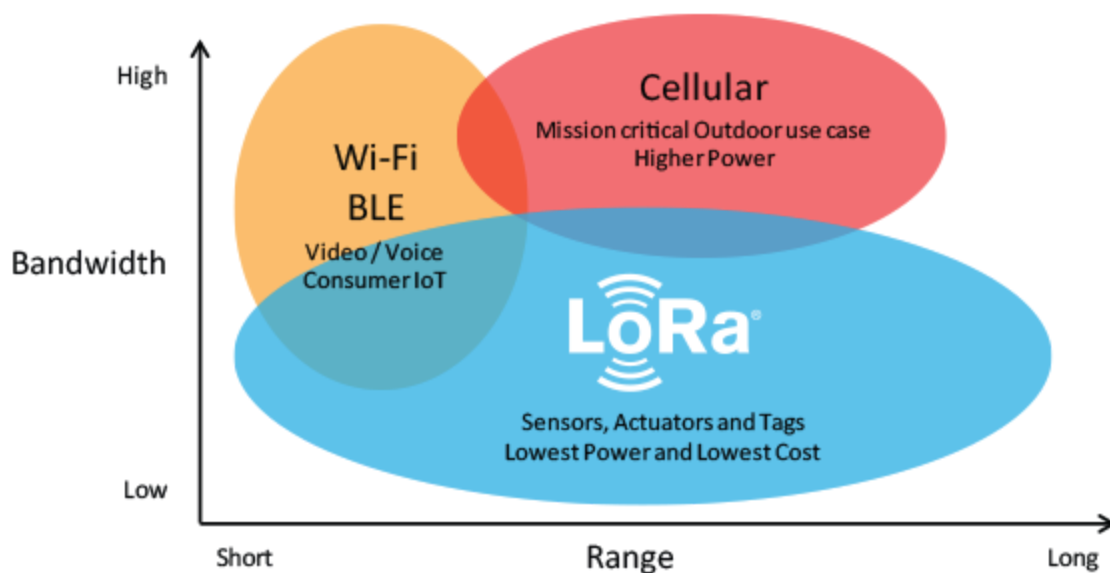


Identification de l'émetteur à partir du préambule LoRa



Réalisation : ANGLADE Alexis & ZHAO Yan

Client : Mme. EL RACHKIDY Nancy (LIMOS)

Tuteur industriel : M. FICKINGER Pascal

Professeur référent : M. LANDRAULT Alexis

Résumé

Ce projet explore l'utilisation du CFO (Décalage de la fréquence porteuse) pour différencier les nœuds dans un réseau LoRa. Le produit final est un modèle de classification basé sur le CFO et le facteur d'étalement (SF), permettant de reconnaître avec précision l'origine des trames LoRa. À ce stade, le projet a permis de valider la faisabilité de cette approche en construisant une base de données et en testant plusieurs algorithmes de machine learning, avec KNN offrant les meilleures performances. Le projet, proposé dans le cadre des recherches académiques, vise à démontrer des compétences interdisciplinaires pour le programme de génie électrique de Polytech Clermont.

Mots clés

LoRa, CFO, Machine Learning, SF, Classification, Communication sans fil.

Abstract

This project explores the use of CFO (Carrier Frequency Offset) to differentiate nodes in a LoRa network. The final product is a classification model based on CFO and the Spreading Factor (SF), enabling precise identification of the origin of LoRa frames. At this stage, the project has validated the feasibility of this approach by building a database and testing multiple machine learning algorithms, with KNN delivering the best performance. The project, proposed as part of academic research, aims to demonstrate interdisciplinary skills within the Electrical Engineering program at Polytech Clermont.

Key words

LoRa, CFO, Machine Learning, SF, Classification, Wireless Communication.

Remerciements

Nous tenons tout d'abord à exprimer notre sincère gratitude envers notre cliente, Mme. EL RACHKIDY Nancy, pour son suivi tout au long de notre projet. Ses retours constructifs et ses attentes bien définies nous ont permis de mieux comprendre les objectifs du projet et de travailler de manière efficace pour les atteindre.

Nous souhaitons également remercier notre tuteur industriel, M. FICKINGER Pascal, pour son soutien organisationnel tout au long du projet. Son aide dans la gestion de projet a grandement contribué à la réussite du notre.

Enfin, nous adressons nos derniers remerciements à notre professeur référent, M. LANDRAULT Alexis, pour son accompagnement académique. Ses précieux conseils ont été essentiels pour explorer des pistes nouvelles, ce qui a permis d'assurer la réussite du projet.

Sommaire

Introduction.....	1
1. Contexte.....	2
1.1. LIMOS.....	2
1.2. LoRa.....	3
1.3. Problématique.....	6
2. Gestion de projet.....	7
2.1. Cahier des Charges.....	7
2.2. WBS.....	7
2.3. Gantt.....	9
3. Réalisation du projet.....	11
3.1. Matériel utilisé.....	11
3.2. Point à point.....	12
3.3. Utilisation du SDR comme récepteur.....	13
3.4. Machine Learning.....	15
4. Résultats et analyses.....	17
Conclusion.....	18

Table des figures

Figure 1 : Fonctionnement du Chirp Spread Spectrum.....	3
Figure 2 : Rapport entre le SF, la portée et le débit du signal émis.....	4
Figure 3 : Représentation du CFO.....	5
Figure 4 : Fonctionnement d'un réseau utilisant LoRa.....	5
Figure 5 : Forme générale d'une trame LoRa.....	6
Figure 6 : WBS du projet.....	8
Figure 7 : Gantt du projet.....	9
Figure 8 : BladeRF Micro ASMB REV1.3.....	11
Figure 9 : Wifi LoRa 32.....	12
Figure 10 : Communication point à point LoRa entre 2 modules Arduino.....	12
Figure 11 : Configuration du BladeRF avec GNU Radio.....	13
Figure 12 : Affiche trames reçues par le BladeRF.....	14
Figure 13 : Affichage trames reçues et CFOs décodés par le BladeRF.....	15
Figure 14 : Nombre de modules correctement identifiés par l'algorithme kNN.....	16
Figure 15 : Précision de cinq algorithmes d'apprentissage supervisé.....	17

Introduction

De nos jours, la communication sans fil joue un rôle majeur dans le développement des nouvelles technologies, telles que l'Internet des Objets (IoT) par exemple. Parmi les nombreuses technologies de communication sans fil disponibles, LoRa (Long Range) se distingue par sa capacité à offrir une communication longue portée avec une faible consommation d'énergie, ce qui en fait une solution idéale pour les applications IoT. Cependant, l'identification et la classification des objets émetteurs de trames LoRa dans un réseau multi-émetteurs posent encore des défis techniques, notamment dans des environnements complexes.

