



HAL
open science

Instrumentation embarquée avec correction déterministe et exploitation par IA de capteurs électroniques : application à la surveillance de ruches.

Augustin Bricout

► To cite this version:

Augustin Bricout. Instrumentation embarquée avec correction déterministe et exploitation par IA de capteurs électroniques : application à la surveillance de ruches.. Energie électrique. Université de Toulouse, 2024. Français. NNT : 2024TLSEI019 . tel-04955046v2

HAL Id: tel-04955046

<https://theses.hal.science/tel-04955046v2>

Submitted on 18 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Doctorat de l'Université de Toulouse

préparé à l'INSA Toulouse

Instrumentation embarquée avec correction déterministe et exploitation par IA de capteurs électroniques : application à la surveillance de ruches.

Thèse présentée et soutenue, le 10 décembre 2024 par

Augustin BRICOUT

École doctorale

GEETS - Génie Electrique Electronique, Télécommunications et Santé : du système au nanosystème

Spécialité

Génie Electrique

Unité de recherche

LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par

Jean-Yves FOURNIOLS

Composition du jury

Mme Daniela DRAGOMIRESCU, Présidente, INSA Toulouse

M. Camel TANOUGAST, Rapporteur, Université de Lorraine

M. Toufik BAKIR, Rapporteur, Université de Bourgogne

M. Pascal ACCO, Examineur, INSA Toulouse

Mme Eunika MERCIER-LAURENT, Examinatrice, Université de Reims Champagne-Ardenne

M. Jean-Yves FOURNIOLS, Directeur de thèse, INSA Toulouse

Membres invités

M. Rémi Floquet, RF Innovation

Titre : Instrumentation embarquée avec correction déterministe et exploitation par IA de capteurs électroniques : application à la surveillance de ruches.

Mots clés : Surveillance des ruches, Développement Matériel et logiciel embarqué, Apiculture, Amélioration d'un capteur, Acoustique des abeilles, Machine Learning

Résumé : L'objectif de cette thèse est de développer des solutions de smart sensing à bas coût, permettant d'améliorer des capteurs peu onéreux grâce à des corrections logicielles. Plutôt que de concevoir des capteurs performants mais coûteux, cette approche vise à créer des capteurs low cost qui sont ensuite corrigés et optimisés via des algorithmes embarqués. Les architectures électroniques récentes offrent désormais une puissance de calcul suffisante pour effectuer ces corrections directement au plus proche de la mesure, edge computing, tout en conservant une consommation énergétique extrêmement faible rendant les systèmes sur batterie viables.

Deux approches de correction logicielle sont explorées : une méthode basée sur un algorithme déterministe, et une seconde méthode s'appuyant sur l'intelligence artificielle. Après avoir conçu une architecture de collecte de données adaptée aux ruches d'abeilles, ces deux approches sont mises en œuvre. La première approche, déterministe, est utilisée pour corriger les données provenant des jauges de contrainte dans le cadre d'une balance à ruches. La seconde méthode est appliquée à des capteurs audio MEMS, afin d'extraire des métriques de santé des colonies d'abeilles grâce à des techniques de machine learning.

Title: IA and smart sensing embedded electronics system dedicated to bee's activity recognition.

Key words: Hive Monitoring, Hardware development and Embedded Software, Beekeeping, Low Cost sensors, Bee sounds, Machine learning

Abstract: The objective of this thesis is to develop low-cost smart sensing solutions, enhancing inexpensive sensors through software-based corrections. Rather than designing high-performance but expensive sensors, this approach aims to create low-cost sensors that are then corrected and optimized via embedded algorithms. Recent electronic architectures now offer sufficient computational power to perform these corrections directly at the measurement source, known as edge computing, while maintaining extremely low energy consumption, making battery-powered systems viable.

Two software correction approaches are explored: a method based on a deterministic algorithm, and a second method relying on artificial intelligence. After designing a data collection architecture suited for beehives, both approaches are implemented. The first, deterministic approach is used to correct data from strain gauges in the context of a hive scale. The second method is applied to MEMS audio sensors, to extract bee colony health metrics using machine learning techniques.

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à Rémi Floquet pour avoir cru en moi et m'avoir accueilli chez RF Innovation. Merci également pour le financement du projet et l'accompagnement tout au long de cette thèse Cifre, son soutien et sa confiance ont été essentiels à la réussite de ce travail.

Merci au LAAS-CNRS et à l'équipe S4M pour l'accueil, les moyens mis à disposition et les discussions, échanges et moments riches de science avec les doctorants, les chercheurs, des femmes et des hommes passionnés et inspirants qui se reconnaîtront.

Je remercie également Jean-Yves Fourniols et Pascal Acco associés en codirection de thèse pour leur encadrement et pour avoir su me redonner de l'élan dans les moments difficiles.

Un remerciement tout particulier à Jean-Yves d'avoir accepté d'héberger les ruches, malgré son aversion initiale pour les insectes et aussi de m'avoir permis d'enseigner en me confiant certains de ses cours à l'INSA, au DGEI, département de spécialité où j'ai été formé en tant qu'ingénieur.

Mes remerciements vont aussi à Philippe Leleux, pour ses réponses toujours claires à mes questions et son aide précieuse dans la rédaction des articles scientifiques.

Merci également à Christophe Escriba pour nos échanges enrichissants aux coins des couloirs du LAAS ou de l'INSA à n'importe quelle heure de la journée ou de la nuit.

Je tiens à remercier chaleureusement les rapporteurs de cette thèse, M. Tanougast et M. Bakir, pour avoir accepté d'évaluer mon travail et également merci aux membres du jury, Mme Dragomirescu et Mme Mercier-Laurent.

Je tiens à exprimer toute ma gratitude à ma famille et à mes amis pour leur soutien. Leur présence et leurs encouragements ont été une source de force et d'inspiration inestimables. Un clin d'œil à Xavier.

Un merci spécial à Pauline, présente au quotidien.

Table des matières

Remerciements	i
Table des matières	iii
Introduction	1
1 Électronique et abeilles	5
1.1 Instrumenter en électronique	6
1.2 Qu'est ce qu'une abeille?	7
1.2.1 Colonie d'abeille	8
1.2.1.1 Biologie et fonctionnement de l'abeille	8
1.2.1.2 Cycle naturel d'une colonie d'abeilles	10
1.2.1.3 Cycle dans une ruche en production	11
1.2.2 Pour un apiculteur professionnel	13
1.2.2.1 Structure d'une ruche	13
1.2.2.2 Organisation d'une saison	14
1.2.2.3 La balance : métrique la plus répandue	15
1.2.2.4 Les besoins d'un apiculteur professionnel	16
1.3 Conclusion	16
2 Collecte et affichage des données	19
2.1 Approche générale	20
2.1.1 Approche de la littérature et de l'industrie	20
2.1.2 Architecture présentée	22
2.2 Choix des technologies radio	24
2.2.1 Protocole de communication au sein du rucher	24
2.2.2 Protocole de communication longue portée	25
2.2.3 Format des données : API standard	25
2.3 Développement matériel et logiciel	27
2.3.1 La Gateway	27
2.3.1.1 Architecture matérielle	27
2.3.1.2 Architecture logicielle	29
2.3.1.3 Consommation et durée de vie	30
2.3.1.4 Durée de vie de la batterie sans le lien 868MHz	33
2.3.2 Les capteurs	34

2.3.2.1	Architecture standard	34
2.3.2.2	La balance	35
2.3.2.3	La capteur audio	39
2.3.2.4	Autonomie sur pile des capteurs	40
2.4	Déploiement du système	41
2.4.1	Données de masse	41
2.4.2	Données audio	45
2.5	Transmission et Stockage des données	46
2.5.1	Protocole MQTT	47
2.5.2	Base de données	48
2.5.3	Web architecture pour le stockage des données	48
2.6	Visualisation des données	49
2.6.1	Utilisation des données pour la recherche	50
2.6.2	Application Web pour les apiculteurs	50
2.7	Conclusion	53
3	Amélioration d'un capteur low cost	55
3.1	L'origine de la dérive thermique	56
3.1.1	Caractérisation de la dérive issue de la chaîne d'acquisition	56
3.1.2	Qualité des jauges de contraintes	58
3.1.3	Correction et calibration par sonde et globale	61
3.1.4	Visualisation de la dérive thermique	62
3.2	Correction dynamique par régression polynomiale	64
3.2.1	Modèle dynamique de correction	65
3.2.2	Identification du polynôme de correction	66
3.2.3	Performance de correction et mesure de sensibilité	69
3.2.4	Extrapolation du modèle	72
3.2.4.1	Exagération du modèle pour illustrer le sur-apprentissage	72
3.2.4.2	Limite de l'ordre du polynôme pour éviter le sur-apprentissage	73
3.2.5	Conclusion de la régression linéaire sur toutes les données	74
3.3	Compensation circadienne avec une mesure par jour IMJ	74
3.3.1	Première visualisation de la dérive journalière	75
3.3.2	Séparation temporelle par filtres complémentaires	77
3.3.3	Corrections linéaires de l'IMJ	78
3.3.4	Résultats de l'IMJ	79
3.4	Discussion sur la possibilité d'embarquer le modèle dans un microcontrôleur	81
3.5	Conclusion	81
4	Exploiter l'audio par l'IA	83
4.1	Importance du signal audio pour la santé de la ruche et applications actuelles	84
4.2	Revue de la littérature	86

4.2.1	Diversité des données audio utilisées dans la littérature scientifique	86
4.2.2	Labels utilisés pour la classification	89
4.2.3	Extraction de features pour la représentation temps/fréquence des fichiers audio	91
4.2.4	Modèles de classifications	93
4.2.5	Problème d’extrapolation vers une nouvelle ruche	94
4.3	Méthodologie	96
4.3.1	Unification des datasets publics : Construction du framework BeeTogether	96
4.3.1.1	Standardisation des données brutes	96
4.3.2	Extraction des features	98
4.3.2.1	Choix des features extraites pour application au modèle	99
4.3.3	Augmentation de données	103
4.3.4	Modèle d’apprentissage issu de l’état de l’art	104
4.3.5	Nouvelle méthode de classification inspirée du contrastive learning	106
4.3.6	Méthode de validation	107
4.3.6.1	Validation classique avec ensembles d’entraînement et de test :	107
4.3.6.2	Validation par extrapolation de type k-hive-fold :	107
4.3.7	Mesures de performance du modèle	107
4.4	Reproduction des performances de l’état de l’art et mise en lumière du problème d’extrapolation	108
4.4.1	Validation de l’approche CNN + Mel-Spec	108
4.4.2	Le label Abeille–PasAbeille est-il utile ?	110
4.4.3	Sur la difficulté d’extrapoler la classification de la présence de la reine à des ruches non vues	111
4.4.3.1	Non-extrapolation des modèles <code>classical</code> et <code>pairwise</code>	111
4.4.3.2	La classification peut être axée sur de mauvais paramètres	115
4.5	Extrapolation avec un modèle inspiré du contrastive learning	119
4.5.1	Prospective sur une base de données non labellisée : piste de validations	121
4.6	Évaluation de la possibilité d’embarquer le modèle dans un microcontrôleur	124
4.7	Conclusion	126
	Conclusion générale et perspectives	129
	Bibliographie	133

Introduction

Contexte

L'apiculture comme source de sucre est utilisée depuis des milliers d'années. On retrouve les premières peintures rupestres de récolte du miel sauvage entre 5000 et 10 000 ans avant Jésus-Christ. Depuis ces débuts, une filière apicole où sont élevées les abeilles dans le but d'extraire les produits de la ruche s'est fortement développée, jusqu'à créer une profession largement répandue dans le monde. En parallèle de ces développements de l'apiculture pour le miel, la mise en évidence de l'augmentation des productions agricoles en présence d'abeilles, grâce à la pollinisation réalisée par celles-ci, a conduit à des collaborations importantes dans certaines filières, notamment la culture de l'amande, où les ruches augmentent la production de près de 80%. Face aux défis actuels liés à la croissance démographique et au besoin de nourrir la planète, la pratique de l'apiculture devient un enjeu majeur [1, 2].

Pour répondre à cette demande de production, la pratique de l'apiculture moderne a évolué vers la transhumance des ruches pour suivre les floraisons qui sont en décalage en fonction de la latitude et de la météo. Ce fonctionnement est presque symbiotique entre apiculteur et agriculteur car les deux profitent de la présence de l'autre pour produire plus. Dans la pratique de l'apiculture moderne en France on estime en moyenne qu'un apiculteur en production de miel doit fonctionner avec environ 400 ruches pour atteindre la rentabilité de son exploitation. Le nombre de ruches rend le volume de travail important. Les opérations de transhumance nécessitent de nombreuses opérations de manutention et de déplacements. L'activité d'apiculteur professionnel est un véritable challenge physique et économique cela couplé avec les opérations de transhumance de nombreuses opérations de manutention et de déplacement sont nécessaires l'activité d'apiculteur professionnel est un véritable challenge physique et économique.

La modification des pratiques agricoles est toutefois un obstacle à la santé globale des abeilles par l'utilisation massive de pesticides qui ont des répercussions néfastes sur la santé des colonies. De plus, l'augmentation de la surface des cultures a réduit la surface et la diversité des zones non cultivées entourant les cultures réduisant la quantité et la diversité des ressources disponibles menant à des périodes de disette ou de carence alimentaire pour les abeilles qu'il faut compenser en nourrissant les colonies. Ajouté à cela la progression de

maladies et parasites importés par les échanges internationaux (varoa, loque américaine), ainsi que les défis liés au réchauffement climatique, la pratique de l'apiculture devient un véritable challenge.

De nombreuses solutions de suivi de la santé des abeilles ont vu le jour [3, 4] ces dernières années pour monitorer les métriques importantes de la ruche afin d'en déterminer à distance la santé. Cela a pour objectif de permettre une réduction du nombre d'interventions et de déplacements sur les colonies, et de permettre de réagir au plus vite en cas de menace pour la santé du rucher.

La métrique principalement utilisée est le poids de la ruche, indicateur fiable pour identifier les débuts de miellées : facteur crucial pour réaliser une bonne saison. Cependant cette métrique est difficile à déployer à grande échelle du fait de son coût mais aussi de l'ajout de manutention durant les phases de transhumance, ajoutant du volume et du poids à transporter et à manipuler. Les autres métriques utilisées sont principalement la température et l'humidité dans la ruche, indicateurs de ponte, ainsi que la vidéo plus rarement pour le comptage des abeilles entrant et sortant de la ruche, indicateur d'activité, et d'identification des ressources à la couleur du pollen. Un autre produit développé est un antivol de ruche pour détecter et retrouver des colonies subtilisées. Les solutions proposées dans la littérature et le commerce ont une approche par ruche, ou chaque ensemble de capteurs est autonome pour l'acquisition et la transmission des données.

Dernièrement des enregistrements audios des ruches sont réalisés comme une nouvelle métrique prometteuse car extrêmement miniaturisable et peu coûteuse. L'analyse des fichiers audio a connu un essor avec l'augmentation des publications scientifiques dans les 3 dernières années malgré un nombre de données disponibles limitées [5, 6]. Les premiers travaux audios portent sur l'analyse de la présence d'abeilles, la détection de l'essaimage ainsi que la présence de la reine. L'indicateur de présence de la reine est un paramètre crucial de la santé globale d'une colonie et il est également celui qui connaît le plus d'analyses avec de nombreuses techniques de machine learning basées sur la donnée.

Objectif de l'étude

Cette thèse CIFRE a pour objectif de réaliser un système de suivi de la santé de la ruche répondant aux besoins des apiculteurs professionnels, c'est à dire abordable, entièrement autonome sur batterie, capable de fonctionner dans les zones rurales et montagneuses reculées où sont souvent situés les ruches. Ces travaux sont réalisés en collaboration avec l'entreprise "RF Innovation, Toulouse", financeur du projet et spécialiste de la conception de matériel électronique embarqué avec communication sans fil.

Afin de rendre le système cohérent à déployer sur des ruches, il tient de remettre en perspective le prix des colonies avec le prix du système d'instrumentation. Les contraintes de budget poussent donc le développement vers une

solution peu coûteuse, et donc équipée de capteurs dont les performances sont limitées. La logique de l'étude est de ne pas subir ce manque de performances mais d'améliorer les mesures avec des compensations logicielles.

Une approche globale au rucher est choisie avec pour objectif de minimiser le coût et d'augmenter le nombre de ruches instrumentées. Le système de collecte est développé en centralisant toutes les fonctions coûteuses et techniques des capteurs dans une gateway tels que le stockage et la transmission des données. Cela permet de réaliser des capteurs simples au niveau matériel pour instrumenter un nombre important de ruches sans faire exploser le coût de la solution tout en rendant possible la possibilité d'embarquer les corrections et les exploitations logicielles au plus près de la mesure, pour en faire des *smart sensors*. Cela est rendu possible par l'utilisation de microcontrôleurs rapides mais peu énergivores. La présentation de l'architecture et des technologies du système repose sur la lecture de la littérature et l'expérience de développement de RF Innovation.

Le capteur de référence dans la pratique actuelle de l'apiculture étant la balance pour ruche, les premiers développements se concentrent sur la reproduction des performances observées sur le système de référence existant (Balance Capaz). La conception de la balance et l'amélioration de ses performances constituent une première réalisation de cette étude. Le principal problème de précision de la balance provient de la dérive thermique, avec un déphasage entre les variations thermiques et les mouvements de la mesure. Ces variations non souhaitées sont compensées par des méthodes de régression polynomiales avec retard appliquées aux mesures. Le besoin spécifique des apiculteurs d'une mesure unique par jour et par ruche permet le sous échantillonnage des données pour créer un indice de masse journalier (IMJ) encore plus fiable. Les cas d'usage de la mesure de masse en continu sont très rares, et la littérature ne fait pas mention de travaux similaires dans d'autres applications que la mesure du poids des ruches mais sans aborder le problème de dérive, ce qui en fait un projet particulièrement complexe.

Le second volet de cette étude se concentre sur l'exploitation des enregistrements audio des colonies, avec comme objectif de substituer les balances par des capteurs audio, moins coûteux et plus embarqués. Les travaux de la littérature scientifique présentent des méthodes qui donnent de très bons résultats pour identifier la présence de la reine dans les colonies en utilisant les méthodes classiques de machine learning, spécialement les réseaux de neurones convolutifs (CNN), ainsi que les machines à vecteurs de supports (SVM). Cependant la question du volume de données et de la généralisation des modèles à de nouvelles ruches ne sont aujourd'hui pas traités dans la littérature et font donc de ces études de mauvais candidats pour la réalisation d'un produit industriel où, par définition, de nombreuses ruches non utilisées pour entraîner les modèles seront évaluées. L'objectif de l'étude devient donc d'évaluer les méthodes existantes en extrapolation et de proposer une solution à ce problème. Le développement est réalisé sur la totalité des données disponibles

publiquement, rassemblées en un dataset normalisé, pour maximiser les différences entre ruches et rendre la solution robuste. Malgré cette collecte de données, la diversité de ruches reste faible.

Dans la totalité des travaux, l'approche est toujours d'essayer d'obtenir les meilleurs résultats possibles tout en ayant une solution viable en termes de calcul, énergie dépensée et temps de développement, pour rester proche de l'industrialisation de la solution développée et donc de réaliser une solution ayant du sens pour le marché cible : un apiculteur professionnel.

Plan de l'étude

Ce mémoire de thèse est composé de quatre chapitres :

Le premier chapitre introduit la logique d'instrumentation utilisée en électronique ainsi les concepts généraux de ce qu'est une abeille, sa biologie, sa vie en colonie et les opérations réalisées par les apiculteurs sur les ruches. Il présente également les besoins des apiculteurs professionnels et donnent ainsi la direction de l'étude pour répondre à ce besoin.

Le deuxième chapitre présente une architecture standard pour réaliser la collecte de données de différentes natures sur les ruches. Il se concrétise par la réalisation de deux capteurs, une balance et un microphone audio. Les données collectées sont montrées pour illustrer le succès du déploiement de la solution.

Le troisième chapitre met en évidence les défauts de la balance *low cost* en caractérisant les dérives thermiques de la chaîne d'acquisition et du capteur employé. Puis des méthodes de régression linéaire avec profondeur mémoire sont mises au point pour compenser cette dérive, sur les données en continues, puis en sous échantillonnant à une journée pour améliorer encore les performances.

Le quatrième chapitre est une exploitation des fichiers audio de colonies d'abeilles pour déterminer automatiquement la présence de la reine dans une ruche. Un dataset de ces données rassemblées est présenté pour faciliter l'utilisation des techniques d'apprentissage. Après avoir reproduit les performances de la littérature pour identifier le label de la présence de la Reine sur les fichiers audios et montré un problème d'extrapolation à de nouvelles ruches, une méthode inspirée du contrastive learning est utilisée pour palier à ce problème et généraliser le modèle.

La conclusion et les perspectives présentent une synthèse des travaux réalisés, des résultats obtenus mais aussi des limites rencontrées par l'étude. Ces limites sont ensuite présentées comme point de départ de futurs travaux de recherche.

Chapitre 1

Électronique et abeilles

Contents

1.1	Instrumenter en électronique	6
1.2	Qu'est ce qu'une abeille?	7
1.2.1	Colonie d'abeille	8
1.2.1.1	Biologie et fonctionnement de l'abeille	8
1.2.1.2	Cycle naturel d'une colonie d'abeilles	10
1.2.1.3	Cycle dans une ruche en production	11
1.2.2	Pour un apiculteur professionnel	13
1.2.2.1	Structure d'une ruche	13
1.2.2.2	Organisation d'une saison	14
1.2.2.3	La balance : métrique la plus répandue	15
1.2.2.4	Les besoins d'un apiculteur professionnel	16
1.3	Conclusion	16

Ce chapitre explore la logique de mesure et l'exploitation des données dans les réseaux de capteurs intelligents, puis fournit des éléments de base sur le fonctionnement des colonies d'abeilles. Il vise à offrir un aperçu vulgarisé et non exhaustif, nécessaire à la compréhension des choix opérés dans cette étude. La première partie présente la logique de réalisation des systèmes d'instrumentation intelligents ("smart sensing") qui associent capteurs en réseau à des architectures reconfigurables matérielles et/ou logicielles ultra low power. Puis les fondements biologiques et les dynamiques d'une colonie d'abeilles en tant que super-organisme au cours d'une saison sont expliqués. Ensuite, le chapitre expose la logique de gestion des colonies par les apiculteurs pour assurer une production apicole efficace. Enfin, il décrit les principales métriques utilisées par les apiculteurs professionnels et les méthodes de mesure actuellement en place pour identifier les verrous et guider la démarche de recherche de ces travaux de thèse.

1.1 Instrumenter en électronique

Avec l'avènement des architectures versatiles et ultra lowpower en électronique, de nombreux systèmes de métrologie se sont développés, allant des applications industrielles (usines de production, gestion des stocks, supervision de l'environnement...) jusqu'à l'usage domestique (sport, domotique...). Ces solutions reposent sur des capteurs de moins en moins coûteux, autonomes en énergie et embarqués. L'essor des applications IoT s'est accompagné d'une facilité accrue pour collecter des données. En partant du principe que la donnée a de la valeur, même si celle-ci n'est pas encore clairement définie, de nombreux systèmes sont instrumentés pour collecter et stocker les données dans le cloud, en vue de leur analyse ultérieure. Bien que ces solutions de prospection soient utiles, elles restent énergivores : mesure, transmission, stockage, traitement et mise à disposition des données via des interfaces web ou des applications mobiles.

Pour réduire l'impact énergétique et environnemental de ces systèmes, une alternative consiste à intégrer l'intelligence directement au niveau du capteur et à ne transmettre et stocker que la valeur extraite. Ces approches sont connues sous le nom de *Smart Sensing* ou *Edge Computing*.

Le *Edge Computing* désigne une approche décentralisée de traitement des données, réalisée au plus près de la mesure dans l'objectif de réduire la latence et la consommation de bande passante en effectuant les traitements localement. Cette méthode améliore la réactivité des systèmes, renforce la confidentialité des données et assure une plus grande résilience des infrastructures [7, 8].

