilisés pour les tâches de classification et de prédiction en raison de leur efficacité et de leur robustesse.

Cependant, l'efficacité d'un modèle d'apprentissage automatique ne dépend pas uniquement de l'algorithme utilisé, mais également de la manière dont ses performances sont évaluées. Pour cette raison, différentes métriques d'évaluation sont employées afin de mesurer la qualité des prédictions, telles que Accuracy, Précision, Recall et F1-score.

2. L'Intelligence Artificielle (IA)

L'intelligence artificielle (IA) est l'une des évolutions scientifiques les plus importantes du monde numérique. Elle consiste à développer des systèmes informatiques capables d'imiter certaines capacités de l'intelligence humaine, telles que l'apprentissage, le raisonnement et la prise de décision. L'IA vise à doter les machines de capacités avancées leur permettant de reproduire certaines compétences scientifiques et cognitives, afin d'assister l'être humain et d'améliorer l'efficacité dans plusieurs domaines. Grâce à l'utilisation d'algorithmes, de grandes quantités de données et de puissantes capacités de calcul, l'intelligence artificielle contribue aujourd'hui au développement et à l'innovation dans de nombreux secteurs et disciplines scientifiques. [1]

L'intelligence artificielle (IA) regroupe plusieurs sous-domaines qui permettent aux machines d'imiter certaines capacités humaines comme l'apprentissage, la perception, la compréhension du langage et la prise de décision. Ces domaines travaillent souvent ensemble pour créer des systèmes intelligents capables d'analyser des données et d'agir de manière autonome. [2]

2.1 Les domaines d'application

Le domaine de l'intelligence artificielle s'étend à de nombreux secteurs, offrant des solutions capables d'analyser, comprendre et prédire des données complexes [2].

2.1.1 Analyse prédictive

L'analyse prédictive utilise des techniques statistiques et des algorithmes d'apprentissage automatique pour analyser les données historiques et prédire les événements futurs. Elle est utilisée dans plusieurs domaines comme la finance, le marketing ou la maintenance prédictive.

2.1.2 Systèmes experts

Les systèmes experts sont des programmes d'intelligence artificielle qui reproduisent le raisonnement d'un expert humain dans un domaine spécifique. Ils utilisent une base de connaissances et des règles logiques pour résoudre des problèmes complexes et aider à la prise de décision.

2.1.3 Traitement automatique du langage naturel (NLP)

Le traitement automatique du langage naturel (Natural Langage Procession – NLP) permet aux machines de comprendre, analyser et générer le langage humain. Ses principales applications sont :

- ✓ La traduction automatique
- ✓ Les assistants virtuels et chatbots
- ✓ L'analyse des sentiments
- ✓ La classification de texte
- ✓ L'extraction d'informations

2.1.4 Reconnaissance vocale

La reconnaissance vocale permet de transformer la parole humaine en texte. Cette technologie est utilisée dans les assistants vocaux, les systèmes de commande vocale et les applications de transcription automatique.

2.1.5 Vision par ordinateur

La vision par ordinateur est un domaine de l'IA qui permet aux machines d'interpréter et d'analyser des images ou des vidéos. Elle permet notamment :

- ✓ La reconnaissance d'images
- ✓ La classification d'images
- ✓ La détection et le suivi d'objets

Cette technologie est utilisée dans la surveillance, les véhicules autonomes et l'imagerie médicale.

2.1.6 Intelligence robotique

L'intelligence robotique combine l'intelligence artificielle et la robotique pour concevoir des robots capables de percevoir leur environnement, de prendre des décisions et d'effectuer des actions de manière autonome. Les robots intelligents sont utilisés dans l'industrie, la médecine, l'exploration… etc.

3. Machine Learning (Apprentissage automatique)

Le Machine Learning est une branche de l'intelligence artificielle basée sur l'utilisation des données pour permettre aux machines d'apprendre automatiquement sans être programmées explicitement. Les modèles apprennent à partir des données afin de réaliser des prédictions ou de prendre des décisions.

Il existe trois principaux types d'apprentissage :

- Apprentissage supervisé
- Apprentissage non supervisé
- Apprentissage par renforcement

Le machine Learning est largement utilisé dans l'analyse de données, la prédiction et la classification. [2]

3.1 Le fonctionnement du machine Learning

Le Machine Learning peut être mis en œuvre selon deux approches distinctes : l'application d'un modèle préalablement entraîné à de nouvelles données, ou l'entraînement d'un nouveau modèle à partir de données brutes.

La première approche, dite d'**inférence**, consiste à exploiter les paramètres d'un modèle déjà optimisé afin de générer des prédictions sur de nouvelles entrées. Cette méthode présente l'avantage d'être moins coûteuse en ressources computationnelles, dans la mesure où la phase d'évaluation des performances a été réalisée en amont, lors de l'entraînement. Elle requiert néanmoins une préparation rigoureuse des données d'entrée, dont le format doit être strictement conforme à celui des données utilisées durant l'apprentissage.

La seconde approche, qui consiste à construire un modèle s'articule autour d'un ensemble d'étapes structurées et interdépendantes, la figure 1 montres ces étapes. [5]

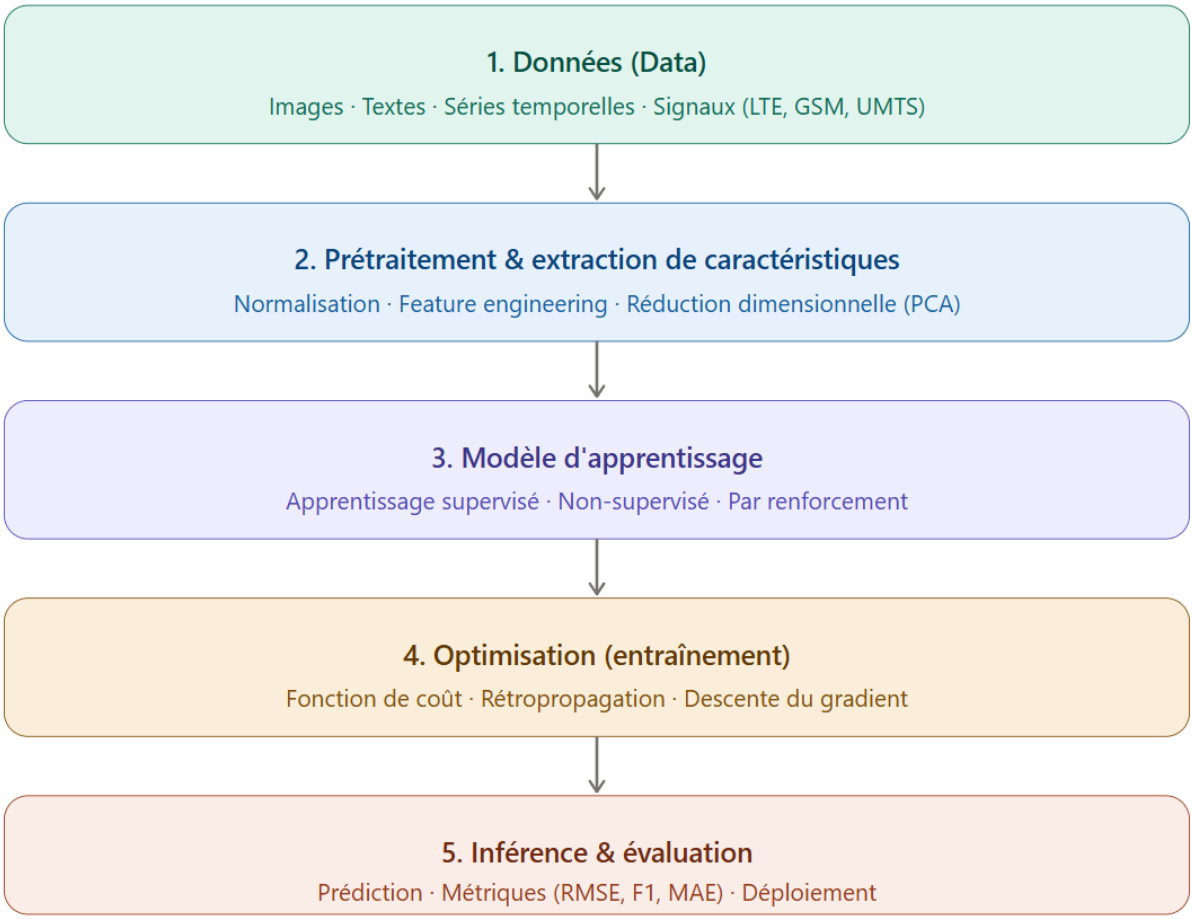


Figure 1 : Les étapes pour entraîner un nouveau modèle à partir de données brutes.

3.1.1 Collecte des données

La première étape consiste à constituer un ensemble de données pertinent et représentatif du problème étudié. Les sources de données sont diverses et peuvent inclure des journaux système, des métriques de performance, des séries temporelles financières (cours boursiers, taux de change), des données de mobilité et de transport, ou encore des indicateurs comportementaux liés à l'utilisation de produits ou de services numériques. Il est impératif que les données sélectionnées soient en adéquation avec la problématique traitée et représentatives de la population cible sur laquelle le modèle sera amené à effectuer des prédictions.

3.1.2 Prétraitement des données

Une fois les données collectées, une phase de prétraitement est indispensable afin de les rendre exploitables par les algorithmes d'apprentissage automatique. Cette étape englobe notamment l'étiquetage des observations, la gestion des valeurs manquantes, la transformation des séries temporelles par agrégation, ainsi que la normalisation ou la mise à l'échelle des variables, de manière à garantir des plages de valeurs homogènes entre les différentes caractéristiques. Il convient de souligner que les architectures profondes, telles que les grands modèles de langage, nécessitent des volumes de données étiquetées considérablement plus importants que les modèles d'apprentissage supervisé classiques.

3.1.3 Sélection des caractéristiques

Cette étape vise à identifier et à retenir les variables les plus pertinentes au regard de la problématique posée. L'analyse des corrélations constitue une méthode élémentaire couramment employée à cet effet. Par ailleurs, la plupart des cadres logiciels dédiés au Machine Learning proposent des méthodes automatisées de sélection des caractéristiques, permettant d'affiner l'optimisation du modèle.

3.1.4 Choix du modèle

Sur la base des caractéristiques retenues, il convient de sélectionner l'architecture algorithmique la mieux adaptée à la nature du problème. Parmi les options disponibles figurent les modèles de régression, les arbres de décision et les réseaux de neurones artificiels, chacun présentant des propriétés spécifiques en termes de complexité, d'interprétabilité et de capacité de généralisation.

3.1.5 Entraînement

L'entraînement constitue la phase centrale du processus, au cours de laquelle l'algorithme extrait les structures et les relations latentes présentes dans les données, et les encode sous forme de paramètres du modèle. Il s'agit d'un processus itératif impliquant le réglage des hyperparamètres, ainsi que des ajustements successifs du pipeline de traitement des données et de la sélection des caractéristiques, dans l'objectif d'atteindre des performances optimales.

3.1.6 Évaluation et tests

À l'issue de la phase d'entraînement, le modèle est soumis à une évaluation sur des données inédites, distinctes de celles utilisées pour l'apprentissage. Cette étape permet de mesurer la capacité de généralisation du modèle et de comparer ses performances à celles d'architectures alternatives. Il est à noter que seules des données non vues lors de l'entraînement fournissent une estimation fiable des performances attendues en conditions réelles de déploiement.

3.1.7 Déploiement

Lorsque les performances du modèle sont jugées satisfaisantes au regard des critères définis, celui-ci est intégré dans un environnement de production, où il est en mesure d'effectuer des prédictions ou de prendre des décisions en temps réel. Cette intégration peut s'effectuer au sein de systèmes existants ou d'applications logicielles, les principaux fournisseurs d'infrastructure cloud proposant des cadres facilitant ce déploiement.

3.1.8 Surveillance et mise à jour

Enfin, le suivi continu des performances du modèle en production est indispensable pour garantir sa fiabilité dans le temps. En fonction de l'évolution des données disponibles ou de la dynamique du problème traité, des mises à jour périodiques peuvent s'avérer nécessaires, qu'il s'agisse d'un réentraînement sur de nouvelles données, d'un ajustement des paramètres ou d'une révision du choix algorithmique, la figure 2 montre les étapes du fonctionnement d’un algorithme de machine Learning.

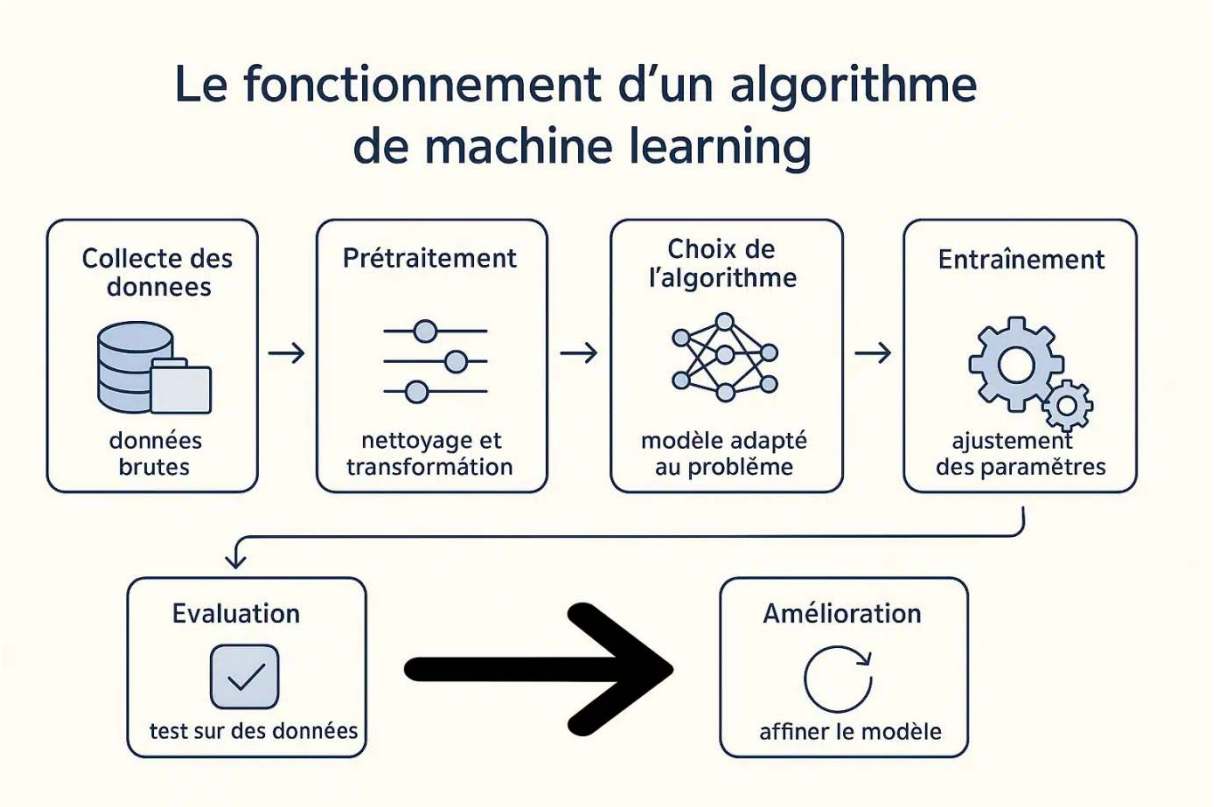


Figure 2: Le fonctionnement du machine Learning [6]

4. Les différents types de Machine Learning

Les méthodes d'apprentissage automatique sont généralement classifiées selon deux critères fondamentaux : le degré d'intervention humaine requis lors de leur mise en œuvre, et la nature des données exploitées, qu'elles soient étiquetées ou non. On distingue ainsi quatre paradigmes d'apprentissage principaux, illustrés dans la Figure 3.[5]

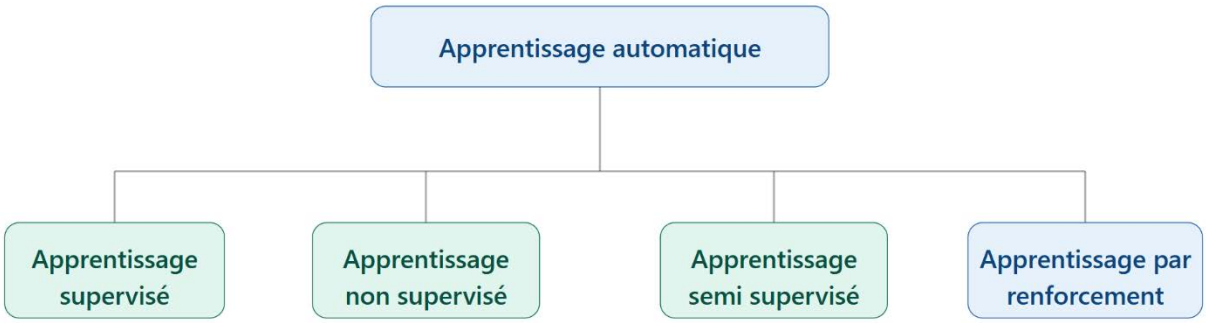


Figure 3: Les types de Machine Learning

4.1 L'apprentissage supervisé

L'apprentissage supervisé repose sur l'utilisation d'un corpus de données d'entraînement étiquetées, dans lequel chaque observation est associée à une sortie cible connue. L'algorithme a pour objectif d'apprendre une fonction de mapping reliant les variables d'entrée à la variable de sortie, de manière à pouvoir généraliser cette relation à des données inédites. Selon la nature de la variable cible, l'apprentissage supervisé se décline en deux sous-catégories : la régression, lorsque la sortie est une valeur continue, et la classification, lorsqu'il s'agit d'attribuer une observation à une catégorie discrète prédéfinie.[5]

4.2 L'apprentissage non supervisé

Contrairement au paradigme précédent, l'apprentissage non supervisé opère sur des données non étiquetées, en l'absence de toute variable cible explicite. L'algorithme est chargé de découvrir de manière autonome la structure intrinsèque des données, en identifiant des regroupements, des corrélations ou des représentations latentes. Ce paradigme est particulièrement adapté aux situations où l'annotation des données est inexistante ou inaccessible, et trouve des applications dans des tâches telles que le clustering, la réduction de dimensionnalité ou la détection d'anomalies.[5]

4.3 Apprentissage Semi-Supervisé

L’apprentissage semi-supervisé est une technique qui fusionne des aspects des deux types précédents. Dans ce type d’apprentissage, un petit ensemble de données étiquetées est combiné avec un ensemble plus vaste de données non étiquetées pour entraîner un modèle. Les données étiquetées servent de guide lors du processus d’apprentissage, tandis que les données non étiquetées aident le modèle à généraliser et à acquérir des fonctionnalités plus robustes. Cette approche est particulièrement précieuse dans les situations où l’obtention de grandes quantités de données étiquetées peut s’avérer difficile ou coûteuse.[5]

4.4 Apprentissage par Renforcement

L’apprentissage par renforcement est une approche où un agent interagit avec un environnement afin d’apprendre à prendre des actions qui maximisent une récompense cumulative. L’agent apprend par essais et erreurs en recevant des commentaires sous forme de récompenses ou de pénalités en fonction de ses actions. Cette technique implique trois éléments clés : l'agent, l’environnement et le signal de récompense. L’agent prend des mesures dans l’environnement en fonction de son état actuel et reçoit un signal de récompense qui indique à quel point ces actions étaient bonnes ou mauvaises. L’agent utilise ses commentaires pour mettre à jour sa politique et améliorer ses actions futures [5].

5. Les algorithmes d’apprentissage supervisé

De nombreux algorithmes et techniques de Machine Learning sont disponibles. ; le choix d'un algorithme approprié est conditionné par la nature du problème à résoudre ainsi que par les caractéristiques intrinsèques des données disponibles.