C'est dans ce contexte que, pour notre cliente Mme. El Rachkidy qui travail sur ce sujet depuis de nombreuses années au sein du laboratoire LIMOS situé sur le campus des Cézeaux à Aubière, nous avons eu comme mission, pour notre projet de 5e année en génie électrique à Polytech Clermont, de valider un raisonnement selon lequel l'émetteur d'une trame LoRa peut être identifié à partir d'une caractéristique physique de cette trame.

À travers une série d'expériences, nous avons donc étudié les principes fondamentaux de la transmission LoRa, validé la communication entre des émetteurs LoRa et un récepteur SDR (Software Defined Radio), et exploité les données collectées pour développer un modèle de classification basé sur des algorithmes de machine learning.

La démarche suivie ainsi que les résultats obtenus tout au long de notre projet sont ainsi présentés dans ce rapport.

1. Contexte

Avant de rentrer dans le vif du sujet, et pour mieux appréhender le sujet, complexe, de notre projet, il est essentiel de poser un contexte à celui-ci en présentant le laboratoire dans lequel travaille notre cliente, ainsi qu'en définissant plus en détails ce qu'est LoRa et comment il fonctionne. A la suite de ces définitions, nous pourrons donc mieux définir la problématique globale du projet, de laquelle découle le Cahier des Charges qui sera présenté dans la partie 2.1.

1.1. LIMOS

Le Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes est une Unité Mixte de Recherche (UMR 6158) en informatique, et plus généralement en Sciences et Technologies de l'Information et de la Communication (STIC).

Le LIMOS est principalement rattaché à l'Institut des Sciences de l'Information et de leurs Interactions (INS2I) du CNRS et de façon secondaire à l'Institut des Sciences de l'Ingénierie et des Systèmes (INSIS). Il a pour tutelles académiques l'Université Clermont Auvergne et l'Ecole Nationale Supérieure des Mines de Saint-Etienne (EMSE), et comme établissement partenaire l'école d'ingénieur SIGMA. Le LIMOS est membre des labex IMOBS3 et ClercVolc et de la fédération de recherche en Environnement FR 3467 (qui regroupe 17 laboratoires UCA et INRA du site de Clermont-Ferrand). Il est membre associé de la fédération MODMAD (MODélisation Mathématique et Aide à la Décision, FED 4169) portée par l'Université Jean Monnet de Saint-Etienne.

Le positionnement scientifique du LIMOS est centré autour de l'Informatique, la Modélisation et l'Optimisation des Systèmes Organisationnels et Vivants. Les principaux thèmes de recherche développés au sein du laboratoire sont :

- Optimisation Combinatoire
- Algorithmique, Graphes, Complexité
- Données, Services, Intelligence
- Réseau et sécurité
- Modélisation et Optimisation des Systèmes Manufacturiers
- Conception et Planification de services
- Métamodélisation, Optimisation Continue et Applications (MOCA)

Parmi ces thèmes, c'est celui du réseau et de la sécurité qui nous intéresse puisque c'est dans ce domaine que notre cliente, Mme. EL RACHKIDY, s'intéresse au protocole de communication LoRa, que nous allons définir à présent.

1.2. LoRa

LoRa, qui signifie Long Range, est un protocole de communication sans fil conçu pour les transmissions longue distance à faible consommation d'énergie. Son fonctionnement repose sur une modulation en fréquence du signal émis appelée Chirp Spread Spectrum (ou CSS), qui offre une excellente immunité au bruit et permet ainsi de transmettre des données sur plusieurs kilomètres.

1.2.1. Chirp Spread Spectrum

Le fonctionnement du CSS est plutôt simple : le signal est généré en modifiant progressivement sa fréquence au fil du temps, créant une variation continue appelée chirp. Ce chirp commence à une fréquence basse et augmente de manière linéaire jusqu'à une fréquence plus élevée, comme le montre la Figure 1. L'amplitude du signal reste ainsi constante, seule sa fréquence change. La durée d'un chirp est déterminée par la quantité de données à transmettre.

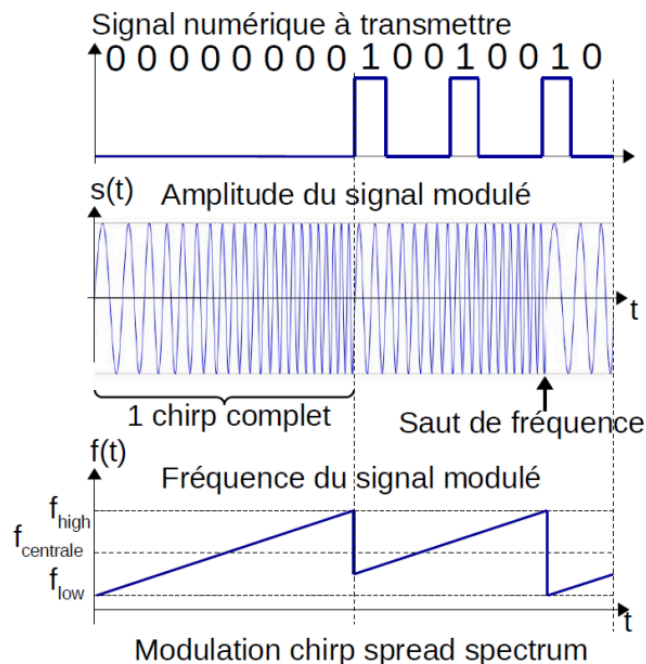


Figure 1 : Fonctionnement du Chirp Spread Spectrum

Lorsqu'un récepteur capte ce signal, il analyse donc les changements de fréquence au fil du temps. En synchronisant correctement la réception avec la fréquence du signal, il est possible de décoder les informations envoyées, même en présence d'interférences.

1.2.2. Spreading Factor

Intéressons-nous à présent à un paramètre clé du protocole LoRa, qui est étroitement lié avec le CSS : le SF, pour Spreading Factor.

Il s'agit d'un nombre, compris entre 7 et 12, qui détermine le nombre de bits qui doivent être transmis durant un chirp. Le SF détermine donc la durée, ainsi que la fréquence, à laquelle les chirps sont générés, influençant ainsi la portée et la capacité de transmission.