Le *Smart Sensing* consiste à intégrer des capacités de traitement et d'analyse avancées directement dans les capteurs. Cette approche permet de collecter, traiter et interpréter les données de manière autonome, en combinant des capteurs à bas coût avec des algorithmes embarqués. Elle vise à exploiter localement les mesures, optimiser la consommation d'énergie et prendre des décisions localement sans nécessiter de traitement externe intensif [9, 10].

Déployer ces méthodes pose toutefois des défis, notamment en ce qui concerne le dimensionnement du capteur rendu intelligent, qui doit être capable d'embarquer l'intelligence nécessaire à l'extraction de la valeur des données. Cette difficulté est d'autant plus grande dans les domaines exploratoires, où la valeur des données est supposée, mais où les solutions d'extraction ne sont pas encore définies ni mises en œuvre. Procéder à un double déploiement — d'abord pour collecter les données, puis un second pour ajuster les spécifications matérielles à la solution logicielle développée — est rarement réaliste en électronique. Il est donc essentiel de concevoir un réseau de capteurs optimisé dès le premier déploiement, sans surdimensionner les caractéristiques (mémoire, vitesse d'exécution, accélérations matérielles...), ce qui augmente-

rait les coûts.

Ces solutions sont aujourd’hui possibles et simplifiées grâce aux micro-contrôleurs récents, qui offrent de plus en plus de puissance de calcul et des mémoires RAM et Flash internes miniaturisées importantes et à faible coût. Ces avancées permettent d’exécuter des logiciels plus complexes [11, 12], très souvent reconfigurables Over The Air, tout en maintenant une consommation énergétique de la solution extrêmement faible. Il devient ainsi viable de déployer des solutions de *Smart Sensing* ou de *Edge Computing*.

Dans un capteur, la transmission des données via un lien radio est souvent le poste le plus coûteux en énergie. Lorsque le traitement local des données est moins énergivore que leur transmission, le bilan énergétique est favorable, augmentant ainsi l’autonomie du capteur.

Réduire la quantité de données transmises diminue également les besoins énergétiques au niveau du réseau et réduit l’infrastructure de stockage nécessaire, ce qui minimise l’impact global de la solution. Cette démarche s’inscrit dans la logique de la *Right Tech* [13, 14], qui vise à développer les technologies adaptées aux bons endroits, dans le but de concevoir des solutions économiques en termes de coûts et d’énergie, et donc plus écologiques.

L’étude vise à améliorer les pratiques apicoles en instrumentant les ruches, afin de faciliter la prise de décision des apiculteurs grâce à l’accès à des données pertinentes. L’objectif est d’augmenter la qualité de la production de la ruche, d’anticiper les besoins des abeilles et d’alléger le travail des apiculteurs en proposant un suivi à distance et pour chaque ruche. L’approche repose sur l’utilisation de deux capteurs (masse et audio), avec une logique de *Edge Computing* pour traiter les données à la source.

Au cours des dix dernières années, diverses solutions de suivi à distance des ruches ont été développées, principalement destinées aux apiculteurs amateurs. Issues de la recherche ou de l’industrie, ces solutions sont souvent coûteuses et focalisées sur la collecte de grandes quantités de données, sans véritablement faciliter la prise de décision. L’étude propose une nouvelle approche, à destination des professionnels, visant à transmettre des mesures précises et pré-traitées en extrayant leur valeur pour en simplifier l’interprétation, tout en minimisant leur volume.

1.2 Qu’est ce qu’une abeille ?

La section suivante fournit aux néophytes quelques bases sur l’apiculture, le contexte dans lequel évoluent les apiculteurs et leurs objectifs au fil de la saison afin de comprendre les motivations derrière les décisions de développement de l’étude.

1.2.1 Colonie d'abeille

Une colonie d'abeilles est un super organisme composé d'une reine, d'abeilles ouvrières et de faux bourdons (le mâle de l'abeille). Ce super organisme s'organise pour assurer la survie de la colonie ainsi que sa multiplication. La population et les activités de la colonie évoluent en fonction de la saison afin d'assurer les différentes tâches nécessaires au bon développement de la ruche. Ainsi une colonie d'abeilles est un système évolutif avec lequel un apiculteur va composer pour essayer d'extraire un produit de ses ruches. Les productions possibles d'une ruche sont nombreuses et variées, du miel, de la cire, du pollen et de la gelée royale (très connus et rependus) , au venin et à la propolis. Dans certaines cultures les larves sont consommées comme source de protéine. Les informations proposées par ce chapitre sont issues des livres introductifs à l'apiculture [15, 16, 17].

1.2.1.1 Biologie et fonctionnement de l'abeille

Une colonie est composée de trois variétés d'abeilles :

- La reine : il y a une reine d'abeilles par ruche, c'est elle qui est la mère de toutes les abeilles de la colonie, sa principale fonction est de pondre les oeufs et d'assurer la population suffisante dans la colonie pour sa survie. La reine se fait féconder par plusieurs mâles afin d'assurer une diversité génétique au sein des abeilles de la colonie. Une reine d'abeilles a une durée de vie d'environ 3 ans dans une colonie naturelle et sont exploitées en moyenne 2 ans dans les ruches en production professionnelle contre 5 ans dans les années 80.
- Les abeilles ouvrières : abeilles femelles non matures sexuellement, ne peuvent pas pondre. Elles assurent les fonctions productives de la ruche telles décrites dans la figure 1.1 Leur durée de vie varie en fonction de la saison, d'environ 30 jours en saison active (printemps, été) et jusqu'à 6 mois pendant la période hivernale.
- Le faux bourdon : Abeille mâle de la colonie. Sa tâche principale est de féconder une reine vierge au moment du remérage des colonies. De ce fait les mâles ne sont pondus qu'au moment de la période de reproduction de la ruche.

Le schéma 1.2 illustre la durée de développement et de maturation des différentes castes d'abeilles. Ces durées jouent un rôle crucial en définissant l'inertie de fonctionnement d'une ruche. En période d'abondance de ressources, un décalage de 21 jours se produit avant une augmentation significative de la population de la ruche, une donnée essentielle pour les apiculteurs dans une optique de production, afin de ne pas manquer les périodes de rendement en raison d'un décalage du pic de population des colonies. De plus, ces durées fournissent une indication sur la période sans ponte en cas de problème lié à la reine (maladie, décès...), où il faut compter environ 25 à 30 jours avant

La vie d'une ouvrière

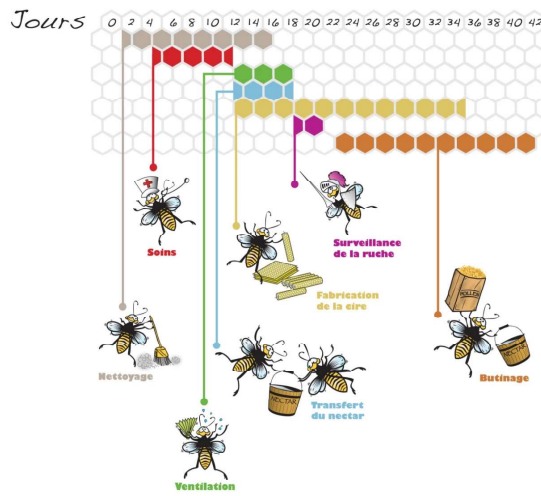


FIGURE 1.1 – Description des métiers de l'abeille en fonction de son âge en saison

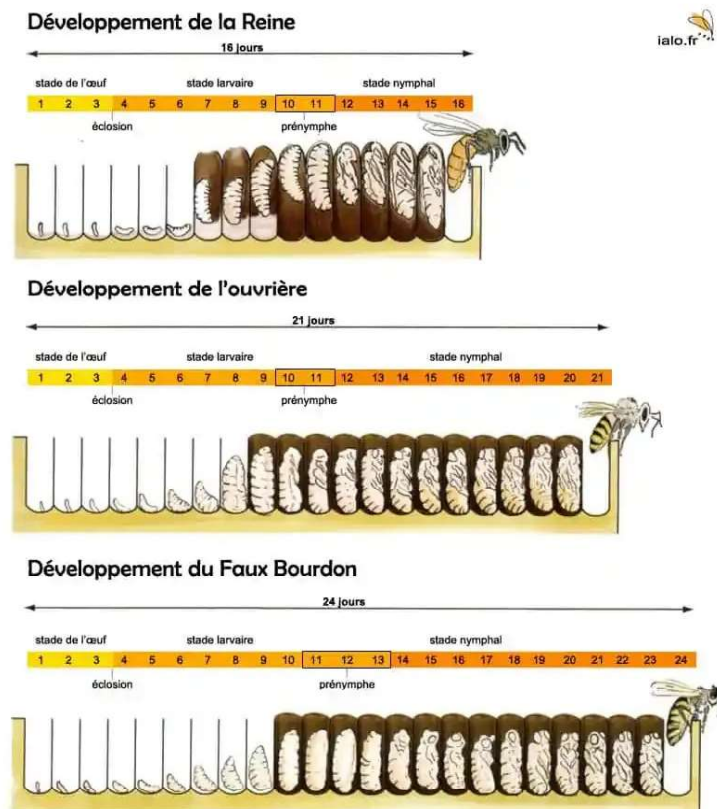


FIGURE 1.2 – Cycle et temps de naissance des abeilles, faux bourdon et reine

que la nouvelle reine ne reprenne la ponte, et 21 jours supplémentaires pour la naissance des nouvelles ouvrières. Cette période prolongée de stress pour la colonie souligne l'importance pour l'apiculteur de minimiser les risques liés à la santé de la reine. Lors d'un remérage naturel, l'ancienne reine quitte la ruche seulement à la naissance de la nouvelle reine, ce qui réduit la période sans ponte pendant la fécondation, mais entraîne tout de même un ralentissement temporaire de la colonie.

1.2.1.2 Cycle naturel d'une colonie d'abeilles

Dans la nature, l'objectif d'une colonie d'abeilles est de survivre et de se reproduire afin d'assurer la continuité de l'espèce. Le cycle de fonctionnement d'une colonie est décrit dans la figure 1.3. Chronologiquement, à la sortie de l'hiver, la colonie reprend progressivement son activité avec l'augmentation des ressources disponibles (nectar, pollen). Avec l'arrivée du printemps, la population croît de manière exponentielle, et la reine pond jusqu'à 2000 œufs par jour. Une fois la colonie en plein fonctionnement, la reine commence à pondre des mâles pour préparer la période de reproduction. Environ 45 jours après la ponte des œufs mâles, ceux-ci deviennent sexuellement matures, marquant le début de la période de reproduction de la colonie.

De nouvelles reines sont alors élevées par les abeilles. Au moment de la naissance de la nouvelle reine, l'ancienne reine quitte la ruche avec une partie des abeilles : c'est l'essaimage. L'essaimage a souvent lieu en anticipation d'un manque d'espace dans la ruche ou le développement de la colonie est limité par la quantité d'espace disponible. L'ancienne reine et son essaim partent à la recherche d'une nouvelle zone pour développer une colonie. Cet essaimage se produit au moment où les ressources sont abondantes, ce qui permet de soutenir le développement de la nouvelle ruche. La nouvelle reine reste dans l'ancienne ruche avec les abeilles les plus jeunes. Elle est fécondée dans les 15 jours suivant sa naissance. Une fois fécondée, elle commence à pondre et la ruche reprend alors un cycle normal de fonctionnement.

Il est important que la reproduction se déroule durant une période favorable, car pendant la fécondation de la nouvelle reine, celle-ci ne pond pas. Cela entraîne une diminution des naissances et un ralentissement temporaire de l'activité de la colonie. De même, l'essaim qui s'installe aura une période sans naissance de 21 jours, correspondant au temps nécessaire à l'éclosion des œufs pondus par la reine (déjà fécondée).

En fin de saison, les deux colonies ainsi créées se concentrent sur la création de réserves de miel pour passer l'hiver. Lorsque les ressources diminuent et que les températures baissent, les abeilles restent dans la ruche et forment une "grappe". La reine interrompt la ponte, et la colonie passe l'hiver en consommant les réserves de miel, nécessaires à leur alimentation et à l'énergie dépensée pour maintenir la chaleur de la colonie. La population décroît progressivement au fil de l'hiver. Le cycle recommence l'année suivante.

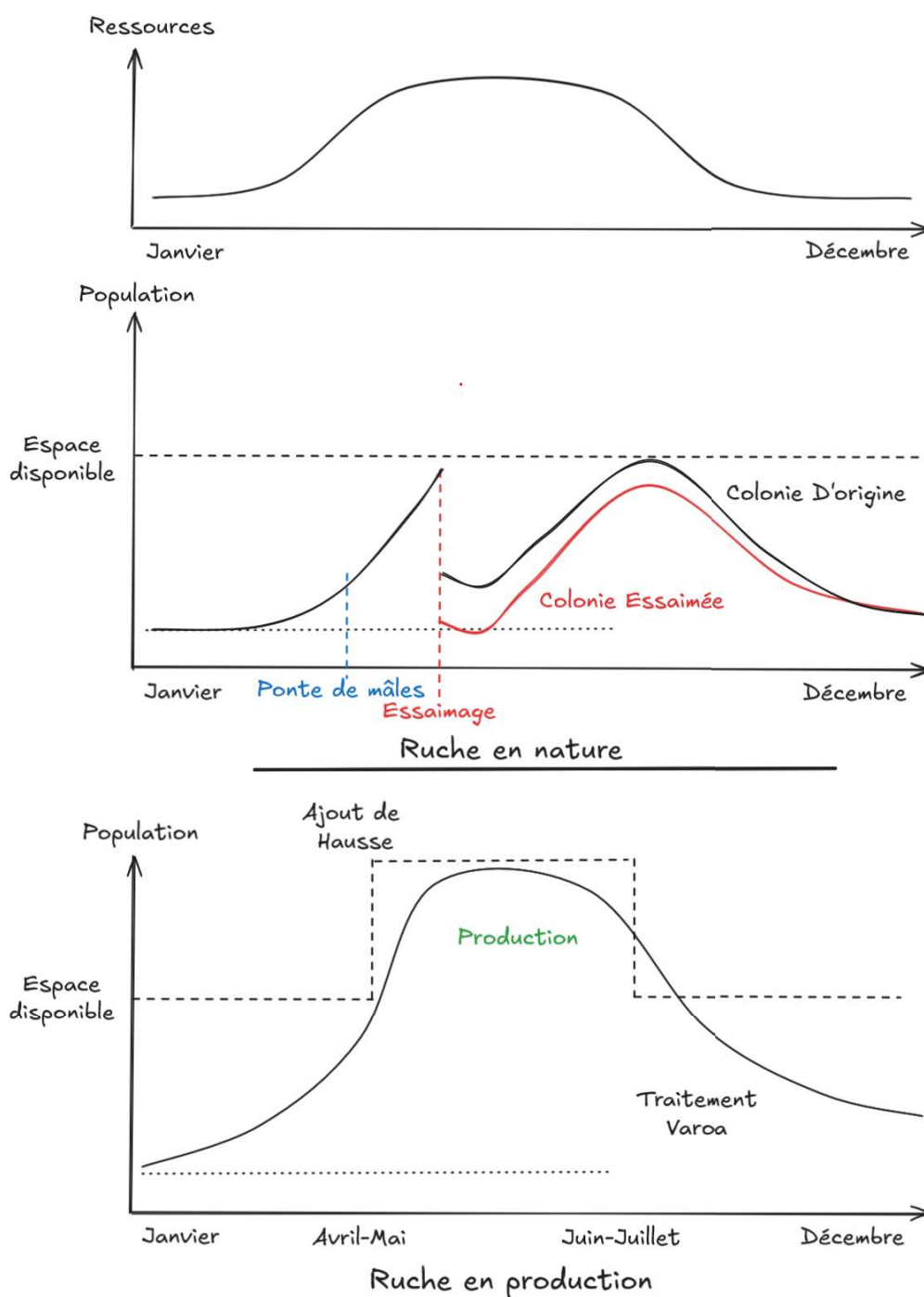


FIGURE 1.3 – Cycle de vie d’une ruche en fonction de la saison, pour une ruche en autonomie et une ruche en production

1.2.1.3 Cycle dans une ruche en production

Cas d’une ruche en production de miel Dans une ruche en production, la période la plus favorable est celle d’abondance de ressources, qui correspond

également à la période d'essaimage. Or, un essaimage conduit à la séparation de la ruche en deux, et donc à la perte d'une partie de la colonie. L'apiculteur cherche à éviter l'essaimage pour profiter pleinement de cette période pour la production de miel. La méthode utilisée consiste à ajouter une hausse sur le dessus de la ruche, ce qui augmente l'espace disponible pour les abeilles, qui continuent ainsi d'augmenter en population 1.3. Cette augmentation de population se traduit par une augmentation des ouvrières qui travaillent dans la ruche, produisant du miel en abondance au fil des miellées. Les hausses ajoutées sont ainsi remplies de miel. En fin de saison, les hausses sont retirées et extraites, puis le miel est mis en pot. Une période est laissée aux abeilles pour stocker des réserves de miel dans le corps de ruche, nécessaires pour passer l'hiver.

En pratique, les ressources disponibles ne forment pas un plateau constant, mais oscillent en fonction des floraisons décalées des différentes fleurs et essences (colza, acacia, châtaigner, tournesol, etc.). Pour maximiser les périodes de forte ressource, l'apiculteur pratique la transhumance. De plus, ces décalages de périodes de floraison permettent aux apiculteurs de produire différents types de miel.

Reproduction et multiplication des ruches Étant donné que l'apiculteur cherche à éviter les essaimages sur ses ruches en production, il lui faut développer une autre méthode pour reproduire ses ruches, et remplacer les reines vieillissantes. Pour cela la méthode la plus courante est de réaliser un élevage de reines.

Le principe repose sur le fait que lorsqu'une colonie se retrouve sans reine (maladie, vieillesse...), les abeilles ouvrières élèvent plusieurs nouvelles reines sur des oeufs déjà pondus pour remplacer la reine défaillante. Les abeilles élèvent plusieurs reines afin de s'assurer au moins un succès. En pratique, quand la première reine naît, elle élimine les autres candidates et la ruche reste donc avec une reine unique. L'apiculteur recrée ces conditions en orphelinant une ruche en en prélevant la reine. Les abeilles élèvent alors plusieurs nouvelles reines. L'apiculteur empêche l'élimination des différentes reines en prélevant les reines vierges de la ruche. Les reines vierges sont placées dans des nucléis (ruches miniatures) durant la période de fécondation. Une fois la ponte reprise, les reines peuvent être utilisées pour remplacer une reine défaillante, ou bien créer de nouveaux essaims sans risquer l'essaimage. Cette méthode évite le départ d'une partie de la colonie car l'ancienne reine est retirée artificiellement. Réaliser un élevage de reine permet de reproduire son cheptel, mais également de réaliser de la sélection génétique sur certains caractères : production, douceur, qualité des constructions, nettoyage...)

1.2.2 Pour un apiculteur professionnel

1.2.2.1 Structure d'une ruche

Afin de pouvoir exploiter au mieux ses abeilles tout en minimisant la gêne apportée par les visites et les récoltes, un standard de fonctionnement semble avoir émergé : la ruche verticale avec cadres. Ce standard se décline en plusieurs factures, Dadant, Langstroth et Warré. Le concept est le même pour ces formats, qui est représenté en figure 1.4

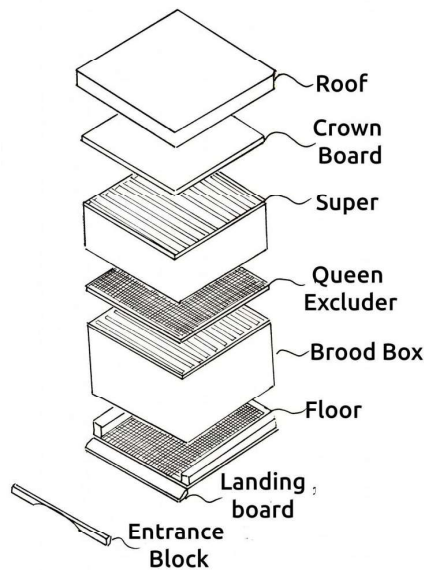


FIGURE 1.4 – Éléments d'une ruche en format vertical

Une ruche verticale se compose de plusieurs éléments, chacun ayant une fonction pour la gestion des colonies d'abeilles.

- **Le corps de ruche** : Il s'agit de la partie principale de la ruche où les abeilles développent la colonie et stockent les ressources pour l'hiver. La ponte est réalisée dans cette partie de la ruche.
- **Les cadres** : Le corps de ruche et les hausses sont divisés par des cadres, support pour la cire dans laquelle sont pondus les oeufs et stockées les ressources. Ces cadres sont amovibles et permettent l'inspection des ruches ainsi que l'extraction des ressources sans abîmer les structures ni déranger les abeilles. Il s'agit de l'élément fondateur du principe de l'apiculture moderne.
- **La hausse** : Placée sur ou sous le corps de ruche, elle est utilisée pour proposer aux abeilles un surplus d'espace dans lequel elles vont stocker le miel excédentaire produit. C'est dans la hausse que l'apiculteur prélève le miel sans perturber le développement du couvain. La hausse est souvent séparée du corps de ruche par une grille à reine, avec des intervalles dimensionnés pour que les abeilles puissent passer au travers, mais pas

la reine, plus grosse. Cette grille garantit l'absence de couvain dans la hausse et permet donc de récolter uniquement du miel.

- **Le plateau de fond et couvre cadre** : Le plancher, situé à la base de la ruche, est grillagé pour permettre la ventilation pour la régulation thermique et l'évacuation des débris de la colonie. Le couvre cadre ferme la ruche par le dessus. Souvent doublé d'un isolant, il protège des fortes chaleurs en été et propose une isolation thermique pour l'hiver.
- **Le toit** : Il recouvre l'ensemble de la ruche et la protège des intempéries. Généralement en métal ou en bois, il isole la ruche et assure une protection contre la pluie, le vent ou le soleil.

Chaque composant de la ruche Dadant est essentiel à la bonne gestion de la colonie, permettant à la fois de maintenir un environnement optimal pour les abeilles et de faciliter le travail de l'apiculteur.

1.2.2.2 Organisation d'une saison

Organisation des ruches : L'apiculteur rassemble plusieurs ruches en un seul lieu nommé le rucher. Cela lui permet de grouper les colonies ensemble et ainsi maximiser la production tout en limitant les déplacements. En revanche il n'est pas possible de placer trop de ruches en un seul lieu sans prendre le risque de voir les colonies se faire de la concurrence sur les ressources à proximité, allant même jusqu'au pillage des colonies les plus faibles en temps de disettes. La taille moyenne d'un rucher est d'environ 40 à 60 ruches en un seul lieu. Un apiculteur professionnel doit donc avoir plusieurs ruchers différents pour gérer ses colonies.

transhumance Afin de réaliser une production importante, un apiculteur peut réaliser de la transhumance, c'est à dire déplacer les ruches à proximité des ressources. Les cultures et essences fortement mellifères ont des périodes de floraisons légèrement différentes en fonction de la latitude et de la météo. En plaçant les ruches du sud vers le nord au fil des floraisons, cela permet de maximiser le temps de production tout en évitant les chutes de ressources quand les monocultures fanent alors que les ruches sont au maximum de leur population.

Les principales opérations de l'apiculteur L'apiculteur qui cherche à réaliser une production sur ses ruches doit s'assurer de la santé et de la force de ses colonies et exploiter la saison au mieux de ses possibilités. Les principales actions sur les ruches sont les suivantes :

- **Nourrissement des colonies** : en période de manque de ressources, typiquement en fin d'hiver moment où les réserves peuvent manquer si le printemps tarde à arriver.

- **Traitement des maladies** : Étape obligatoire au moins une fois par an depuis que le varoa, parasite de l'abeille, est arrivé en France.
- **Remplacement des vieux cadres** : bien que les abeilles nettoient et entretiennent leur ruche, il est nécessaire de remplacer régulièrement les vieux cadres car la qualité de la cire, corps gras qui accumule les déchets, se dégrade avec le temps.
- **Placement des hausses** : au moment opportun de forte miellée, l'apiculteur place les hausses sur ses ruches afin de réaliser une production de miel. Une fois les hausses pleines ou au terme de la miellée, les hausses sont récoltées et extraites.
- **Élevage des colonies pour la reproduction** : afin d'entretenir une génétique de qualité sur les ruchers et compenser la mortalité des ruches, un élevage est presque systématiquement réalisé permettant la création de nouvelles colonies et le remplacement des reines vieillissantes mettant en péril les colonies.