5.1 K-Nearest Neighbors

K-Nearest Neighbors (KNN) est un algorithme d'apprentissage automatique supervisé généralement utilisé pour la classification, mais peut aussi être utilisé pour des tâches de régression. Il fonctionne en trouvant les « k » points de données les plus proches (voisins) d'une entrée donnée et effectue des prédictions basées sur la classe majoritaire (pour la classification) ou la valeur moyenne (pour la régression) comme le montre la figure 4 et 5. Puisque KNN ne fait aucune hypothèse sur la distribution sous-jacente des données, il en fait une méthode d'apprentissage non paramétrique et basée sur des instances. [7]

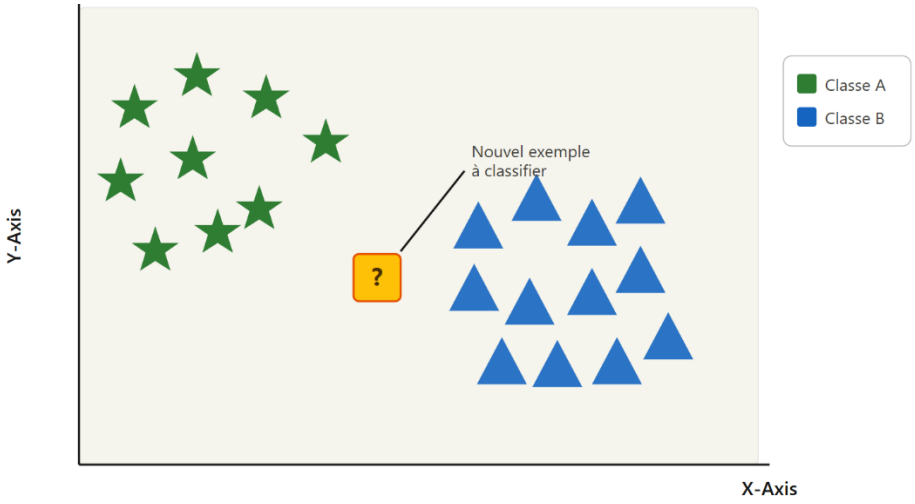


Figure 4: Représentation des données initiales et d'un nouvel échantillon à classifier dans l'algorithme KNN.

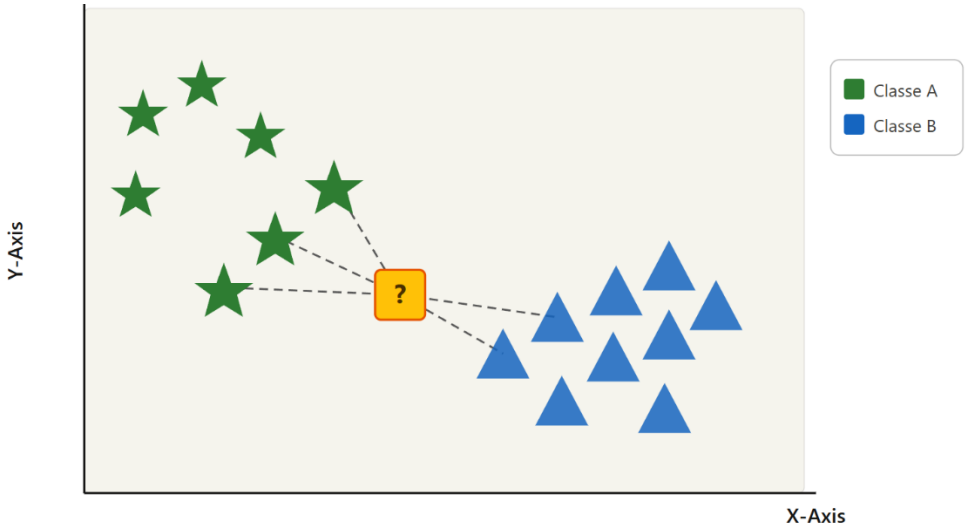


Figure 5: Calcul des distances et identification des voisins les plus proches pour la classification dans l'algorithme KNN.

KNN fonctionne en utilisant la proximité et le vote majoritaire pour faire des prédictions. Dans l'algorithme k-des plus proches voisins, k est simplement un nombre qui indique à l'algorithme combien de points ou voisins proches il doit regarder lorsqu'il prend une décision. [7]

L'algorithme KNN basé sur des métriques de distance afin d'identifier les voisins les plus proches d'un point de requête. Ces voisins sont ensuite exploités pour accomplir des tâches de classification ou de régression. Les métriques de distance les plus couramment employées sont :

La distance euclidienne : représente la longueur du segment rectiligne reliant deux points dans un espace. Elle correspond au chemin le plus court entre deux points et constitue la métrique de distance la plus intuitive et la plus répandue en apprentissage automatique.

$$d(x, X_i) = \sqrt{\sum_{j=1}^{d} \left(x_j - X_{i_j}\right)^2} \qquad (1)$$

avec :

x : Le point de requête (le nouvel exemple à classifier), représenté par un vecteur de d caractéristiques.

X_i : Le i-ème point d'entraînement dans le dataset, avec lequel on calcule la distance.

d : Le nombre de dimensions (caractéristiques) de l'espace. Par exemple, si chaque point est décrit par 3 variables (taille, poids, âge), alors $d = 3$.

x_j : La valeur de la j-ème caractéristique du point de requête x.

X_{i_j} : La valeur de la j-ème caractéristique du point d'entraînement X_i.

La distance de Manhattan : correspond à la somme des valeurs absolues des différences entre les coordonnées de deux points. Contrairement à la distance euclidienne, elle mesure le déplacement total effectué en ne se déplaçant qu'horizontalement et verticalement. Elle est formalisée comme suit :

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \qquad (2)$$

La distance de Minkowski : constitue une généralisation des deux métriques précédentes. Elle introduit un paramètre p permettant de moduler le comportement de la métrique en fonction du contexte applicatif. Sa formule générale est donnée par :

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \qquad (3)$$

Cette formulation unifiée englobe les distances euclidiennes et de Manhattan comme cas particuliers : lorsque $p = 2$, elle est équivalente à la distance euclidienne, et lorsque $p = 1$, elle se réduit à la distance de Manhattan. [7]

5.2 La forêt aléatoire (Random Forest)

Est un algorithme d'apprentissage automatique largement utilisé développé par Leo Breiman et Adele Cutler, qui combine la sortie de plusieurs arbres de décision pour obtenir un seul résultat. Sa facilité d'utilisation et sa flexibilité, associées à son efficacité en tant que classificateur de forêts aléatoires, ont alimenté son adoption, car il traite à la fois des problèmes de classification et de régression. [8]

5.2.1 Arbres de décision

Un arbre de décision est une structure de type organigramme où les nœuds internes représentent les décisions basées sur des caractéristiques, les branches représentent les résultats de ces décisions, et les nœuds feuilles représentent les prédictions finales. Bien que les arbres de décision soient faciles à comprendre et à interpréter, ils peuvent être sujets à un surajustement, surtout lorsque l'arbre est profond et complexe, ce principe est montré dans la figure 6. [8]

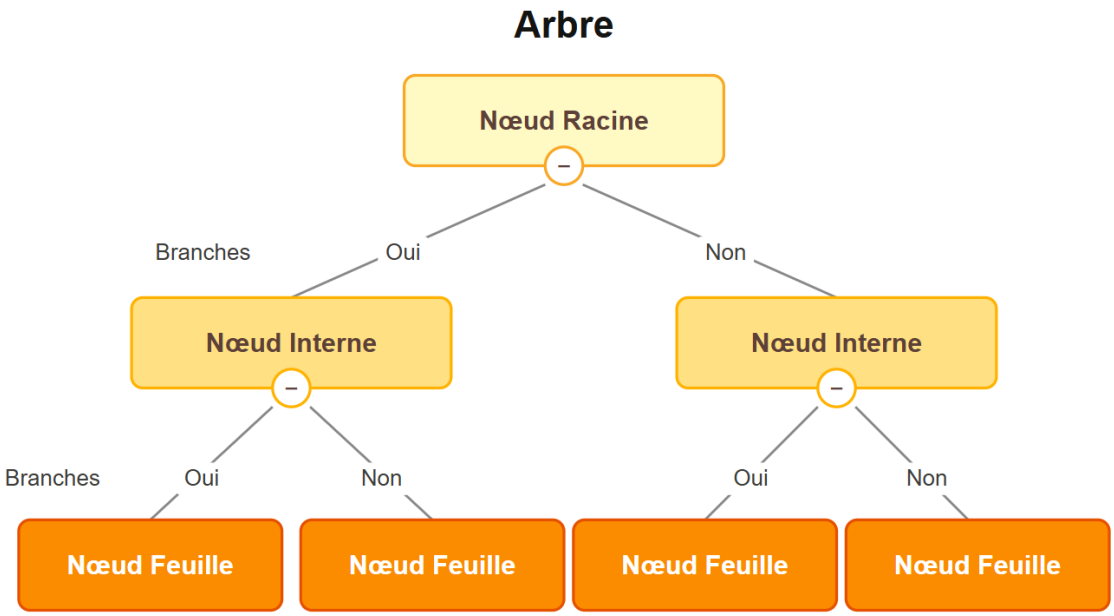


Figure 6: Représentation d'un arbre de décision

5.2.2 Principe de la forêt aléatoire

L'apprentissage en ensemble est une stratégie de modélisation qui consiste à combiner les prédictions de plusieurs modèles afin d'obtenir des performances supérieures à celles d'un modèle unique. La forêt aléatoire constitue une illustration emblématique de cette approche, elle agrège les sorties d'un grand nombre d'arbres de décision entraînés indépendamment,

produisant ainsi des prédictions plus précises et plus stables, notamment en termes de robustesse face aux variations des données.

Au sein de la forêt aléatoire, la technique du bagging (*Bootstrap Aggregating*) est employée pour introduire de la diversité entre les arbres. L'algorithme génère plusieurs sous-ensembles du jeu de données d'origine par échantillonnage aléatoire avec remise, selon le principe du bootstrap. Chaque arbre de décision est alors entraîné sur un sous-ensemble distinct, ce qui permet de réduire la variance globale du modèle et de limiter le risque de sur-apprentissage. Vue d'ensemble de la forêt aléatoire est montrée dans la figure 7.

Lors de la construction de chaque arbre, la forêt aléatoire n'évalue pas l'ensemble des variables disponibles à chaque nœud de décision, mais sélectionne aléatoirement un sous-ensemble restreint de caractéristiques candidates. Ce mécanisme garantit une diversité structurelle entre les arbres et prévient la domination du modèle par un nombre limité de prédicteurs fortement corrélés, renforçant ainsi la capacité de généralisation de l'ensemble [8].

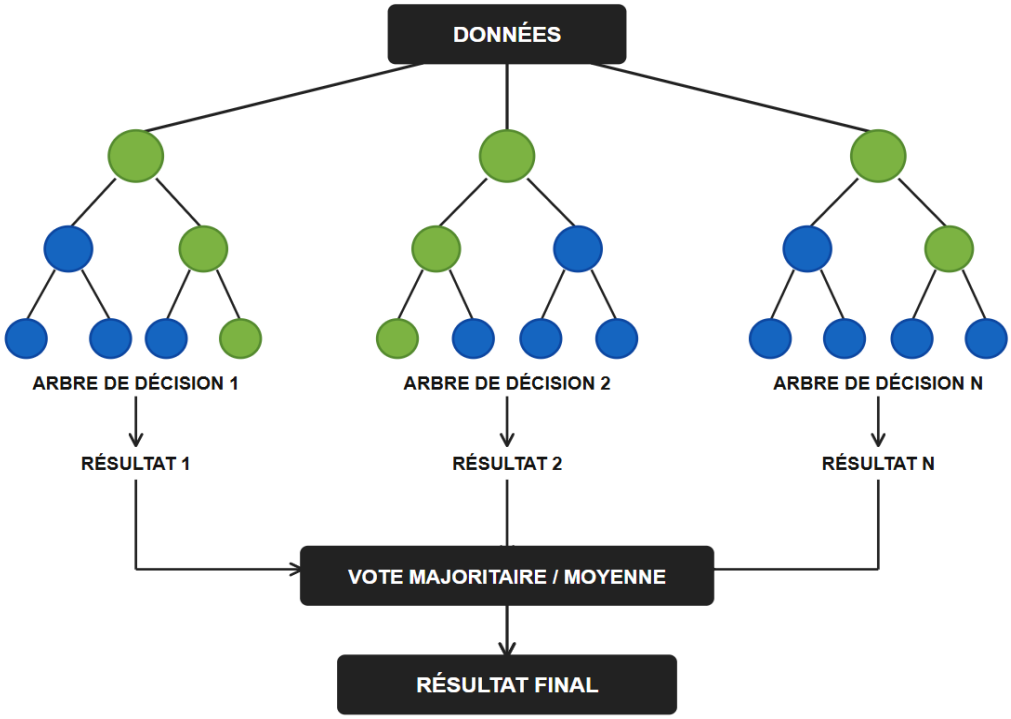


Figure 7: Vue d'ensemble de la forêt aléatoire

La forêt aléatoire repose ainsi sur quatre composantes fondamentales :

Les arbres de régression et de classification : Les modèles élémentaires qui constituent la forêt sont des arbres de décision profonds, capables de capturer des relations complexes et non-linéaires dans les données.

L'échantillonnage bootstrap : Chaque arbre est construit à partir d'un échantillon aléatoire du jeu de données d'entraînement tiré avec remise. Cette procédure garantit que chaque arbre est exposé à une représentation légèrement différente des données, favorisant ainsi la diversité au sein de l'ensemble.

La sélection aléatoire de variables. À chaque nœud de chaque arbre, un sous-ensemble aléatoire de caractéristiques est constitué, et la meilleure règle de division est recherchée exclusivement parmi ces variables candidates. Ce mécanisme décorrèle les arbres entre eux et prévient la concentration des décisions autour des variables les plus informatives.

L'agrégation des prédictions. À l'instar du bagging, les prédictions individuelles de l'ensemble des arbres sont combinées pour produire une prédiction finale. Dans le cadre de la régression, on procède généralement à la moyenne des sorties individuelles. Dans le cadre de la classification, la décision finale est déterminée par vote majoritaire ou par la moyenne des probabilités prédites pour chaque classe [9].

5.3 XGBoost (Extreme Gradient Boosting)

XGBoost, acronyme de Extreme Gradient Boosting, est un algorithme d'apprentissage automatique supervisé fondé sur le principe du gradient boosting appliqué aux arbres de décision. Il s'agit d'une implémentation hautement optimisée et scalable de cette technique, conçue pour offrir à la fois des performances prédictives élevées et une efficacité computationnelle remarquable.

Sur le plan algorithmique, XGBoost construit séquentiellement un ensemble d'arbres de décision, chaque arbre étant entraîné pour corriger les erreurs résiduelles de l'ensemble des arbres précédents. Cette procédure d'optimisation itérative repose sur la minimisation d'une fonction de perte régularisée, ce qui confère à l'algorithme une capacité naturelle à contrôler le sur-apprentissage tout en maintenant une grande précision prédictive [10].

5.3.1 Principes de l’algorithme XGBoost

Le principe de fonctionnement de XGBoost repose sur une approche itérative. Dans un premier temps, une prédiction initiale est réalisée. Ensuite, les résidus, correspondant à la différence entre les valeurs réelles et les valeurs prédites, sont calculés. Un arbre de décision est alors construit pour modéliser ces erreurs, en s’appuyant sur des mesures de similarité afin de déterminer les meilleures divisions des données.

Au cours de la construction de l’arbre, les gains de similarité sont évalués afin de sélectionner les variables et les seuils optimaux pour chaque nœud. Les valeurs de sortie des feuilles sont ensuite calculées à partir des résidus. Dans le cas de la classification, ces valeurs sont généralement exprimées sous forme de probabilités ou de log-vraisemblances.

Chaque nouvel arbre est construit à partir des erreurs des arbres précédents, ce qui permet une amélioration progressive du modèle. Ce processus est répété jusqu’à atteindre un nombre d’itérations défini ou jusqu’à ce que les erreurs deviennent négligeables. Contrairement à la méthode de la forêt aléatoire, les arbres dans XGBoost ne sont pas indépendants, mais interdépendants.

Lors de la phase de prédiction, les sorties de tous les arbres sont combinées. Chaque contribution est pondérée par un taux d’apprentissage (learning rate), puis additionnée à la prédiction initiale afin d’obtenir la valeur finale ou la classe prédite. Le processus d’apprentissage automatique de l’algorithme XGBoost est montré dans la figure 8.

Afin d'optimiser les performances du modèle, XGBoost intègre plusieurs techniques avancées :

- **Régularisation** : un paramètre (λ) est utilisé pour limiter la complexité du modèle et réduire le risque de surapprentissage.
- **Élagage (pruning)** : un paramètre (γ) permet de supprimer les branches peu significatives, simplifiant ainsi les arbres.
- **Approximation par quantiles pondérés** : cette méthode accélère le choix des seuils de division en réduisant le nombre de valeurs testées.
- **Parallélisation** : les calculs sont répartis sur plusieurs unités de traitement, ce qui améliore la vitesse d'apprentissage.
- **Gestion des données manquantes** : l'algorithme prend en compte la rareté des données en déterminant automatiquement la meilleure direction de division.
- **Optimisation du cache** : l'utilisation de la mémoire cache permet d'accélérer les calculs et d'améliorer les performances globales.
- **Traitement out-of-core** : cette technique permet de traiter des ensembles de données très volumineux stockés sur disque, en les divisant en blocs compressés.[12]

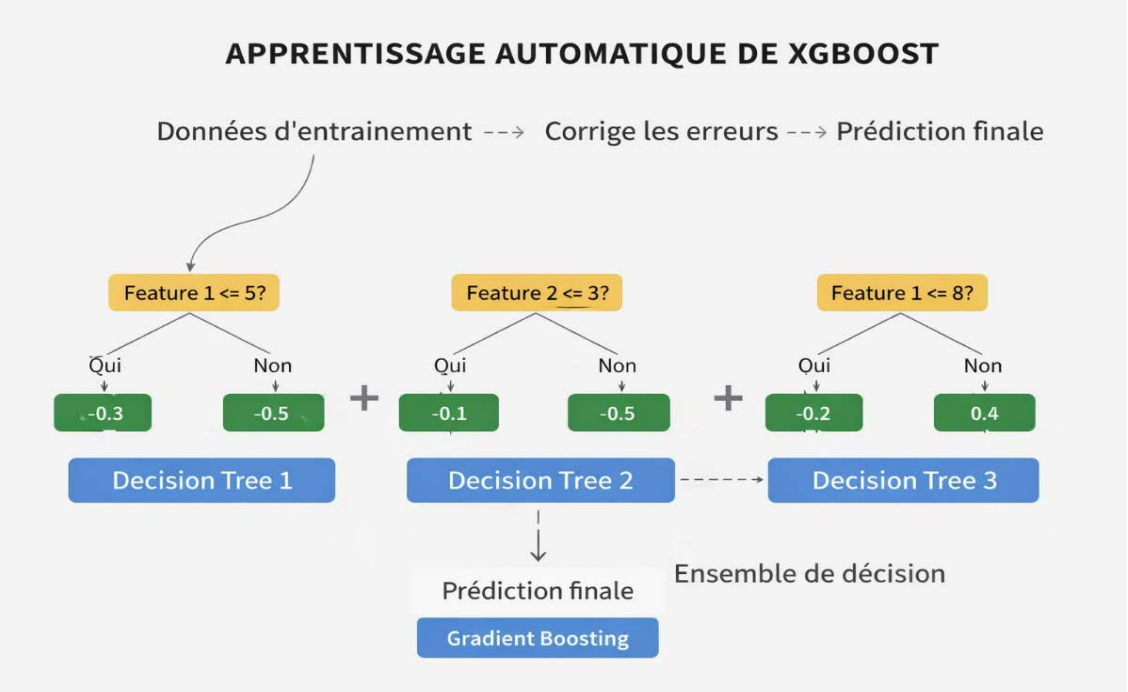


Figure 8: Illustration du processus d'apprentissage automatique de l'algorithme XGBoost basé sur la méthode du Gradient Boosting.

XGBoost présente plusieurs avantages importants qui expliquent sa popularité en apprentissage automatique. Il offre d'abord une excellente précision prédictive, grâce à des techniques avancées de régularisation qui réduisent le surapprentissage. Il se distingue aussi par sa grande efficacité de calcul, utilisant le traitement parallèle et des optimisations de mémoire, ce qui permet d'entraîner rapidement des modèles même sur de grands ensembles de données. Enfin, XGBoost est flexible et facile à utiliser, il peut traiter différents types de données, propose des outils intégrés de validation croisée, d'optimisation des hyperparamètres et fournit des mesures d'importance des variables, utiles pour interpréter les résultats et comprendre les données. Malgré ses nombreux avantages, XGBoost présente certaines limites. Il peut être très gourmand en ressources de calcul, surtout avec de grands ensembles de données ou un grand nombre d'itérations de boosting, de plus, l'algorithme possède beaucoup d'hyperparamètres, ce qui rend son optimisation plus complexe et parfois longue et sans un réglage approprié, cela peut aussi entraîner un surapprentissage (overfitting). Enfin, même si XGBoost propose des outils d'interprétation, ses modèles restent plus difficiles à comprendre que des méthodes plus simples comme la régression linéaire ou un arbre de décision unique [11].