En effet, comme le prouve la Figure 2, un SF élevé signifie qu'un chirp dure plus longtemps, ce qui augmente la portée de la communication car cela permet au signal d'être plus facilement détecté, même dans des conditions difficiles. Cependant, un SF élevé réduit la vitesse de transmission des données, car il faut donc plus de temps pour transmettre chaque chirp. A contrario, un SF faible réduit la durée du chirp, augmentant ainsi le débit de données, mais au détriment de la portée et de la capacité à résister aux interférences.

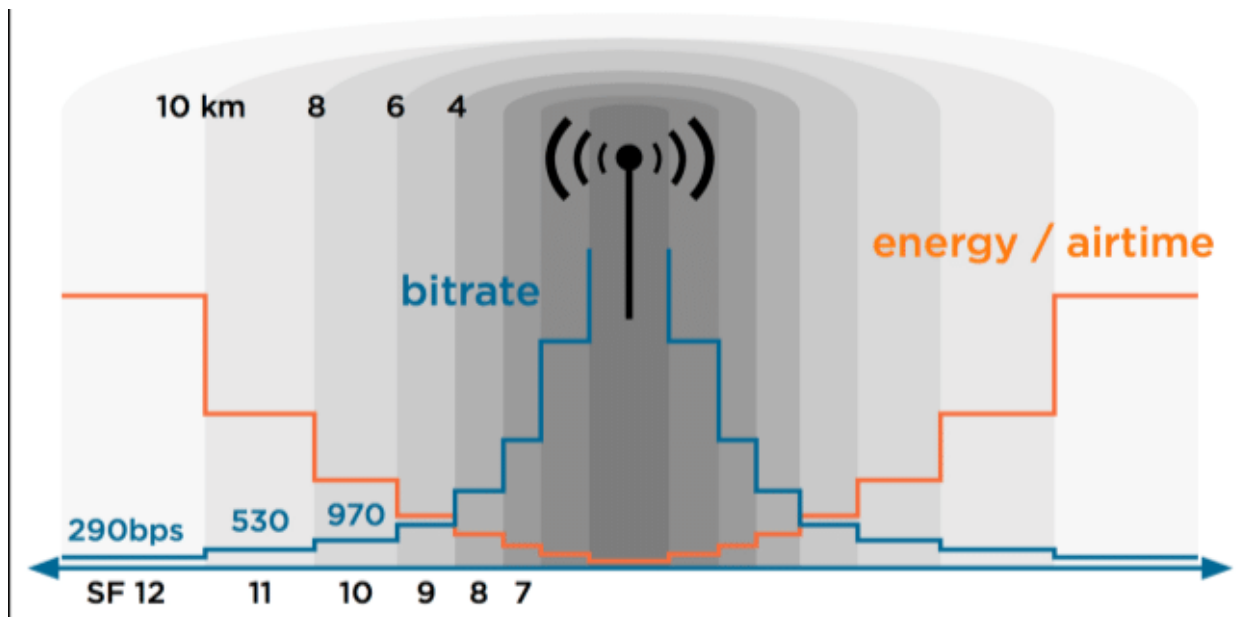


Figure 2 : Rapport entre le SF, la portée et le débit du signal émis

1.2.3. Carrier Frequency Offset

Enfin, et c'est le paramètre qui nous intéresse le plus, définissons ce qu'est le CFO, pour Carrier Frequency Offset.

La traduction littérale de ce terme, “Décalage de la fréquence porteuse”, nous donne déjà un bon indice sur ce qu’est ce paramètre. Il s’agit donc, comme le montre la Figure 3, de la différence de fréquence (centrale) entre le signal émis par un émetteur et le signal reçu par un récepteur.

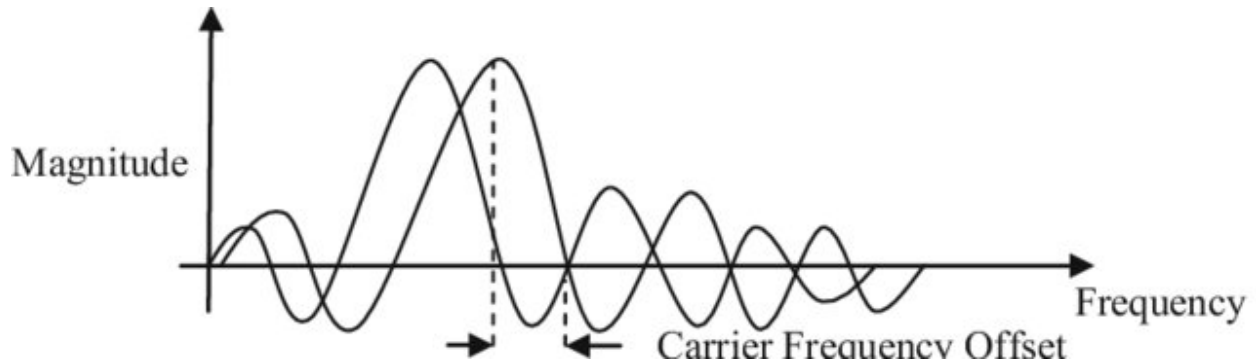


Figure 3 : Représentation du CFO

Selon les résultats de plusieurs études, cette différence semble être unique et constante entre un émetteur et un récepteur, en l'occurrence une passerelle transformant ce signal LoRa en un signal TCP/IP interprétable par un serveur comme le présente la Figure 4, mais cela n’as jamais été prouvé.