1.2.2.3 La balance : métrique la plus répandue

Le nombre de capteurs adressant les problématiques des apiculteurs professionnels sont limités en diversité de mesure mais également en choix de matériel. Après échange avec quelques apiculteurs professionnels, il apparaît que la métrique utilisée est presque exclusivement la masse (poids de la ruche) sur le système Capaz, précis mais onéreux. Le système de mesure de masse le plus utilisé aujourd'hui par les apiculteurs professionnels est le suivant :

- Utilisation d'une ou deux balances de précision extrêmement chère par rucher
- Placer la ruche la plus forte sur cette balance, celle qui est en meilleure santé avec le plus d'abeilles
- Observer l'évolution de la masse journalière de la ruche pour en déduire la présence d'une miellée
 - Si la ruche gagne du poids, les abeilles font du miel
 - Si la ruche perd du poids, il est possible que quelque chose n'aille pas bien

Le coût important de chaque balance (ordre de grandeur 1000€) est à remettre en perspective avec le prix d'un essaim d'abeilles (ordre de grandeur 130€). Pour rentabiliser au maximum les balances, les apiculteurs vont parfois placer deux ruches l'une sur l'autre sur la balance afin de créer un moyennage sur plus d'une ruche et ainsi éviter un faux positif ou négatif. Cette solution est efficace dans la prospection d'une miellée : placer une ruche sur une balance dans un terrain où on attend une miellée prochaine, et lorsque la ruche commence à prendre du poids, on apporte les ruches à ce lieu pour produire le miel avec le meilleur timing possible. Cela permet aussi d'identifier la fin de cette même miellée et indiquer à l'apiculteur que la transhumance de ses

ruches est à reprendre en un autre lieu. En revanche, le système est mauvais pour avoir des informations sur la santé de la ruche : si le lieu rencontre un problème important, ce seront les ruches les plus faibles qui auront des difficultés en premier. De plus, si jamais la ruche placée sur la balance perd du poids, cela peut mener à l'interprétation que la miellée est terminée alors qu'il s'agit d'un élément isolé lié à une mauvaise santé passagère de la ruche observée. Dans un contexte de transhumance, il faut éviter à l'apiculteur d'effectuer des déplacements inutiles.

1.2.2.4 Les besoins d'un apiculteur professionnel

Après échanges avec des apiculteurs professionnels, il apparaît clairement que la recherche de multiplicité des données n'est pas un objectif. Un apiculteur professionnel en production de miel possède en moyenne 400 ruches. Si chacune de ces ruches produit de nombreuses données le temps de traitement et d'analyse de ces données devient alors une charge au lieu d'un support car il ajoute un temps de travail à des journées déjà chargées en saison . Il ressort donc que la demande est d'avoir une mesure unique à un horaire fixe pour l'entièreté des ruches, et à défaut d'une mesure, idéalement un indicateur déjà interprété, bien que de nombreux apiculteurs ne souhaitent pas avoir "une machine" pour réaliser des diagnostics et prendre des décisions à leur place. La consultation des métriques standard réalisée par les apiculteurs est le soir en fin de journée afin de déterminer le planning du lendemain en gérant les priorités dans l'ordre d'importance.

La logique de mesures mise en place est de remonter des mesures à intervalle régulier, puis d'extraire les informations utiles de ces mesures pour essayer de proposer des indicateurs simples et fiables. Dans un mode avancé, les mesures complètes sont proposées pour réaliser des analyses et debriefing de la saison apicole à la période hivernale lorsque la charge de travail est moindre.

Le second besoin essentiel est de réaliser une gamme de produits abordables en prix. Les outils de mesure sont à mettre en perspective avec le coût d'une colonie d'abeilles 160-180€. Au regard de ce prix, il devient inintéressant de placer un système de surveillance qui rendrait le modèle économique de l'apiculture non viable. Chaque capteur doit donc avoir un prix raisonnable pour que le service rendu ne dépasse pas les coûts économisés en déplacement sur les ruchers dans le cadre de la transhumance ainsi que le prix d'une colonie perdue.

1.3 Conclusion

En saison apicole favorable en terme de ressources, la population d'une colonie d'abeilles augmente, puis se divise afin de créer plusieurs colonies, assurant ainsi la survie de l'espèce. Afin de préparer l'hiver, les colonies emmagasinent des réserves pour se nourrir durant la période hivernale. Le travail

d'un apiculteur est de gérer la population de ses colonies pour essayer d'éviter la division naturelle de ses colonies et exploiter le comportement naturel de sur-emmagasiner les ressources des abeilles pour extraire de ses colonies une production de la ruche (miel, pollen, cire...). De plus, l'apiculteur veut protéger ses colonies des maladies et des aléas de santé. Pour cela, il met en place des visites régulières pour l'inspection de ses colonies et appliquer les opérations nécessaires à la production. Pour faciliter ce travail et limiter les déplacements inutiles, des capteurs de mesures peuvent être placés pour diagnostiquer à distance la situation de santé des ruches, spécialement en périodes de transhumance (où les ruches peuvent être placées à plusieurs centaines de kilomètres). La métrique la plus utilisée est une balance. Elle est la mesure cible de cette étude couplée à la mesure audio.

1.3. Conclusion

Chapitre 2

Collecte et affichage des données

Contents

2.1	Approche générale	20
2.1.1	Approche de la littérature et de l'industrie	20
2.1.2	Architecture présentée	22
2.2	Choix des technologies radio	24
2.2.1	Protocole de communication au sein du rucher	24
2.2.2	Protocole de communication longue portée	25
2.2.3	Format des données : API standard	25
2.3	Développement matériel et logiciel	27
2.3.1	La Gateway	27
2.3.1.1	Architecture matérielle	27
2.3.1.2	Architecture logicielle	29
2.3.1.3	Consommation et durée de vie	30
2.3.1.4	Durée de vie de la batterie sans le lien 868MHz	33
2.3.2	Les capteurs	34
2.3.2.1	Architecture standard	34
2.3.2.2	La balance	35
2.3.2.3	La capteur audio	39
2.3.2.4	Autonomie sur pile des capteurs	40
2.4	Déploiement du système	41
2.4.1	Données de masse	41
2.4.2	Données audio	45
2.5	Transmission et Stockage des données	46
2.5.1	Protocole MQTT	47
2.5.2	Base de données	48
2.5.3	Web architecture pour le stockage des données	48
2.6	Visualisation des données	49
2.6.1	Utilisation des données pour la recherche	50

2.6.2 Application Web pour les apiculteurs	50
2.7 Conclusion	53

Le premier challenge du monitoring des ruches à distance est la collecte de données fiables. Dans cet esprit, le développement d'une architecture versatile est pensé pour instrumenter les ruches en s'inspirant de la littérature et en observant des produits industriels utilisés par les apiculteurs. Cette architecture de compensation et d'observation sur des temps longs est mise en œuvre autour de deux jauges de contraintes ; réalisant la fonction balance qui est le capteur de référence du milieu apicole, puis étendue à un capteur audio pour corrélérer l'activité production de miel et vie de la ruche. Le développement de l'architecture se heurte à des contraintes de cas d'usage "apiculture" : un environnement parfois éloigné d'une couverture réseau classique, des besoins en autonomie énergétique, des fluctuations climatiques et un impératif bas coût puisque un rucher est composé de plusieurs dizaines de ruches à instrumenter.

Dans une optique de réduction des coûts, le choix est fait d'utiliser des capteurs "dits modestes" que nos travaux vont chercher à améliorer et exploiter par le *edge computing*, nécessitant une puissance de calcul suffisante pour exécuter les algorithmes embarqués, sans pour autant compromettre l'autonomie du produit.

Ce chapitre se décompose en plusieurs temps :

- La présentation d'une approche en rupture avec les méthodes proposées dans la littérature
- Le développement matériel et logiciel des capteurs et de la passerelle vers le réseau
- La description de l'architecture cloud et de l'application web pour le stockage et l'exploitation des données

2.1 Approche générale

2.1.1 Approche de la littérature et de l'industrie

Pour acquérir les données nécessaires au diagnostic des ruches, divers systèmes ont été présentés dans la littérature. Les métriques les plus couramment surveillées incluent la température, l'humidité, l'audio et le poids [18]. Ces mesures sont généralement obtenues via un système tout-en-un qui intègre plusieurs capteurs avec des capacités de stockage et de transmission sur un même microcontrôleur.

Le système le plus répandu à ce jour est une solution filaire tout-en-un, présenté sur la figure 2.1. Ce type d'architecture a été décrit par plusieurs auteurs [19, 20, 21, 22, 23, 24, 25, 26, 27]. Ces systèmes sont souvent construits à partir de cartes d'évaluation assemblées pour réaliser une preuve de concept et initier la collecte de données. Cependant, ces approches présentent des limites, notamment en termes de longévité, avec des problèmes liés à la durée de vie

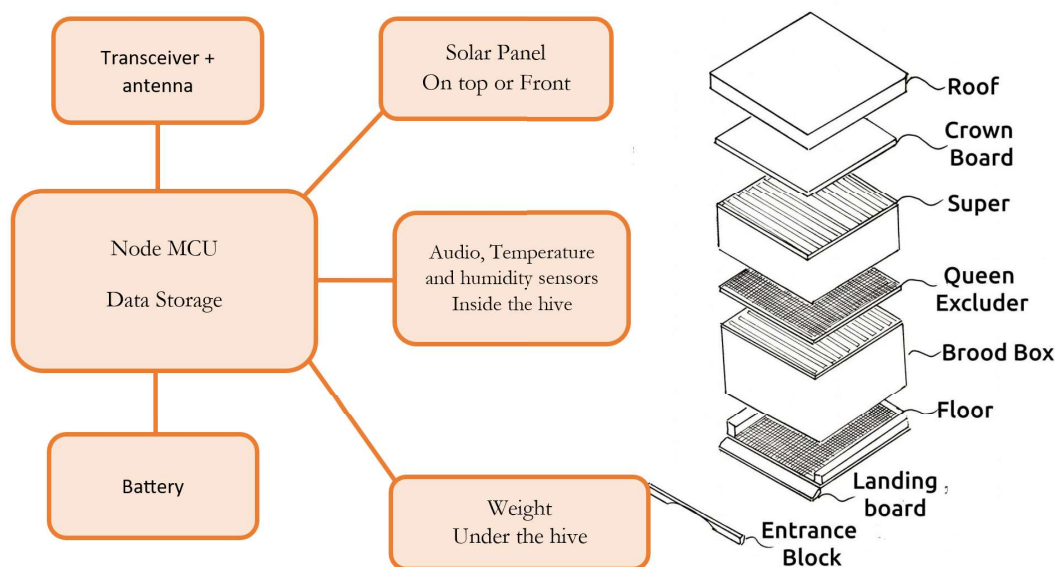


FIGURE 2.1 – Architecture "classique" de collecte des données en système tout-en-un filaire

des batteries et à la durabilité mécanique. Des systèmes extraits de solutions commerciales [28, 29], présentent également majoritairement cette architecture avec une limite en nombre de capteurs par rucher et des connexions filaires bien que la réalisation des capteurs soit plus industrialisée.

Les solutions tout-en-un sont simples à développer au niveau hardware et logiciel car l'ensemble des fonctions sont rassemblées sur un seul nœud qui assure collecte, stockage et transmission. En revanche, elles ont l'inconvénient d'être coûteuses puisque toutes les fonctions sont embarquées et pour instrumenter plusieurs ruches sur un rucher, l'ensemble des fonctions de stockage et transmission, sont dupliquées. Ces solutions sont donc plus adaptées pour le suivi de ruchers de petite taille, en loisir ou pour de la prospection de miellées, quand une ou deux ruches sont placées dans des zones éloignées afin de détecter le début des miellées par la prise de masse et ainsi déplacer le reste des ruchers dans la zone avec le bon timing pour optimiser la production. De même, ces systèmes peuvent indiquer les fin de périodes de fastes ressources.

Lorsqu'on aborde la question des ruches connectées, la source d'énergie est une considération centrale, car elle détermine la fréquence d'échantillonnage réalisable, chaque mesure consommant de l'énergie pour la polarisation du capteur, le temps de mesure et la transmission des données. Une solution évidente consiste à utiliser des panneaux solaires pour fournir de l'énergie à la demande, permettant ainsi une flexibilité totale dans la mesure.

L'utilisation de panneaux solaires comme principale source d'énergie est utilisée dans de nombreuses conceptions pour alimenter le système [20, 21, 25, 27, 23]. Cette approche est efficace si l'énergie produite par les panneaux solaires est égale ou supérieure à l'énergie consommée. Pour tenir compte des

jours nuageux et pluvieux, une marge de sécurité est nécessaire, ce qui exige l'utilisation de panneaux solaires de taille importante. De plus, dans les scénarii où les ruches sont à l'ombre, la point de puissance maximale des panneaux (MPPT) n'est pas atteint, rendant cruciale la marge de taille des panneaux. Dans les applications professionnelles, les ruches sont fréquemment déplacées lors de la transhumance, ce qui augmente le risque de dégâts des panneaux solaires pendant les manutentions.

En revanche, certains systèmes se concentrent sur des solutions basées sur des batteries [22, 24], qui sont plus fiables dans toutes les conditions, mais nécessitent un design précis pour garantir une durée de vie plus longue que la saison apicole. En particulier, les balances autonomes pour ruches sont alimentées par des batteries [30, 31, 29] car leur positionnement sous la ruche empêche une exposition directe à la lumière du soleil. Ces exemples montrent qu'il est possible de se passer de panneaux solaires, ce qui augmente la robustesse et réduit les coûts.

Une exception se trouve dans l'étude [26], où le système est conçu pour un environnement urbain, ayant un accès au réseau électrique, et ne fait donc face à aucune contrainte énergétique. Dans la solution présentée, les panneaux solaires ne sont pas utilisés, une batterie et des piles leur sont préférées pour alimenter le système durant toute la saison apicole, comme cela a été démontré dans 2.3.1.3 et 2.3.2.4.

2.1.2 Architecture présentée

Une approche orientée rucher a été privilégiée afin de pouvoir déployer un plus grand nombre de capteurs sur un même rucher. Ce système repose sur une passerelle (gateway) capable de stocker les données puis de les transmettre avec un serveur distant. Cette solution permet de déployer plusieurs capteurs sur une zone géographique sans devoir répliquer le réseau de communication, ce qui serait plus coûteux et énergivore. Pour simplifier les connexions entre les capteurs et la gateway, nous avons sélectionné des protocoles de communication sans fil. Cette stratégie permet également de placer la passerelle à un emplacement offrant une connectivité de réseau optimale.

Ainsi, dans le cas où les ruches sont dans une zone de "mauvaise réception réseau", la délocalisation à la centaine de mètres de la gateway permise par le lien radio local (par exemple, lien BLE et positionnement sur un arbre), permet de retrouver une bonne connectivité.

Bien que peu utilisée en apiculture, cette approche est bien établie dans d'autres domaines de l'industrie [32, 33] où la donnée issue de capteurs représente une grande valeur. Dans ces applications, l'un des sujets majeurs est la sécurité apportée sur l'authenticité et la confidentialité des données collectées. Pour la solution proposée, la notion de sécurité a été adressée pour parer aux menaces les plus courantes, mais étant donné que les données apicoles sont de faible volume et que le risque de perte de données ponctuelles est acceptable,

les plus gros efforts ont été concentrés sur d'autres aspects. La gateway est présentée en section 2.3.1.

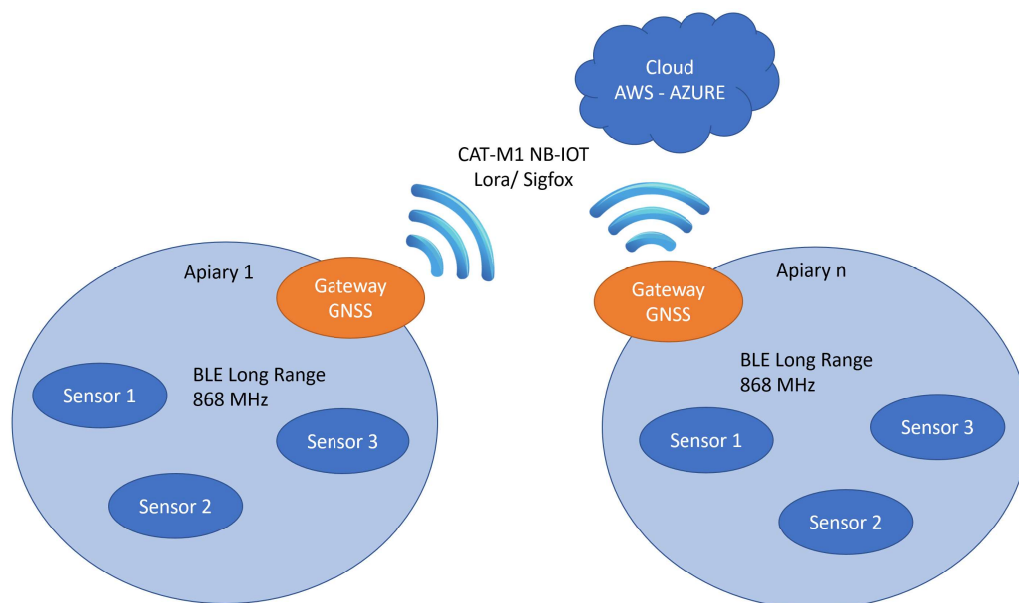


FIGURE 2.2 – Architecture du système de collecte et protocoles de communication

La figure 2.2 présente l'architecture globale du système d'acquisition de données, qui se compose de trois éléments clés :

- **Capteurs d'instrumentation** : placés sur les ruches, ils collectent les données des colonies. Ces capteurs peuvent être de plusieurs natures en fonction du besoin de l'apiculteur
- **Concentrateur de données (gateway)** : recueil des données des capteurs et sert de relais vers la base de données hébergée sur un serveur distant.
- **Base de données et web application** : centralise et stocke les mesures obtenues. Les données sont ensuite accessibles pour pouvoir être exploitées et présentées à l'apiculteur

L'ensemble des capteurs sont standalone et fonctionnent sans fil. Les capteurs sont placés directement à l'intérieur de la ruche. La section 2.3.2.1 présente une architecture standard pour réaliser un capteur compatible.

Cette architecture est flexible et permet également d'adapter le choix des capteurs pour surveiller l'ensemble d'un rucher en réalisant un panachage de différentes métriques dans les ruches. Cette approche permet d'optimiser le coût de la solution tout en instrumentant un maximum de ruches. Ensuite, en corrélant les données des différentes métriques, il est possible de suivre la santé et la tendance globale de la vie des ruches à plus large spectre que sur une approche capteur tout-en-un.

Les protocoles de communication mis en place sont détaillés dans les sections 2.2.1 pour la communication au sein du rucher et 2.2.2 pour la communication entre la passerelle et le serveur.

2.2 Choix des technologies radio

2.2.1 Protocole de communication au sein du rucher

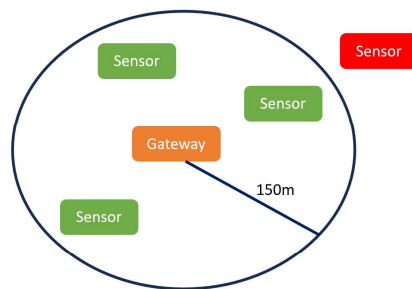


FIGURE 2.3 – Protocole de communication à courte portée : au sein du rucher

Dans un rucher typique, les ruches sont espacées au maximum d'une centaine de mètres les unes des autres. Par mesure de sécurité, le système doit garantir un fonctionnement jusqu'à 150m comme présenté dans 2.3. Deux technologies de communication à courte portée sont sélectionnées avec une utilisation dépendant des besoins en termes de portée et de consommation en énergie :

- **Bluetooth Low Energy Long Range (BLE)** : Cette technologie offre une compatibilité native avec de nombreux capteurs existants tout en garantissant une portée suffisante pour les besoins d'un rucher. Selon les fabricants, la portée théorique du BLE Long Range peut atteindre jusqu'à 1 300 mètres [34], ce qui couvre largement les distances typiques entre les ruches [35].
- **Protocole 868 MHz** : Dans les situations où le BLE Long Range ne permettrait pas d'assurer une couverture suffisante, la gateway est également équipée d'un émetteur-récepteur 868 MHz. Un protocole personnalisé est alors utilisé pour garantir une communication fiable entre les capteurs et la gateway.

Le processus de collecte des données est conçu pour minimiser la consommation d'énergie des capteurs. Ceux-ci stockent les mesures localement et activent périodiquement une séquence de récupération de données qui comprend les étapes suivantes :

- La passerelle scanne les capteurs à proximité en BLE.
- Récupération et stockage des données de chaque capteur détecté.
- Transmission des données collectées.

2.2.2 Protocole de communication longue portée

Pour la transmission des données collectées par la passerelle vers la base de données, les technologies NB-IoT et LTE-M (CAT-M1) ont été sélectionnées. Ces solutions optimisent le compromis entre débit de données et couverture réseau, en particulier dans les zones éloignées.

CAT-M1 et NB-IoT sont des réseaux à bande passante moyenne, offrant une alternative intéressante aux réseaux 2G-4G, souvent très énergivores, tout en surpassant les limitations des réseaux LoRa et Sigfox en termes de bande passante. Le modem utilisé est capable de se connecter à l'un ou l'autre des réseaux, en fonction de la couverture disponible.

La stratégie de transmission repose sur des rendez-vous programmés. Les capteurs accumulent et stockent les données dans leur mémoire interne. Lorsque cette mémoire atteint la moitié de sa capacité, les capteurs commencent à émettre un message signalant la disponibilité de données à transmettre. La passerelle scanne régulièrement ces messages de statut. Lorsqu'un message est reçu, la passerelle se connecte au capteur pour récupérer les données. En cherchant à transmettre les données à mi-mémoire de stockage, cela permet de pallier le cas où un rendez-vous est manqué entre la gateway et le capteur, et les données seront transmises en totalité sur le rendez-vous suivant. Dans le cas où plus d'un rendez-vous est manqué, une alerte est transmise par e-mail aux apiculteurs pour réaliser la maintenance du système. Les capteurs et la gateway partagent l'heure à chaque connexion pour assurer la synchronicité, et éviter d'éventuelles dérives. Cette synchronisation minimise les échanges radio et optimise la consommation énergétique. Une fois toutes les données recueillies, la passerelle les envoie à un serveur MQTT.

Le processus d'envoi des données au serveur suit les étapes suivantes :

- Démarrage du modem.
- Établissement de la connexion réseau.
- Authentification auprès du serveur.
- Envoi des données stockées dans la mémoire flash.
- Extinction du modem.

2.2.3 Format des données : API standard

Une API standardisée a été développée pour simplifier la transmission des données des capteurs vers le serveur, garantissant ainsi l'indépendance entre le capteur et la passerelle. Le tableau 2.1 présente la structure de la trame de données utilisée pour le transfert des données capteurs. Le type de données échangées entre le capteur et la passerelle, puis entre la passerelle et le serveur, n'a pas d'incidence sur le fonctionnement de la passerelle. La structure de la trame reste constante à travers les protocoles BLE (capteur vers passerelle) et MQTT (passerelle vers serveur), permettant à la passerelle de transmettre

les données sans avoir besoin de les interpréter. Les couches de protocole assurent l'authenticité des trames, éliminant ainsi la nécessité d'une vérification par somme de contrôle. Le système est optimisé pour réduire au maximum le volume de données transmises, sans bits inutilisés. Des méthodes de compression au niveau des capteurs peuvent également être mises en place pour réduire encore plus ce volume, diminuant ainsi le temps de transfert, la consommation d'énergie et les coûts associés (par exemple, l'utilisation de données mobiles).

En plus des trames de données, des commandes standardisées facilitent les interactions entre la passerelle et les capteurs. Ces commandes incluent la récupération des données, l'effacement des données, la définition de la fréquence d'échantillonnage et la configuration du timestamp des capteurs. La standardisation de ces commandes permet une compatibilité entre différents types de capteurs, simplifiant ainsi leur gestion.