6. Conclusion

Ce chapitre a permis de poser les bases théoriques du Machine Learning en mettant en lumière son importance dans le domaine de l'intelligence artificielle. Après avoir présenté les concepts généraux et les domaines d'application de l'IA, nous avons détaillé le processus complet de développement d'un modèle d'apprentissage automatique, depuis la préparation des données jusqu'à son déploiement.

Nous avons également étudié les différents types d'apprentissage, en insistant particulièrement sur l'apprentissage supervisé et ses algorithmes les plus utilisés, tels que KNN, Random Forest et XGBoost, qui offrent des performances élevées dans les tâches de classification et de prédiction.

Chapitre 2 : Acquisition des Signaux Radio

1.Introduction

Le développement de méthodes robustes et intelligentes pour l'acquisition, le traitement et l'analyse des signaux radiofréquences s'impose comme une nécessité incontournable, afin de garantir des communications fiables, performantes et adaptées aux environnements radio de plus en plus denses et hétérogènes.

L'acquisition des signaux radio constitue la première étape de cette chaîne de traitement. Elle permet de capter les ondes présentes dans l'environnement et de les convertir en données exploitables. La qualité de cette étape est déterminante, car elle influence directement la précision des analyses réalisées par la suite. Un signal mal acquis peut entraîner des erreurs importantes dans l'interprétation des informations.

Le prétraitement des signaux intervient ensuite pour améliorer leur qualité. Il consiste à réduire les effets du bruit et des interférences, et à préparer les données pour les étapes d'analyse. Ces opérations sont indispensables dans les environnements réels, où les conditions de transmission sont souvent variables et perturbées.

Par ailleurs, l'étude de l'activité spectrale permet de comprendre comment les différentes bandes de fréquences sont utilisées. Elle met en évidence que certaines parties du spectre sont fortement occupées, tandis que d'autres restent disponibles. Cette observation est importante pour optimiser l'utilisation des ressources et éviter les interférences entre systèmes.

Dans ce cadre, la détection de signal joue un rôle central. Elle permet de déterminer si un signal est présent ou non dans une bande de fréquence donnée. Cette opération est essentielle pour de nombreuses applications, notamment dans les systèmes intelligents qui doivent prendre des décisions en temps réel.

Enfin, les avancées technologiques ont permis l'apparition de solutions modernes comme la radio définie par logiciel. Ces systèmes offrent une grande flexibilité et facilitent l'analyse des signaux radio à l'aide d'outils accessibles, comme le RTL-SDR

2.Activité spectrale et détection de signal

2.1 Activité spectrale

L'activité spectrale représente l'étude de l'occupation du spectre électromagnétique par les signaux radio dans une bande de fréquences donnée et sur une durée temporelle déterminée. Elle permet d'analyser la manière dont les ressources fréquentielles sont exploitées par différents systèmes de communication sans fil. Cette analyse est essentielle dans les environnements modernes où plusieurs technologies coexistent dans des bandes de fréquences communes [13][14].

Dans les systèmes actuels, le spectre radio est considéré comme une ressource rare et limitée. Il est partagé entre différents services tels que les réseaux cellulaires (GSM, LTE, 5G), les réseaux Wi-Fi, les systèmes radar et les communications satellitaires. Cette coexistence entraîne une utilisation dynamique du spectre, caractérisée par des variations constantes dans le temps et la fréquence [16].

L'occupation du spectre n'est pas uniforme. En effet, certaines bandes de fréquences sont fortement utilisées par les utilisateurs autorisés, tandis que d'autres restent partiellement ou totalement inoccupées. Ces zones libres sont appelées **trous spectraux** (*spectrum holes* ou *white spaces*). Elles représentent une opportunité importante pour les systèmes intelligents capables de détecter ces espaces et de les exploiter sans interférer avec les utilisateurs principaux [13][14].

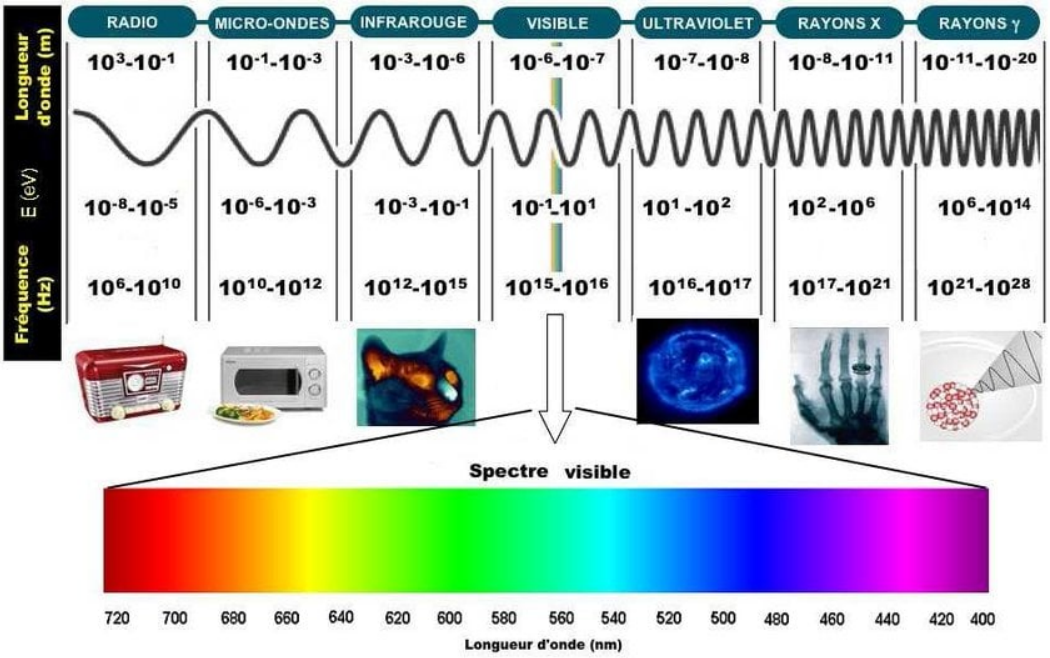


Figure 9 : Occupation du spectre électromagnétique.

2.2 Détection de signal dans les systèmes radio

La détection de signal est une opération fondamentale dans les systèmes de communication sans fil. Elle consiste à déterminer si un signal utile est présent ou absent dans une bande de fréquence donnée, malgré la présence du bruit et des interférences. Cette étape est essentielle pour assurer une utilisation efficace et fiable du spectre radio [15][16].

D'un point de vue théorique, le problème de détection peut être formulé comme un test d'hypothèse binaire. Le système doit choisir entre deux hypothèses :

- **H_0 : absence de signal**, où le signal reçu contient uniquement du bruit
- **H_1 : présence de signal**, où le signal reçu est constitué du signal utile plus du bruit

Mathématiquement, cela s'écrit :

$$H_0: x(t) = n(t)$$
$$H_1: x(t) = s(t) + n(t) \quad (4)$$

Où $x(t)$représente le signal reçu, $s(t)$le signal utile et $n(t)$le bruit additif [15].

La difficulté principale de la détection réside dans le fait que le bruit et les interférences peuvent masquer complètement le signal utile, surtout lorsque sa puissance est faible. Ainsi, le système de détection doit être suffisamment sensible pour détecter les signaux faibles tout en évitant les erreurs de décision [17].

2.3 Performance et critères de détection

Les performances d'un système de détection de signal sont généralement évaluées à l'aide de plusieurs critères statistiques importants. Le premier est la **probabilité de détection (Pd)**, qui représente la capacité du système à détecter correctement un signal lorsqu'il est réellement présent. Une valeur élevée de (Pd) est souhaitée afin de garantir une bonne sensibilité du système [15].

Le deuxième critère est la **probabilité de fausse alarme (Pfa)**, qui correspond à la détection d'un signal alors qu'il est absent. Une valeur élevée de (Pfa) entraîne une mauvaise utilisation du spectre, car le système considère à tort une bande occupée [15][17].

Enfin, on définit également la **probabilité de non-détection (Pmd)**, qui représente l'échec du système à détecter un signal réellement présent. Ce cas est particulièrement critique, car il peut entraîner des interférences avec les utilisateurs primaires [17].

Un système de détection efficace doit donc trouver un compromis optimal entre ces différents paramètres afin de garantir à la fois fiabilité et efficacité spectrale [15][17].

2.3.1 Techniques de détection

Les techniques de détection se divisent en méthodes de traitement du signal et en méthodes coopératives, ces dernières améliorant la fiabilité grâce à la collaboration entre plusieurs nœuds.

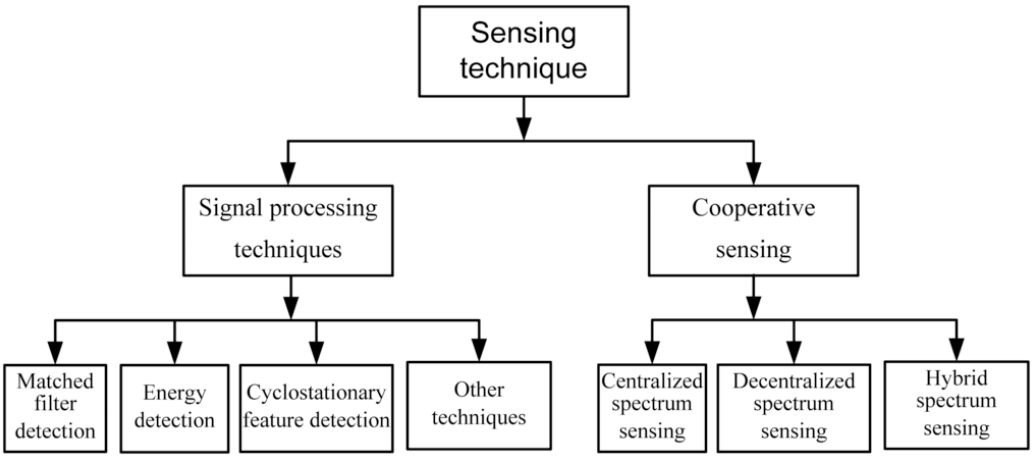


Figure 10 : les techniques de détection

Les techniques de traitement du signal dans les réseaux radio cognitifs reposent principalement sur trois méthodes de détection : le filtre adapté, la détection d'énergie et la détection des caractéristiques cyclos stationnaires.

La **détection par filtre adapté** offre une grande rapidité et une bonne efficacité en termes de temps et de nombre d'échantillons requis, mais elle nécessite une connaissance préalable

complète du signal de l'utilisateur primaire et présente une forte complexité ainsi qu'une consommation énergétique élevée.

La **détection d'énergie** est la méthode la plus simple, car elle ne requiert aucune information préalable sur le signal. Toutefois, elle présente une faible précision, surtout en présence d'un faible rapport signal/bruit, et ne permet pas de distinguer les différents types d'utilisateurs.

La **détection cyclo stationnaire** repose sur l'exploitation des propriétés périodiques des signaux modulés. Elle offre de meilleures performances dans des environnements à faible SNR et permet une bonne classification des signaux, mais elle est plus complexe et plus coûteuse en termes de calcul.

En raison des limitations des conditions de propagation et des contraintes matérielles, les capteurs radio cognitifs utilisent également des techniques de **détection coopérative**. Celle-ci peut être centralisée, décentralisée ou hybride. Elle permet d'améliorer la fiabilité de la détection en combinant les décisions de plusieurs capteurs, mais elle augmente la complexité du système et les ressources nécessaires. [18]

2.4 Importance de la détection spectrale

La détection spectrale joue un rôle central dans les systèmes de communication modernes, en particulier dans les environnements dynamiques où le spectre est partagé entre plusieurs utilisateurs. Elle permet d'identifier les ressources fréquentielles disponibles et d'assurer une utilisation efficace du spectre radio [13][14].

Dans les systèmes de radio cognitive, comme le montre la figure 8, la détection de signal constitue la première étape du processus intelligent. Ce processus peut être résumé comme suit : perception de l'environnement radio, analyse du spectre, prise de décision et adaptation des paramètres de transmission [13].

Grâce à ce mécanisme, les systèmes peuvent s'adapter automatiquement aux conditions du canal radio. La détection permet également de réduire les interférences entre utilisateurs, d'améliorer la qualité de service (QoS) et de garantir la coexistence entre différents systèmes de communication [14].

Cependant, elle reste une tâche complexe en raison de la présence de bruit, de fading et de la faible puissance de certains signaux [17].

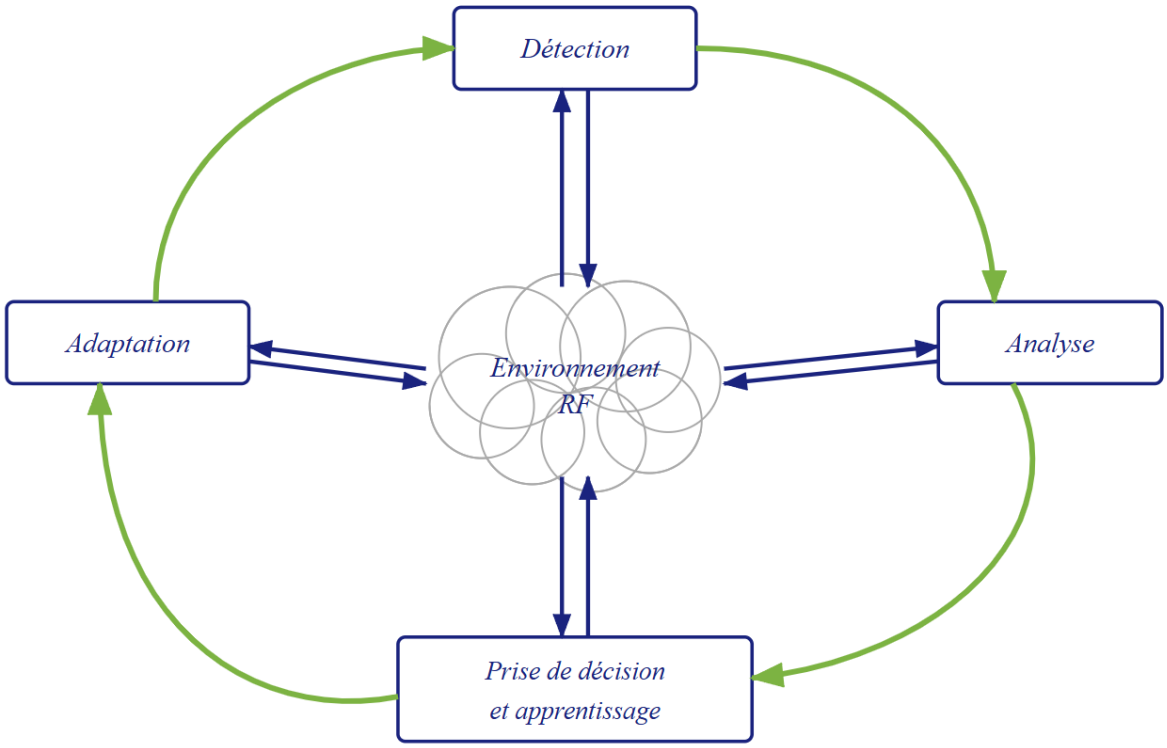


Figure 11 : Cycle de la radio cognitive-interaction avec l'environnement RF

2.5 Applications

Les techniques de détection de signal et d'analyse spectrale trouvent des applications dans plusieurs domaines des télécommunications modernes [16].

Dans les réseaux mobiles tels que GSM, LTE et 5G, elles sont utilisées pour la gestion dynamique des ressources radio, la planification des cellules et la réduction des interférences entre stations de base. Elles permettent ainsi d'améliorer les performances globales des réseaux [16].

Dans le domaine de la radio cognitive, ces techniques permettent une utilisation intelligente du spectre en détectant les bandes de fréquences libres et en les exploitant sans perturber les utilisateurs primaires [13]. Cela améliore considérablement l'efficacité spectrale.

Les organismes de régulation utilisent également ces techniques pour la surveillance du spectre, afin de détecter les interférences, les brouillages et les transmissions illégales [16]. Enfin, dans le domaine militaire, elles sont utilisées pour la détection de radars, la guerre électronique et l'interception des signaux [17].

3. La Radio Logiciel (Software Defined Radio - SDR)

3.1 Présentation du RTL-SDR

La radio définie par logiciel (Software Defined Radio – SDR) constitue l'une des avancées majeures dans le domaine des communications sans fil. Contrairement aux systèmes radio traditionnels reposant sur des composants matériels dédiés, le SDR utilise des traitements

logiciels pour effectuer la modulation, la démodulation et l'analyse des signaux, offrant ainsi une grande flexibilité et une réduction significative des coûts.

Parmi les implémentations les plus répandues de cette technologie, le RTL-SDR se distingue comme une solution économique et performante pour la réception et l'analyse des signaux radio. Il s'agit d'un récepteur basé sur des clés USB, utilisant principalement des puces telles que le RTL2832U et le tuner R820T, et pouvant être connecté à un ordinateur via un port USB afin de fonctionner comme un récepteur large bande.

Initialement conçus pour la réception de la télévision numérique terrestre (DVB-T), ces dispositifs ont été détournés de leur usage initial suite à la découverte de la possibilité d'accéder directement aux données brutes I/Q. Cette avancée, rendue possible grâce aux contributions de la communauté open source, a permis de transformer ces tuners en récepteurs SDR polyvalents via des pilotes logiciels spécifiques.

Les dispositifs RTL-SDR montrée dans la figure 9 permettent d'explorer une grande variété de signaux radio. Ils sont utilisés pour la réception des stations FM, l'écoute des communications aéronautiques et maritimes, l'analyse des transmissions radioamateur, ainsi que le décodage de signaux numériques tels que les données ADS-B des aéronefs ou encore les signaux issus des satellites météorologiques.

En raison de sa simplicité d'utilisation, de son faible coût et de sa grande polyvalence, le RTL-SDR est devenu un outil incontournable dans les domaines de l'enseignement, de la recherche et de l'expérimentation en télécommunications et en traitement du signal. [19]

Figure 12 **:** Clé RTL-SDR

La mise en place d'un système RTL-SDR ne requiert qu'un équipement de base accessible et peu coûteux. Les composants indispensables à son fonctionnement sont les suivants :

- ✓ Un ongle RTL-SDR constituant l'élément central du dispositif ;

- ✓ Une antenne adaptée aux fréquences ciblées par l'application
- ✓ Un câble coaxial accompagné d'un adaptateur assurant la liaison physique entre l'ongle et l'antenne ;
- ✓ Un ordinateur équipé d'un processeur au minimum double cœur, ou un système embarqué fonctionnant sous Linux ;
- ✓ Les pilotes et logiciels associés au RTL-SDR, dont la grande majorité est disponible gratuitement.

Au-delà de cette configuration minimale, il est possible d'étendre les capacités du système par l'adjonction d'équipements complémentaires. Un convertisseur de fréquence ascendant (*up*-converter) permet d'étendre la plage de réception aux bandes hautes fréquences (HF), couvrant la gamme de 0 à 30 MHz, qui n'est pas nativement accessible au RTL-SDR. Un amplificateur à faible bruit (*Low Noise Amplifier* – LNA) peut être intégré en tête de chaîne afin d'améliorer la sensibilité de réception, notamment dans les environnements à faible niveau de signal. Enfin, des filtres de présélection (preselector filtrs) permettent d'atténuer les interférences hors bande et de réduire le bruit de fond, contribuant ainsi à une meilleure qualité spectrale du signal reçu [20].

3.2 Fonctionnement des dongles RTL-SDR

Le principe repose sur la conversion d'un signal radio analogique en données numériques traitables par un ordinateur. L'antenne capte d'abord les ondes électromagnétiques, qui sont amplifiées par un **LNA** (amplificateur à faible bruit) pour en préserver la clarté. Ce signal est ensuite mélangé à une fréquence générée localement (**LO**) dans la puce de tuner pour abaisser sa fréquence vers une plage intermédiaire plus facile à manipuler. Une fois filtré et ajusté en gain, ce signal analogique entre dans la puce de traitement où un convertisseur **ADC** le transforme en signal numérique. Ce flux est alors séparé en deux composantes complexes, nommées **I (In-phase)** et **Q (Quadrature)**, puis filtré et réduit en débit (**décimation**) avant d'être transmis via **USB** au PC. C'est ensuite un logiciel dédié qui se charge de démoduler ces données brutes pour restituer le son ou les données de la station radio choisie [20].