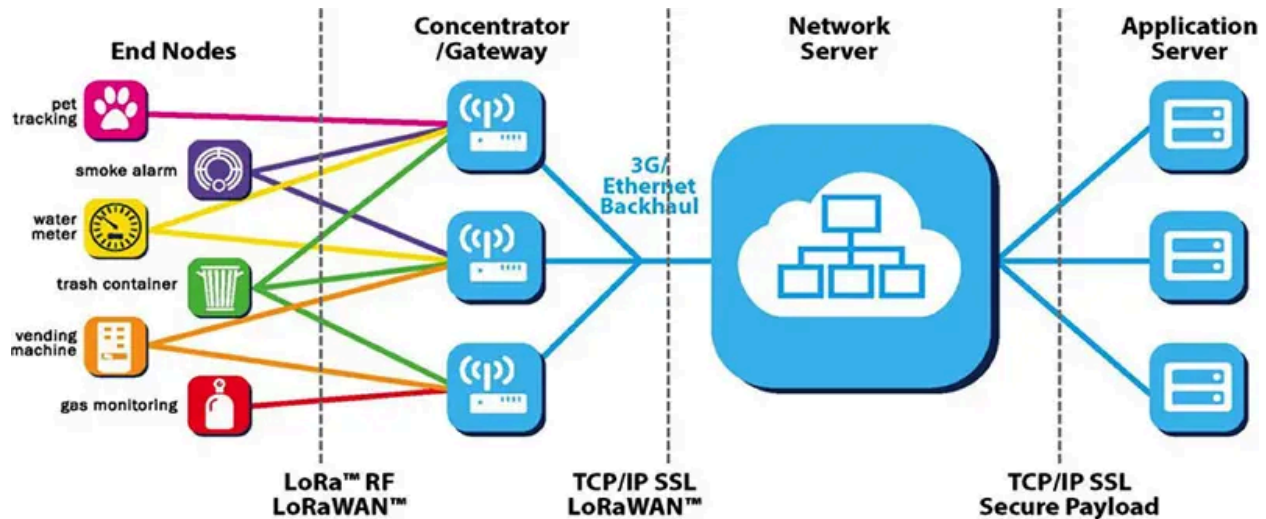


Figure 4 : Fonctionnement d'un réseau utilisant LoRa

Cette unicité et cette stabilité seraient pourtant très utile si elles s’avèrent être vraies car cela permettrait de résoudre un des principaux problème du protocole LoRa, qu’est le temps dédié au décodage de l’identifiant de l’émetteur, présent dans l’en-tête de chaque trame LoRa (dont la forme est explicité en Figure 5).

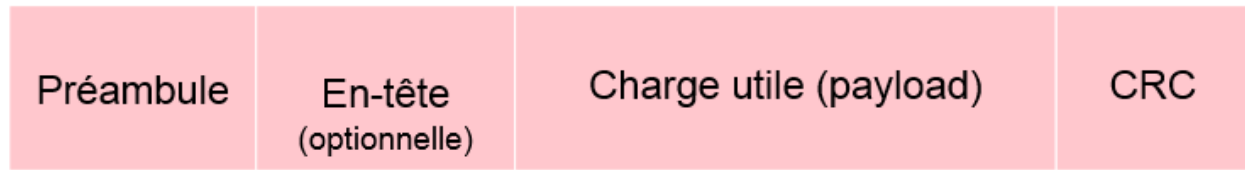


Figure 5 : Forme générale d'une trame LoRa

En effet, ce temps est extrêmement long, d'autant plus, qu'en fonction de la quantité de données utiles envoyées dans chaque trame, l'identifiant peut représenter entre 17 et 57% de cette trame, car il est toujours codé sur 4 octets (1 octets comportant 8 bits).

1.3. Problématique

L'objectif de notre projet a donc été de déterminer, et de prouver si c'est le cas, s'il était possible d'identifier l'émetteur d'une trame LoRa à partir de son CFO.

2. Gestion de projet

Maintenant que le contexte est clairement défini, détaillons notre organisation du projet en commençant par la rédaction d'un Cahier des Charges. Celui-ci se présente sous la forme d'une suite de problématiques auxquelles répondre pour mener à bien le projet.

2.1. Cahier des Charges

La première problématique à laquelle nous devons répondre est la problématique générale du projet, c'est-à-dire déterminer s'il est possible d'identifier l'émetteur à partir de son CFO.

L'objectif est donc d'étudier si le CFO est bien unique et constant entre un émetteur et un récepteur. Cela permettrait alors de l'envisager comme un identifiant unique pour chaque émetteur LoRa. Cette première étape implique donc d'analyser les variations du CFO entre différents émetteurs et d'évaluer leur stabilité dans des conditions variées.

Cette première problématique nous amène directement à la deuxième : étudier l'impact des paramètres du SF sur la continuité du CFO.

En effet, parmi les conditions à testées, il y a la modification du SF, qui pourrait influencer la stabilité du CFO. Cette deuxième étape consiste donc à faire varier le SF et à voir si l'unicité et la stabilité, si elles ont été déterminées lors de la première étape, sont conservées ou non.

Enfin, si les deux premières problématiques donnent des résultats positifs, l'idée est ensuite de déterminer un modèle d'identification de l'émetteur à partir du CFO. Cette étape consiste alors à entraîner un réseau de neurones pour faire du Machine Learning en utilisant une base de données, créée pour l'occasion, dans l'objectif d'obtenir un modèle d'identification de l'émetteur à partir du CFO.

2.2. WBS

Une fois le Cahier des Charges rédigé, nous avons ensuite pu construire un WBS (Work Breakdown Structure), présenté en Figure 6, pour avoir un meilleur aperçu du travail à réaliser pour le projet.

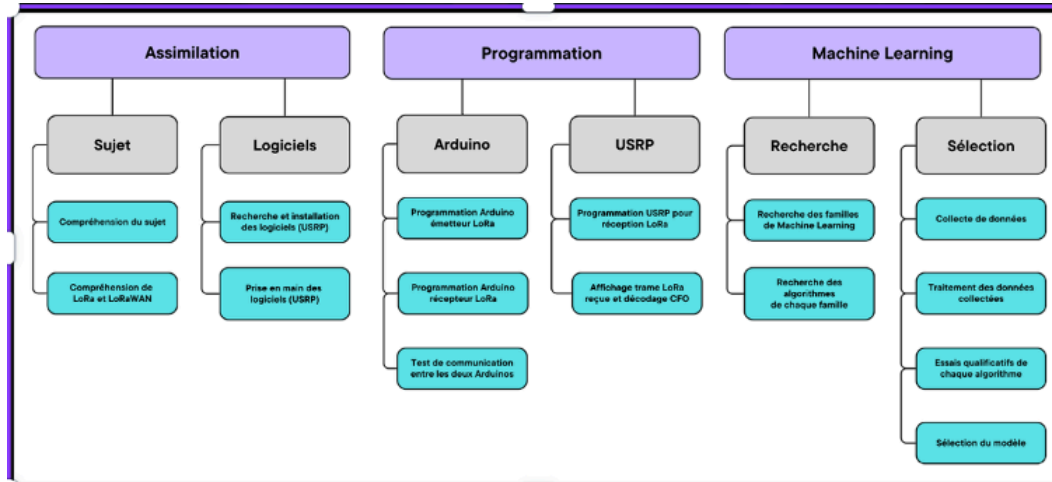


Figure 6 : WBS du projet

2.2.1. Assimilation

La première partie du projet consiste donc à assimiler les concepts clés et les outils nécessaires. Cela inclut une compréhension approfondie de LoRa et LoRaWAN, ainsi que des recherches et l'installation des logiciels nécessaires, notamment pour l'utilisation d'une USRP (ou SDR), qui sera présentée dans la partie 3.1. Une prise en main des outils logiciels est également nécessaire pour assurer une base solide en vue des phases suivantes du projet.