L'intégration de nouveaux capteurs avec de nouvelles métriques est simplifiée grâce à ce système. Les développeurs n'ont qu'à respecter l'API pour créer le capteur et les trames de données correspondantes, puis implémenter une fonction lambda sur le serveur pour analyser les données spécifiques au capteur. Ces données sont ensuite stockées dans une base de données. Actuellement, ce système prend en charge le stockage des mesures de température, de poids et des échantillons audio provenant des ruches, démontrant ainsi sa flexibilité à gérer différents types de données.

TABLE 2.1 – Modèle de trame pour le transfert des données capteurs en BLE.

Position	Taille (bits)	Valeur	Description
0	32	Timestamp	Timestamp 32 bits de la première mesure
1	48	Adresse MAC*	Adresse MAC du capteur pour identification
2	8	Type de mesure	Type de mesure du capteur
3	16	Fréquence de mesure	Intervalle de temps entre deux mesures
4	16	N	Nombre de mesures transmises
5	8	K	Nombre de bits par mesure
6	K	Mesure 1	Première mesure à transmettre
...			
N-5	K	Mesure N	N-ième mesure à transmettre

Note : L'adresse MAC peut être remplacée par un identifiant unique si le BLE n'est pas utilisé au profit d'un protocole 868 MHz.

2.3 Développement matériel et logiciel

2.3.1 La Gateway

2.3.1.1 Architecture matérielle

La gateway est développée pour pouvoir être industrialisée. La figure 2.2 présente une image du produit dans son boîtier, ainsi que la batterie utilisée. Elle illustre également l'architecture globale retenue pour la passerelle. Les justifications thermiques de chaque composant sont présentées ci-après. Une comparaison des caractéristiques techniques des composants existant sur le marché est réalisée en amont de la réalisation et un exemple de tableau comparatif est présenté en 2.4. Le prix de la solution est également un enjeu pour garder un produit commercialisable.

TABLE 2.2 – Image et architecture matérielle de la gateway

Schéma d'architecture	Image de la passerelle
<pre> graph TD Batterie[Batterie + Alimentation block] --- NRF52833[NRF52833 BLE Long Range] BG95[BG95-M5 NB IoT/CAT-M1/GNSS/GPRS] --- NRF52833 CMT2300A[CMT2300A 868MHz] --- NRF52833 I76[I.76 GNSS] --- NRF52833 ESP8266[ESP8266 WIFI (for positioning only)] --- NRF52833 </pre>	

Microcontrôleur avec BLE intégré Le microcontrôleur constitue le noyau central de l'architecture, responsable de la gestion des tâches et de l'optimisation de la consommation d'énergie en mode veille. Un microcontrôleur intégrant la stack Bluetooth a été sélectionné pour fournir une solution compacte, économe en énergie et économiquement intéressante. Les critères de sélection principaux sont présentés dans 2.4 avec une comparaison entre la solution retenue et d'autres options disponibles sur le marché. Les critères retenus sont

2.3. Développement matériel et logiciel

la faible consommation d'énergie en mode veille et des performances optimales en termes de communication BLE. Le nRF52833 est choisi pour ses caractéristiques techniques, avec une consommation d'énergie en veille faible, et une sensibilité de réception BLE atteignant -103 dBm, tout en ayant une puissance de transmission maximum de +8 dBm. La taille mémoire de 512 KB Flash et 128 KB RAM est jugée suffisante pour une gestion efficace du code applicatif. Si une plus grande capacité devait être nécessaire, le nRF52840 de la même famille propose des options de mémoire étendues tout en limitant le nombre de modifications logicielles et matérielles requises.

Reference	nRF52833	STM32WB55CG	EFR32BG12
Fabricant	Nordic Semiconductor	STMicroelectronics	Silicon Labs
Package	aQFN73 42 pins, 7 x 7 mm WLCSP53 42 pins, 3.2x3mm QFN40, 18 pins, 5x5mm	UFQFPN 48 pins 7x7mm VFQFPN68 8 x 8 mm WLCSP100 4.8 x 5.1 mm BGA129	QFN32 4x4 mm Package
Cœur	Cortex M4 with FPU @ 64MHz	Cortex M4 with FPU @ 64MHz	Cortex M33 with FPU @ 80MHz
Architecture mémoire	Flash: 512 KB RAM: 128 kB	Flash: up to 1MB RAM: up to 256KB	Flash: up to 1MB RAM: up to 96KB
Fonctionnement Radio	Stack Bluetooth intégrée a la flash 152kB	Dedicated 32-bit Cortex M0 + CPU for real-time Radio layer	Stack Bluetooth intégrée a la flash 80-100kB
Consommation sleep	1.3 μ A @ 3 V in OFF mode + full RAM Retention	2.1 μ A @3V in stop mode + full RAM retention +RTC	4.5 μ A @ 3V in deep sleep + full ram retention
TX current	6 mA @ 0 dBm with DC/DC	5.2 mA @ 0dBm	8.8mA @ 0dBm
RX current	6 mA @ 1 Mbps	4.5 mA @ 1Mbps Avec assez peu de détail	9.3 mA @ 1Mbps
Périphérique	USB 2.0, 2 UART, 2 SPI, 2 I2C, ADC@12bit, PDM	USB2.0, 2 UART, 2 SPI, QSPI, 2 I2C, ADC@12bit	3 UART, 3 SPI, 2 I2C, ADC@12bit
Puissance max de sortie	TX Power -20 to +8 dBm in 4 dB steps	TX power up to +6 dBm with 1 dB steps	Up to +19dBm
Receiver sensitivity Bluetooth	103 dBm @ 1Mbps	96 dBm @ 1Mbps	97.5 dBm @ 1Mbps
Supply voltage	Range 1.7 V to 5.5 V	Range 1.71 to 3.6	Range 1.71 V to 3.8 V
Prix	3.02€ @ 1000	3.92€ @ 1000	3.05€ @ 1000

FIGURE 2.4 – Paramètres de choix du microcontrôleur de la gateway

Choix du modem : Le modem Quectel BG95-M5 a été choisi pour sa prise en charge des bandes NB-IoT et CAT-M1, ainsi que pour ses capacités de géolocalisation GPS intégrées, un élément clé pour cartographier automatiquement les apiaries. Ce modem offre des performances réseau fiables, avec

une sensibilité de réception de -106 dBm pour CAT-M1 et de -114 dBm pour NB-IoT, et une puissance de transmission de +23 dBm. Bien que sa consommation en mode économie d'énergie (PSM) soit théoriquement de 6 μ A, une stratégie a été mise en place pour l'éteindre complètement lorsqu'il n'est pas utilisé, éliminant ainsi toute consommation résiduelle. Le modem utilise une carte SIM fournie par un courtier réseau, permettant une connectivité flexible à travers divers réseaux et assurant une couverture maximale, même dans des zones éloignées.

Émetteur-récepteur 868 MHz : L'émetteur-récepteur CMT2300A a été sélectionné pour sa compatibilité avec les bandes Sub-GHz, qui offrent une communication longue portée et une meilleure pénétration des obstacles, un atout majeur dans les zones rurales ou isolées. Il se distingue par sa faible consommation d'énergie, avec un courant de réception de seulement 4,8 mA et un courant de transmission de 19 mA à une puissance de sortie de +13 dBm, ce qui le rend particulièrement adapté aux systèmes alimentés par batterie.

Mémoire FLASH externe : Une mémoire SPI NOR externe standard, logée dans un boîtier 8-USON, a été choisie pour sa flexibilité dans la sélection du fabricant et l'adaptation de la taille de la mémoire selon les besoins en stockage. La taille de la mémoire est par défaut de 32 Mo, suffisante pour stocker divers types de données capteur, y compris des échantillons audios. Si des données vidéos étaient ajoutées à partir d'un capteur externe, la capacité de mémoire pourrait être augmentée en conséquence.

Batterie et alimentation : La passerelle utilise une batterie lithium-ion standard de 3,6 V, couplée à un régulateur de tension pour garantir une alimentation stable. La batterie a une capacité de 4000 mAh, avec la possibilité d'augmenter cette capacité en fonction de l'espace disponible dans le boîtier pour prolonger la durée de vie du produit.

Composants optionnels : Des options supplémentaires incluent l'installation d'un ESP8266 pour la géolocalisation Wi-Fi en scannant les adresses MAC, ainsi qu'un module GNSS L76 pour des données de localisation plus précises. Ces composants sont particulièrement utiles si la passerelle doit être utilisée comme dispositif anti-vol avec suivi en temps réel. Par défaut, ces composants ne sont pas inclus sur la carte.

2.3.1.2 Architecture logicielle

La structure du programme exécuté est plutôt simple sur le principe : à un intervalle régulier, la gateway collecte les données des capteurs présents à portée. Ces données sont stockées dans la mémoire flash externe permettant de rassembler un grand volume de données qui sont ensuite transmises sur la

base de données au travers du modem mis à disposition. Cette architecture est présentée en 2.5. Plus en détails, le fabricant du microcontrôleur mettant à disposition un software développement kit (SDK) pour le développement applicatif de la fonction BLE, le travail est relativement simple pour réaliser une application fonctionnelle. Le challenge repose dans la réalisation du driver du modem, afin de garantir le post des données, et ce même en couverture réseau limitée. En pratique, une machine à état software est conçue et implémentée. Il a également fallu arbitrer entre le "temps réel" du système et la consommation d'énergie liée aux tâches à effectuer, sujet abordé dans 2.3.1.3

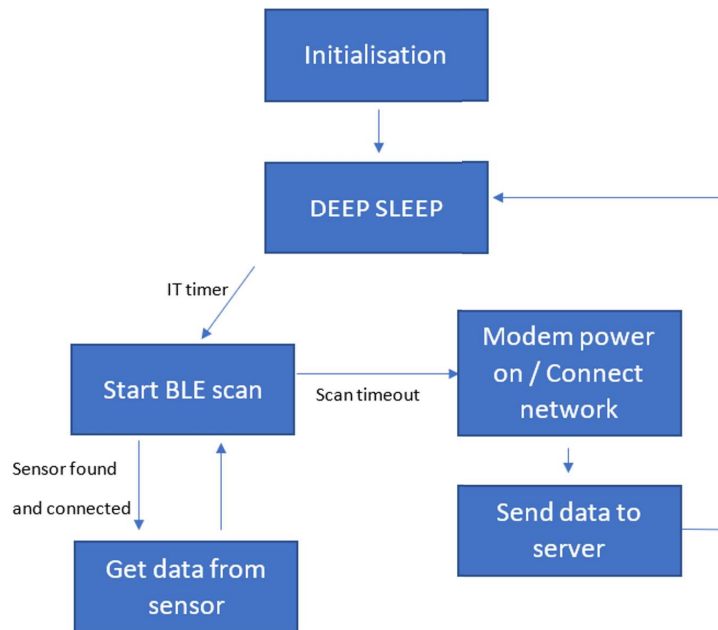


FIGURE 2.5 – Diagramme de fonctionnement du logiciel

2.3.1.3 Consommation et durée de vie

Afin de valider la compatibilité de la gateway avec une utilisation sur une saison complète apicole, il est nécessaire de confirmer que sa durée de vie couvre cette période de temps. Pour valider ce comportement, la consommation de la gateway est mesurée pour toutes ses étapes de fonctionnement à l'aide d'un power profiler, le produit utilisé est l'alimentation OTII [36]. En mode de veille, la consommation de fond de la gateway est de $4\mu\text{A}$. Cela nous permet d'avoir une grande autonomie sur la batterie choisie dont la capacité est de 4000mAh. Il est alors évident que les éléments coûteux en énergie sont la collecte des données locales des capteurs ainsi que le post des données sur la base de données via le modem. Ce sont eux qui conditionneront la durée de vie du produit.

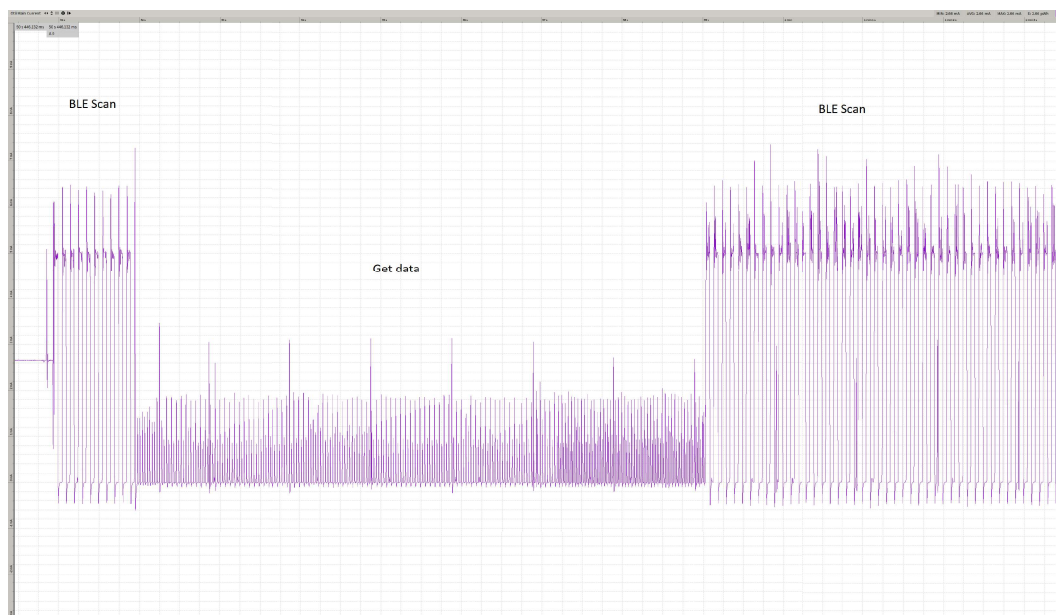


FIGURE 2.6 – Consommation lors de la collecte des données d’une balance par le lien BLE

Collecte des données capteurs La consommation de la séquence de récupération des données issue des capteur en BLE est présentée dans 2.6, les valeurs numériques de cette consommation sont détaillées dans 2.3. Afin de minimiser l’empreinte énergétique de la collecte des données, la taille des packets BLE qui transitent et la période de rendez vous ont été optimisés. Les paramètres sont les suivants :

- Augmentation de la taille des packets BLE à leur maximum soit 247 octets/messages.
- Diminution au minimum de la fréquence de rendez vous à 7.5ms.
- Pas de transmission de bit superflu, chaque bit transmis est utile 2.2.3

TABLE 2.3 – Consommation et temps pour chaque étape

	Temps (s)	Consommation
Conso Scan BLE	$2s \cdot N + 1\text{min}$	2,55mA
Conso Get Data	$8s \cdot N$	190 μ A

Note : N est le nombre de capteurs utilisés.

La durée de transfert des données pour le calcul de consommation est fixée à 8 secondes, correspondant au temps nécessaire pour transférer 8 échantillons audios successifs, qui est le cas de transfert maximal conformément aux spécifications décrites en 2.3.2.3. Un échantillon audio occupe 32 ko. Par BLE, il est possible de transmettre 247 octets utiles par trame, avec une période de

rendez-vous de 7,5 ms. Cela permet un temps de transfert de 1 seconde par échantillon. Chaque capteur pouvant stocker jusqu'à 8 échantillons audio, la durée maximale de transfert est donc de 8 secondes. Le volume de données des autres capteurs étant plus réduit, cela diminue encore le temps de transfert, ce qui contribue à prolonger la durée de vie de la batterie.

Transmission vers le serveur : La séquence de post des données sur le serveur est décrite ci après avec le profil de consommation associé (figure 2.7). Les mesures sont réalisées dans une cellule TEM afin de réduire la qualité de la couverture réseau sur le lieu de test et se placer dans le pire cas de couverture. Ainsi les calculs de durée de vie de la batterie sont une limite basse. Les résultats de ces mesures sont présentés dans 2.4

- Démarrage du modem
- Accroche réseau et ouverture de la connexion
- Authentications vers le serveur
- Post des données stockées en mémoire flash

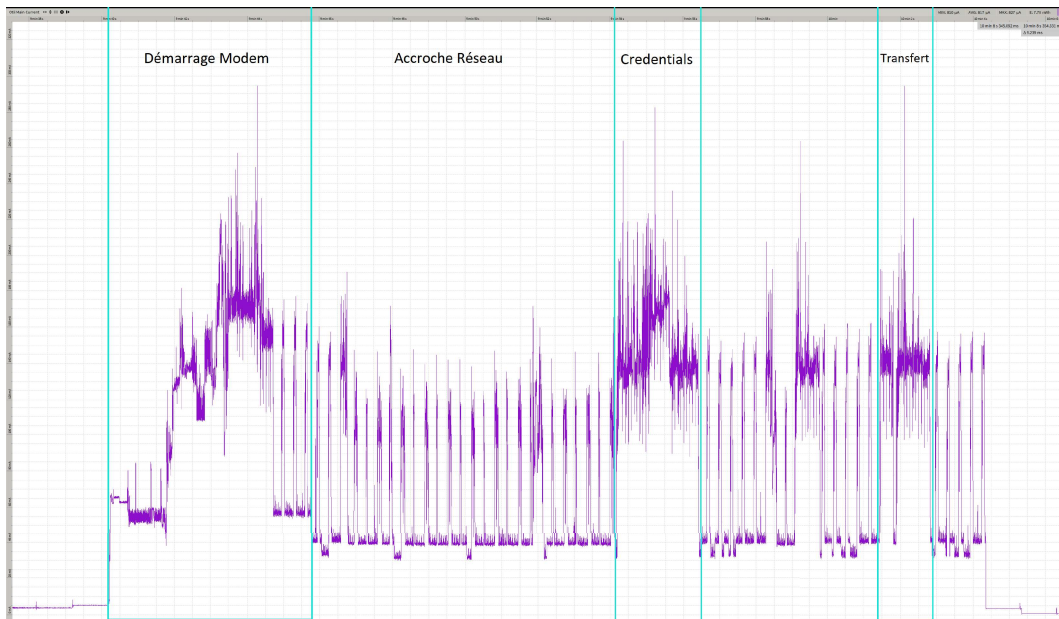


FIGURE 2.7 – Consommation d'un post de données MQTT

Lors d'un envoi de données (post), deux étapes distinctes peuvent être identifiées : une partie fixe comprenant le démarrage du modem, l'établissement de la connexion au réseau et l'authentification auprès du serveur à l'aide des credentials, et une partie variable qui correspond à l'envoi effectif des données, dont la consommation dépend du volume de données à transmettre. La consommation de la partie fixe est évaluée à 86,43 mA pour une durée de 16,138 secondes. Cette consommation fixe représente un total proche de 100 mAh si les données sont postées toutes les 4 heures.

TABLE 2.4 – Consommation et temps pour chaque étape

	Temps (s)	Consommation (mA)
Démarrage modem	5,618	114
Accroche réseau	8,332	52,2
Credentials	2,188	146
Transfert 10Ko	1,1	139
Total d'une transaction	22,8	83,1

Pour minimiser cette consommation, les modes de conservation de la connexion réseau du modem sont utilisés (LTE-M : PSM, NB IoT : eDRX). Dans ces modes, le modem maintient la connexion active tout en consommant une quantité de courant limitée, environ 6 μ A pour le Quectel BG95-M5. Ces modes reposent sur des périodes de rendez-vous pendant lesquelles le modem confirme qu'il conserve la connexion ouverte. Les paramètres associés à ces modes dépendent des options fournies par le réseau et n'ont pas été caractérisés dans cette étude. Par conséquent, les calculs de durée de vie sont effectués dans le pire des cas, c'est-à-dire en prenant en compte un redémarrage complet du modem à chaque occurrence de transmission des données.

2.3.1.4 Durée de vie de la batterie sans le lien 868MHz

Les calculs de la consommation énergétique de la passerelle ont été réalisés pour un total de 20 capteurs, comprenant 10 balances et 10 capteurs audios. La consommation de la passerelle est divisée en trois principales phases d'activité dont les valeurs sont décrites dans 2.5.

- Mode IDLE
- Récupération des données des capteurs 2.3.1.3
- Envoi des données au serveur 2.3.1.3

Sur la base de ces valeurs, la consommation moyenne de la passerelle est calculée en pondérant chaque valeur de consommation par la durée de la phase correspondante. Cela aboutit à une consommation moyenne de 0,3 mA. Avec une batterie de 4000 mAh, comme indiqué dans la section 2.3.1, la durée de vie théorique de la passerelle, avec 20 capteurs actifs, est calculée à 1,5 an. Cette durée dépasse l'objectif initial de la durée totale d'une saison apicole. Ces résultats calculés pour chaque étape dans le pire cas de consommation montrent qu'il est possible d'ajouter davantage de capteurs sans compromettre la durée de vie de la batterie de la gateway.

TABLE 2.5 – Consommation de la gateway en fonction de son activité .

Événement	Durée	Consommation
Scan BLE	30s + 2s*N	2,55 mA
Récupération des données BLE	8s*N	190 μ A
Démarrage modem et connexion serveur	16.138s	86,43 mA
Envoi des données au serveur	1s*N	139 mA
IDLE	4H - temps des autres phases	6,4 μ A
Consommation moyenne pour 20 capteurs		0,3 mAh

Note : N est le nombre de capteurs utilisés.

2.3.2 Les capteurs

2.3.2.1 Architecture standard

Matériel La figure 2.8 illustre l'architecture standard utilisée pour le développement des capteurs. Ces capteurs peuvent fonctionner de manière autonome sur batterie et être équipés d'une ou plusieurs métriques de mesures. Par exemple, un capteur installé dans une ruche peut être équipé pour mesurer l'humidité et la température, tandis qu'un capteur de poids sera uniquement configuré pour cette fonction. Sur cette architecture, les fonctions fixes sont :

- **nRF Family microcontrôleur** : Le microcontrôleur applicatif intègre la stack BLE et exécute l'application du capteur.
- **CMT2119A** : Transceiver radio 868MHz en palliatif au BLE long range.

Les options configurables incluent :

- **Alimentation** : En fonction des besoins énergétiques des capteurs réalisant les mesures, la taille de la source d'alimentation peut être ajustée, allant d'une pile CR2016 à un pack lithium-ion multi cellules.
- **Mémoire flash interne du microcontrôleur** : La famille nRF offre une gamme de microcontrôleurs avec des tailles de mémoire FLASH et RAM variables tout en maintenant des performances radio stables. L'utilisateur peut choisir le modèle le plus adapté pour optimiser les coûts, sans nécessiter de modifications majeures du logiciel.

Logiciel Le logiciel implémenté pour les capteurs est structuré en deux parties principales : l'une commune, la transmission radio, l'autre personnalisée : la mesure. L'architecture du software des capteurs est présentée en 2.9. La transmission radio est identique pour tous les capteurs et fonctionne en synchronisation avec la gateway via l'advertising Bluetooth. Plus précisément,

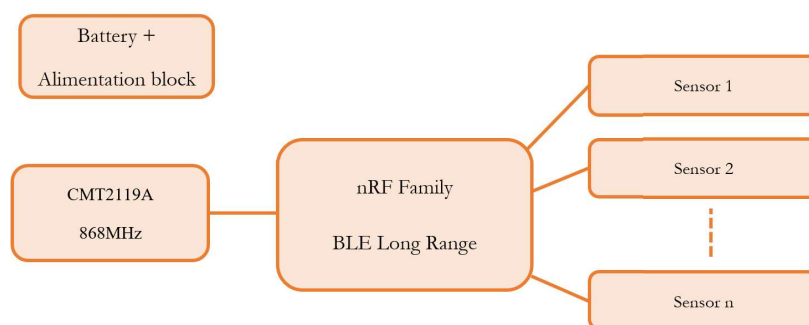


FIGURE 2.8 – Architecture matérielle standard d'un capteur

chaque capteur annonce en BLE et, lors de la connexion initiée par la gateway (jouant le rôle de maître), les données sont transmises. Cette partie du logiciel reste inchangée pour chaque capteur et est donc transférable de l'un à l'autre. Le développement spécifique de chaque nouveau capteur est la réalisation du driver spécifique au capteur pour la mesure, ainsi qu'à la gestion de l'alimentation des composants de la chaîne d'acquisition.