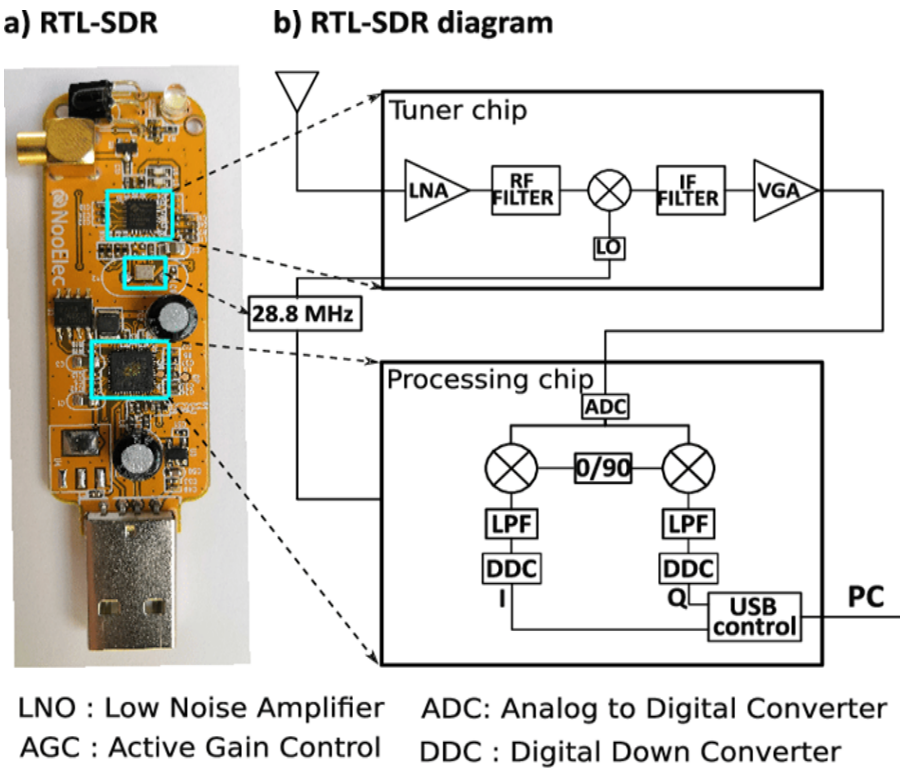


Figure 13 : Schéma fonctionnel de la clé RTL-SDR.

3.3 Les applications du RTL-SDR

La clé RTL-SDR permet de réaliser de nombreux projets technologiques et scientifiques. Parmi les principales applications, on peut citer quelques-unes [21]:

- Écoute des conversations non cryptées de la police, des ambulances, des pompiers et des services d'urgence médicale.
- Écoute des communications du contrôle aérien. Suivi des positions des aéronefs (comme un radar) avec décodage ADS-B.
- Décodage des messages courts ACARS des aéronefs.
- Analyse des communications radio commerciales à ressources partagées.
- Décodage des transmissions vocales numériques non cryptées.
- Suivi des positions des bateaux (comme un radar) avec décodage AIS.
- Décodage du trafic des pagers POCSAG/FLEX.

3.4 Partie logicielle

La partie logicielle du RTL-SDR constitue l'élément central du système de radio définie par logiciel. Contrairement aux architectures radio classiques où le traitement est assuré par des circuits matériels dédiés, le RTL-SDR repose sur des programmes informatiques pour effectuer l'ensemble des opérations de traitement du signal. Ces logiciels permettent de configurer les paramètres du récepteur tels que la fréquence, le gain et la bande passante, tout en assurant la visualisation du spectre de fréquence à travers des outils informatiques. Ils offrent également des fonctions avancées telles que la démodulation de signaux analogiques et numériques (FM, AM, GSM), ainsi que l'enregistrement et l'analyse des données IQ. Cette approche logicielle

rend le système extrêmement flexible et adaptable à différentes applications, notamment dans les domaines des télécommunications, de la recherche et de la surveillance spectrale. [23]

Les logiciels de traitement du signal occupent une place essentielle dans l'écosystème RTL-SDR, en particulier dans le cadre de la recherche scientifique et du développement d'algorithmes avancés. Des outils tels que GNU Radio et MATLAB/Simulink permettent de concevoir des chaînes de traitement du signal complexes à partir de blocs fonctionnels. GNU Radio, en tant que framework open source, offre une grande flexibilité pour implémenter des systèmes SDR personnalisés, tandis que MATLAB propose un environnement puissant pour la simulation, l'analyse et le traitement des données IQ. Ces logiciels sont largement utilisés dans le milieu académique pour développer et tester des algorithmes de détection, de filtrage et de modulation, contribuant ainsi à l'évolution des technologies de communication modernes.

Les logiciels spécialisés de décodage permettent d'exploiter les données capturées par le RTL-SDR afin d'extraire des informations spécifiques à partir de signaux radio. Contrairement aux logiciels généralistes, ces outils sont conçus pour traiter des protocoles particuliers tels que le GSM, les communications aéronautiques ou les capteurs IoT. Par exemple, gr-gsm est utilisé pour analyser les communications GSM, tandis que RTL_433 permet de décoder les signaux émis par des dispositifs fonctionnant dans la bande ISM. Ces logiciels reposent sur l'analyse des données IQ pour reconstruire les informations transmises, ce qui les rend particulièrement utiles dans les domaines de la sécurité, de la recherche et de l'ingénierie des télécommunications 24].

Les logiciels de réception généralistes représentent la première catégorie d'outils utilisés avec le RTL-SDR. Ils permettent à l'utilisateur de capter et de visualiser en temps réel les signaux radio présents dans l'environnement. Des applications comme SDR#, HDSDR, SDR Console et SDR++ offrent une interface graphique intuitive qui affiche le spectre de fréquence et le waterfall, facilitant ainsi l'identification des signaux actifs. Ces logiciels intègrent des fonctions de réglage de la fréquence centrale, du gain et des filtres, ainsi que des options de démodulation pour différents types de signaux. Leur simplicité d'utilisation et leur efficacité en font des outils indispensables pour les débutants comme pour les utilisateurs avancés souhaitant analyser rapidement l'activité spectrale. [25]

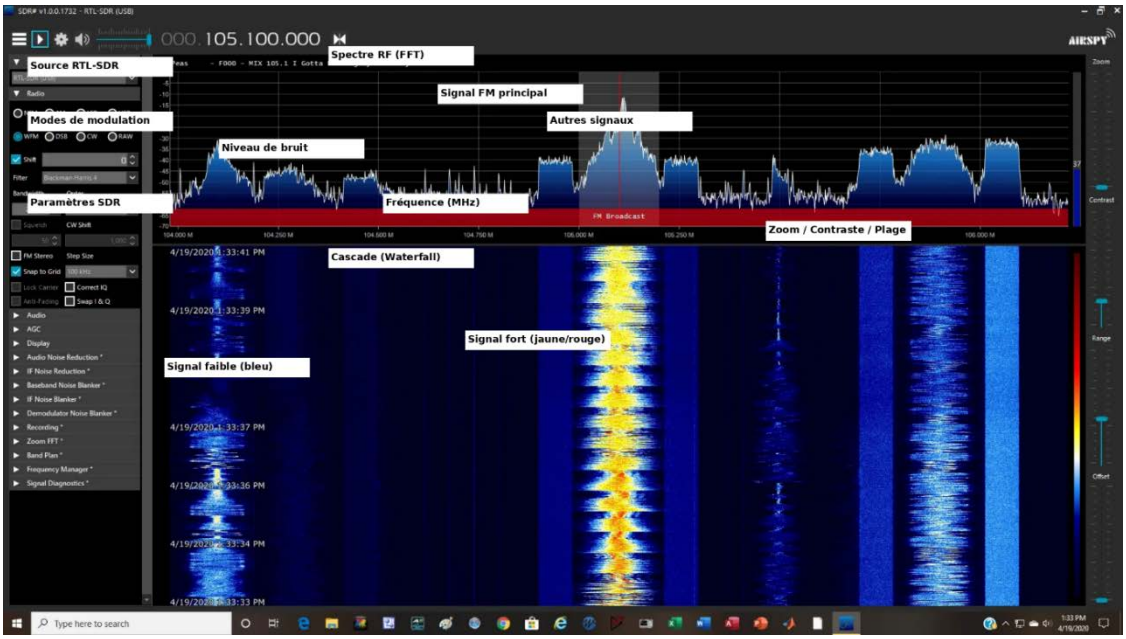


Figure 14 : SDRSharp (or SDR#)

4. Types de signaux étudiés

4.1 Signal FM (Frequency Modulation)

La **modulation de fréquence (FM)** est une technique de modulation de signal utilisée en communication électronique, initialement pour transmettre des messages via des ondes radio. Dans ce procédé, la fréquence instantanée d'une onde porteuse est ajustée en fonction d'une caractéristique du signal à transmettre, généralement l'amplitude instantanée d'un signal.

Dans le cas de la **modulation de fréquence analogique**, utilisée par exemple pour la diffusion radio de la voix ou de la musique, la déviation instantanée de la fréquence (c'est-à-dire la différence entre la fréquence de la porteuse et sa fréquence centrale) est proportionnelle à l'amplitude du signal modulant [22].

4.1.1Bande de fréquence

Dans le monde, la diffusion FM se situe dans la partie VHF du spectre radio. Généralement, la plage de fréquences de 87,0 à 108,0 MHz est utilisée, ou une portion de celle-ci, avec quelques exceptions :

- La plage de 87,0 à 87,5 MHz n'est pas disponible dans la région UIT 1 (Afrique, Europe et Russie).
- Dans les anciennes républiques soviétiques et certains pays de l'ancien bloc de l'Est, l'ancienne bande 65,8–74 MHz est également utilisée, avec des fréquences attribuées par intervalles de 30 kHz. Cette bande, connue sous le nom de bande OIRT, est progressivement abandonnée. Dans les zones où la bande OIRT est encore utilisée, la bande 87,0–108,0 MHz est appelée bande CCIR.
- Au Japon, la diffusion FM utilise la bande 76–95 MHz.
- Au Brésil, jusqu'à la fin des années 2010, les stations FM n'exploitaient que la bande 88–108 MHz. Avec la suppression de la télévision analogique, la bande 76–88 MHz

(anciennes chaînes 5 et 6 en VHF) a été réattribuée aux anciennes stations locales en modulation d'amplitude (MW) ayant migré vers la FM, conformément aux directives de l'ANNATEL.[22]

La figure 16 montre la réception de la station radio d'Oran (El Bahia) sur la fréquence 92.7 Mhz.

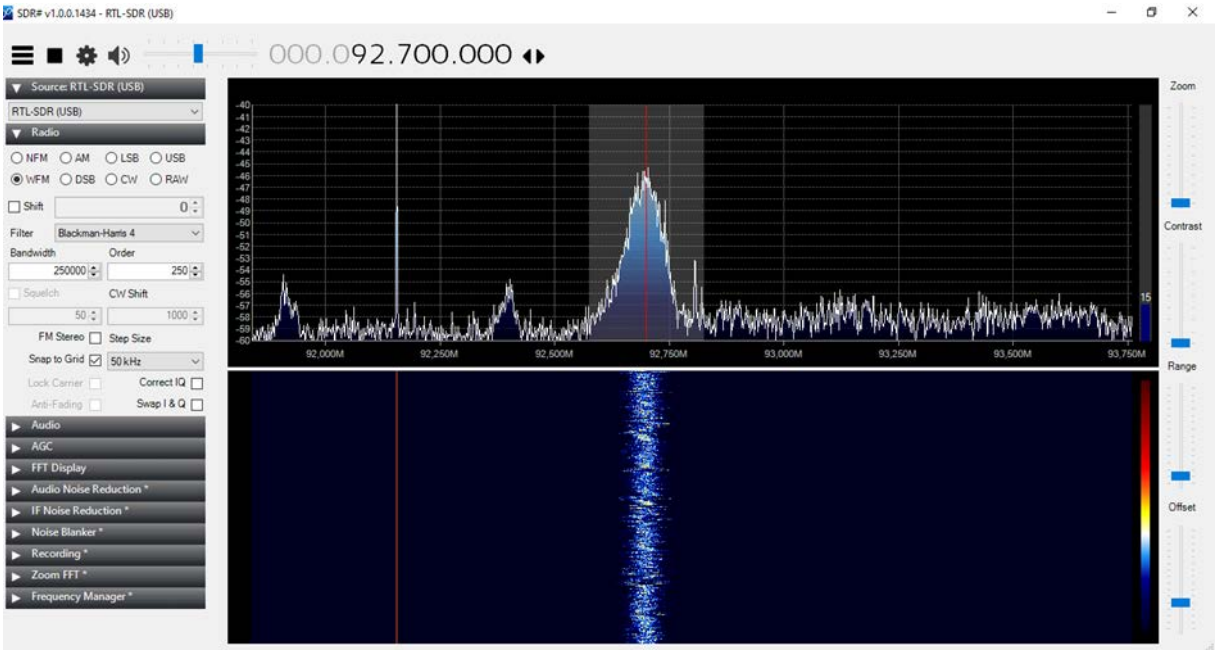


Figure 15 : Spectre de la station radio d'Oran (El Bahia).

La largeur de bande d'une transmission FM peut être estimée selon la règle de Carson, qui indique que la largeur de bande nécessaire correspond à deux fois la déviation maximale plus deux fois la fréquence maximale du signal modulant [24].

5. Signal GSM

Le GSM est un système de communication mobile basé sur une architecture cellulaire. Chaque zone géographique est couverte par une station de base (BTS) qui communique avec les téléphones mobiles via des ondes radio. Le réseau est conçu pour permettre à plusieurs utilisateurs de partager les mêmes ressources spectrales grâce à une organisation en fréquences et en temps.[26]

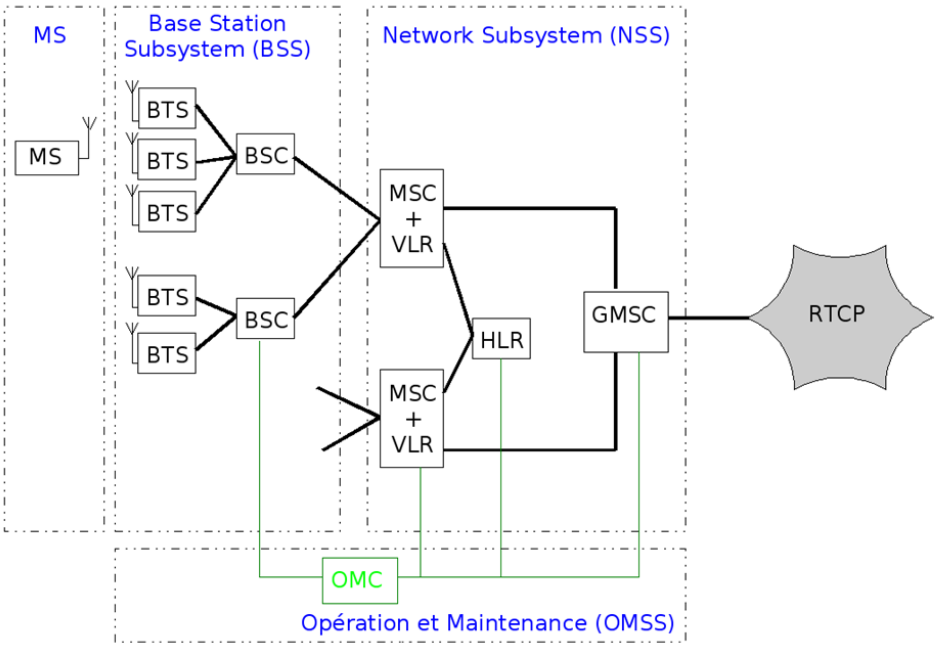


Figure 16 : Structure simplifiée des réseaux GSM

5.1 Bandes de fréquences GSM

La téléphonie mobile par GSM occupe deux bandes de fréquences aux alentours de 900 Mhz.

- De 890 à 915 pour la transmission des signaux des stations mobiles vers la station de base.
- De 935 à 960 pour la transmission inverse.

La bande passante de chaque direction est divisée en 124 canaux d'une largeur de 200 MHz. Ces canaux ne suffisent pas dans les grands canaux, il faut donc allouer une bande supplémentaire autour de 1800 Mhz. Ceci est un système DCS 1800 présente des caractéristiques similaires au GSM en termes de protocoles et de services. Ensuite, la communication montante se situe entre 1710 et 1785 la communication descendante se situe entre 1805 et 1880 [27].

Le tableau 1 montre les caractéristiques techniques du réseau GSM.

	GSM 900	**GSM 1800**
Bande spectrale-Canaux descendant	935 à 960 MHZ	1805 à 1880 MHZ
Bande spectrale-Canaux montant	890 à 915 MHZ	1710 à 1785 MHZ
Nombre de canaux	124	374
Largeur des canaux	200 KHZ	200 KHZ

Tableau 1:Caractéristiques techniques du réseau GSM.

La figure 17 montre la réception de du signal GSM avec RTL-SDR sur la fréquence 935 Mhz

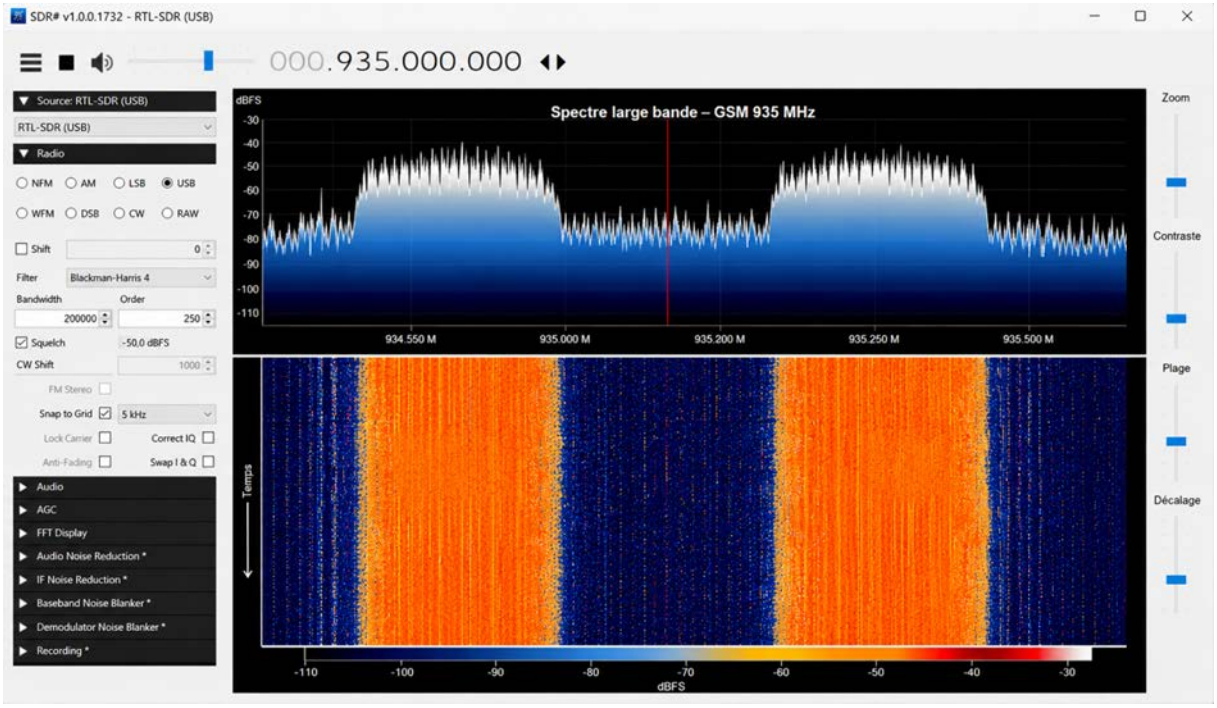


Figure 17 : Spectre de signal GSM sur la fréquence 935Mhz

6. Conclusion

L'analyse des signaux radio dans les systèmes de communication sans fil met en évidence la nécessité de techniques efficaces pour exploiter un spectre électromagnétique de plus en plus saturé. L'ensemble de la chaîne de traitement, de l'acquisition jusqu'à l'analyse spectrale, est essentiel pour assurer une utilisation optimale des ressources fréquentielles.

Dans ce chapitre, nous présentons une approche basée sur la radio logicielle (SDR) en utilisant le dispositif RTL-SDR associé à MATLAB pour l'acquisition et le traitement de signaux radio réels. Cette méthode permet une grande flexibilité dans l'étude des signaux sans nécessiter de modifications matérielles.