2.2.2. Programmation

Ensuite, la deuxième partie du projet est dédiée à la programmation, qui se divise en deux parties principales : Arduino et USRP.

Du côté Arduino, des programmes doivent être développés pour le module émetteur et pour le module récepteur LoRa, présentés dans la partie 3.1, suivis de tests de communication entre les deux modules.

Concernant l'USRP, une programmation doit également être réalisée pour la réception des signaux LoRa, avec un affichage des trames reçues et un décodage du CFO. Cette étape permettra donc d'extraire les données nécessaires pour la suite.

2.2.3. Machine Learning

Cette suite est la partie Machine Learning. Celle-ci commence par une recherche des familles d'algorithmes et une sélection des méthodes adaptées. Ensuite, les données collectées doivent être traitées et utilisées pour tester différents algorithmes. Des essais qualifiés permettent de comparer les performances, et un modèle final doit être sélectionné en fonction de sa précision et de sa robustesse. Cette phase, une fois terminée, marquera ainsi l'aboutissement du projet.

2.3. Gantt

Pour structurer encore plus le projet, et pour pouvoir suivre l'évolution du travail fourni sur la durée du projet, nous avons mis en place un Gantt, visible en Figure 7.

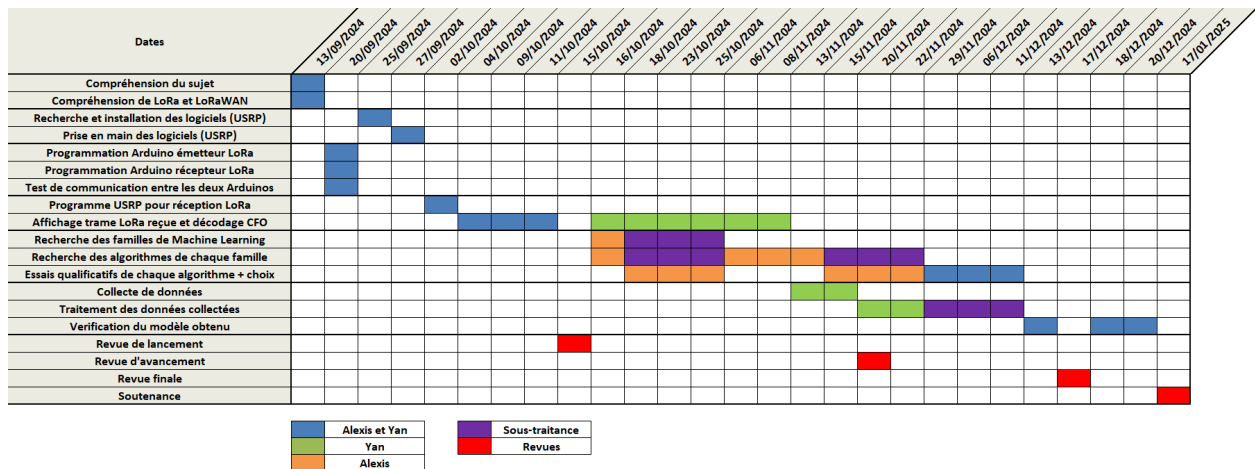


Figure 7 : Gantt du projet

Comme nous pouvons le voir, le projet se déroule sur plusieurs mois, avec une répartition claire des tâches entre les membres et des étapes bien définies.

La phase d'assimilation commence par la compréhension du sujet et de la technologie LoRa/LoRaWAN, suivie de la recherche et de l'installation des logiciels (USRP), ainsi que leur prise en main, qui sont réalisées entre septembre et début octobre.

Ensuite, la phase de programmation débute avec le développement des programmes Arduino pour l'émetteur et le récepteur LoRa, suivi des tests de communication entre ces deux modules.

Une fois la partie Arduino terminée, le travail se sépare en deux avec Yan se concentrant sur l'affichage des trames LoRa reçues et le décodage du CFO, tandis qu'Alexis commence la recherche des familles de Machine Learning et des algorithmes correspondants. Ces activités, ainsi que la programmation de l'USRP pour la réception LoRa, s'étendent jusqu'à la fin novembre.

Enfin, le projet rentre dans sa dernière phase, celle du Machine Learning, avec la collecte et le traitement des données nécessaires, suivis d'essais qualifiants pour tester différents algorithmes. Cela aboutit à la sélection du modèle le plus performant en décembre.

Les revues du projet (lancement, avancement et finale) sont réparties à des dates stratégiques (fixées par le responsable des projets du département Génie électrique : M. LAFFONT) pour assurer un suivi rigoureux, et la soutenance finale est prévue pour mi-janvier 2025.

3. Réalisation du projet

La partie gestion de projet étant à présent bien définie, rentrons dans le vif du sujet en expliquant chaque partie du projet, d'un point de vue technique, et commençons par présenter le matériel que nous avons utilisé.

3.1. Matériel utilisé

3.1.1. BladeRF Micro ASMB REV1.3

Le BladeRF Micro ASMB REV1.3 est une plateforme de radio logicielle (SDR) haute performance et compacte, couvrant une large gamme de fréquences de 47 MHz à 6 GHz. Équipé d'un transcepteur bidirectionnel full-duplex, il permet l'envoi et la réception des signaux en temps réel. Utilisant la puce LMS7002M de Lime Microsystems, il offre une bande passante allant jusqu'à 56 MHz et est compatible avec l'interface USB 3.0.

Cette plateforme est idéale pour la recherche en communication sans fil, l'éducation, la surveillance du spectre et les applications IoT. Grâce à sa conception ouverte, elle est compatible avec des outils open source tels que GNU Radio, offrant une grande flexibilité pour le développement et le test de protocoles sans fil (comme LoRa, Wi-Fi, 4G/5G) données vers cette plateforme SDR, puis afficher ces trames à l'aide de GNU Radio tout en calculant le CFO à l'aide de fonctions internes.