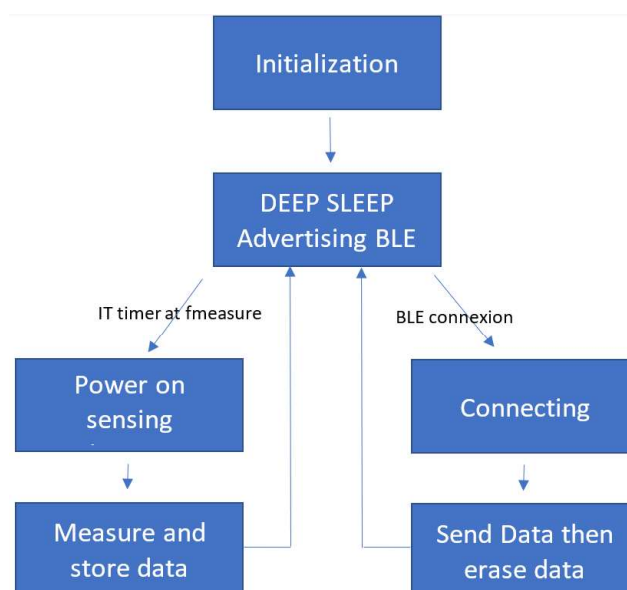


FIGURE 2.9 – Architecture logicielle standard d'un capteur

2.3.2.2 La balance

La principale métrique utilisée par les professionnels est la pesée des ruches. Afin de se différencier de l'existant il est impératif de réaliser un capteur à faible coût tout en répliquant les performances actuellement proposées [29, 28]. Les principaux composants d'une balance sont présentés en suivant :

- Un support métallique pour répartir la charge (le plateau de la balance).

- Deux jauges de contrainte pour la mesure de la masse.
- Une carte électronique équipée d'ADCs (convertisseurs analogique-numériques) pour la lecture des valeurs des jauges de contrainte, ainsi que des protocoles radios pour transmettre les données à la passerelle.

Le critère de choix différenciant par rapport à la concurrence est le « low cost ». Ainsi, des composants aux performances plus limitées, mais économiques, ont été retenus pour la balance. Une compensation logicielle sera ensuite implémentée afin de corriger les limitations matérielles et d'atteindre des performances comparables à celles des balances concurrentes 3.



FIGURE 2.10 – Visualisation des éléments de la balance et sa mise en service

Architecture mécanique : La mesure de la masse se fait en utilisant des jauges de contraintes (sondes) sur lesquelles sont collés des peignes résistifs dont la valeur change en fonction de la contrainte appliquée sur la structure. Il est nécessaire de déterminer le nombre de sondes utilisées en parallèle pour réaliser une mesure de masse unique. Dans le cas de l'utilisation de plusieurs sondes, la masse pesée est la somme des poids mesurés par chaque jauge. Afin de pouvoir peser, il faut récupérer le poids de la ruche et le diriger sur les sondes. Plusieurs options sont possibles sur le nombre de capteurs utilisés :

- **1 Capteur :** Architecture majoritairement utilisée par les balances concurrentes [29]. La difficulté de cette architecture repose sur le risque de faire travailler la sonde en torsion ce qui dé-linéarise le capteur. Il est donc nécessaire de développer une mécanique complexe et coûteuse. Pour ces raisons cette solution n'a pas été testée.
- **4 Capteurs :** Une itération avec 4 jauges a été réalisée mais le cumul des erreurs de chaque sonde ainsi que le manque de stabilité mécanique, (une sonde subit plus d'effort que les autres si le support n'est pas horizontal), a montré que cette solution n'est pas la meilleure.
- **2 Capteurs :** Le produit final est composé de deux capteurs 2.10b. Avec ce montage, la charge est répartie également sur les deux capteurs (même

en légère pente) et travaillent dans le fonctionnement normal de la sonde, en pression.

La structure mécanique du produit réalisé est la suivante :

- Deux jauges de contraintes 2.3.2.2
- Un plateau en aluminium plié
 - Permet de renforcer la résistance à la déformation.
 - L'épaisseur du plateau peut ainsi être réduit réduisant le poids de l'ensemble.
 - Le plateau est percé pour permettre à la ruche d'évacuer ses déchets et d'avoir une bonne ventilation.
- Deux pieds en tube aluminium
 - Simplifient l'équilibre de la balance même sur un support qui n'est pas parfaitement plan
- Des cales en acier de 4mm sont utilisées pour écarter mécaniquement les jauges de contraintes du plateau ainsi que des pieds pour que la déformation de ces dernières puisse se faire de la manière la plus adaptée possible. Le choix du matériaux acier est fait pour une bonne tenue mécanique à la déformation pour avoir un appui stable et fiable pour les jauges de contraintes. De ce fait on évite une déformation des éléments de la balance autre que ceux des jauges de contrainte, déformations voulues pour la mesure.

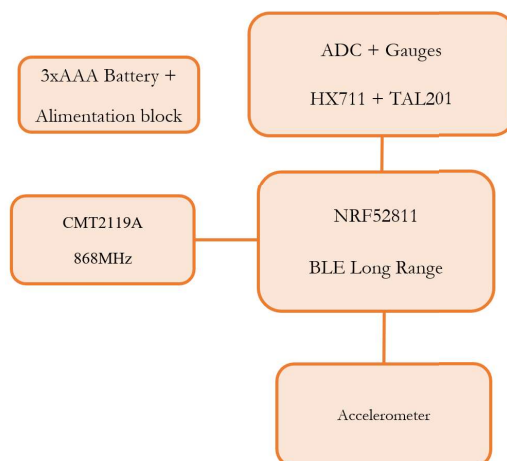


FIGURE 2.11 – Architecture hardware de la balance

Architecture électronique L'architecture électronique reprend le principe de l'architecture présentée en 2.3.2.1. Le microcontrôleur choisi est le nRF52811, petit modèle de la famille mais qui possède une mémoire suffisante pour stocker les mesures de masse. La balance peut faire data logueur sur une durée de 3 jours avec une fréquence d'échantillonnage de 6 mesures/heure.

Le choix des jauges de contrainte Les solutions à jauges de contraintes sont utilisées dans la littérature [19, 30, 31] et dans l'industrie [29]. Le plus gros défaut des jauges de contraintes est la dérive thermique de ces dernières, problème difficilement soluble sans réaliser une caractérisation et une compensation custom en usine pour chaque sonde [37]. Le tableau 2.6 présente les différences théoriques entre une sonde compensée avec un test unitaire et une sonde "low cost". Ces sondes sont également présentées sur 2.12. Pour des questions de coût, la TAL203 est choisie, issue d'une famille déjà utilisée dans la littérature [19]. L'objectif principal étant de réaliser un capteur low cost, le prix des jauges qualitatives bien compensées les disqualifie. Une compensation software sera présentée pour ces jauges dans 3. Les TAL203 sont qualifiés pour les charges allant jusqu'à 150kg chacune, ce qui permet de supporter des ruches allant jusqu'à 300kg, poids maximum compatible avec l'approche d'utiliser la balance en moyenneur de variations avec plusieurs ruches placées l'une sur l'autre.

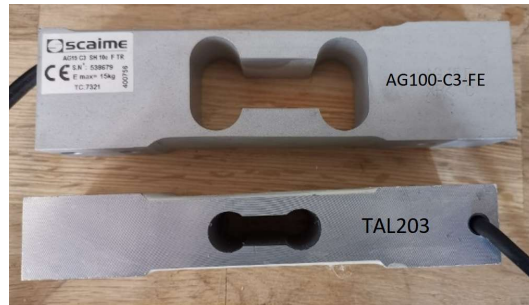


FIGURE 2.12 – Image des corps d'épreuve équipés des jauges de contraintes

TABLE 2.6 – Comparaison de deux jauges de contraintes de calibre différent

	AG100-C3-FE	TAL203
Max charge	100kg	150kg
Dérive thermique théorique	0.14g/°C	3g/°C
Prix	100€@1000	5€@1000

Le choix des ADCs Le choix de l'ADC est d'importance car il définira la qualité du signal enregistré pour la mesure du poids. L'ADC HX711 est couramment référencé dans les architectures incluant une balance pour les ruches d'abeilles[19, 26, 20, 21, 22, 31]. Ce composant est un amplificateur ADC 24 bits à très faible coût et à gain élevé (128). Afin de confirmer que le composant correspond bien au besoin, une comparaison avec des ADC de différentes factures sont comparées dans 2.7. Pour comparer le bruit, un pont diviseur est placé entre l'alimentation, la masse et les deux entrées différentielles des ADCs

afin de les polariser en leur point milieu. Les résistances sont choisies pour correspondre aux valeurs du pont diviseur des jauges de contraintes (400Ω) et ainsi faire travailler les ADCs dans les mêmes conditions de courant que pour une sonde. La quantité de bruit, s'il est gaussien, peut être réduit en moyennant plusieurs mesures. Dans notre cas, la vitesse de variation de la masse étant négligeable au regard de la vitesse d'échantillonnage de tous les ADCs, il est possible de réaliser un filtre en moyennant et donc de revenir à des seuils de bruit réduit. Le NAU7802 est disqualifié par la nature de son bruit non gaussien, en plus du nombre de bits en mouvement. Les ADC de chez Analog Device (AD,) bien que plus performants sont aussi réellement plus coûteux. Au regard de la contrainte économique et de la confirmation de la littérature, le HX711 est choisi sur la solution. Deux ADCs sont utilisés, un pour chaque sonde. Bien que l'ADC ait deux entrées analogiques possibles, seule une entrée fournit un gain d'amplification de 128. Sur la seconde voie de l'ADC ayant un gain de 64, une thermistance est câblée pour fournir la valeur de la température et pouvoir par la suite réaliser la compensation thermique des sondes.

TABLE 2.7 – Comparaison du bruit en bits de différents ADC au regard de leur prix

	NAU7802	ADS1232	AD7124	HX711
Bruit (bits)	8 (non gaussien)	6	5	7
Prix (€)	0.72@1000	5.90@1000	6.35@1000	0.4@1000

2.3.2.3 La capteur audio

Pour le capteur audio, un microphone MEMS PDM de 16 bits est choisi, car cette résolution est suffisante pour les applications audios. La littérature suggère qu'une durée d'enregistrement de 2 secondes est optimale, et étant donné que les fréquences sonores des abeilles se situent principalement entre 20 et 2000 Hz, un taux d'échantillonnage minimal de 4 kHz est suffisant selon le critère de Shannon-Nyquist. Cependant, afin de conserver des données plus riches pour l'apprentissage automatique et de permettre une flexibilité ultérieure, un taux de 8 kHz a été sélectionné [38, 39, 40, 41, 42, 43].

Pour la fréquence des mesures des capteurs audios, un équilibre est recherché entre la quantité de données collectées et le volume de stockage requis. Un échantillon par heure a été choisi, ce qui permet de représenter efficacement le bruit à l'intérieur de la ruche tout au long de la journée, tout en maintenant un compromis optimal entre la consommation d'énergie et la capacité de stockage.

Le microcontrôleur sélectionné est le nRF52832, qui dispose d'une mémoire flash suffisante pour stocker jusqu'à 16 secondes d'audio, permettant ainsi de stocker 8 heures de données dans chaque capteur. Le capteur audio est alimenté par une pile CR2032 (220 mAh), qui a un très petit format, facilitant le placement du capteur à l'intérieur de la ruche.

FIGURE 2.13 – Image et architecture hardware du capteur audio

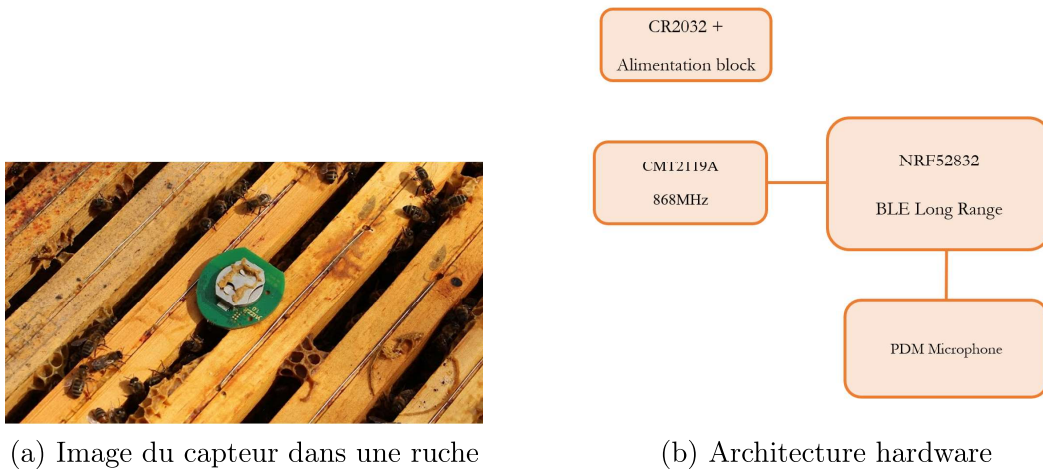


FIGURE 2.14 – Capteur audio et son architecture hardware

2.3.2.4 Autonomie sur pile des capteurs

En utilisant la même méthode décrite dans 2.3.1.3, les différentes phases de consommation de la balance et du capteur audio sont mesurées. Les consommations de veille des capteurs étant respectivement de $6\mu\text{A}$ pour la balance et $4\mu\text{A}$ pour le capteur audio, il est nécessaire de prendre en compte la fréquence de réalisation des mesures. Les fréquences d'échantillonnage choisies sont effectuées en prenant en compte l'impact sur la durée de vie du produit comme présenté en exemple pour la balance dans 2.15.

Le tableau 2.8 présente les calculs de durée de vie des batteries calculées. Avec les taux d'échantillonnage sélectionnés et les intervalles de transmission des données vers la gateway, la durée de vie des batteries dépasse un an, ce qui confirme que les paramètres choisis sont compatibles avec le besoin. Pour établir les paramètres

TABLE 2.8 – Durée de vie des batteries des capteurs de balance et audio. "Consommation moyenne" : consommation moyenne en courant ; "Taille de la batterie" : capacité de la batterie en milliampères-heures (mAh) ; "Durée" : temps de fonctionnement estimé.

Paramètre	Balance	Audio
Consommation moyenne	$115\ \mu\text{A}$	$15\ \mu\text{A}$
Taille de la batterie	1250 mAh	220 mAh
Durée	1,23 ans	1,18 ans

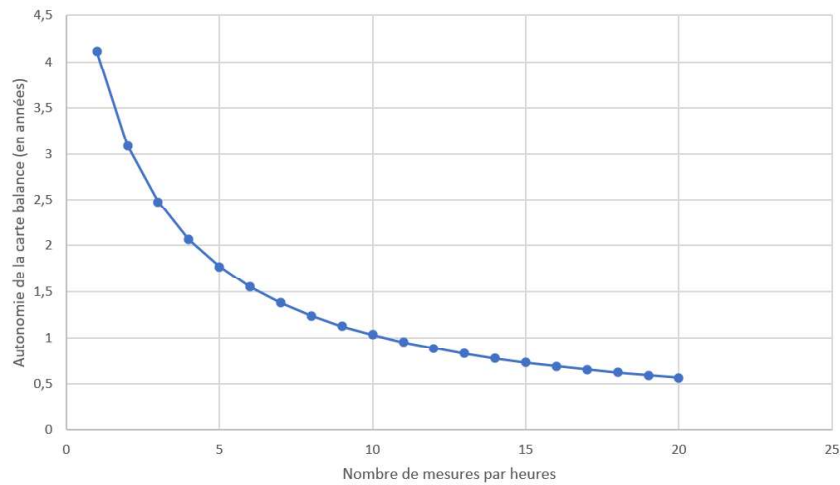


FIGURE 2.15 – Année d'autonomie de la balance en fonction du nombre de mesures par heures

2.4 Déploiement du système

Le système est déployé dans 10 ruches depuis un an dans des conditions apicoles réelles (inspections fréquentes par l'apiculteur, manipulation des cadres, traitements appliqués, récolte de miel, etc.) sans rencontrer de problèmes majeurs. Le rucher avec le système entièrement déployé est présenté sur la figure 2.16.



FIGURE 2.16 – Système déployé sur un rucher

2.4.1 Données de masse

La figure 2.17 présente les mesures de poids d'une ruche sur une période de 5 jours durant une miellée en avril 2023. L'analyse de ces données permet

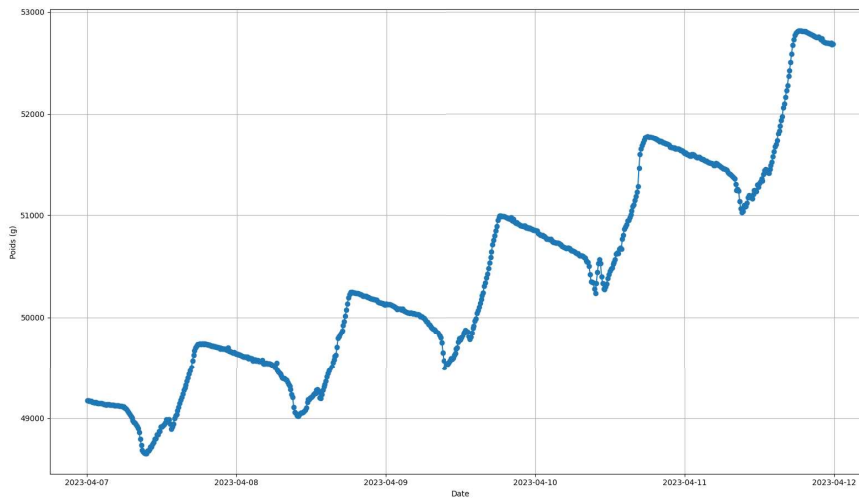


FIGURE 2.17 – Évolution du poids d’une ruche sur une période de 5 jours au cours d’une miellée, pour identification des comportements des abeilles.

d’identifier plusieurs phases dans l’activité des abeilles : une phase de prise de masse, une phase de perte de poids et une phase intermédiaire de creux entre ces deux périodes. On constate que la prise de poids se produit principalement entre 9h00 et 18h00, la phase de plateau de perte de poids s’étend de 18h00 à 8h00, et une perte de poids rapide mais brève survient entre 8h00 et 9h00. Cette période de prise de poids correspond aux heures d’activité de la journée, limitée par la durée de la journée et des températures favorables ($>12^{\circ}\text{C}$) caractéristiques du mois d’avril.

Pendant cette période d’activité, les abeilles collectent des ressources, ce qui entraîne une augmentation de la masse de la ruche. Inversement, la nuit, les abeilles étant présentes dans la ruche, le poids diminue progressivement. Cette perte de masse peut être attribuée à deux phénomènes : l’alimentation des abeilles et le séchage du miel par ventilation pour réduire son taux d’humidité en dessous de 20%. La perte de poids subite observée le matin correspond au départ des abeilles pour collecter des ressources et évacuer les déchets accumulés durant la nuit (déjections, abeilles mortes, etc.). Globalement, cette période de miellée montre que les ressources accumulées durant la journée dépassent les pertes nocturnes, ce qui indique un gain de réserves, un cas favorable. À l’inverse, en période de ressources limitées ou hivernales, la ruche perd progressivement de la masse.

Le système permet ainsi d’identifier plusieurs schémas :

- La prospection des miellées grâce à la prise de masse.
- La quantité de ressources disponibles et le besoin éventuel de nourrissage en cas de carence.

- La détection de problèmes de santé si le comportement de masse d'une ruche diverge de celui des autres (pillage, maladie, essaimage, etc.).

Les observations sur cette période mettent également en lumière des phénomènes courants en apiculture.

Au début du printemps 2023, lors d'une inspection des ruches, il est constaté que toutes les colonies sont pleines avec une abondance de couvain. Des cadres supplémentaires sont ajoutés aux ruches n'ayant pas encore atteint les 10 cadres, tandis que des hausses sont placées sur les ruches déjà fortes pour lancer la production de miel. La figure 2.18 illustre l'évolution du poids de deux ruches, montrant une discontinuité lors de l'ajout des hausses (intervention de l'apiculteur), suivie d'une prise de masse progressive au fil des jours, signe du stockage de miel dans les hausses. À la fin de cette miellée, les hausses sont récoltées pour être extraites.

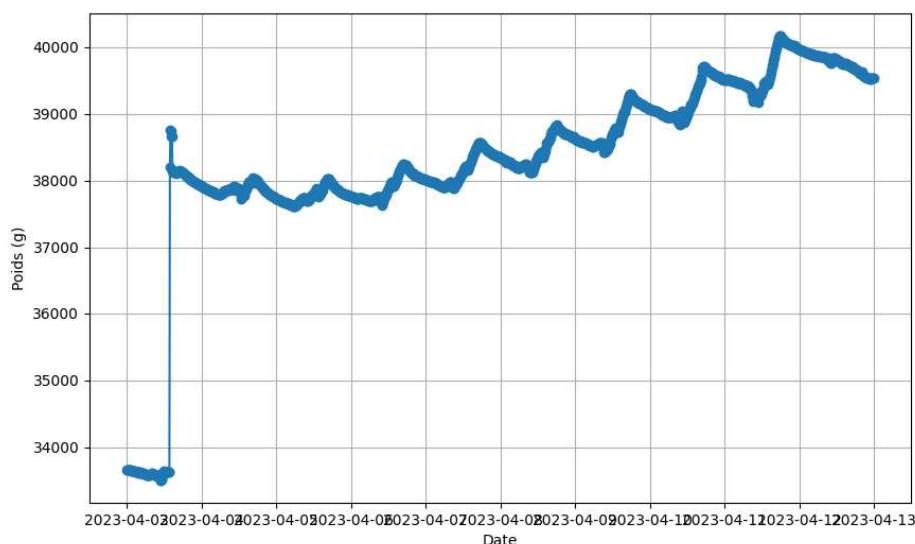


FIGURE 2.18 – Prise de masse progressive d'une ruche après l'ajout d'une hausse en anticipation de la miellée, entraînant une augmentation de poids.

Le 11 avril 2023, un comportement d'essaimage est détecté dans les données de masse enregistrées. L'analyse de ces données, présentée à la figure 2.19, révèle qu'une ruche, en forte prise de poids sur plusieurs jours, subit une perte soudaine de 3,612 kg. Cet événement se produit à 16h00, pendant la période la plus chaude de la journée, à une température de 25°C. La perte de poids est suivie d'une reprise rapide, avec des augmentations journalières de +1,1, +1,35, +1,15, +1,58 et +1,46 kg, culminant avec l'essaimage.

Les premières analyses des données de masse confirment la cohérence des comportements observés sur les ruches en activité, validant ainsi la réalisation matérielle de la balance. Les ordres de grandeur observés correspondent également aux activités relevées par l'apiculteur sur les ruchers. Cependant des

2.4. Déploiement du système

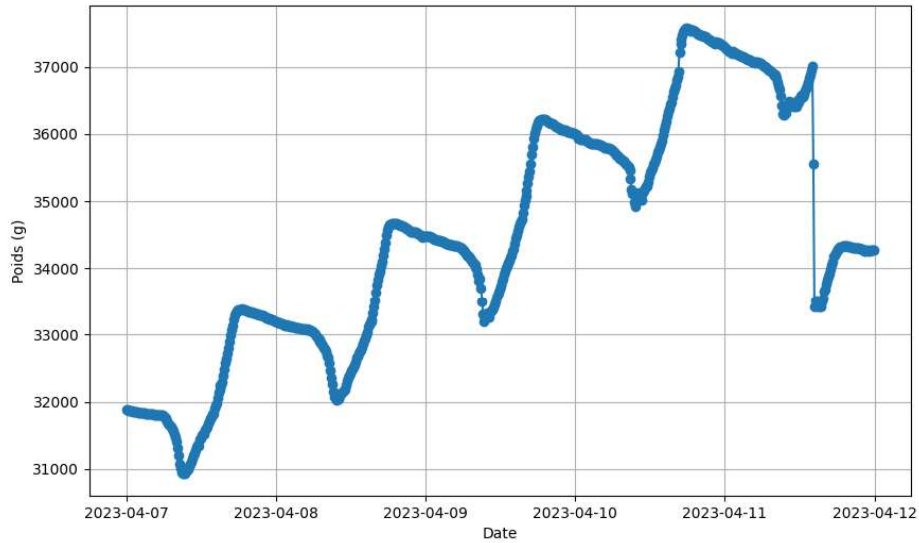


FIGURE 2.19 – Prise de masse progressive d’une ruche durant une miellée menant à un essaimage, identifié par la discontinuité.