Chapitre 3 : Modélisation et Résultats Expérimentaux

1.Introduction

L'analyse des signaux radiofréquences s'impose aujourd'hui comme un enjeu majeur dans de nombreux domaines, notamment les télécommunications, la surveillance du spectre et les systèmes intelligents. La complexité et la variabilité de ces signaux nécessitent la mise en œuvre de chaînes de traitement performantes, capables de convertir des données brutes en informations exploitables pour des tâches d'interprétation et de prise de décision.

Dans ce contexte, l'utilisation de la radio logicielle (Software Defined Radio - SDR) constitue une solution particulièrement adaptée, offrant une grande flexibilité dans l'acquisition et la manipulation des signaux. Les signaux capturés sous forme de données IQ (In-phase et Quadrature) représentent une base riche mais volumineuse, nécessitant des traitements spécifiques afin d'en extraire des caractéristiques pertinentes.

Ainsi, une phase de prétraitement rigoureuse est mise en place, intégrant des opérations essentielles telles que la décimation pour la réduction du volume de données, la transformation de Fourier rapide (FFT) pour l'analyse fréquentielle, ainsi que des techniques d'extraction de caractéristiques permettant de représenter efficacement le contenu spectral. Ces étapes jouent un rôle déterminant dans l'amélioration de la qualité des données et dans la préparation d'un dataset structuré adapté aux algorithmes d'apprentissage automatique.

Par ailleurs, l'intégration de méthodes d'apprentissage supervisé permet d'automatiser la détection d'activité spectrale à partir des caractéristiques extraites. Plusieurs modèles de classification sont considérés, notamment Random Forest, K-Nearest Neighbors (KNN) et XGBoost, chacun présentant des avantages spécifiques en termes de précision, de robustesse et de complexité computationnelle. Une évaluation comparative basée sur des métriques standards, telles que l'accuracy, la précision et la matrice de confusion, permet d'analyser de manière approfondie leurs performances.

2. Enregistrement des données radio

2.1 Architecture du système d'acquisition

L'architecture du système d'acquisition constitue une étape fondamentale dans toute chaîne de traitement des signaux radio. Elle assure la capture, la numérisation et la structuration des signaux électromagnétiques sous une forme exploitable pour des analyses ultérieures. Dans le cadre de ce travail, cette architecture repose sur une approche de radio logicielle (SDR), permettant une grande flexibilité dans l'observation et le traitement du spectre radiofréquence.

L'ensemble du système suit une chaîne fonctionnelle composée de cinq blocs principaux montrés dans la figure 18.

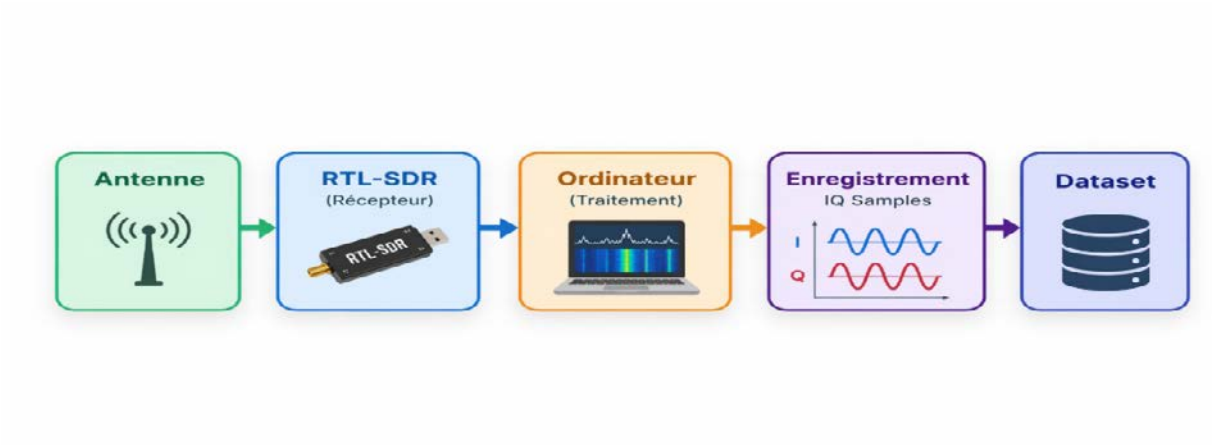


Figure 18 : Architecture du système d'acquisition

2.2 Procédure d'enregistrement

L'acquisition des signaux radio dans ce travail est réalisée à l'aide d'un récepteur **RTL-SDR** piloté par le logiciel **SDR# (SDRSharp)**. Ce dernier permet de visualiser le spectre en temps réel, de choisir la fréquence d'écoute et de régler les paramètres essentiels du récepteur avant l'enregistrement comme le montre la figure 19.

La première étape consiste à choisir la fréquence centrale du signal à capturer :

- Pour un **signal FM radio**, la bande utilisée est généralement comprise entre **88 MHz et 108 MHz**.
- Pour un **signal GSM**, les bandes principales sont :
    - **GSM 900 MHz** (ex : 935 MHz à 960 MHz pour la voie descendante).

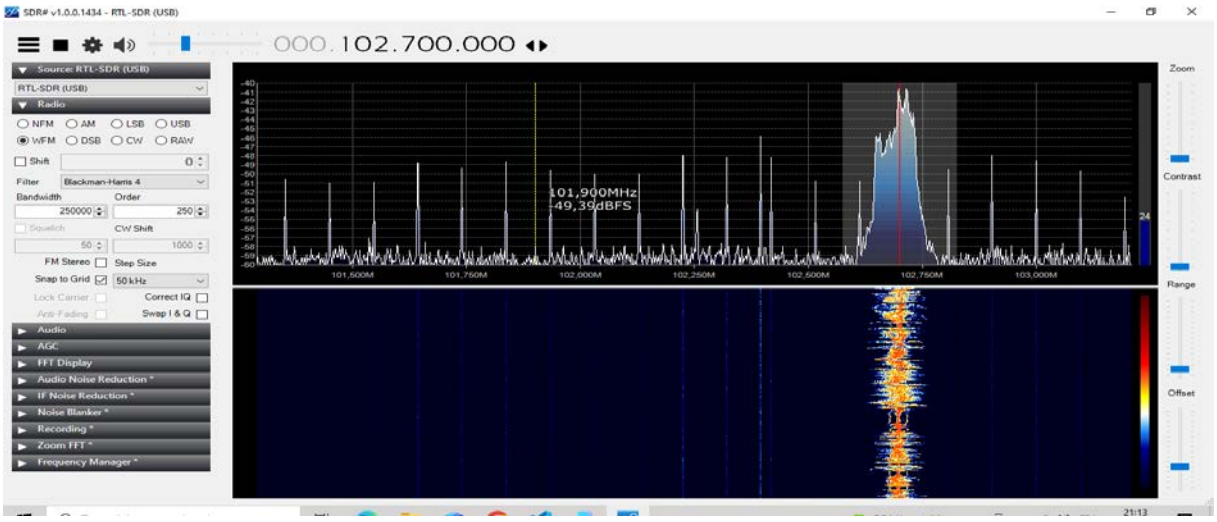


Figure 19 : Visualisation du spectre radiofréquence et du spectrogramme en bande FM à l'aide du logiciel SDRSharp (RTL-SDR)

Dans SDR#, la fréquence est réglée directement dans le champ “Frequency” en entrant la valeur souhaitée. Le spectre permet ensuite de vérifier la présence du signal (pic d’énergie visible).

L'acquisition des signaux radio constitue la deuxième étape du processus. Elle a été réalisée directement via l'interface en ligne de commande (CMD) à l'aide de l'outil **rtl_sdr**, qui permet de capturer les signaux sous forme de données brutes IQ (In-phase et Quadrature). Ce format de représentation du signal, fondamental en traitement du signal numérique, rendant les données exploitables pour les phases ultérieures de traitement et d'analyse.

La procédure d'acquisition a nécessité la configuration rigoureuse de plusieurs paramètres :

La fréquence centrale (f) a été sélectionnée en fonction de la nature du signal ciblé. Pour les signaux de radiodiffusion FM, la fréquence est choisie dans la bande 88–108 MHz (par exemple, $f = 100{,}7\ MHz$), tandis que pour les signaux GSM, elle est fixée dans les bandes 900 MHz (par exemple, $f = 947\ MHz$).

La fréquence d'échantillonnage (s) a été réglée à $2{,}4\ MS/s$ (mégaéchantillons par seconde), offrant une largeur de bande d'observation suffisante pour couvrir les canaux d'intérêt et capturer fidèlement les variations spectrales du signal.

Le gain du récepteur (g) a été ajusté en décibels (dB) de manière à amplifier convenablement le signal reçu tout en évitant la saturation du convertisseur analogique-numérique.

Le nombre d'échantillons (n) définit le volume total de données IQ enregistrées par session d'acquisition et détermine directement la durée de l'enregistrement.

La durée d'enregistrement (TT T) est déduite des paramètres précédents selon la relation :

$$T = n/s \qquad (5)$$

Le format des données adopté est le format binaire brut (extension **.dat**).

La figure 20 montre l’enregistrement des données via l'interface en ligne de commande (CMD) à l'aide de l'outil **rtl_sdr.**

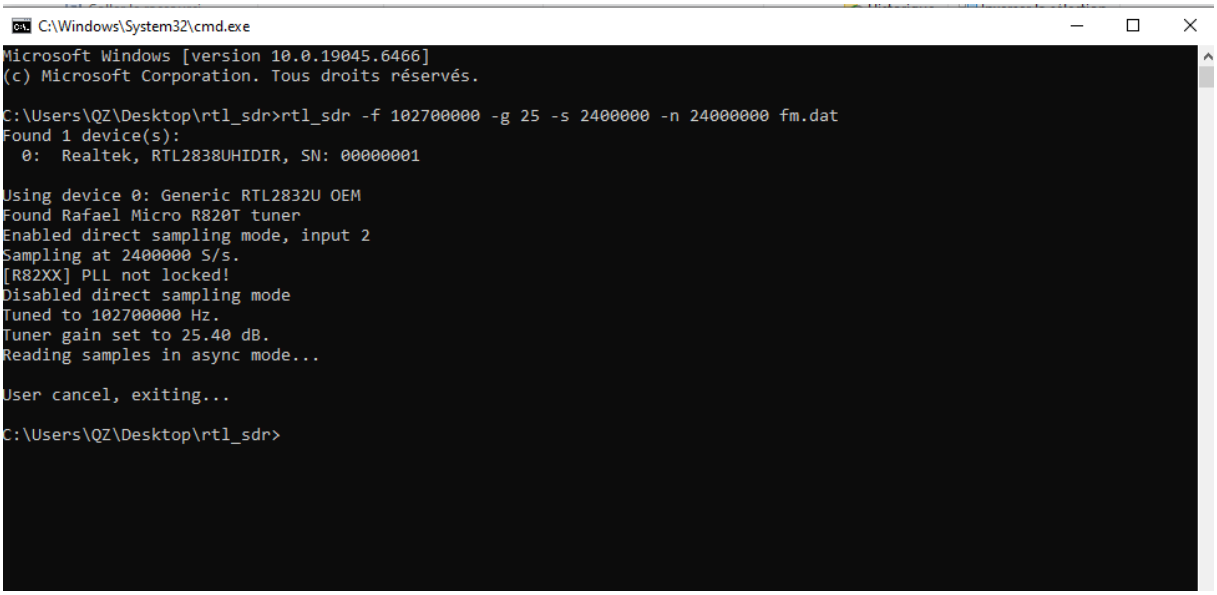


Figure 20: l'enregistrement des données à l'aide de l'outil rtl_sdr

2.3 Prétraitement des données

La phase de traitement des données a été réalisée à l'aide d'un script développé sous l'environnement MATLAB. Ce script assure la chaîne complète de traitement du signal, depuis le chargement des données brutes IQ jusqu'à l'extraction des caractéristiques spectrales et leur sauvegarde dans un fichier structuré destiné à l'apprentissage automatique. Les différentes étapes de ce traitement sont :

Chargement des données brutes : La première instruction du script procède au chargement du fichier binaire contenant les échantillons IQ préalablement enregistrés via l'outil rtl_sdr :

$$Y = loadFile('fm50.dat') \qquad (6)$$

La fonction $loadFile$ lit les données brutes et les convertit en un vecteur complexe de la forme :

$$y = I + jQ \qquad (7)$$

où I et Q désignent respectivement les composantes en phase et en quadrature du signal reçu.

Définition des paramètres de traitement : plusieurs paramètres sont définis afin de contrôler le traitement :

- **Fs = 2.4 MHz** : fréquence d'échantillonnage initiale du SDR
- **Fc = 99.2 MHz** : fréquence centrale du signal reçu
- **M = 8** : facteur de décimation permet de réduire la fréquence d'échantillonnage effective
- **Segment_duration = 1 s** : durée d'un segment
- **Num_segments = 10** : nombre de segments analysés

Le signal est donc découpé en plusieurs blocs temporels pour faciliter l'analyse locale.

Décimation du signal : La décimation est une étape essentielle du prétraitement. Elle consiste à réduire la fréquence d'échantillonnage tout en conservant les informations utiles du signal. Cette opération réduit la complexité computationnelle des étapes ultérieures tout en conservant l'information spectrale pertinente dans la bande d'intérêt. La figure 21 et 22 montre le spectre avant et après la décimation respectivement.

Calcul du spectre de puissance par transformée de Fourier rapide (FFT) : Le spectre du signal décimé est calculé par transformée de Fourier rapide (FFT) sur une fenêtre de $N_{fft} = 8192$ points.

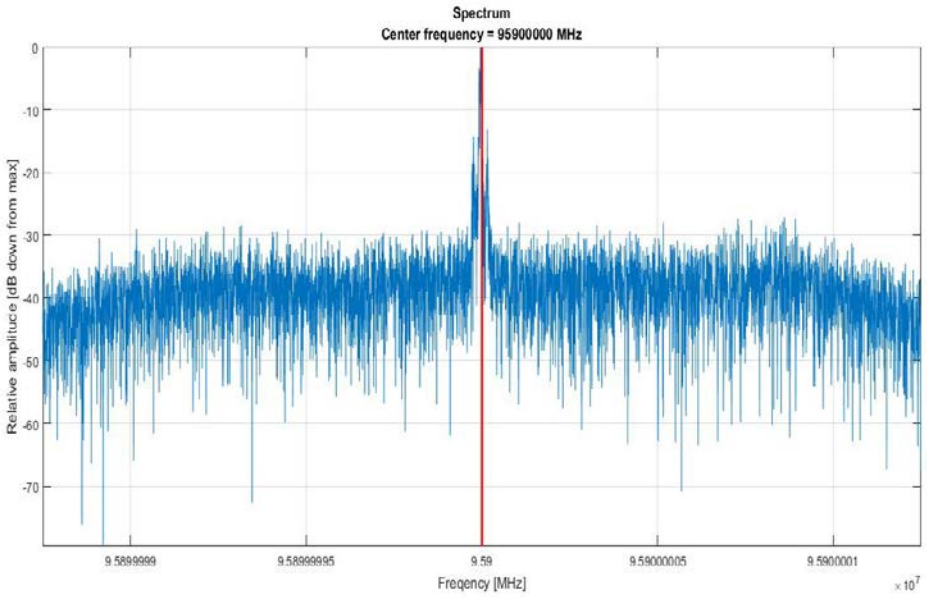


Figure 21: Spectre fréquentiel du signal avant décimation.

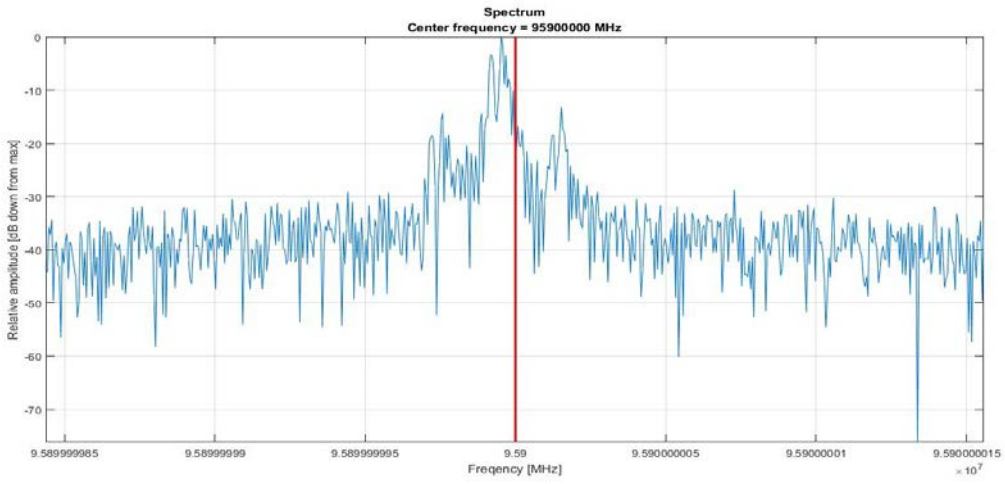


Figure 22: Spectre du signal obtenu après décimation (MATLAB)

Afin de réduire la dimensionnalité des données spectrales tout en conservant leur structure informative, un moyennage par fenêtres glissantes est appliqué sur le spectre calculé, La figure 23 montre le spectre après cette opération.

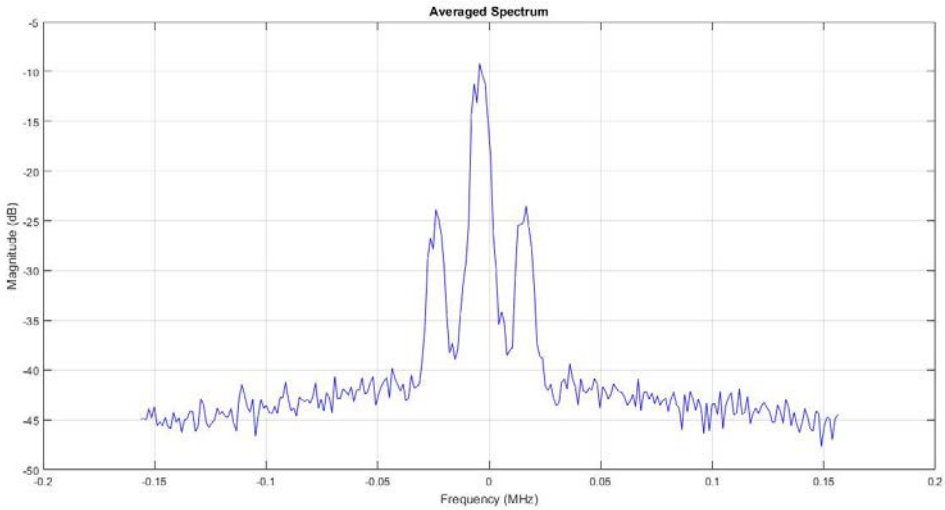


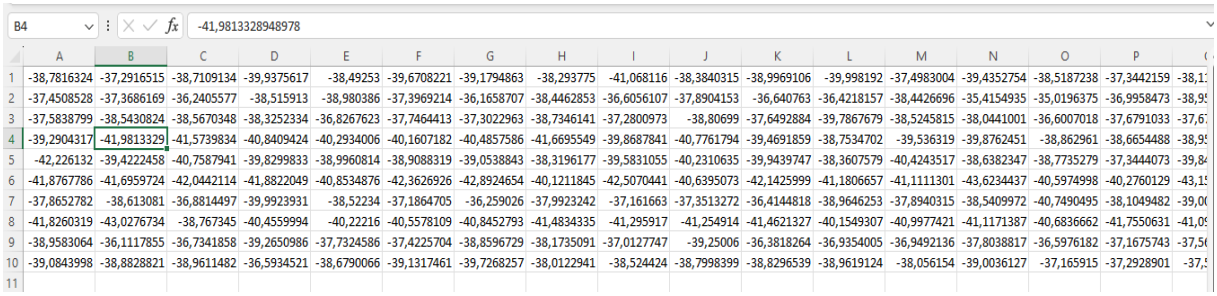
Figure 23:Spectre après le moyennage par fenêtre glissante

Construction du dataset : grâce à ce code MATLAB, nous avons pu générer une base de données structurée à partir des signaux radio bruts. En effet, après les étapes de prétraitement (décimation, FFT et extraction des caractéristiques), les informations pertinentes sont regroupées puis sauvegardées dans un fichier CSV. Cette base de données contient, pour chaque segment de signal, un ensemble de caractéristiques spectrales ainsi que son label associé comme le montre la figure 24.