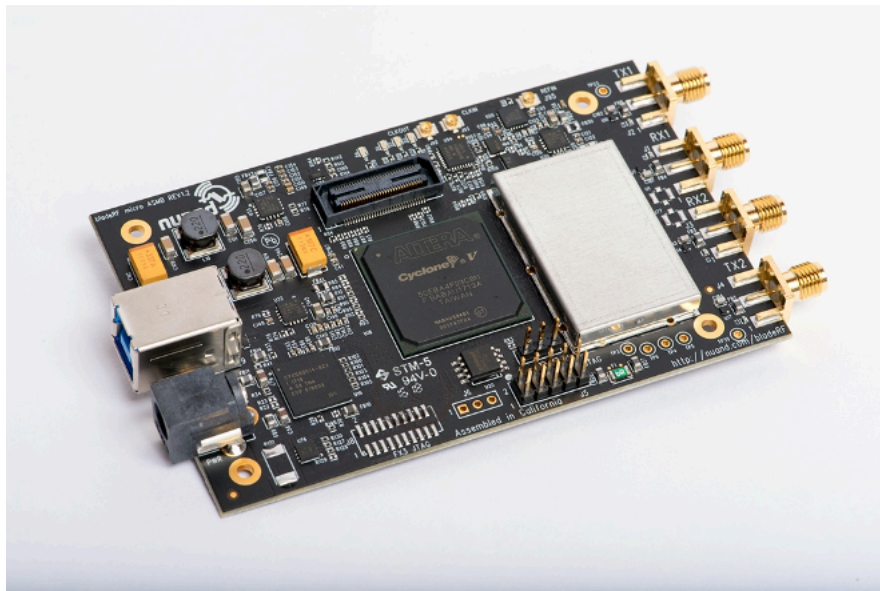


Figure 8 : BladeRF Micro ASMB REV1.3

3.1.2. WiFi LoRa 32

Le WiFi LoRa 32 est une carte de développement IoT classique conçue et produite par Heltec Automation(TM). C'est un produit hautement intégré basé sur ESP32 + SX127x, qui dispose de fonctionnalités Wi-Fi, BLE, LoRa, d'un système de gestion de batterie Li-Po et d'un écran OLED de 0,96 pouce. C'est le choix idéal pour les villes intelligentes, les fermes connectées, les maisons intelligentes et les créateurs IoT. Dans mon projet, j'utilise ce nœud, programmé avec Arduino, afin d'envoyer des trames vers le SDR.

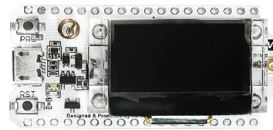


Figure 9 : Wifi LoRa 32

3.2. Point à point

Dans la première étape du projet, nous avons réalisé une expérience de communication point à point en utilisant la plateforme Arduino. L'objectif de cette expérience était de comprendre les principes fondamentaux et le fonctionnement de la transmission LoRa.

Comme illustré dans la Figure 11, deux modules Wifi LoRa 32 ont été configurés pour communiquer directement entre eux. Cela nous a permis de mieux comprendre le mécanisme de transmission des données dans une communication point à point. En réussissant cette expérience, nous avons ainsi établi une base solide pour des expériences de communication plus complexes dans les étapes suivantes du projet.

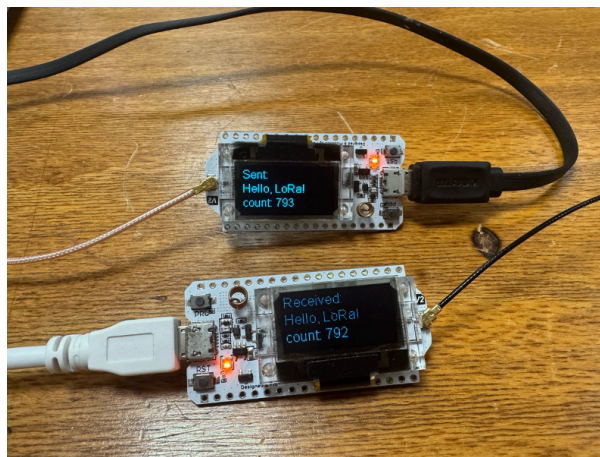


Figure 10 : Communication point à point LoRa entre 2 modules Arduino

3.3. Utilisation du SDR comme récepteur

Après avoir acquis une bonne compréhension de la communication point à point et du protocole LoRa, nous avons réalisé l'étape suivante du projet. L'objectif de cette phase était d'utiliser une radio définie par logiciel (SDR) comme récepteur pour extraire les informations des trames envoyées par un nœud (module) LoRa. Cette expérience nous a permis de valider la capacité de communication entre LoRa et le SDR, tout en posant les bases pour des systèmes plus complexes à développer ultérieurement.

Pour atteindre cet objectif, nous avons d'abord configuré GNU Radio en utilisant BladeRF comme source d'entrée. La fréquence centrale a été réglée sur 868 MHz, avec une bande passante de 125 kHz. BladeRF a servi de récepteur SDR pour capter les signaux LoRa et les décoder via le module LoRa Receiver. En parallèle, nous avons utilisé le module QT GUI Frequency Sink pour surveiller en temps réel la puissance et le spectre des signaux reçus. Cette visualisation a permis de confirmer que le récepteur était correctement synchronisé avec les signaux envoyés.