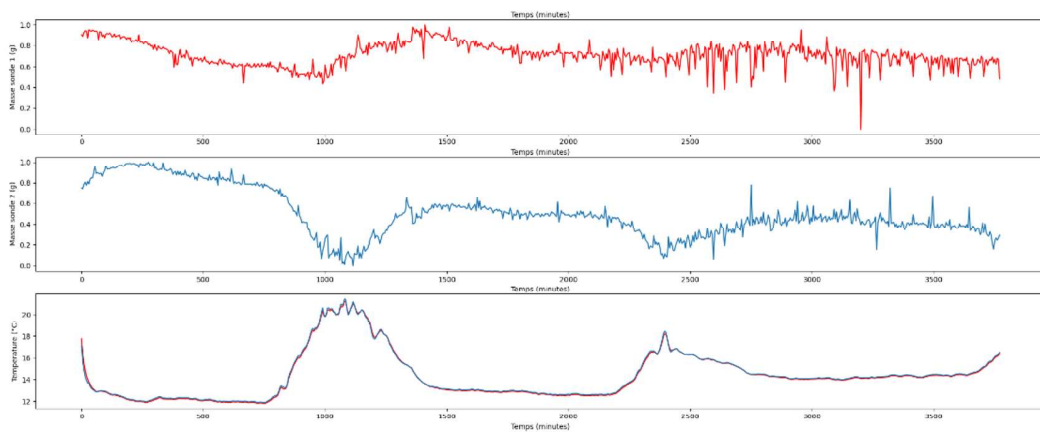


FIGURE 2.20 – Premières observations de la dérive thermique en grammes sur deux jauges de contraintes (rouge et bleue) en fonction de la température (noir) pour une balance a charge constante. L’échelle des abscisses représente le temps en minutes.

essais a charge constante ont montré que la balance subit un dérive thermique. Un exemple de variation est présenté en 2.20, où l’on constate que les mesures des jauges de contrainte d’une balance évoluent en fonction de la température. Les origines de cette dérive ainsi que le travail de compensations pour fiabiliser la mesure de masse seront donc à compenser en 3.

2.4.2 Données audio

Des données audio sont récoltées sur les ruches équipées d'une balance. Dans un premier temps des fichiers de 5 secondes et 16kHz ont été collectés pour valider les choix technologiques de l'acquisition, puis la taille et la fréquence des échantillons ont été réadaptés conformément aux spécifications décrites dans 2.3.2.3.

La figure 2.21 représente le signal audio temporel sur un échantillon de 5 secondes. Le tracé en noir donne la résolution en bits nécessaires au stockage de la mesure sur des intervalles d'une seconde. Cette résolution sur nos fichiers audio confirme le choix de la résolution du microphone de 16 bits qui est suffisante pour capter l'information audio de la ruche. On remarque une lente dérive de la composante DC (rouge) dont la cause n'est pas clairement identifiée, mais semble être inhérente à la technologie des capteurs PDM. Il sera nécessaire d'effectuer un filtrage de cette composante pour éviter des pertes de résolution et ne pas influencer les algorithmes d'IA. Lors de la quatrième seconde du fichier audio on remarque un évènement d'amplitude supérieur au signal typique. Cet évènement est un bruit parasite extérieur à la ruche, soulignant l'importance de repérer les périodes audibles sans interférences prononcées pour garantir l'exploitabilité de l'échantillon audio dans les méthodes de machine learning qui seront développés en chapitre 4.

Pour confirmer la fréquence d'échantillonnage choisie pour les fichiers audio, le tracé de la transformée de Fourier (FFT) et de la Discrete Cosine Trans-

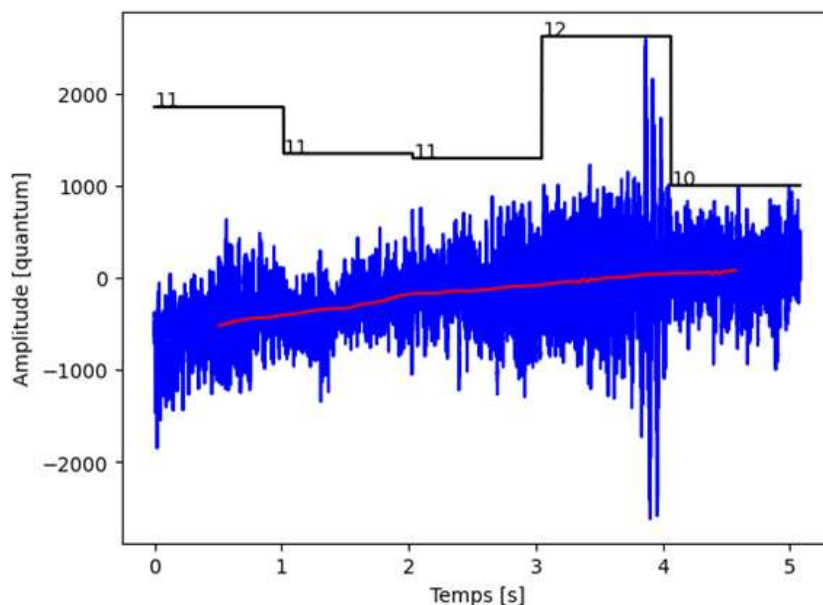


FIGURE 2.21 – Représentation temporelle d'un fichier audio de 5 secondes (bleu). La courbe noire représente la résolution en bits utilisée pour stocker l'échantillon et la courbe rouge correspond à la composante DC du signal audio.

form (DCT) sont tracés en figure 2.22. Le spectre utile semble compris entre 10Hz et 800Hz. Le calcul de la DCT n'est donc pas utile dans ce cas du fait du grand nombre de points. Une décimation par 4 du signal permettrait de descendre à une fréquence de 4kHz tout en gardant un signal utile observable. Ces observations vont dans le sens du choix d'un "sampling" à 8kHz, suffisant tout en gardant une marge de manœuvre si nécessaire.

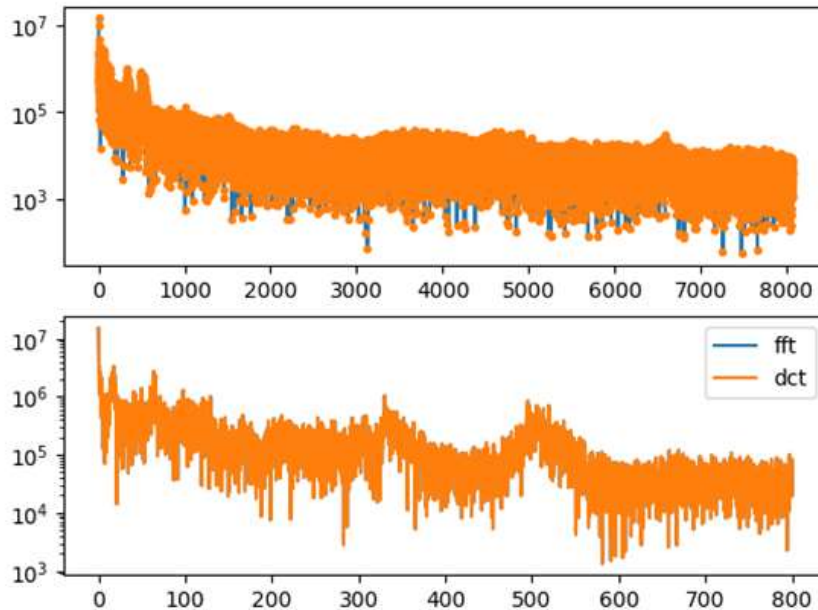


FIGURE 2.22 – FFT et DCT calculés à partir du fichier audio présenté en 2.21. Les abscisses sont la fréquence en Hz, et l'ordonnée est l'amplitude

La figure 2.23 présente le spectrogramme du fichier audio en analyse. Le spectrogramme est la représentation temporelle de la FFT : utile pour créer une représentation 2D, qui peut ensuite être analysée avec les techniques de machine learning développées pour le traitement d'images. Le spectrogramme met efficacement en évidence le phénomène de parasitage visible à la 3^{ème} seconde.

Finalement, les caractéristiques du capteur audio semblent cohérentes avec les premières observations et analyse faites sur les données collectées. Cela permet après constitution d'une base de données conséquente de réaliser des analyses de comportements sur les ruches.

2.5 Transmission et Stockage des données

L'objectif de compenser des imperfections des capteurs passe par la construction de modèles déterministes ou IA à partir de la collecte massive de données. Aussi, il est essentiel de conserver ces données pour constituer un dataset réunissant un ensemble de métriques sur différentes ruches sur plusieurs

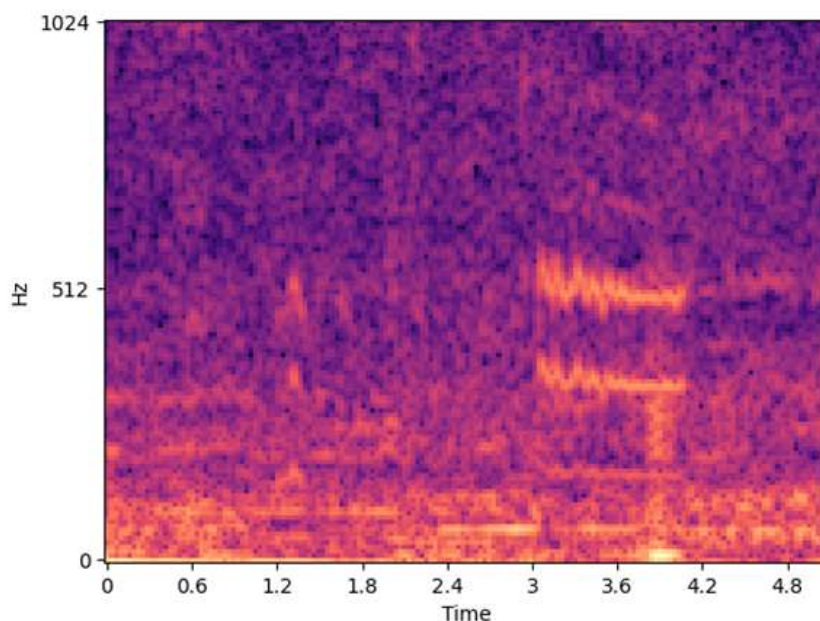


FIGURE 2.23 – Spectrogramme du fichier audio présenté en 2.21.

années. Dans cette optique, une base de données est mise en place. Sa structure est conçue pour faciliter l'accès aux données stockées, dans le but d'abord d'améliorer les mesures par des corrections logicielles, puis de les exploiter à l'aide de méthodes étudiées dans ce travail de thèse, tout en les rendant accessibles aux utilisateurs. La base de données est conçue pour pouvoir être mise à l'échelle avec l'augmentation du nombre de ruches instrumentés, et les services de visualisations et d'exploitation des mesures reposent sur sa structure. Les quelques données de masse sont présentées parmi les données accumulées sur une année de mesure sur deux ruchers totalisant une quinzaine de ruches.

2.5.1 Protocole MQTT

Pour la transmission des données vers Internet, le protocole MQTT a été choisi. Ce protocole est particulièrement adapté à ce cas d'usage grâce à son système de publication-souscription. Chaque type de mesure (température, masse, audio, etc.) dispose de son propre topic MQTT, permettant un traitement et un stockage spécifiques à chaque métrique. De plus, des topics supplémentaires sont utilisés pour transmettre des messages de statut des capteurs et des alertes sur la santé des ruches.

Le système de souscription facilite également le déploiement à distance des configurations et des mises à jour des capteurs, via une interface web qui sera développée ultérieurement. Cela permet d'ajuster les paramètres des capteurs à grande échelle avec un minimum d'effort.

La principale limitation de MQTT est la taille maximale des messages, limitée à 256 Mo (soit 2,27h d'audio), ce qui peut poser problème pour des

enregistrements audio en continu ou pour de la vidéo via ce canal.

2.5.2 Base de données

Les données des capteurs sont stockées dans une base de données hébergée sur le service Amazon RDS 2.5.3. La figure 2.24 présente une visualisation de la base de données pour stocker les données de masse. Cette structure permet l'association des capteurs de pesée avec leurs ruches correspondantes au sein du rucher. Ces associations ruche-rucher sont générées automatiquement en regroupant toutes les ruches dont les données sont transmises par la même gateway, et donc situées en un seul rucher. Cependant, il incombe à l'apiculteur de spécifier l'association des capteurs aux ruches. Suivant la même approche flexible que le reste du système, l'ajout d'un nouveau type de mesure est simple en ajoutant uniquement une table dans la base de données, sans nécessiter de modifications du système existant.

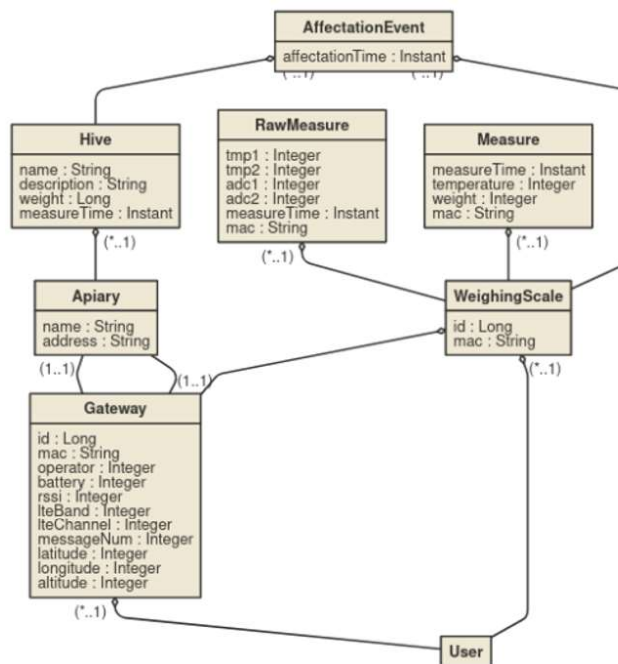


FIGURE 2.24 – Architecture de la base de données pour les capteurs de poids

2.5.3 Web architecture pour le stockage des données

Les services d'Amazon AWS sont utilisés pour ce produit. L'objectif de cette décision est d'accélérer le développement et de garantir un stockage sécurisé et fiable des données de mesure collectées. Compte tenu du choix du protocole de communication, le service IoT Core d'Amazon est directement adapté car il fournit directement un serveur MQTT. La base de données est

localisée sur le service Amazon RDS et différents services sont également employés pour le parsing des données en vue de leur insertion sous le bon format dans la base de données. Ces services sont présentés sur la figure 2.25 :



FIGURE 2.25 – Structure de réception et de stockage des données sur Amazon AWS

- **AWS IoT Core** : Fournit un serveur MQTT qui gère directement les Topic et les abonnements des gateways. Le service permet également l'identification et la sécurisation des transmissions grâce au système de certificats SSL,
- **AWS SQS** : Le service SQS est une file d'attente pour les messages reçus, permettant d'éviter les pertes de données en cas de fort trafic réseau. De plus, il permet de surveiller les messages de non gérés par l'application, en les conservant jusqu'à ce qu'un opérateur puisse intervenir.
- **AWS Lambda** : Le service Lambda est déclenché par des messages entrant dans la file d'attente SQS. Il exécute une fonction spécifique pour chaque message reçu en fonction du topic MQTT. Ces fonctions réalisent le parsing des données reçues puis l'insertion des données dans les tables correspondantes de la base de données. Si besoin, les lambda fonctions peuvent aussi déclencher des alertes (e-mail, SMS, Notification d'application), utiles pour les opérations urgentes de l'apiculteur comme l'essaimage.
- **AWS RDS** : Héberge la base de données où toutes les données sont stockées. La base de données est basée sur MySQL et présentée en 2.5.2

2.6 Visualisation des données

Une fois les données stockées dans la base de données, il est nécessaire de les afficher afin de permettre à l'apiculteur de les exploiter pour décider de ses futures actions sur les ruchers. Les services de visualisation et d'exploitation des données s'interfaçent avec la base de données suivant les services décrits dans la figure 2.24. Deux services sont mis en place, l'un complet ayant accès aux données brutes pour la recherche 2.6.1 et une dédiée aux apiculteurs suivant leurs besoins avec les données processées 2.6.2. La figure 2.26 décrit la structure d'accès aux données.

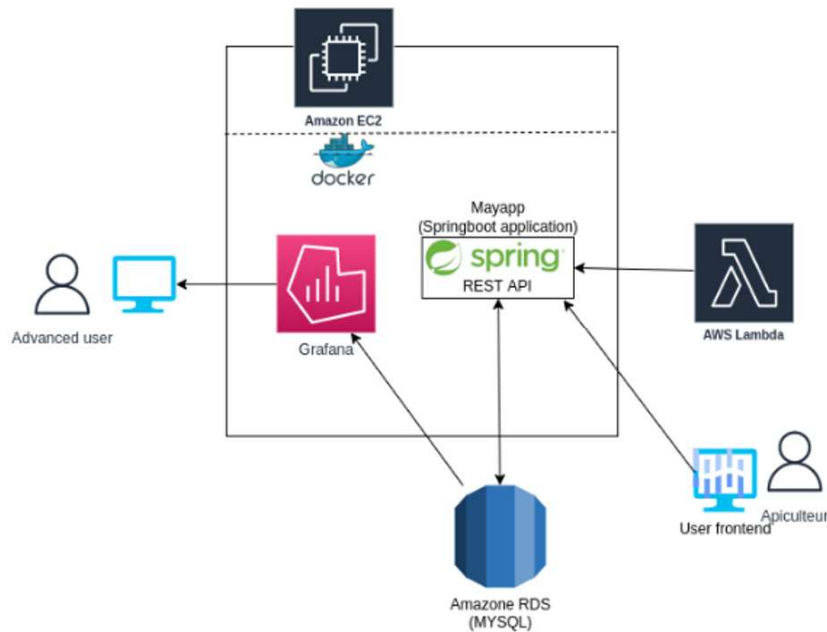


FIGURE 2.26 – Stack applicative des différents services permettant l’exploitation des données des capteurs sur des applications web de recherche et d’exploitation

2.6.1 Utilisation des données pour la recherche

Le service de visualisation des données dédié à la recherche est basé sur Grafana. Le résultat est présenté succinctement en 2.27. Grafana est un outil capable de récupérer les données de la base de données, d’appliquer un pré-traitement, puis de les afficher. Il est utilisé pour la visualisation des données en temps réel et pour identifier les périodes de mesure pertinentes afin de tester les modèles de compensation développés dans le cadre de la recherche sur les ruches.

2.6.2 Application Web pour les apiculteurs

Le service développé pour les apiculteurs est une application web qui permet la visualisation des données de leurs ruches. Les fonctionnalités implémentées sont les suivantes :

- Création et édition de comptes utilisateurs
- Contrôle d’accès (y compris la récupération de mot de passe)
- Accès à la base de données
- Association des capteurs avec un utilisateur
- Gestion des capteurs
- Pré-traitement des données avant affichage
- Visualisation des données sur différentes périodes



FIGURE 2.27 – Visualisation des données pour la recherche. Les graphiques représentent l'évolution du poids en kg (à gauche) et de la variation de masse (droite) au fil du temps sur quatre ruches.

La figure 2.28 montre la visualisation des données de pesée et des informations traitées dans l'application destinée aux apiculteurs. La courbe inférieure présente les données de masse (en orange) et de température (en vert) au fil du temps. Pour simplifier l'interprétation des mesures, la courbe supérieure affiche la variation journalière de la masse de la ruche. Cela permet de visualiser simplement et rapidement les tendances de prise ou de perte de poids d'un jour à l'autre. La figure 2.29 présente les données de quatre ruches dans le panneau de gestion sur une période de météo défavorable entraînant une perte de poids progressive des ruches, à l'exception de la première ruche qui est identifiée comme l'apiculteur comme forte et qui a su profiter d'une journée plus favorable pour accumuler des ressources, gagnant 1.2kg sur cette période. L'analyse de cette période de données permet de constater la présence de ressources mais aussi de confirmer l'état de santé des colonies avec une quantité de variation de masse différentes alors que les ressources à portée sont les mêmes pour chacune des colonies.

2.6. Visualisation des données

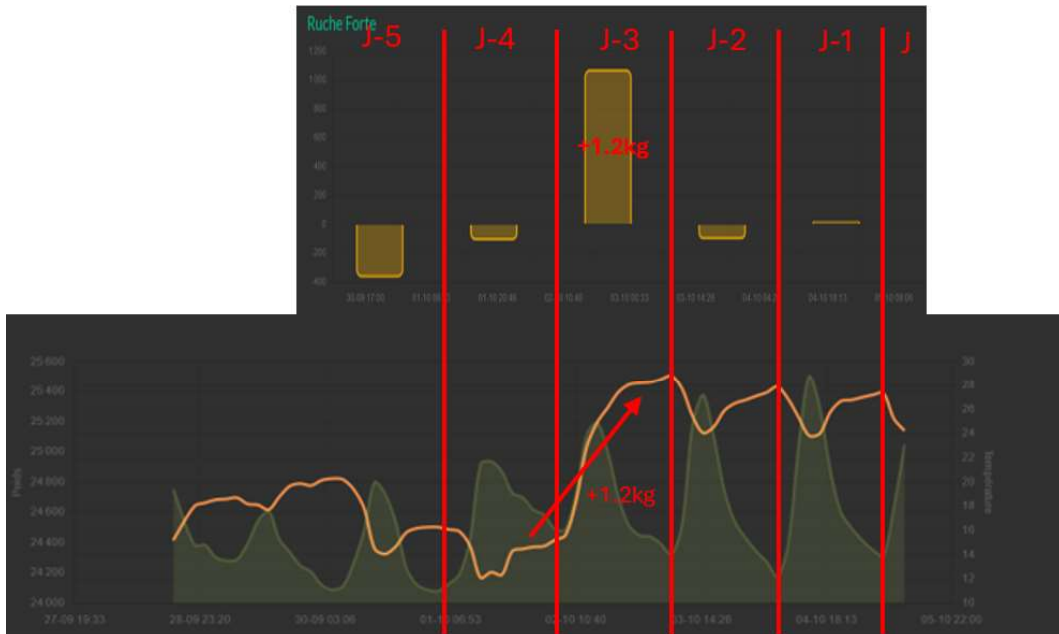


FIGURE 2.28 – Correspondance entre les données de masse (en bas, orange) et la variation journalière de masse (en haut, orange)



FIGURE 2.29 – Visualisation des données de pesée et des données traitées sur l'application dédiée aux apiculteurs. La courbe de droite montre l'évolution du poids (jaune) et de la température au fil du temps (vert). Le graphique de gauche illustre l'évolution quotidienne du poids pour une interprétation rapide des tendances.

2.7 Conclusion

Ce chapitre présente une architecture versatile pour la surveillance des ruches, basée sur un système de capteurs sans fils dont les données sont centralisée et transmises par une gateway. Ce système se distingue des architectures filaires et tout-en-un proposées par la littérature et l'industrie dans le domaine apicole [18]. Le système propose un standard pour créer des capteurs *low cost* à faible consommation d'énergie, *low power*, tout en offrant des capacités de calcul suffisantes pour embarquer des compensations et traitements analytiques. Cette solution s'inscrit dans la suite des stratégies utilisées dans le domaine industriel [44] tout en suivant les recommandations de [45] en matière de choix technologiques, notamment pour les protocoles et technologies radio. Ceci permet d'assurer la compatibilité avec d'autres capteurs présents dans les environnements agricoles. La solution proposée est conçue pour être industrialisable, en répondant aux exigences de coût, consommation énergétique, modularité et de maintenabilité.

En conclusion, l'architecture proposée est non seulement flexible et optimisée pour les conditions de l'apiculture moderne, mais elle est aussi prête pour des applications de *edge computing*, renforçant ainsi sa capacité à accueillir des modèles de compensation des erreurs des capteurs ou des impacts des variations de l'environnement. La collecte de données et leur stockage servira ensuite aux apiculteurs à optimiser la gestion des ruchers en tirant les enseignement des mesures issues des années précédentes.