Le schéma d'étiquetage adopté reflète l'état d'occupation spectrale du canal observé :

- **Étiquette 1** : signifie Activité, indique la présence d'un signal radio actif dans la bande analysée. Elle est assignée aux segments dans lesquels une émission est détectée,
- **Étiquette 0** : Absence d'activité : indique l'absence de signal, c'est-à-dire un canal libre ou occupé uniquement par du bruit de fond.

Cette convention d'étiquetage est essentielle pour les algorithmes d'apprentissage supervisé utilisés dans les phases ultérieures.



B4 : -41,9813328948978

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	-38,7816324	-37,2916515	-38,7109134	-39,9375617	-38,49253	-39,6708221	-39,1794863	-38,293775	-41,068116	-38,3840315	-38,9969106	-39,998192	-37,4983004	-39,4352754	-38,5187238	-37,3442159
2	-37,4508528	-37,3686169	-36,2405577	-38,515913	-38,980386	-37,3969214	-36,1658707	-38,4462853	-36,6056107	-37,8904153	-36,640763	-36,4218157	-38,4426696	-35,4154935	-35,0196375	-36,9958473
3	-37,5838799	-38,5430824	-38,5670348	-38,3252334	-36,8267623	-37,7464413	-37,3022963	-38,7346141	-37,2800973	-38,80699	-37,6492884	-39,7867679	-38,5245815	-38,0441001	-36,6007018	-37,6791033
4	-39,2904317	-41,9813329	-41,5739834	-40,8409424	-40,2934006	-40,1607182	-40,4857586	-41,6695549	-39,8687841	-40,7761794	-39,4691859	-38,7534702	-39,536319	-39,8762451	-38,862961	-38,6654488
5	-42,226132	-39,4222458	-40,7587941	-39,8299833	-38,9960814	-38,9088319	-39,0538843	-38,3196177	-39,5831055	-40,2310635	-39,9439747	-38,3607579	-40,4243517	-38,6382347	-38,7735279	-37,3444073
6	-41,8767786	-41,6959724	-42,0442114	-41,8822049	-40,8534876	-42,3626926	-42,8924654	-40,1211845	-42,5070441	-40,6395073	-42,1425999	-41,1806657	-41,1111301	-43,6234437	-40,5974998	-40,2760129
7	-37,8652782	-38,613081	-36,8814497	-39,9923931	-38,52234	-37,1864705	-36,259026	-37,9923242	-37,161663	-37,3513272	-36,4144818	-38,9646253	-37,8940315	-38,5409972	-40,7490495	-38,1049482
8	-41,8260319	-43,0276734	-38,767345	-40,4559994	-40,22216	-40,5578109	-40,8452793	-41,4834335	-41,295917	-41,254914	-41,4621327	-40,1549307	-40,9977421	-41,1171387	-40,6836662	-41,7550631
9	-38,9583064	-36,1117855	-36,7341858	-39,2650986	-37,7324586	-37,4225704	-38,8596729	-38,1735091	-37,0127747	-39,25006	-36,3818264	-36,9354005	-36,9492136	-37,8038817	-36,5976182	-37,1675743
10	-39,0843998	-38,8828821	-38,9611482	-36,5934521	-38,6790066	-39,1317461	-39,7268257	-38,0122941	-38,524424	-38,7998399	-38,8296539	-38,9619124	-38,056154	-39,0036127	-37,165915	-37,2928901
11																

Figure 24:Extrait du fichier Excel contenant les puissances du signal radio acquis

3. Méthodologie

Dans le cadre de ce travail, plusieurs algorithmes d'apprentissage supervisé ont été implémentés, entraînés et comparés au sein de l'environnement de développement Google Colab. Les modèles retenus pour cette étude comparative sont : la forêt aléatoire (Random Forest), l'algorithme des k plus proches voisins (K-Nearest Neighbors — KNN) et le gradient boosting extrême (XGBoost). Ces trois algorithmes ont été sélectionnés en raison de leurs performances reconnues dans la littérature pour les tâches de classification binaire, ainsi que de leur complémentarité en termes d'approches algorithmiques — méthode d'ensemble par agrégation, apprentissage par similarité et méthode d'ensemble par boosting séquentiel, respectivement.

L'objectif central de cette phase est d'identifier le modèle offrant les meilleures performances pour la tâche de classification binaire de l'activité spectrale, consistant à distinguer automatiquement deux états mutuellement exclusifs : la présence d'un signal actif (Activité, étiquette 1) et l'absence de signal (Absence d'activité, étiquette 0). La comparaison des modèles est effectuée sur la base de métriques d'évaluation standardisées, détaillées dans les sections suivantes.

La méthodologie globale repose sur trois étapes fondamentales :

- Normalisation des données
- Séparation des données en ensembles d’apprentissage et de test
- Évaluation des performances à l’aide de métriques standards

La base de données, constitué de 8460 enregistrements, a été partitionné en deux sous-ensembles distincts selon la répartition classiquement adoptée en apprentissage automatique :

- Ensemble d'entraînement (80% des données) : utilisé pour l'ajustement des paramètres des modèles.
- Ensemble de test (20% des données) : réservé exclusivement à l'évaluation des performances généralisées des modèles sur des données inédites.

Cette partition a été réalisée de manière aléatoire et stratifiée, de façon à préserver la distribution des classes dans chaque sous-ensemble, ce qui est particulièrement important dans le cas d'un jeu de données potentiellement déséquilibré. En Python, cette opération est effectuée via la

fonction $train\_test\_split$ de la bibliothèque scikit-learn. Le Listing 1 présente le code utilisé pour cette opération.

```python
from sklearn.model_selection import train_test_split
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Listing 1: Division des données en ensembles d'apprentissage et de test.

3.1 Entraînement des modèles

Les trois modèles ont été instanciés et entraînés sur l'ensemble d'entraînement selon les configurations suivantes.

Random Forest : Le modèle est configuré avec un nombre d'arbres $n\_estimators = 100$, une profondeur maximale non contrainte permettant aux arbres de se développer jusqu'à la pureté complète des feuilles, et une sélection aléatoire des caractéristiques à chaque nœud fixée à $\sqrt{p}$ où p est le nombre total de caractéristiques. Le Listing 2 présente le code utilisé pour cette opération.

```python
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

Listing 1 : création un modèle de forêt aléatoire.

K-Nearest Neighbors (KNN) : Le modèle KNN est configuré avec les hyperparamètres suivants :

- $K = 5$ Voisins : le nombre de voisins les plus proches pris en compte pour la décision de classification.
- Pondération par distance (weights="distance") : contrairement à la pondération uniforme, cette option attribue un poids inversement proportionnel à la distance à

chaque voisin, de sorte que les voisins les plus proches exercent une influence plus importante sur la décision finale.

- Métrique de Minkowski (metric="minkowski") : avec le paramètre par défaut p =2, cette métrique est équivalente à la distance euclidienne, assurant une mesure de similarité géométrique standard dans l'espace des caractéristiques.

Le Listing 3 présente le code utilisé pour créer un modèle KNN avec ces paramètres.

```python
from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(
    n_neighbors=5, weights="distance",
    metric="minkowski"
)

knn_model.fit(X_train_scaled, y_train)
```

Listing 2: création un modèle KNN

XGBoost : Le modèle est configuré avec un taux d'apprentissage $\eta = 0{,}1$, une profondeur maximale des arbres $max\_depth = 6$, et un nombre d'itérations $n\_estimators = 200$. Le Listing 4 présente le code utilisé pour créer un modèle XGBoost.

```python
from xgboost import XGBClassifier

clf = XGBClassifier(
    n_estimators=200,max_depth=6,learning_rate=0.1,subsample=0.8,
    colsample_bytree=0.8,random_state=42,use_label_encoder=False,
    eval_metric='logloss'
)

clf.fit(X_train, y_train)
```

Listing 3 : Création un modèle XGBoost.

3.2 Évaluation des performances

À l'issue de la phase d'entraînement, les trois modèles ont été évalués sur l'ensemble de test selon les métriques de classification définies précédemment. Les résultats obtenus sont analysés à partir des matrices de confusion et des indicateurs de performance calculés pour chaque modèle.

La matrice de confusion constitue un outil d'analyse détaillé permettant de quantifier la nature et la distribution des erreurs de classification commises par chaque modèle sur les deux classes cibles.

Comme le montre la figure 25, Le modèle Radom Forest commet un total de 2 erreurs sur 1692 observations : 1 faux positif et 1 faux négatif, démontrant une excellente capacité de discrimination entre les deux classes.

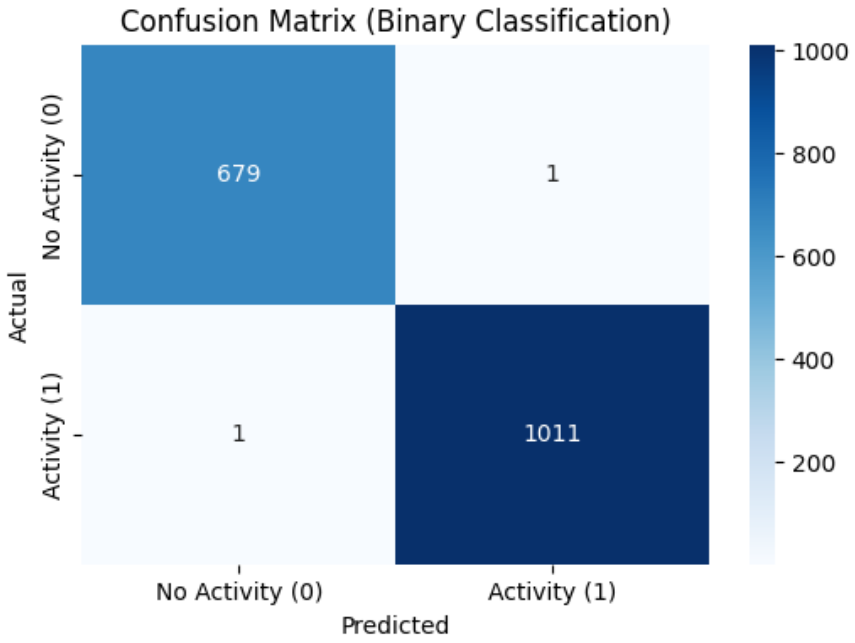


Figure 25: Matrice de confusion obtenue par la classification binaire avec l'algorithme Random Forest.

Le modèle KNN ne commet qu'1 seule erreur sur l'ensemble de test, soit 1 faux négatif, avec un taux de faux positifs nul. Ce résultat remarquable traduit une capacité de classification quasi-parfaite sur les données testées. La figure 26 montre la matrice de confusion pour KNN.

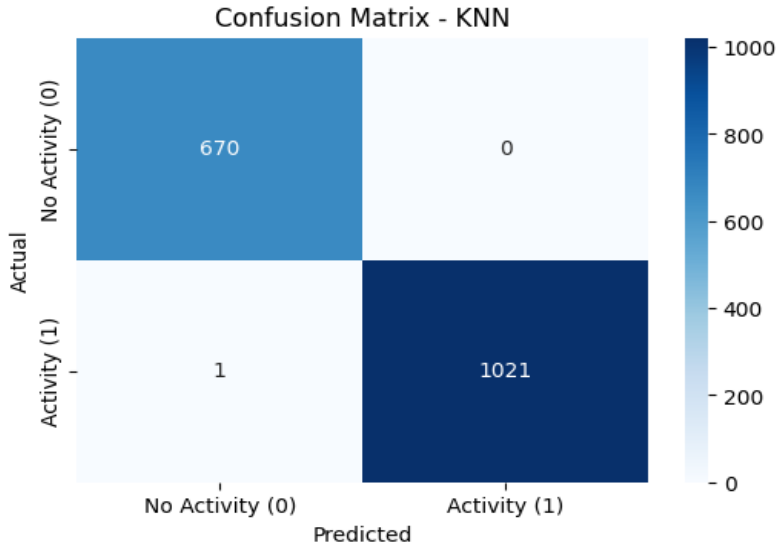


Figure 26:Matrice de confusion obtenue par la classification binaire avec l'algorithme KNN

Le modèle XGBoost commet 3 erreurs au total : 2 faux positifs et 1 faux négatif. Ces erreurs restent marginales au regard du volume total des observations. La figure 27 montre la matrice de confusion pour XGBoost.

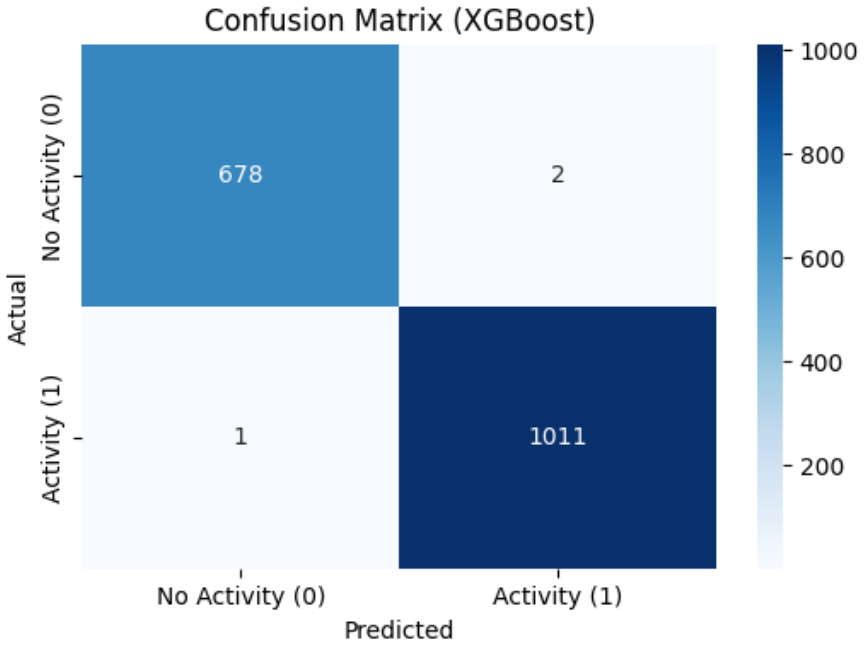


Figure 27: Matrice de confusion obtenue par la classification binaire avec l'algorithme XGBoost.

L'analyse comparative des trois modèles révèle des performances globalement très élevées, tous les modèles atteignant une précision supérieure à 99,8%, ce qui témoigne de la qualité de la

base de données construit et de la pertinence des caractéristiques spectrales extraites pour la tâche de classification de l'activité radio.

Le modèle KNN se distingue comme le plus performant avec une accuracy de 99,94%, surpassant légèrement Random Forest (99,88%) et XGBoost (99,82). Le tableau 2 montre une comparaison entre les trois Modèles.

Modèle	**Accuracy**	**Sensibilité à la normalisation**	**Avantage principal**	**Performance globale**
Random Forest	99.88 %	Faible	Robustesse + stabilité	Très élevée
KNN	99.94 %	Très élevée (obligatoire)	Simplicité + efficacité sur petits jeux de données	Excellente
XGBoost	99.82 %	Faible	Très puissant pour données complexes	Très élevée

Tableau 2:Comparaison entre les trois modèles

4. Résultats expérimentaux et Discussion

Dans cette section, une validation complémentaire des modèles développés est effectuée à l'aide d'une nouvelle base de données, Cette étape constitue une phase essentielle dans tout processus de modélisation supervisée, dans la mesure où elle permet d'évaluer la capacité de généralisation des algorithmes au-delà du cadre expérimental initial. En effet, contrairement à la validation interne réalisée sur une partition issue de la même distribution statistique, l'évaluation externe simule des conditions proches d'un déploiement opérationnel, où les caractéristiques des signaux peuvent présenter des variations significatives.

L'objectif principal de cette phase est de mesurer la robustesse des modèles face à des données issues d'un environnement potentiellement hétérogène. Ces variations peuvent être attribuées à plusieurs facteurs, notamment :

- Les fluctuations du niveau de bruit thermique et environnemental,
- Les variations de puissance des signaux reçus,
- Les conditions de propagation radio (multi-trajets, atténuation),
- Les différences potentielles dans la chaîne d'acquisition.

4.1 Protocole expérimental

La nouvelle basse de donnes de test a été soumis à la même chaîne de prétraitement que celle appliquée aux données d'entraînement, garantissant ainsi la cohérence des descripteurs extraits. Les modèles préalablement entraînés (Random Forest, KNN et XGBoost) ont ensuite été appliqués directement sur ces données sans réajustement des paramètres, afin de garantir une évaluation strictement orientée vers la généralisation.

4.2 Résultats expérimentaux sur les données externes

Le modèle Random Forest conserve une performance stable, avec une augmentation marginale du taux d'erreur, avec 14 erreurs dont 12 faux positifs sur 3489 observations. Cette robustesse s'explique par la nature même de l'algorithme, basé sur une agrégation d'arbres de décision indépendants, ce qui lui confère une bonne résistance au bruit et aux variations des données d'entrée.

La figure 28 présente la matrice de confusion obtenue sur l'ensemble de test externe. On observe une répartition des prédictions correcte avec un faible nombre d'erreurs, confirmant la robustesse du modèle en conditions réelles.

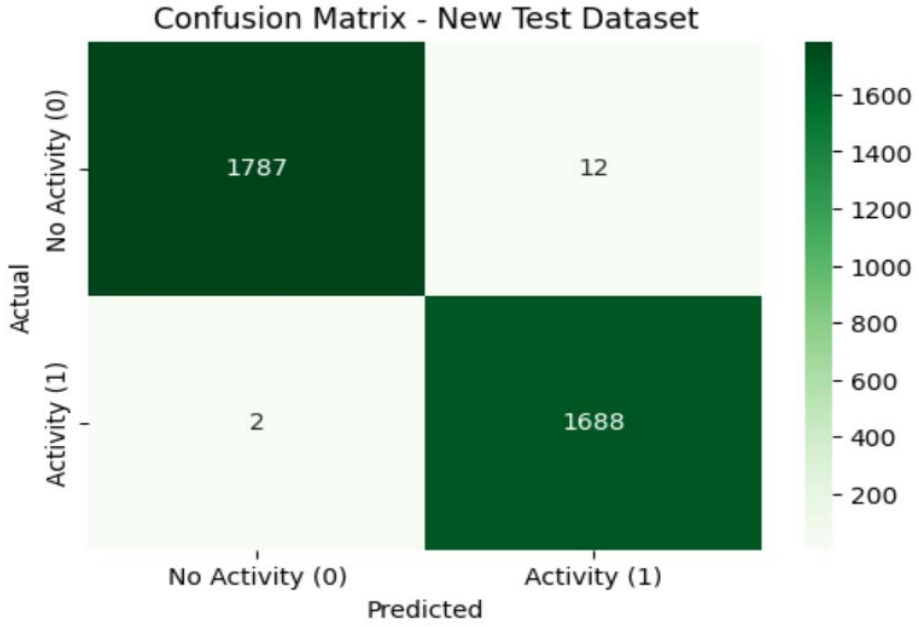


Figure 28:Matrice de confusion obtenue par la classification binaire avec l'algorithme Random Forest sur la nouvelle base de test.

Le modèle KNN ne commet que 2 erreurs sur 3489 observations, 1 faux positif et 1 faux négatif, soit un taux d'erreur global de 0,057% comme le montre la figure 29.

Le modèle KNN maintient une accuracy de 99,94% sur un volume de données presque doublé par rapport au test interne, démontrant une remarquable stabilité et une absence de dégradation des performances avec l'augmentation du volume d'observations.

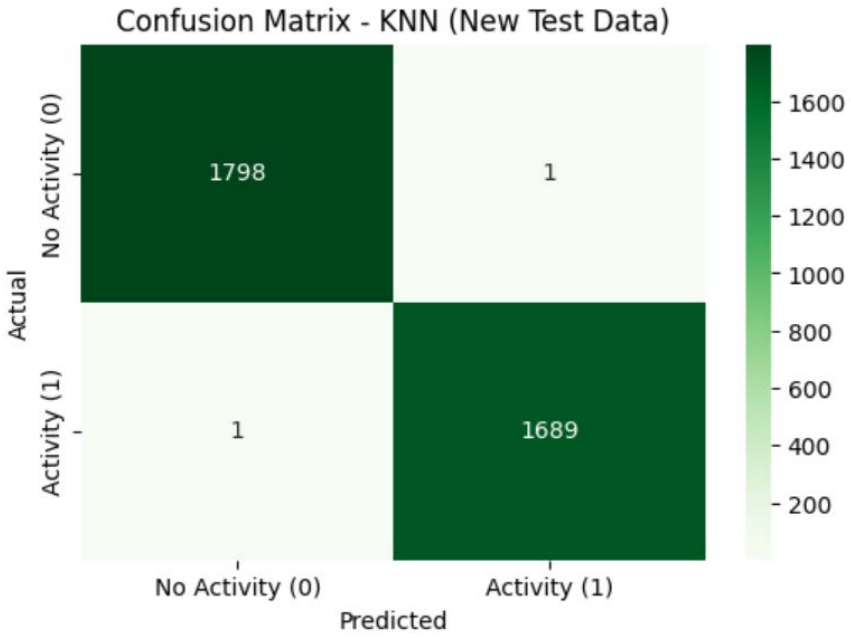


Figure 29: Matrice de confusion obtenue par la classification binaire avec l'algorithme KNN sur la nouvelle base de test.