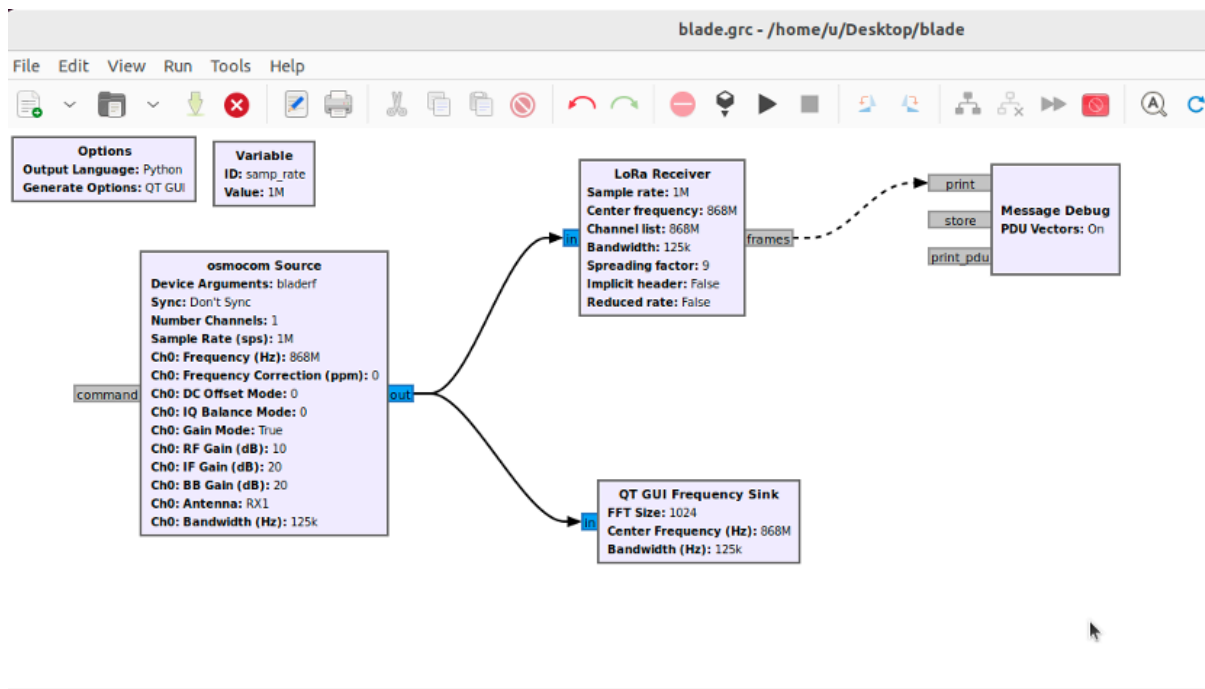


Figure 11 : Configuration du BladeRF avec GNU Radio

3.3.1. Réception des données

Du côté de l'émetteur, nous avons utilisé une carte Arduino associée à un module LoRa pour envoyer les trames de données. Les paramètres de l'émetteur ont été configurés pour correspondre à ceux du récepteur, assurant ainsi une communication fluide. Le nœud Arduino a été programmé pour envoyer des trames en continu, chaque trame contenant un format de données fixe destiné à être vérifié par le récepteur.

Les résultats expérimentaux ont montré qu'avec une configuration appropriée, BladeRF peut recevoir de manière stable les trames LoRa envoyées par l'émetteur. Grâce au module Message Debug de GNU Radio, nous avons pu observer les données décodées des trames, qui correspondaient exactement aux données originales de l'émetteur. En outre, le module QT GUI Frequency Sink a affiché un spectre de signaux stable, confirmant davantage la fiabilité du système.

```
*****
01 20 10 33 (3)
***** MESSAGE DEBUG PRINT *****
#[0 0 0 0 0 0 0 0 0 0 0 0 50 0 1 32 16 51]
*****
01 20 10 34 (4)
***** MESSAGE DEBUG PRINT *****
#[0 0 0 0 0 0 0 0 0 0 0 0 50 0 1 32 16 52]
*****
01 20 10 35 (5)
***** MESSAGE DEBUG PRINT *****
#[0 0 0 0 0 0 0 0 0 0 0 0 50 0 1 32 16 53]
*****
01 20 10 36 (6)
***** MESSAGE DEBUG PRINT *****
#[0 0 0 0 0 0 0 0 0 0 0 0 47 0 1 32 16 54]
*****
01 20 10 37 (7)
***** MESSAGE DEBUG PRINT *****
#[0 0 0 0 0 0 0 0 0 0 0 0 49 0 1 32 16 55]
*****
```

Figure 12 : Affiche trames reçues par le BladeRF

3.3.2. Calculer le CFO et l'afficher

Après avoir confirmé avec succès que le BladeRF pouvait recevoir sans problème les trames de données provenant du nœud émetteur, nous avons commencé à calculer le CFO. En modifiant les fonctions intégrées fournies par GNU Radio, nous avons pu effectuer le calcul du CFO et afficher les résultats en temps réel.

Deux nœuds ont ensuite été utilisés pour envoyer des trames de données au BladeRF. Après le calcul à l'aide des fonctions intégrées, il est clairement visible que les CFO des deux nœuds sont différents. Ces données de CFO peuvent ensuite être utilisées pour créer une base de données.

```
01 20 10 37 (7)
***** MESSAGE DEBUG PRINT *****
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 54 0 1 32 16 55]
*****
etting CFO 8027.26
etting CFO -7889.96
01 20 10 38 (8)
***** MESSAGE DEBUG PRINT *****
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 55 0 1 32 16 56]
*****
etting CFO 7702.44
etting CFO -5058.59
02 20 e0 32 33 (23)
***** MESSAGE DEBUG PRINT *****
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 52 0 2 32 224 50 51]
*****
etting CFO 4584.65
```

Figure 13 : Affichage trames reçues et CFOs décodés par le BladeRF

3.4. Machine Learning

Ensuite, nous avons commencé les expérimentations de Machine Learning et la collecte des données nécessaires pour l'entraînement et les tests.

La base de données créée contient 600 échantillons, répartis équitablement entre les nœuds noirs et blancs (en référence à la couleur du câble associé à chaque module), avec 300 échantillons chacun, 100 par SF (7, 8 et 9). Chaque échantillon contient un SF, un CFO et un nœud émetteur (noir ou blanc).

Cette base de données a ensuite permis, en utilisant le CFO et le SF comme données d'entrées et le module émetteur comme donnée de sortie, de tester grâce à Matlab cinq algorithmes d'apprentissage supervisé.

Matlab, nous a ainsi permis d'obtenir, pour chaque algorithme, une représentation du nombre de modules correctement identifiés ou non par l'algorithme, dont un exemple est donné en Figure 15 (pour l'algorithme kNN).

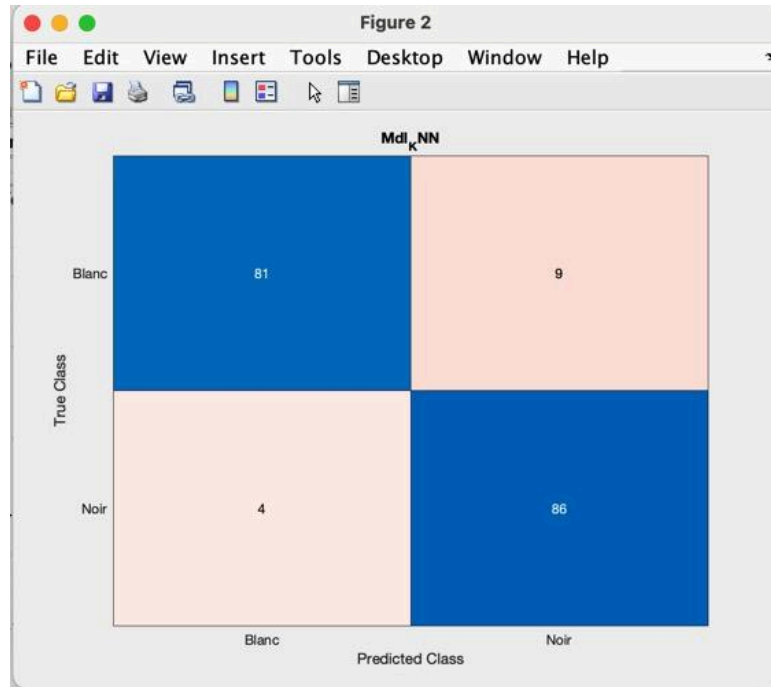


Figure 14 : Nombre de modules correctement identifiés par l'algorithme kNN

4. Résultats et analyses

En concaténant dans un seul graphique, présenté en Figure 16, toutes les représentations du nombre de modules correctement identifiés ou non par chaque algorithme, nous avons pu constater que l'algorithme KNN (k Nearest Neighbour) obtient la meilleure précision, avec presque 90% de bonnes prédictions du module émetteur.