2.7. Conclusion

Chapitre 3

Amélioration d'un capteur low cost

Contents

3.1	L'origine de la dérive thermique	56
3.1.1	Caractérisation de la dérive issue de la chaîne d'acquisition	56
3.1.2	Qualité des jauges de contraintes	58
3.1.3	Correction et calibration par sonde et globale	61
3.1.4	Visualisation de la dérive thermique	62
3.2	Correction dynamique par régression polynomiale	64
3.2.1	Modèle dynamique de correction	65
3.2.2	Identification du polynôme de correction	66
3.2.3	Performance de correction et mesure de sensibilité	69
3.2.4	Extrapolation du modèle	72
3.2.4.1	Exagération du modèle pour illustrer le sur-apprentissage	72
3.2.4.2	Limite de l'ordre du polynôme pour éviter le sur-apprentissage	73
3.2.5	Conclusion de la régression linéaire sur toutes les données	74
3.3	Compensation circadienne avec une mesure par jour IMJ	74
3.3.1	Première visualisation de la dérive journalière	75
3.3.2	Séparation temporelle par filtres complémentaires	77
3.3.3	Corrections linéaires de l'IMJ	78
3.3.4	Résultats de l'IMJ	79
3.4	Discussion sur la possibilité d'embarquer le modèle dans un microcontrôleur	81
3.5	Conclusion	81

Les premières mesures réalisés sur des balances à charge constante suggèrent une dérive thermique. Ces observations sont cohérentes avec le choix pragmatique des jauges de contraintes "à faible coût". Après avoir confirmé l'origine de la dérive thermique, une compensation logicielle déterministe est mise en place pour améliorer le capteur, compensation basée sur des méthodes de régressions linéaires. Dans un second temps un indicateur journalier qui améliore encore les performances du capteur au prix de la fréquence des mesures est mis en œuvre, pour une application en accord avec le besoin apiculteur.

3.1 L'origine de la dérive thermique

La totalité des capteurs connaît une dérive en température, ainsi l'on observe des dérives sur les capteurs piezorésistifs [46], les centrales inertielles [47], mesures de courant [48], capteurs de pression [49], et cetera. Pour la plupart de ces capteurs, le problème de mesure provient de l'auto-échauffement du capteur pendant sa période de mesure en plus des variations thermiques extérieures. Dans le cas de la balance proposée, avec une fréquence d'échantillonnage de dix minutes, l'auto échauffement de la chaîne d'acquisition est peu probable. Il reste donc à caractériser la dérive thermique liée aux variations extérieures. Or, dans le cas d'une ruche en extérieur, les variations thermiques sur une journée peuvent être importantes ($\pm 15^{\circ}\text{C}$) et encore plus grandes sur la totalité d'une saison (-10°C à $+45^{\circ}\text{C}$). Malheureusement, les capteurs disponibles ne caractérisent pas la dérive thermique de leurs produits [29] mais se contentent de donner une plage de température de fonctionnement, probablement liée aux limites du hardware.

3.1.1 Caractérisation de la dérive issue de la chaîne d'acquisition

Pour s'assurer que la dérive thermique n'est pas issue de la chaîne d'acquisition électronique : référence de tension et ADC. Le protocole de test suivant, illustré dans la fig. 3.1, est mis en place : un montage en pont diviseur sur les entrées de l'ADC est réalisé. Le pont diviseur place la tension de mesure au centre de la plage de mesure de l'ADC. Il est réalisé avec deux résistances à basse tolérance ($<1\%$) de même référence. Ces deux résistances issues du même lot connaîtront la même dérive thermique et garantissent donc une stabilité du point de tension présenté à l'entrée de la chaîne. Cela permet d'identifier la variation de la référence de tension et de l'acquisition des mesures ADC.

Les manipulations sont placées dans une étuve avec régulation de température fixée sur 30°C . La qualité de la régulation en mode "tout ou rien" de l'étuve étant de mauvaise qualité (gros cycle d'hystérésis), des cycles thermiques de 5°C environ sont observés. Sur ces amplitudes, on caractérise la

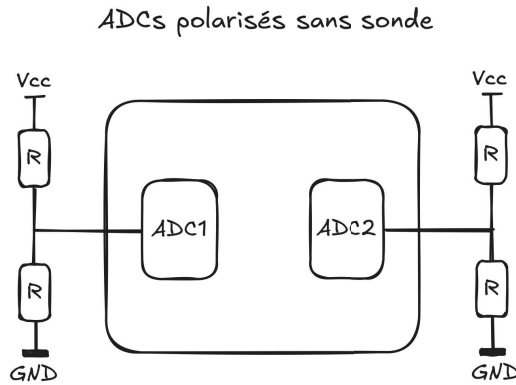


FIGURE 3.1 – Montage de polarisation de l'ADC sans les sondes pour caractériser la dérive thermique de l'assemblage

dérive en grammes par degrés de l'ensemble. Dans un second temps, le même test est réalisé avec les sondes placées à la place du pont de résistances pour caractériser la dérive en incluant les sondes.

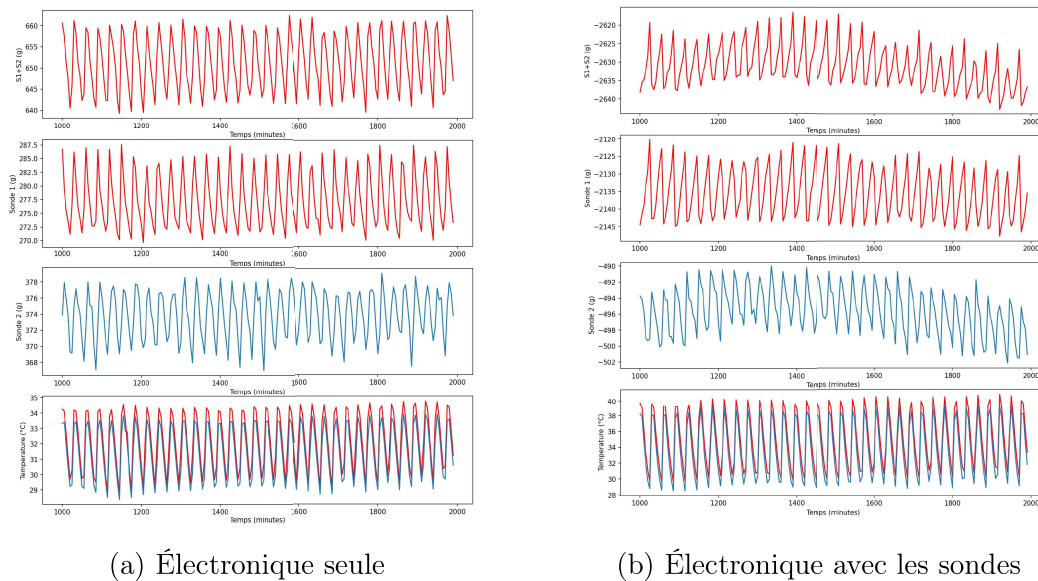


FIGURE 3.2 – Comparaison de la dérive thermique de la chaîne d'acquisition seule 3.2a et avec les jauges de contraintes connectées 3.2b

Les courbes présentées en 3.2a illustrent que la dérive thermique de l'électronique seule est directement proportionnelle à la température, en termes de valeurs mesurées. En revanche, lorsqu'une sonde est intégrée à la chaîne d'acquisition (voir 3.2b), celle-ci présente une inertie thermique plus importante. Cela se traduit par une constante de temps plus longue avant que le système n'atteigne une valeur stabilisée.

Une comparaison quantitative des résultats de la dérive thermique de la chaîne d'acquisition est mesuré en étuve avec et sans sonde sur 4 balances de

deux capteurs chacunes.

La plage de variation dynamique des signaux de masse (m) s_1 et s_2 de chaque sonde est calculée en prenant l'excursion maximale $\Delta_m(i) = \max_{0 \leq k < N} (s_i[k]) - \min_{0 \leq k < N} (s_i[k])$ où k est le numéro d'échantillon de la série temporelle, N le nombre de mesures et i le numéro de capteur. La plage dynamique de variation thermique est évalué à partir des signaux des capteurs T_1 et T_2 avec $\Delta_T = \max_{0 \leq k < N} \left(\frac{T_1[k] + T_2[k]}{2} \right) - \min_{0 \leq k < N} \left(\frac{T_1[k] + T_2[k]}{2} \right)$. Finalement la sensibilité d'un capteur unique est donnée par le ration des plages dynamiques $S_i = \frac{\Delta_m(i)}{\Delta_T}$ en $g/^\circ\text{C}$. Dans le cas d'une balance mesurant une masse avec deux sondes, la sensibilité est celle de la plage dynamique de la somme des signaux :

$$S = \frac{\max_{0 \leq k < N} (s_1[k] + s_2[k]) - \min_{0 \leq k < N} (s_1[k] + s_2[k])}{\Delta_T}$$

La moyenne des 8 sensibilités par capteur et des 4 balances est donnée dans la table 3.1.

TABLE 3.1 – Comparaison des dérives pour l'électronique seule et l'électronique avec sonde. Les essais ont été réalisés avec quatre balances soit 8 ADCs. La balance complète est la somme des mesures des deux ADCs composant une balance

	Élec seule ($\pm g/^\circ\text{C}$)	Élec et sonde ($\pm g/^\circ\text{C}$)
Par ADC	2,1	10.3
Balance complète	3,5	12.5

La sonde dégrade d'au moins un facteur 4 la sensibilité de la chaîne de mesure et est donc la principale source de dérive. De plus, un déphasage est observé entre la variation de température et la réponse thermique du système avec sonde, ce qui constitue un facteur supplémentaire à estimer lors de la compensation de la dérive thermique. En revanche, la dérive mesurée pour la chaîne d'acquisition sans sonde est très corrélée à la variation de température, ce qui la rend facilement compensable par régression linéaire.

La caractérisation de la dérive de la chaîne d'acquisition confirme que le problème de dérive thermique provient principalement des sondes. Un déphasage entre les variations rapides de température (observées dans les cycles thermiques de l'étuve) et la mesure de masse révèle un phénomène d'inertie thermique. L'estimation et la compensation de ce phénomène est abordé dans la sec. 3.1.4.

3.1.2 Qualité des jauges de contraintes

La balance est équipée de deux jauges de contrainte sur lesquelles sont apposées des sondes de déformation sous forme de peignes résistifs. Lorsque

la partie métallique de la jauge se déforme, elle entraîne une variation de la résistance des peignes. À partir de cette variation, il est possible de déterminer la masse.

Les premières observations, décrites en 3.1.1, révèlent que les jauges de contraintes présentent une dérive en fonction de la température. Une série de mesures avec une masse constante confirme que cette dérive thermique affecte l'ensemble de la balance. Les principales sources de cette dérive sont le coefficient de température de la résistance des jauges de contrainte et celui du module d'élasticité du matériau de l'élément [37].

Les observations présentées en 2.20 montrent une dérive thermique affectant les jauges de contrainte, avec des amplitudes et des directions de dérive différentes selon les sondes. Ces jauges étant supposées compensées en température, la couche protectrice des composants électroniques a été retirée pour mieux comprendre si cette dérive est due au support mécanique des jauges ou aux peignes résistifs. Une analyse plus détaillée de la dérive thermique des sondes est présentée en 3.1.4, corroborant les résultats de la littérature [37], qui décrivent une dérive non linéaire des jauges de contrainte.

L'image 3.3 représente une jauge de contrainte comme elle se présente à l'achat. Les peignes se situent sous une couche protectrice siliconée (visible en blanc sur la figure). Cette couche est retirée pour atteindre les peignes de résistances collés à la jauge.



FIGURE 3.3 – Présentation d'une jauge de contrainte

La photo 3.4 montre le collage des peignes de résistances sur la mécanique et révèle ainsi un placement irrégulier des peignes de résistances d'un côté à l'autre d'une même jauge de contrainte. Le détail des peignes résistifs est visible dans la photo 3.5b. Les peignes de résistances qui mesurent la déformation du support aluminium sont les peignes collés à l'extérieur de chaque côté de la sonde.

On reproduit ainsi le montage électrique de la jauge de contrainte, voir la figure 3.6, permettant de comprendre la correction thermique hardware mise

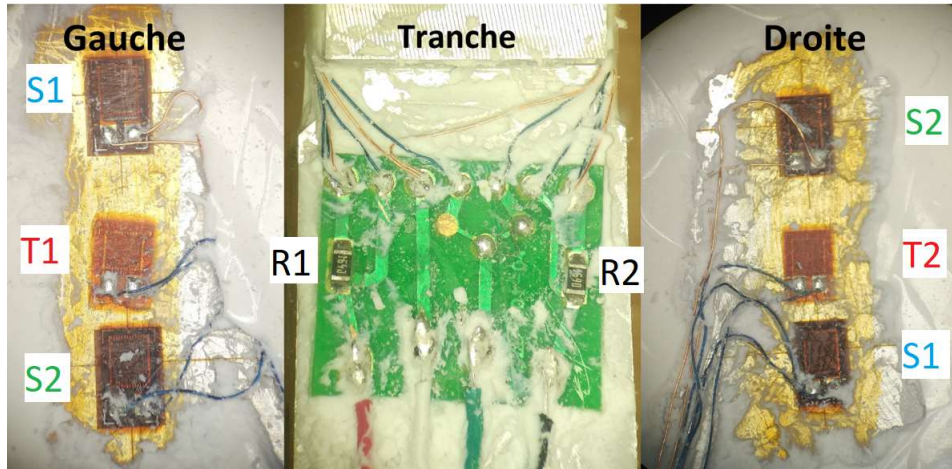
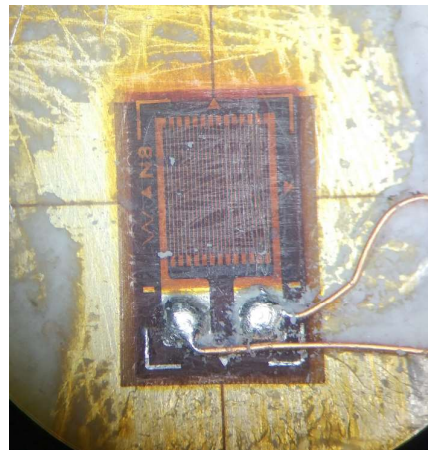


FIGURE 3.4 – Collage des peignes de résistances sur la jauge de contrainte



(a) Peigne central de la jauge de contrainte – compensation thermique T1 et T2



(b) Peignes résistifs extérieurs de la jauge de contrainte – Mesure de déformation S1 et S2

FIGURE 3.5 – Comparaison entre les peignes de la jauge de contrainte

en place. Ces peignes sont montés en pont de Wheatstone pour mesurer au mieux la déformation.

Les deux peignes centraux sont placés en encadrement du pont de Wheatstone pour moduler l'amplitude de la tension de mesure en fonction de la dérive thermique. Cela a pour objectif de compenser la dérive thermique par un ajustement de la tension de polarisation de la mesure. Le collage non régulier et non parallèle des peignes de résistances sur le support aluminium explique les variations thermiques des sondes ainsi que l'aspect différent en amplitude et sens de dérive.

La littérature présente les mêmes dérives thermiques sur les capteurs issus de technologie similaire [37]. En revanche il y est également spécifié que pour

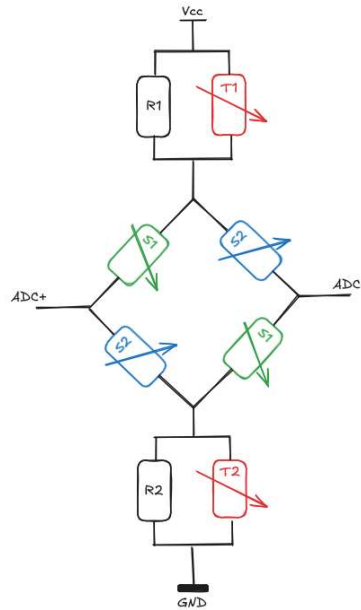


FIGURE 3.6 – Montage électronique des peignes de résistances sur la sonde. S1 : Compensation Thermique, S2 : Mesure poids

réaliser une compensation efficace de la dérive thermique, la meilleure méthode est de compenser chaque sonde individuellement par le hardware, et d'utiliser un signal en tension d'excitation de la sonde [50, 51]. Cependant, la compensation thermique n'est de toute manière pas possible étant donnée la dispersion de variation thermique des différents peignes de résistances ainsi que leur non linéarité. De ce fait, les jauges de contraintes dans les applications modernes fonctionnent plutôt pour des applications où la mesure est différentielle, ou lorsque les variations de températures sont lentes en fonction des événements à mesurer [52, 53]

3.1.3 Correction et calibration par sonde et globale

Les dispersions dans le processus de fabrication affectent le système de compensation des dérives thermiques présenté ci-dessus, et laissent ainsi une dérive thermique résiduelle pouvant même en changer le sens de variation, dans le cas de sur-compensation thermique. On nomme par la suite *correction*, l'étape qui consiste à réduire encore la dérive résiduelle en se basant sur un modèle statistique identifié dans une phase nommée *calibration*.

Chaque sonde peut être calibrée en usine et corrigée électroniquement ou bien corrigée numériquement. Dans ce dernier cas, la calibration fournit les coefficients d'un polynôme de correction en fonction de la température et le calculateur doit disposer d'une mesure de température qui soit au plus près de la jauge de contrainte. La mesure corrigée est alors de la forme :

$$M(t) = m(t) + p(T) \quad (3.1)$$

où M est la mesure corrigée, m la mesure brute et $p(T)$ la fonction de correction thermique.

Une telle calibration effectuée en usine représente un facteur de coût que la sonde soit fournie avec le système de correction électronique ou bien qu'elle soit fournie avec les coefficient du polynôme à appliquer au niveau du calculateur. De plus, l'électronique de mesure externe à la sonde ainsi que la mécanique de montage du plateau de pesée de la ruche sur deux jauges de contraintes apportent des dérives supplémentaires qui ne peuvent être prises en compte par cette correction.

Une phase de calibration du système de la balance complète sur une large plage thermique est nécessaire. Afin de caractériser la dérive globale de ces mesures en fonction de la température et pour pouvoir corriger cette dérive thermique, une session de mesures à masse constante est réalisée sur cinq balances. Ces mesures sont réalisées dans un hangar couvert mais exposé aux variations de température journalières extérieures. Afin de placer les balances dans le système typique d'utilisation, elles sont chargées avec des poids fixes et non évolutifs (Bidons d'eau). Les mesures sont faites sur une période de 7 jours consécutifs permettant d'observer 7 cycles thermiques circadiens (d'amplitude journalière de 30 degrés) et une tendance de variation lente dans la semaine.

La section suivante fait une analyse statistique de ces données et mets en évidence un cycle d'hystérésis limitant les performances d'une correction statique habituelle de la forme de l'eq. 3.1.

3.1.4 Visualisation de la dérive thermique

Dans un premier temps, la visualisation de la dérive thermique est présentée afin de bien dimensionner le problème et comprendre par la suite les méthodes de compensations mises en place. Dans le graphique 3.7, un axe temporel est utilisé :

- à gauche, le temps s'étend du jour 288 au jour 295, couvrant ainsi les 7 jours d'enregistrement ;
- à droite, le temps circadien est indiqué en heures, allant de 0 (minuit) à 24 heures.

Les données sont en quantum de l'ADC non converties en grammes pour la masse et en °C pour la température. La thermistance est une CTN ce qui explique l'inversion dans la dynamique de température, ainsi le minimum thermique observé a midi, correspond en fait à la température maximum. Les couleurs représentent les différentes balances utilisées pour la mesure qui sont identifiées par leur adresse MAC BLE unique. Ces cinq balances étant installées sur le même site, les diagrammes de température (en bas) fournissent des informations similaires d'une balance à l'autre. Les capteurs de température ne

semblent pas présenter de biais ni de dispersion significative entre les balances. En observant les données en fonction du temps circadien, des décalages temporels apparaissent entre les balances, attribuables aux effets d'éphémérides liés à un ensoleillement légèrement décalé de quelques minutes, en raison des distances de quelques mètres entre les balances.

L'observation des mesures de masse révèle plusieurs phénomènes :

- Des biais variables allant de 5000 à 30000 quantums ADC, soit plus de 200% de variation. Ce phénomène n'est toutefois pas problématique, car seule l'évolution journalière est pertinente pour l'utilisateur.
- Des sensibilités très disparates entre les capteurs. Par exemple, la balance violette (MAC f76d...) affiche des cycles d'environ 20000 quantums, tandis que la balance orange (fe5...) présente des variations dix fois moindres pour un même stimulus thermique.

Ces écarts sont attribués aux défauts de montage du système de compensation thermique, comme expliqué dans la section 3.1.

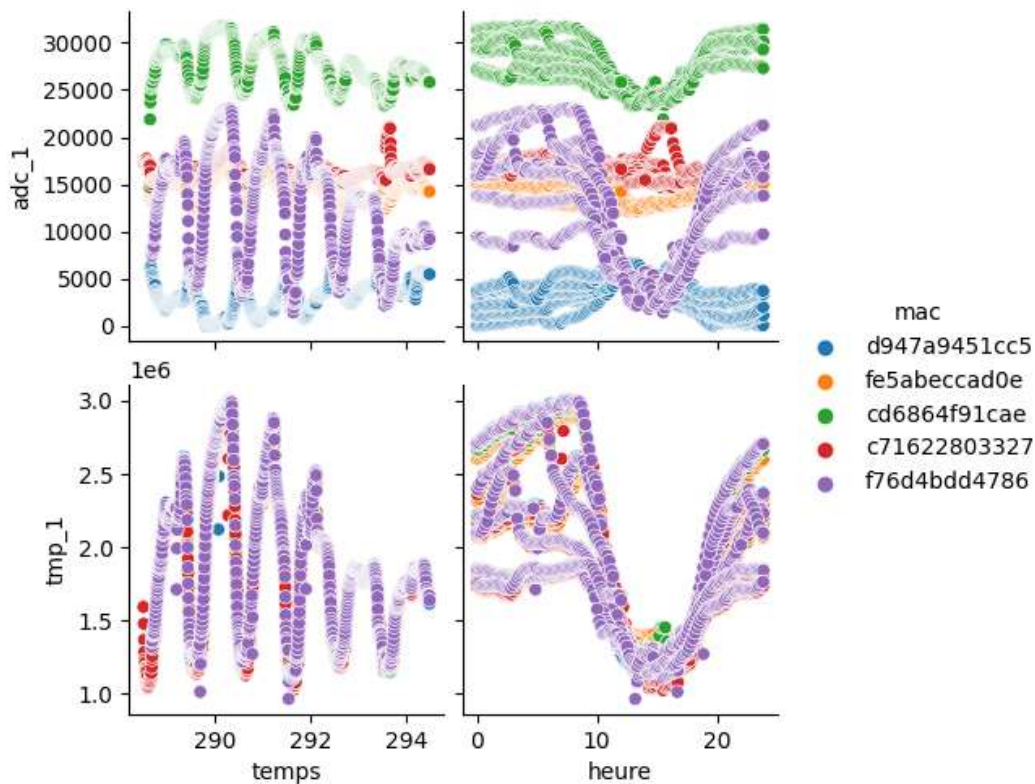


FIGURE 3.7 – Visualisation de la dérive thermique sur les cinq balances en test. L'axe des abscisses de gauche représente le temps durant la période de mesure, tandis que celui de droite indique l'heure au cours d'une journée. L'axe des ordonnées en bas montre les variations de température, tandis que celui en haut illustre les mesures de l'ADC issues d'une sonde sur chacune des cinq balances en test.

Deux diagrammes permettent d'illustrer les conclusions précédentes présentés dans 3.8 :

- En haut à gauche : ce diagramme représente la densité de répartition des mesures de température pour chaque balance. Il est confirmé que le cycle thermique mesuré est sensiblement similaire d'une balance à l'autre.
- En bas à droite : ce diagramme montre la densité de répartition des mesures de masse ADC1 pour chaque balance. De fortes différences de biais sont observées, notamment entre la balance bleue (MAC d947...) et la balance verte (MAC cd686...). Des écarts de sensibilité apparaissent également, avec les balances orange et rouge (MAC FE5A... et C716...) qui présentent une très faible sensibilité aux variations thermiques, tandis que la balance violette (MAC F76d4) affiche près de 100% de variation pour un même cycle thermique.