Le modèle XGBoost maintient des performances élevées, bien que légèrement inférieures à celles observées en validation interne, avec 22 erreurs réparties entre 14 faux positifs et 8 faux négatifs comme le montre la figure 30. Cette légère baisse peut être attribuée à la complexité des structures apprises, qui peuvent être partiellement sensibles à des variations non représentées lors de l'apprentissage.

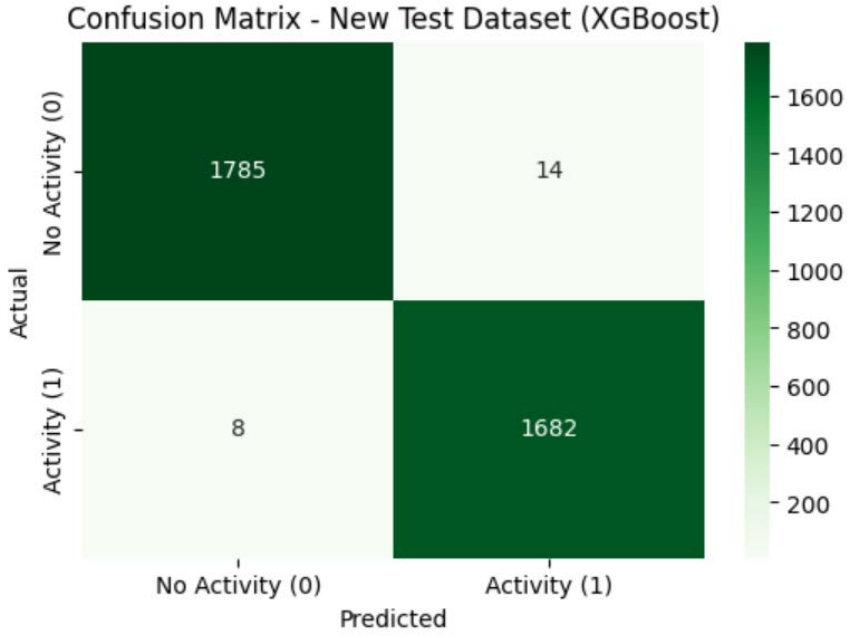


Figure 30: Matrice de confusion obtenue par la classification binaire avec l'algorithme XGBoost sur la nouvelle base de test.

Dans l'ensemble, ces résultats confirment que les trois modèles conservent une précision supérieure à 99 %, validant la qualité du prétraitement spectral et la pertinence des descripteurs extraits.

Les résultats obtenus confirment la validité de la chaîne de traitement proposée, depuis l'acquisition des signaux RF jusqu'à leur classification par apprentissage supervisé. L'ensemble du processus démontre une capacité significative à transformer des données brutes complexes en représentations discriminantes exploitables pour la détection automatique d'activité spectrale.

5. Conclusion

Les résultats obtenus mettent en évidence l'efficacité de la chaîne de traitement proposée, depuis l'acquisition des signaux jusqu'à leur classification automatique. Les modèles étudiés présentent des performances très élevées avec un faible taux d'erreur.

Dans ce chapitre, l'analyse comparative a permis d'identifier les spécificités de chaque algorithme, KNN se distingue par sa précision par rapport Random Forest et XGBoost.

Le modèle KNN se distingue comme la solution la plus robuste et la plus fiable pour la tâche de détection automatique de l'activité spectrale radio, avec les meilleures performances aussi bien en test interne qu'en conditions réelles de déploiement.

Par ailleurs, l'évaluation réalisée sur une base de données externe a permis de confirmer la capacité de généralisation des modèles, tout en mettant en évidence une légère dégradation des performances, attendue dans un contexte hors distribution. Ce résultat souligne l'importance d'intégrer des données variées lors de la phase d'apprentissage afin d'améliorer la robustesse des modèles.

Chapitre 4 : Développement de l'Application Web pour La Détection d'Activité Spectrale

1. Introduction

Après les phases d'étude théorique, ce chapitre est consacré à la réalisation pratique de l'application web développée dans le cadre de ce projet. Cette étape représente la phase d'implémentation du système proposé, en mettant en œuvre les différentes méthodes étudiées afin de concevoir une plateforme complète dédiée à l'analyse et à la détection automatique d'activité spectrale.

L'objectif principal de cette réalisation est de développer une application capable d'assurer l'acquisition, le traitement et d'activité spectrale des signaux RF, soit à partir d'un récepteur Software Defined Radio de type RTL-SDR, soit à partir de fichiers de données au format CSV. Le système conçu permet ainsi d'effectuer une analyse spectrale automatisée tout en offrant une interface utilisateur intuitive facilitant l'exploitation des résultats obtenus.

L'architecture adoptée est basée sur une séparation entre la partie backend et la partie frontend. Le backend, développé en Python, assure les opérations liées au traitement des données, au prétraitement des signaux, à l'extraction des caractéristiques ainsi qu'à l'exécution des modèles de classification. Le frontend, développé avec ReactJS, fournit une interface graphique interactive permettant à l'utilisateur de visualiser les données, de lancer les analyses et d'interagir avec les différentes fonctionnalités de l'application.

Dans ce chapitre, seront présentés de manière détaillée l'architecture globale de l'application, l'environnement et les outils de développement utilisés, la réalisation des différentes composantes du backend et du frontend, ainsi que les étapes de test et de validation permettant d'évaluer les performances et le bon fonctionnement du système développé.

2. Architecture Globale du Système

L'application réalisée repose sur une architecture modulaire de type client–serveur permettant une séparation claire entre les différentes composantes logicielles. Cette approche améliore considérablement la maintenabilité, la réutilisabilité ainsi que l'évolutivité du système. L'architecture générale est composée principalement de deux couches :

- Une couche backend dédiée aux traitements
- Une couche frontend dédiée à l'interaction utilisateur

La communication entre ces deux composantes est assurée via des interfaces de programmation API basées sur le protocole HTTP.

2.1 Architecture Backend

Le backend constitue le noyau fonctionnel de l'application. Il assure l'ensemble des opérations relatives au traitement des données radiofréquences ainsi qu'à l'exécution des modèles de Machine Learning. Les principales fonctionnalités implémentées au niveau du backend sont :

- L'acquisition des signaux RF depuis le récepteur RTL-SDR
- Le chargement et le traitement des fichiers CSV

- Le prétraitement des données spectrales
- L'extraction des caractéristiques
- L'exécution du modèle de classification (Activité ou Absence d'activité)
- La génération des résultats de prédiction

Le choix du langage Python pour le développement du backend est motivé par la richesse de son environnement scientifique ainsi que par la disponibilité de nombreuses bibliothèques dédiées au traitement numérique du signal et à l'apprentissage automatique.

2.2 Architecture Frontend

Le frontend représente l'interface graphique de l'application et constitue le point d'interaction principal avec l'utilisateur. Cette partie du système permet :

- La configuration des paramètres SDR (fréquence, gain, etc.)
- Le lancement des acquisitions
- L'importation des fichiers CSV
- L'affichage des résultats de classification
- La visualisation des paramètres et statistiques.

3. Fonctionnement Global du Système

Le fonctionnement du système repose sur une chaîne de traitement séquentielle et cohérente, orchestrant l'ensemble des composants logiciels depuis l'acquisition des données jusqu'à la restitution des résultats à l'utilisateur. Cette chaîne se déroule selon les étapes suivantes.

En premier lieu, l'utilisateur interagit avec l'interface graphique afin de configurer les paramètres de capture du signal radio ou d'importer directement un fichier de données au format CSV contenant les caractéristiques spectrales préalablement extraites. Les données saisies ou importées sont ensuite transmises au backend par le biais de requêtes HTTP conformes à l'architecture REST, assurant une communication structurée et standardisée entre les deux couches applicatives.

À la réception des données, le backend procède aux opérations de prétraitement nécessaires, incluant la normalisation des caractéristiques à l'aide du normalisateur StandardScaler sauvegardé lors de la phase d'entraînement, garantissant ainsi la cohérence avec les données sur lesquelles le modèle a été calibré. Les caractéristiques spectrales sont ensuite extraites et structurées sous la forme d'un vecteur conforme au format d'entrée attendu par le modèle. Ce vecteur est alors soumis au modèle Random Forest chargé en mémoire, qui effectue l'inférence et retourne la décision de classification binaire — Activité (1) ou Absence d'activité (0) — accompagnée des probabilités associées à chaque classe.

Enfin, les résultats produits par le modèle sont transmis au frontend sous forme de réponse JSON, où ils sont interprétés et affichés dynamiquement sur l'interface utilisateur, permettant

une visualisation immédiate et intuitive de l'état d'occupation spectrale détecté. Cette architecture en couches distinctes est montré dans la figure 31.

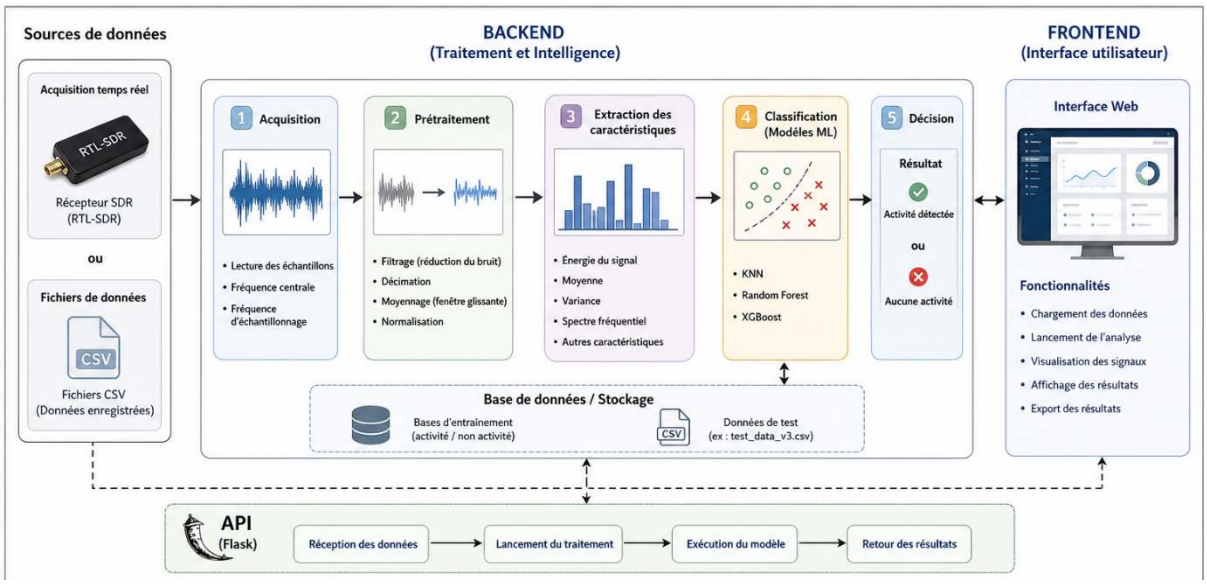


Figure 31:Architecture globale du système

4. Outils de Développement

La réalisation de l'application web de détection d'activité spectrale a nécessité la mobilisation d'un ensemble de technologies logicielles, de bibliothèques scientifiques et d'outils de développement soigneusement sélectionnés en fonction des exigences techniques du projet.

4.1 Python

Python constitue le langage principal du backend de l'application. Son utilisation couvre l'ensemble des traitements côté serveur, à savoir le traitement numérique du signal radio, l'analyse et la manipulation des données spectrales, l'implémentation et l'exploitation des modèles d'apprentissage automatique, ainsi que la communication avec le périphérique RTL-SDR pour l'acquisition des signaux radio.

Le choix de Python comme langage de développement backend se justifie par plusieurs avantages. Sa syntaxe claire et expressive favorise la lisibilité et la maintenabilité du code, tandis que sa flexibilité permet de couvrir des domaines applicatifs variés au sein d'un même projet, allant du traitement du signal au déploiement d'API web. Par ailleurs, la richesse et la maturité de son écosystème de bibliothèques scientifiques — notamment NumPy, SciPy, pandas et scikit-learn — en font un environnement particulièrement adapté aux projets à forte composante de traitement de données et d'apprentissage automatique, réduisant significativement le temps de développement tout en garantissant des performances computationnelles satisfaisantes. [28]

4.2 JavaScript / ReactJS

JavaScript, au travers de la bibliothèque ReactJS, a été retenu comme technologie de développement du frontend de l'application. ReactJS est une bibliothèque open-source développée et maintenue par Meta (anciennement Facebook), largement adoptée dans l'industrie pour la conception d'interfaces utilisateur modernes, réactives et performantes [29].

Dans le cadre de ce projet, ReactJS assure l'ensemble des responsabilités de la couche de présentation, notamment :

- La conception et le rendu de l'interface graphique utilisateur ;
- La gestion des interactions et des événements utilisateur ;
- La communication asynchrone avec le backend via des requêtes HTTP ;
- L'affichage dynamique et en temps réel des résultats de classification retournés par le modèle.

5. Présentation de l'interface utilisateur

L'application a été créée pour être facile à utiliser. Elle est organisée de manière logique et intuitive, ce qui signifie que les paramètres sont bien classés et que l'on peut facilement interagir avec le système. Les résultats sont également présentés de manière claire et facile à comprendre. L'application comporte deux principales parties : une pour capter des signaux en direct avec RTL-SDR et une autre pour importer et analyser des fichiers CSV.

5.1 Interface de capture en temps réel (Live Capture RTL-SDR)

La première interface, illustrée dans la Figure 32, est dédiée à l'acquisition et à l'analyse des signaux radiofréquences en temps réel à partir de la clé RTL-SDR. Elle se compose d'une zone de configuration des paramètres d'acquisition et de configuration de la clé, d'un bouton de lancement de la capture, et assure une communication temps réel avec le backend Python via des requêtes HTTP asynchrones.

Les paramètres configurables par l'utilisateur sont les suivants :

- ✓ Center Frequency : Fréquence de réception
- ✓ Sample Rate : Fréquence d'échantillonnage
- ✓ Gain : Gain du récepteur SDR
- ✓ Segment Duration : Durée de chaque capture
- ✓ Segments : Nombre total de segments acquis
- ✓ FFT Size : Taille utilisée pour la Transformée de Fourier Rapide
- ✓ Decimation Factor : Facteur de réduction de la fréquence d'échantillonnage (décimation)

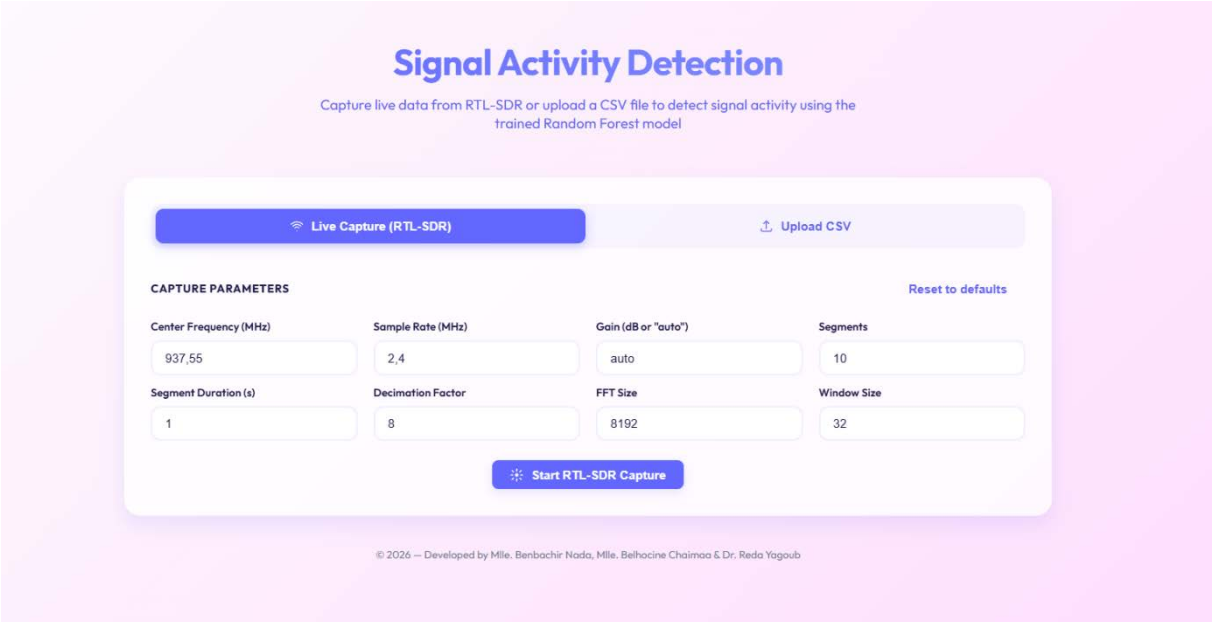


Figure 32:interface Live Capturer des signaux RTL-SDR

Une fois que les paramètres sont configurés, l'utilisateur peut lancer l'acquisition. Il clique sur le bouton « Start RTL-SDR Capture ». Cela déclenche tout le processus de traitement. Le processus comprend l'acquisition IQ, la décimation, le calcul FFT, l'extraction des caractéristiques spectrales, l'inférence par le modèle Random Forest et enfin l'affichage des résultats.

5.2 Interface d'importation et d'analyse de fichiers CSV

La deuxième interface, que l'on voit dans la figure 33, sert à examiner des données spectrales qui ont déjà été enregistrées et organisées dans des fichiers CSV. Cette interface possède une zone où vous pouvez glisser et déposer des fichiers, ce qui signifie que vous pouvez directement faire glisser le fichier de données vers l'emplacement prévu, ou bien le sélectionner manuellement en utilisant le navigateur de fichiers du système.

Le format attendu par le système est le suivant : chaque ligne du fichier CSV doit contenir les valeurs spectrales des caractéristiques (Power1; Power2; ...; Power512), avec une colonne optionnelle Label permettant d'évaluer les prédictions du modèle par rapport aux étiquettes réelles.

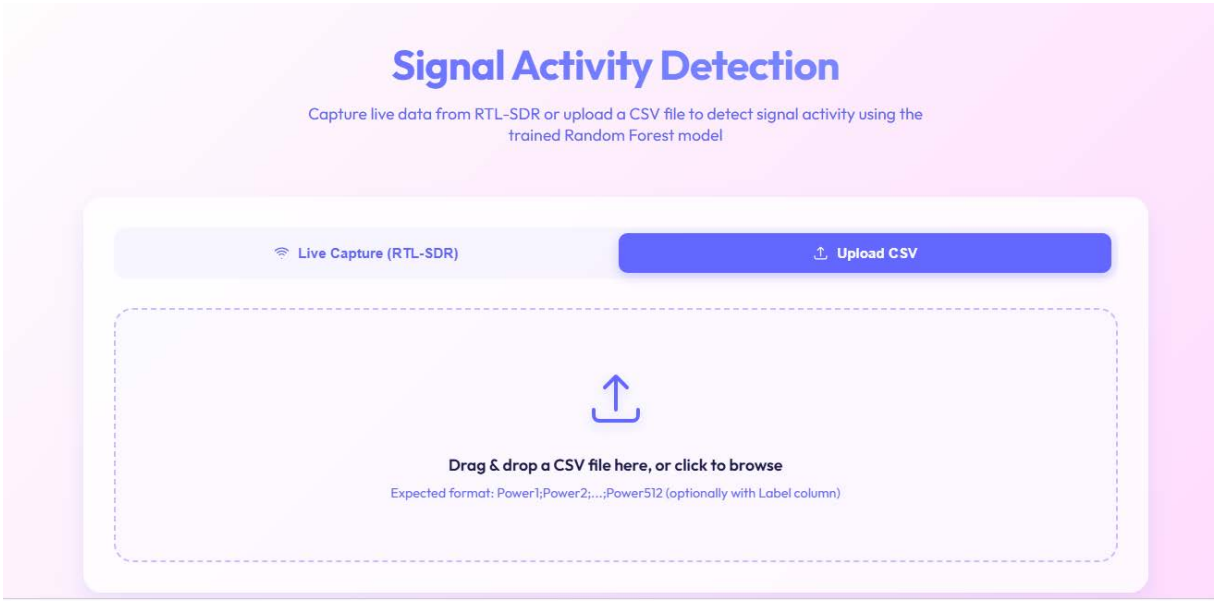


Figure 33:interface de d'importation de fichier CSV

5.3 Tests et Validation de l'Application

Plusieurs tests ont été réalisés afin d'évaluer les performances, la stabilité et le bon fonctionnement du système développé.