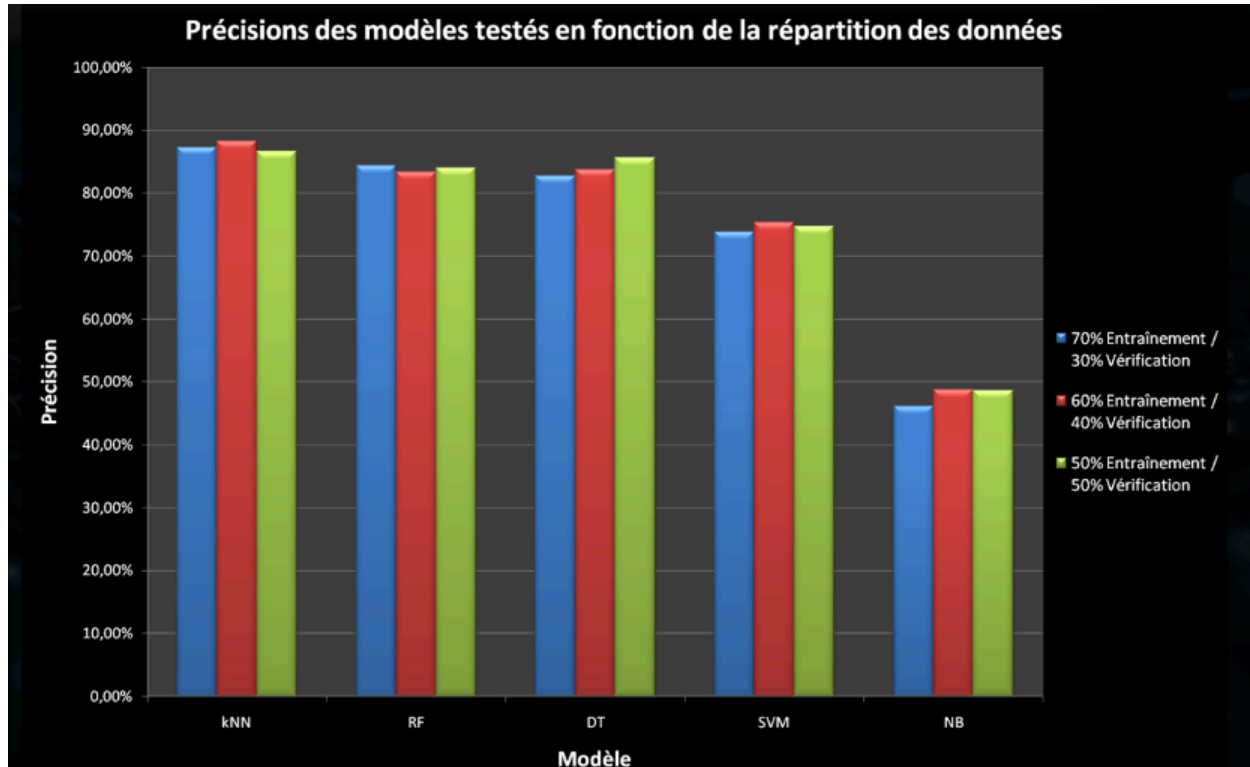


Figure 15 : Précision de cinq algorithmes d'apprentissage supervisé

Ce graphique nous permet également de confirmer qu'il est possible de distinguer les différents nœuds en utilisant le CFO. Ce résultat représente donc une avancée prometteuse pour le développement futur de la technologie LoRa, mettant en évidence le potentiel du CFO dans les tâches de classification des nœuds.

Il peut néanmoins être noté qu'en augmentant le nombre de données de la base utilisée il serait possible, mais ce n'est qu'une hypothèse, d'obtenir de meilleurs résultats sur la partie Machine Learning.

Conclusion

Dans ce projet, nous avons exploré comment utiliser le CFO pour différencier les différents nœuds LoRa, et nous avons réussi à en démontrer la faisabilité.

À travers une série d'expériences, nous avons progressivement réalisé une transition de la communication point à point vers la réception par une SDR. avec cette série d'expériences nous avons pu construire une base de données contenant 600 échantillons à laquelle nous avons appliqué plusieurs algorithmes d'apprentissage supervisé pour la classification des nœuds, et nous avons finalement constaté que l'algorithme KNN offrait les meilleures performances avec une précision optimale.

Bien que ce projet soit de nature exploratoire, les expériences et les analyses de données nous ont permis de mieux comprendre les principes de la communication LoRa, notamment l'impact des paramètres clés tels que le SF et le CSS sur la portée des transmissions et la robustesse des signaux.

L'utilisation du CFO a, quant à elle, non seulement démontré son potentiel dans la classification des nœuds, mais a également ouvert de nouvelles perspectives pour le développement futur de la technologie LoRa.

Ce projet ayant abouti à une conclusion définitive, il a permis de valider des hypothèses et d'optimiser les résultats par une approche expérimentale. Ce processus a considérablement renforcé nos compétences en conception expérimentale, en analyse des données et en résolution de problèmes. Nous avons également pris conscience de l'importance de l'intégration des connaissances interdisciplinaires, telles que la communication sans fil, le traitement du signal et le machine learning, dans la résolution de problématiques complexes.

À l'avenir, le projet pourrait être étendu par une augmentation de la taille des données, l'optimisation des paramètres des algorithmes, ou encore l'exploration d'applications dans des environnements multi-nœuds plus complexes. En somme, ce projet a non seulement atteint ses objectifs, mais a également jeté des bases solides pour des recherches futures et démontré le potentiel prometteur du CFO dans les applications de la technologie LoRa.