Les résultats obtenus démontrent une bonne stabilité de l'application. La figure 34 montre l'analyse de 10 segments temporels d'une durée d'une seconde chacun, acquis en temps réel via le périphérique RTL-SDR sur la fréquence centrale de 947,8 MHz, appartenant à la bande GSM-900.

L'interface de résultats comporte :

- Total Samples : 10 — le nombre total de segments analysés au cours de la session ;
- Activity (1) : 10 — le nombre de segments classifiés comme actifs par le modèle, affiché en vert pour signaler la présence d'une émission radio ;
- No Activity (0) : 0 — le nombre de segments classifiés comme inactifs, affiché en orange, indiquant l'absence d'une émission radio.

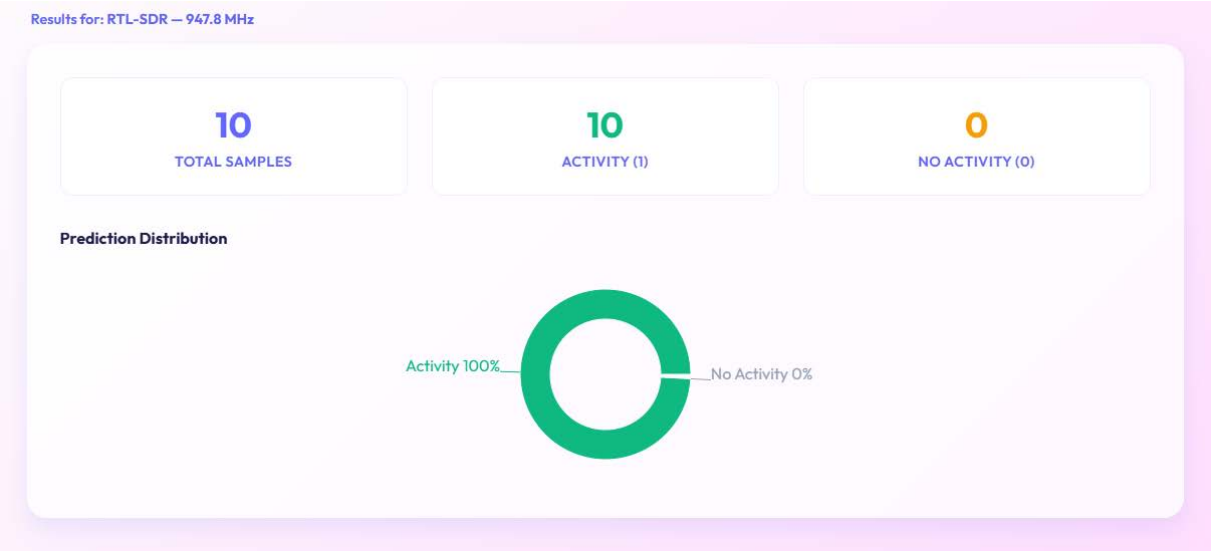


Figure 34:tableau de bord des résultats de classification des signaux

En complément des indicateurs de classification, l'application intègre un module de visualisation spectrale interactive, illustré dans la figure 35. Ce composant graphique offre une représentation du spectre de puissance du signal acquis, permettant à l'utilisateur d'observer directement la structure fréquentielle du signal analysé et de corroborer visuellement les décisions de classification retournées par le modèle.

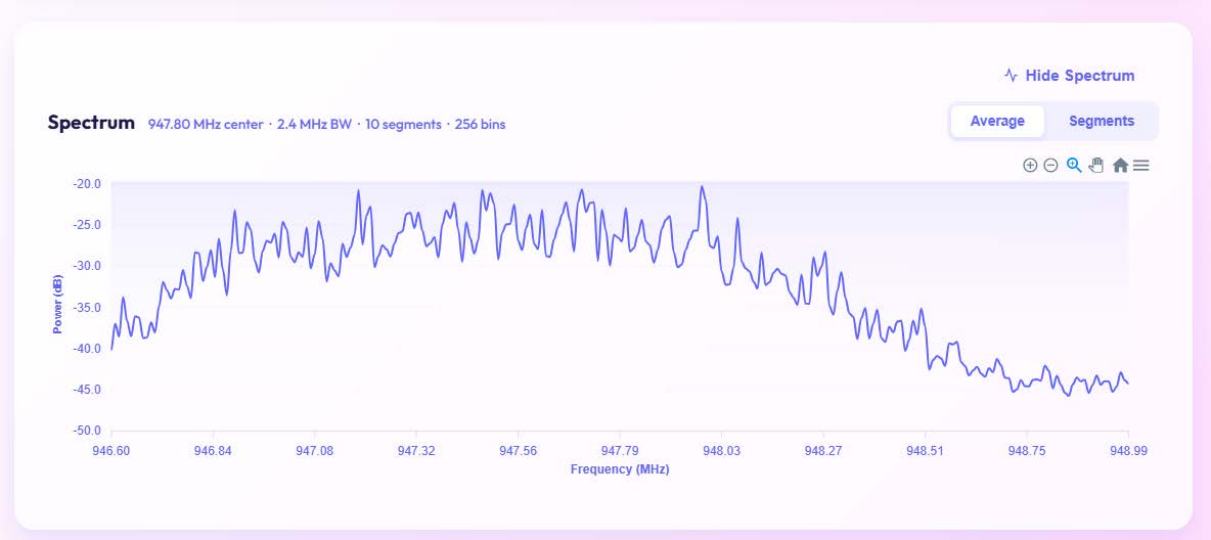


Figure 35:Spectre de signal GSM sur la fréquence 947.8Mhz.

Afin de valider le fonctionnement de l'interface d'importation CSV, un fichier de test contenant 7 segments des signaux différents a été soumis au système, 3 segments actifs (Activity = 1) et 4 segments inactifs (No Activity = 0). La figure 36 montre l'analyse du fichier CSV, L'accuracy de 100% confirme que le modèle Random Forest a correctement classifié l'ensemble des 7 segments sans commettre la moindre erreur. L'application donne aussi le tableau de ***Prediction***

Results, qui détaille segment par segment la prédiction du modèle, l'étiquette réelle (True Label) et le score de confiance (Confidence) associé à chaque décision.

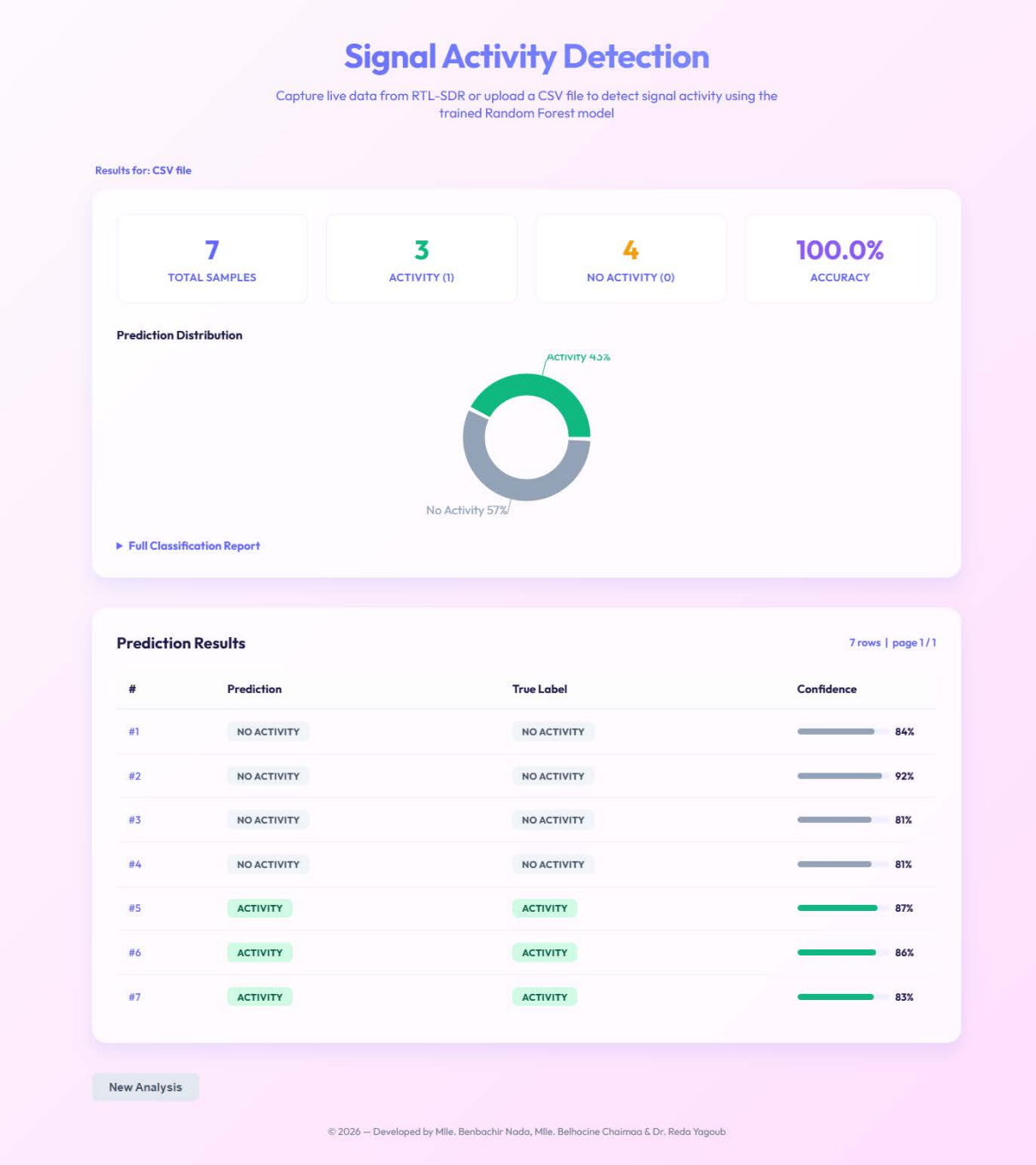


Figure 36:interface d'analyse des fichier CSV et des résultats de prédiction

6. Perspectives et travaux futurs

Bien que l'application développée dans le cadre de ce projet représente une solution efficace et opérationnelle pour la détection automatique de l'activité spectrale, plusieurs axes

d'amélioration et d'élargissement ont été repérées, ouvrant des perspectives de recherche et de développement prometteuses.

Intégration multi-modèles : actuellement, l'application s'appuie uniquement sur le modèle Random Forest pour classer l'activité spectrale. Une extension naturelle implique d'incorporer les deux autres modèles analysés dans ce travail, à savoir le K-Nearest Neighbors (KNN) et le XGBoost, pour donner à l'utilisateur la possibilité de choisir dynamiquement le modèle de classification désiré depuis l'interface graphique. Cette combinaison de plusieurs modèles donne la possible de faire une comparaison instantanée des décisions générées par les divers algorithmes sur un même signal, simplifiant l'analyse comparative et la validation croisée des résultats.

Compatibilité avec d'autres récepteurs SDR : L'application développée repose actuellement sur le périphérique RTL-SDR comme unique source d'acquisition du signal radio. Bien que ce récepteur offre un excellent rapport qualité-prix pour les applications d'observation spectrale à faible coût, ses caractéristiques techniques — notamment sa plage de fréquences limitée (24 MHz – 1,7 GHz), sa résolution en bits et sa sensibilité — peuvent constituer des contraintes pour certaines applications avancées de surveillance spectrale. L'extension de la compatibilité de l'application à d'autres récepteurs SDR de gamme supérieure représente ainsi un axe de développement majeur tel que : USRP, BLADE RF, HACK RF, etc.

Figure 37 : USRP-2943R

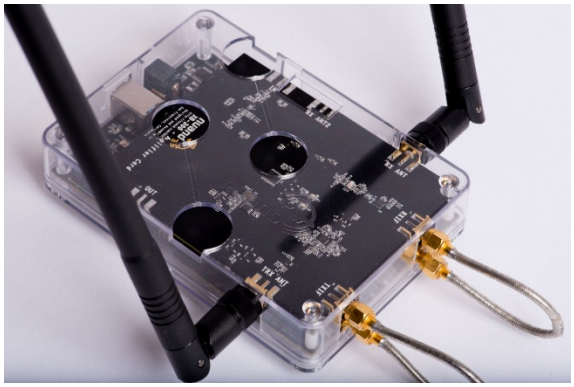


Figure 38 :LimeSDR avec antennes RF

7. Conclusion

Ce chapitre a présenté la mise en œuvre complète de l'application web dédiée à la détection d'activité spectrale, développée dans le cadre de ce projet.

L'intégration conjointe des technologies SDR, des méthodes de traitement numérique du signal, des algorithmes de Machine Learning et des technologies web modernes a permis la conception d'une plateforme robuste, performante et adaptée à l'analyse automatique des signaux radiofréquences.

L'architecture modulaire adoptée constitue un choix pertinent, offrant plusieurs avantages majeurs, notamment en termes de maintenabilité, d'évolutivité et de flexibilité du système. Elle facilite également l'intégration de nouveaux modules fonctionnels ainsi que l'amélioration continue des modèles de classification utilisés.

Les tests et validations réalisés ont confirmé la fiabilité de l'application et sa capacité à répondre efficacement aux objectifs fixés. Ainsi, la solution proposée représente un outil pertinent pour la détection et l'analyse des activités spectrales dans des environnements radiofréquences complexes et dynamique.

CONCLUSION GENERALE

L’évolution rapide des systèmes de communication sans fil a conduit à l’émergence de nouvelles approches basées sur la flexibilité, la reconfigurabilité et l’intelligence des systèmes d’analyse. Dans ce contexte, les technologies de Software Defined Radio (SDR), notamment RTL-SDR, associées aux techniques de traitement numérique du signal et aux méthodes d’intelligence artificielle, constituent une solution moderne et performante pour l’analyse des environnements radiofréquences complexes.

L’utilisation de la représentation IQ (In-phase / Quadrature) et des techniques de transformation fréquentielle telles que la Fast Fourier Transform (FFT) permet d’extraire des caractéristiques pertinentes décrivant le comportement spectral des signaux. Ces informations constituent une base essentielle pour l’application des algorithmes de Machine Learning, qui améliorent les capacités de classification et de détection automatique des activités spectrales.

L’intégration des modèles d’apprentissage supervisé tels que Random Forest, XGBoost et K-Nearest Neighbors (KNN) démontre l’efficacité des approches basées sur les données pour traiter des signaux bruités et dynamiques. Cette combinaison entre SDR, traitement du signal et intelligence artificielle permet de développer des systèmes capables d’analyser et d’interpréter les signaux radio de manière plus autonome, précise et rapide.

Ainsi, les résultats obtenus mettent en évidence le potentiel important de ces technologies dans le domaine des télécommunications et de la surveillance du spectre, où la demande en solutions intelligentes et adaptatives ne cesse de croître. Cette approche ouvre également la voie à de futures améliorations, notamment l’intégration de modèles plus avancés d’intelligence artificielle et l’optimisation des performances en temps réel

BIBLIOGRAPHIE

[1] S. Berbra , K. Cherfeddine, « L'usage de l'intelligence artificielle dans l'enseignement-apprentissage du FLE », Mémoire de Master, Université Kasdi Merbah Ouargla, Ouargla, Algérie, 2024.

[2] S. Legg et M. Hutter, « Universal Intelligence: A Definition of Machine Intelligence », *arXiv preprint* arXiv:0712.3329, 2007.

[3] F. Pedregosa *et al.*, « Scikit-learn: Machine Learning in Python », *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2e éd., O'Reilly Media, 2019.

[5] M. N. U. Alam *et al.*, « Machine Learning in Biological Research: Key Algorithms, Applications, and Future Directions », *BMC Biology*, vol. 23, 2025.

[6] A. Mellouk , A. Chebira, *Machine Learning*, IntechOpen, 2009.

[7] K. Q. Weinberger , L. K. Saul, « Distance Metric Learning for Large Margin Nearest Neighbor Classification », *Journal of Machine Learning Research*, vol. 10, 2009.

[8] L. Probst, A.-L. Boulesteix , B. Bischl, « Tunability: Importance of Hyperparameters of Machine Learning Algorithms », *Journal of Machine Learning Research*, vol. 20, 2019.

[9] A. Cutler, D. R. Cutler and J. R. Stevens, « Random Forests », *Ensemble Machine Learning*, Springer, 2012.

[10] T. Chen , C. Guestrin, « XGBoost: A Scalable Tree Boosting System », *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2016, pp. 785–794.

[11] Université d'Oran 2 Mohamed Ben Ahmed, « Mémoire de fin d'études : Détection et traitement des signaux (SDR / IA) », Département Informatique, Université d'Oran 2, Oran, Algérie, 2024.

[12] T. Chen , C. Guestrin, « XGBoost: A Scalable Tree Boosting System », *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2016, pp. 785–794.

[13] S. Haykin, « Cognitive Radio: Brain-Empowered Wireless Communications », *IEEE Journal on Selected Areas in Communications*, vol. 23, n° 2, pp. 201–220, févr. 2005.

[14] T. Yücek et H. Arslan, « A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications », *IEEE Communications Surveys & Tutorials*, vol. 11, n° 1, pp. 116–130, 1er trimestre 2009.

[15] H. Urkowitz, « Energy Detection of Unknown Deterministic Signals », *Proceedings of the IEEE*, vol. 55, n° 4, pp. 523–531, avr. 1967.

[16] A. Goldsmith, *Wireless Communications*, Cambridge University Press, Cambridge, Royaume-Uni, 2005.

[17] D. Cabric, A. Mishra et R. W. Brodersen, « Implementation Issues in Spectrum Sensing for Cognitive Radios », *Proceedings of the IEEE Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 2004, pp. 772–776.

[18] I. F. Akyildiz, W. Y. Lee, M. C. Vuran and S. Mohanty, « NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey », *Computer Networks*, vol. 50, n° 13, pp. 2127–2159, 2006.

[19] O. J. Bent , M. A. Ingram, « Software Defined Radio: A Modern Approach to Wireless Communication Systems », *IEEE Communications Surveys & Tutorials*, vol. 15, n° 2, pp. 575–589, 2013.

[20] H. Urkowitz, « Energy Detection of Unknown Deterministic Signals », *Proceedings of the IEEE*, vol. 55, n° 4, pp. 523–531, 1967.

[21] T. Hentschel, M. Henker and G. Fettweis, « The Digital Front-End of Software Radio Terminals », *IEEE Personal Communications*, vol. 6, n° 4, pp. 40–46, 1999.

[22] B. P. Lathi et Z. Ding, *Modern Digital and Analog Communication Systems*, 4e éd., Oxford University Press, New York, NY, USA, 2009.

[23] Stewart, Robert W. and Barlee, Kenneth W. and Atkinson, Dale S. W. and Crockett, Louise H., *Software Defined Radio Using MATLAB & Simulink and the RTL-SDR*, University of Strathclyde Research Portal, Glasgow, Royaume-Uni.

[24] J. Mitola III, « Software Radios: Survey, Critical Evaluation and Future Directions », *IEEE Aerospace and Electronic Systems Magazine*, vol. 8, n° 4, pp. 25–36, 1993.

[25] T. Hentschel, M. Henker and G. Fettweis, « The Digital Front-End of Software Radio Terminals », *IEEE Personal Communications*, vol. 6, n° 4, pp. 40–46, 1999.

[26] European Telecommunications Standards Institute (ETSI), *Digital Cellular Telecommunications System (Phase 2+) — GSM Technical Specification (GSM 03.03 / 3GPP TS 23.003)*, Norme ETSI, Sophia Antipolis, France.

[27] R. A. Chipman, *Transmission Lines and Antenna Principles*, McGraw-Hill, New York, NY, USA.

[28] D. Beazley, *Python Essential Reference*, 4e éd., Addison-Wesley Professional, Indianapolis, IN, USA, 2009.

[29] A. Banks , E. Porcello, *Learning React: Functional Web Development with React and Redux*, O'Reilly Media, Sebastopol, CA, USA, 2020.