Le graphique en bas à gauche illustre les évolutions des mesures en fonction de la température. On peut noter :

- des différences de sens de variation, certaines balances présentant une dérive positive avec la température tandis qu'une autre (balance bleue MAC d947...) voit ses valeurs décroître avec l'augmentation de la température ;
- un phénomène d'hystérésis plus ou moins prononcé, avec des dérives distinctes pendant les phases de croissance et de décroissance thermiques.

Une correction polynomiale appliquée à la température semble être une solution potentielle pour corriger les problèmes de biais et de sensibilité (notamment en ce qui concerne le sens de variation). Cependant, le phénomène d'hystérésis suggère que l'utilisation d'une mémoire temporelle pour compenser le déphasage entre température et dérive thermique 3.1.1 pourrait être nécessaire pour modéliser la dynamique thermique ainsi que son sens de variation.

3.2 Correction dynamique par régression polynomiale

Les deux approches couramment utilisées pour compenser la dérive thermique des capteurs sont basées soit sur la régression linéaire ou polynomiale [49, 54], soit sur des méthodes de machine learning [46, 55] lorsque des volumes de données suffisants sont disponibles. Idéalement, pour des raisons de volume de données et de simplicité de calcul sur le microcontrôleur embarqué de la balance, la régression linéaire est privilégiée par défaut. En outre, afin d'éviter l'accumulation de grandes quantités de données, qui impliquerait de laisser la balance fonctionner sur de longues périodes, les méthodes de machine learning ne sont pas les plus adaptées dans ce contexte.

Le cycle d'hystérésis impose un modèle dynamique où le temps intervient afin de modéliser la dynamique thermique affectant le corps de la sonde et

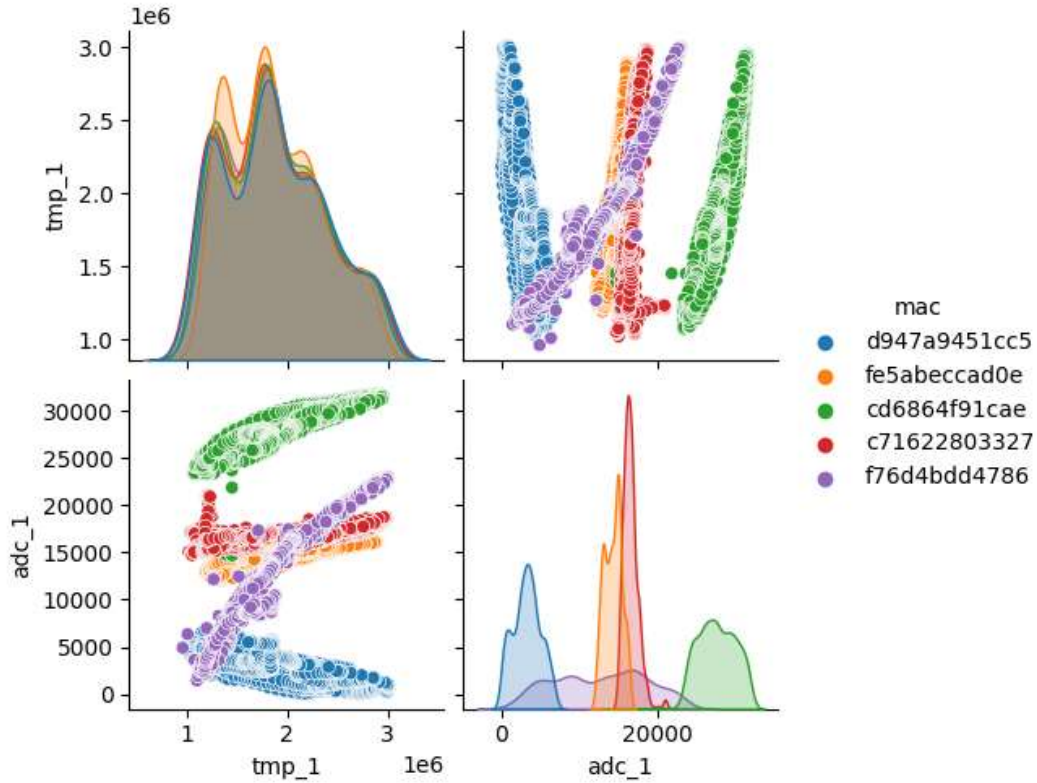


FIGURE 3.8

l'électronique. Ainsi un modèle dynamique du premier ordre pourrait être utilisé pour représenter cette dynamique de transfert de chaleur traversant le capteur de température vers le corps d'épreuve de la sonde.

3.2.1 Modèle dynamique de correction

L'équation statique de correction 3.1 devient alors dynamique de la forme :

$$M(t) = m(t) + p(T(t) - \tau.T'(t)) \quad (3.2)$$

où τ est la constante de temps du système et T' la dérivée temporelle de la température mesurée.

Si l'on considère par exemple le cas d'une correction polynomiale d'ordre 2, $p(x) = a.x^2 + b.x + c$ et que l'on discrétise les mesures, ainsi que la dérivée, on obtient alors un terme correcteur de la forme suivante :

$$M[k] = m[k] + a. \left(T[k] - \tau. \left(\frac{T[k] - T[k-1]}{T_e} \right) \right)^2 + b. \left(T[k] - \tau. \left(\frac{T[k] - T[k-1]}{T_e} \right) \right) + c \quad (3.3)$$

Quelle que soit la méthode de différentiation discrète utilisée, le terme de correction utilisé se simplifie en un polynôme de la variable $T[k]$ et de sa mémoire $T[k - 1]$, soit l'expression suivante :

$$M[k] = m[k] + a_{00}.T[k]^2 + a_{10}.T[k].T[k - 1] + a_{11}.T[k - 1]^2 + b_0.T[k] + b_1.T[k - 1] + c \quad (3.4)$$

$$= m[k] + p(T[k], T[k - 1]) \quad (3.5)$$

Ainsi une dynamique d'ordre deux, ou plus, nécessite une représentation par un polynôme de 3, ou plus, variables $T[k], T[k - 1], T[k - 2], \dots$

On généralise donc le modèle statique de l'eq.3.1 en un modèle dynamique constitué d'un polynôme de degré N de M variables choisies parmi les mémoires T_j de la température T :

$$M[k] = m[k] + p(\Theta[k]) \quad (3.6)$$

$$\text{où } \Theta[k] = (T_0[k], T_1[k], \dots, T_{M-1}[k]) = (T_j)_{0 \leq j < M} \quad (3.7)$$

$$\text{et } T_j[k] = T[k - N_j] \quad \text{avec } 0 \leq N_0 < N_1 < \dots < N_{M-1} \quad (3.8)$$

L'intérêt d'un tel modèle additif polynomial est de pouvoir identifier des dynamiques d'ordre arbitraire M en constituant un vecteur des retards de $T[k]$ inférieurs à M tout en gardant une correction non-linéaire d'ordre N. De plus, l'identification conjointe de la dynamique thermique et du polynôme de correction se formule sous une simple optimisation linéaire aux moindres carrés dont on peut garantir une bonne stabilité numérique.

3.2.2 Identification du polynôme de correction

Pour identifier à la foi le modèle dynamique de température et les coefficients de correction, nous proposons d'utiliser un nombre $P+M$ de mesures consécutives d'une température $T[k]$ relevées au plus proche de la sonde qui mesure les masses $m[k]$ non-corrigées. Dans le cas de la campagne de mesure menée dans ces travaux, le nombre P correspond à la durée d'apprentissage pouvant aller jusqu'à 7 jours de mesures échantillonnées à la période $T_e = 10mn$ soit $P=7.6.24=1008$ mesures de masses $m[k]$ consécutives. Et M est le nombre de mémoire d'échantillons de températures lié à l'ordre de la dynamique de la température.

Comme la masse est constante lors de ces mesures d'étalonnage, et que l'on ne désire pas obtenir de mesure absolue de cette masse (seule une mesure relative stable en température est désirée), nous établissons le vecteur des corrections thermiques désirées $b \in \mathbb{R}^P$ par le vecteur des écarts à la moyenne :

$$b = (m[k] - \bar{m})_{0 \leq k < P} \quad \text{où } \bar{m} = \frac{\sum_{0 \leq k < P} x[k]}{P} \quad (3.9)$$

Ainsi, nous cherchons le polynôme d'ordre N de M variables, minimisant l'erreur quadratique de correction :

$$\|b - \underbrace{(p(\theta[k]))_{0 \leq k < P}}_y\|_2 \quad (3.10)$$

Où $y \in \mathbb{R}^P$ est le vecteur des P évaluations consécutives du polynôme de correction appliqué au vecteur des M mémoires de la température.

On note x le vecteur des coefficients du polynôme $p_x : \mathbb{R}^M \rightarrow \mathbb{R}$ d'ordre N qui calcule la valeur de la correction pour un vecteur de M mémoires de températures. Le nombre R des coefficient de x correspond au nombre de monômes différents du polynôme p_x . On définit l'application de $\mathbb{R}^R \rightarrow \mathbb{R}^P$ qui associe à chaque vecteur de coefficients x le vecteur $y = (p_x(\Theta[k]))_{0 \leq k < P}$ des P évaluations du polynômes aux points $\Theta[0]$ à $\Theta[P-1]$. Cette application est linéaire et sa matrice associée A permet de calculer le vecteur de correction y de l'ensemble des données d'étalonnage.

$$\begin{aligned} y &= A \cdot x \\ [P \times 1] &= [P \times R] \cdot [R \times 1] \end{aligned} \quad (3.11)$$

où $[a \times b]$ indique une matrice de a rangées par b colonnes.

Ainsi au coefficient α_r du polynôme qui est associé au monôme de forme $T_0^{p_0(r)} \cdot T_1^{p_1(r)} \cdot \dots = \prod_{0 \leq j < M} T_j^{p_j(r)}$ (avec $\sum_{0 \leq j < M} p_j(r) \leq N$ pour un polynôme d'ordre N) correspond une colonne $y_r \in \mathbb{R}^P$ de la matrice A qui contient les valeurs de ce monôme aux différents instants k allant de 0 à $P-1$, soit :

$$A = (y_r)_{0 \leq r < R} \quad \text{avec } y_r = \left(\prod_{0 \leq j < M} T[k - N_j]^{p_j(r)} \right)_{0 \leq k < P} \quad (3.12)$$

Pour éclaircir cette écriture, reprenons l'exemple du polynôme d'ordre $N=2$ avec une mémoire dynamique $M=1$ de l'eq.3.5 que l'on rappelle ici :

$$p(T[k], T[k-1]) = a_{00} \cdot T[k]^2 + a_{10} \cdot T[k] \cdot T[k-1] + a_{11} \cdot T[k-1]^2 + b_0 \cdot T[k] + b_1 \cdot T[k-1] + c$$

Dans ce cas le polynôme $p(T[k], T[k-1])$ (car la mémoire $M = 1$) d'ordre 2, donne $R = 6$ coefficients (3 d'ordre 2, 2 d'ordre 1 et la constante) avec un vecteur des coefficients $x = (a_{00}, a_{10}, a_{11}, b_0, b_1, c)$. La matrice A comporte donc 6 colonnes avec P rangées correspondant aux P corrections calculées de l'instant $k = 1$ à $k = P$ pendant la phase d'étalonnage. Soit la matrice des monômes de température suivante :

$$\underbrace{\begin{pmatrix} p_x(T[1], T[0]) \\ p_x(T[2], T[1]) \\ \vdots \\ p_x(T[k], T[k-1]) \\ \vdots \\ p_x(T[P], T[P-1]) \end{pmatrix}}_y = \underbrace{\begin{pmatrix} T[1]^2 & T[1].T[0] & T[0]^2 & T[1] & T[0] & 1 \\ T[2]^2 & T[2].T[1] & T[1]^2 & T[2] & T[1] & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T[k]^2 & T[k].T[k-1] & T[k-1]^2 & T[k] & T[k-1] & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T[P]^2 & T[P].T[P-1] & T[P-1]^2 & T[P] & T[P-1] & 1 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} a_{00} \\ a_{10} \\ a_{11} \\ b_0 \\ b_1 \\ c \end{pmatrix}}_x$$

En faisant apparaître ainsi la linéarité du problème de régression polynomiale, les coefficients optimaux du polynômes \hat{x} au sens des moindres carrés :

$$\hat{x} = \arg \min_{x \in \mathbb{R}^R} \left\| \begin{array}{c} \underbrace{m - A.x}_{\text{mesure corrigée}} - \underbrace{\bar{m}}_{\text{mesure cible}} \end{array} \right\|_2 = \arg \min_{x \in \mathbb{R}^R} \|b - A.x\|_2 \quad (3.13)$$

La solution analytique est obtenue par la matrice pseudo inverse de Moore-Penrose :

$$\hat{x} = ({}^T A.A)^{-1} . {}^T A.b \quad (3.14)$$

La matrice orthogonale ${}^T A.A$ pouvant être mal conditionnée numériquement, le calcul de cet optimal n'est pas effectué directement mais en utilisant la librairie python `scikit-learn`. Le conditionnement numérique est amélioré en effectuant une normalisation préalable des mesures.

De plus une contrainte de type Lasso ou Ridge peut être utilisée pour compresser les coefficient \hat{x} trouvés. L'optimisation devient alors une optimisation sous contraintes ce qui donne un critère (exprimé avec le Lagrangien) :

$$\hat{x} = \arg \min_{x \in \mathbb{R}^R} \|b - A.x\|_2 + \lambda \|x\|_1 \quad (3.15)$$

Ainsi, la sensibilité du polynôme aux mesures est réduite favorisant la propriété de généraliser la correction avec de nouvelles données.

L'algorithme de calibration s'effectue donc selon les étapes :

- **Normalisation des mesures :** La valeur moyenne et l'écart type de la mesure brute m et de la température T sont calculés sur les P+M échantillons enregistrés lors de l'étalonnage. Chaque mesure est ensuite centrée et normalisée avec $\frac{m[k]-\bar{m}}{\sigma_m}$ et $\frac{T[k]-\bar{T}}{\sigma_T}$. Les coefficients de normalisation sont mémorisés pour pouvoir être appliqués lors des corrections de mesures, ils peuvent être communs ou spécifique à chaque balance.
- **Détermination des hyper-paramètres N, M et λ** En fonction de la durée d'étalonnage P disponible, les hyper-paramètres N et M du polynôme optimaux pour l'apprentissage sont déterminés.

- **Calcul du polynôme optimal** : La matrice A des monômes de températures normalisées est construite, et les coefficients \hat{x} du polynôme sont calculés avec la méthode du Lasso de paramètre λ

L'algorithme de correction thermique embarqué dans chaque balance nécessite la connaissance des 4 paramètres de normalisations et des R coefficients du polynôme correcteur de chacune des deux sondes. À chaque nouvelle itération, chacune des deux mesures de masse sont corrigées :

- **Normalisation et mémorisation** : Les mesure $m[k]$ et $T[k]$ sont normalisées avec $\frac{m[k]-\bar{m}}{\sigma_m}$ et $\frac{T[k]-\bar{T}}{\sigma_T}$. La nouvelle valeur est mémorisée avec les M mémoires de la température.
- **Calcul des monômes thermiques** : les nouveaux monômes normalisés impliquant $T[k], T[k]^2, T[k-1].T[k]$ sont calculés et insérés dans la table a des R monômes normalisés. Cette table de monômes possède la dimension et le format d'une rangée de la matrice A
- **Calcul de la correction** $p(a) = a.\hat{x}$: Le calcul du polynôme est un simple produit scalaire de R termes.
- **Application de la correction à la mesure** : La mesure corrigée de chaque sonde est calculée par simple soustraction : $M[k] = m[k] - p(a)$. L'étape inverse de la normalisation est appliquée avec $M[k].\sigma_m + \bar{m}$

3.2.3 Performance de correction et mesure de sensibilité

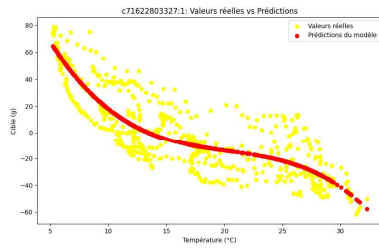
La modélisation de l'erreur peut être réalisée soit comme une fonction affine linéaire, soit comme un polynôme de la température mesurée, en minimisant l'erreur quadratique ou bien l'erreur absolue maximale (H_∞). Les paramètres principaux de cette modélisation sont les suivants :

- **ordre** : Dans le cas de méthodes polynomiales, ce paramètre correspond à l'ordre du polynôme. Un ordre de 1 revient à une estimation affine.
- **mémoire** : Ce paramètre indique le nombre d'échantillons de température précédents à inclure dans l'équation linéaire ou polynomiale. Lorsque mem=0, seul un échantillon de température est utilisé, que ce soit de manière linéaire ou polynomiale. Ce paramètre sert à limiter le déphasage thermique présenté et observé en 3.1.4

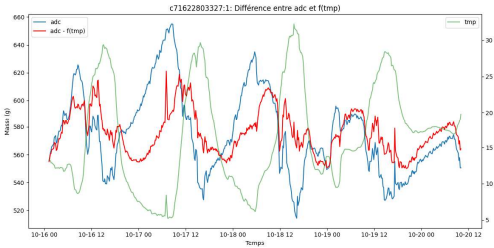
La fig.3.9 représente les performances de corrections d'une balance.

Les hyper paramètres utilisés pour les résultats présentés en 3.9 sont les suivants : un modèle polynomial d'ordre 3 (N=3) et une mémoire nulle (M=0) pour ignorer l'hystérésis. L'étalonnage est réalisé sur la totalité de la période. Les résultats obtenus à partir des calculs de compensation thermique révèlent une fois de plus le problème d'hystérésis associé aux variations rapides de température. Sur la courbe 3.9d, il est clairement visible que la correction (en rouge) est très précise, sauf durant les périodes de forte variation de température où la masse fluctue également de manière importante. La sonde 1 de cette

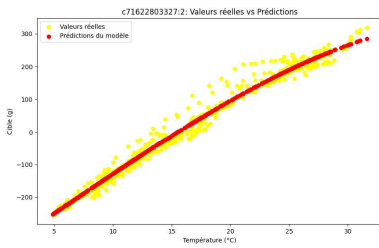
3.2. Correction dynamique par régression polynomiale



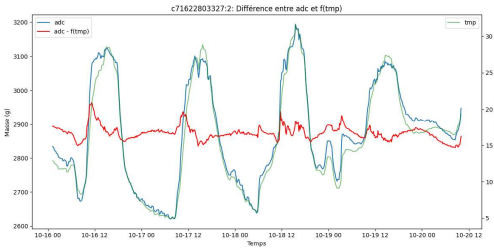
(a) Cible et correction sonde 1



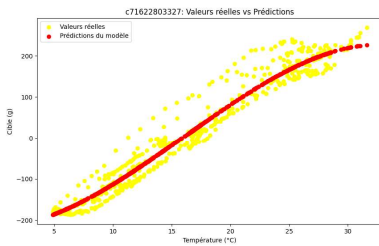
(b) Résultat compensation sonde 1



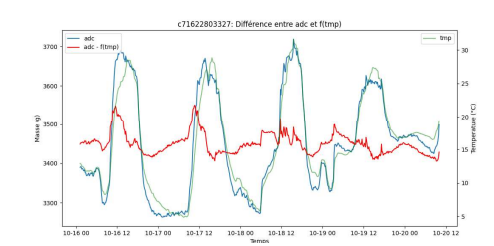
(c) Cible et correction sonde 2



(d) Résultat compensation sonde 2



(e) Cible et correction balance



(f) Résultat compensation balance totale

FIGURE 3.9 – Représentation de la régression polynomiale appliquée aux sondes de la balance c71622803327. Les graphiques à gauche illustrent la comparaison entre la valeur cible de la correction (courbe jaune) et la prédiction obtenue par la fonction polynomiale implémentée (courbe rouge). Les graphiques à droite montrent les données brutes de l'ADC (en bleu), les valeurs compensées (en rouge), et la température (en vert). Les sous-figures 3.9a et 3.9b correspondent aux résultats pour la sonde 1, 3.9c et 3.9d pour la sonde 2, et 3.9e et 3.9f représentent l'ensemble de la balance avec l'agrégation des deux sondes.

balance présente une amplitude de dérive thermique beaucoup plus faible et son profil de variation, déjà bien compensé par le hardware, montre une plus grande inertie face à la dérive thermique, ce qui explique l'absence des zones de plateau observées sur la sonde 2. Cela constitue un défi lors de l'implémentation de la fonction de correction, car si les variations au centre de la plage de température sont trop dispersées, l'amélioration globale obtenue n'est pas aussi significative qu'espéré. En observant la figure 3.9a, il apparaît qu'un polynôme

d'ordre 3 est nécessaire pour correspondre au mieux au schéma observable. Cet ordre a donc été retenu et fixé comme une valeur appropriée. La valeur du déphasage par défaut est de 2, ce qui représente 30 minutes de déphasage, cohérent avec la dérive observée en étuve et présentée dans la section 3.1.1.

La plage de température analysée, comprise entre 5 et 30°C, représente environ la moitié de la plage totale d'utilisation, qui s'étend de -10 à +45°C. Il est donc raisonnable de supposer que la tendance polynomiale ne subira pas de variations importantes sur le reste de la plage, qui s'étend symétriquement de chaque côté de la courbe utilisée pour l'analyse. Après examen des courbes spécifiques des sondes, il a été constaté que la compensation est particulièrement efficace pour la sonde présentant l'amplitude la plus élevée de dérive thermique (sonde 2). La table 3.2 montre l'amplitude maximale de la dérive thermique pour la balance c71622803327, dont les courbes sont présentées, ainsi que les résultats moyens obtenus pour l'ensemble des sondes et balances sur lesquelles le test a été réalisé.

Ces résultats confirment une compensation efficace pour les balances avec la plus grande dérive thermique, tandis que l'amélioration reste minime pour les sondes bénéficiant déjà d'une bonne compensation thermique matérielle. Toutefois, la compensation globale par balance, plutôt que par sonde, ne semble pas offrir un avantage significatif en termes de performance brute. En effet, les performances compensées par balance reproduisent essentiellement celles compensées par sonde, avec de légères variations selon que les dérives des sondes se compensent ou s'amplifient mutuellement. Le véritable problème de la compensation demeure ce cycle d'hystérésis causé par les variations rapides de température, qui pose particulièrement problème au centre de la plage de mesure.

La compensation réalisée sur la ruche totale, c'est à dire la somme des mesures issue des deux ADCs

TABLE 3.2 – Dérive brute et dérive compensée des sondes et de la balance.

	Dérive brute (\pm g/°C)	Dérive compensée (\pm g/°C)
c71622803327 Sonde 1	5.21	2.63
c71622803327 Sonde 2	21.42	4.99
c71622803327 Balance	33.88	5.66
Moyenne par sonde	14.96	5.36
Écart Type par sonde	10.14	2.25
Moyenne par Balance	28.96	5.95
Écart Type par Balance	7.75	2.17

La régression polynomiale permet de réduire la dérive thermique des sondes qui sont très mal compensées thermiquement et ont une plage de variation très large. En revanche, sur les sondes qui sont déjà bien compensées et où l'hystérésis représente une plus grosse proportion de la dérive, les améliorations ne

sont pas aussi nettes. Pour cette raison, la mise en place d'un indice de variation journalier est mis en place pour essayer d'éliminer ces phases de variations rapides et conserver uniquement la tendance pour obtenir une correction plus efficace.

3.2.4 Extrapolation du modèle

La méthode d'apprentissage semble être la bonne mais repose sur une régression linéaire sur l'entièreté de la période de test. Pour valider que l'apprentissage sur une période plus courte va pouvoir s'extrapoler sur les mesures suivantes, la technique de cross-validation par "n-folds" est mise en place. Sur ces sets de mesures, un fold est une journée de mesure, et avec sept jours d'enregistrement il est possible de réaliser la méthode avec $n=7$. La méthode est présentée en 3.10. L'un des risques pour la non extrapolation est le sur-apprentissage, c'est à dire que la fonction de correction est tellement précise sur la période donnée lors de l'apprentissage, qu'elle ne parvient pas du tout à corriger des données légèrement différentes. La seconde observation faite est la baisse des performances en utilisant moins de volume de données pour déterminer les coefficients de correction. Il est tout de même important de réaliser une correction sur une période courte pour réaliser un produit car la phase de caractérisation des balances pour établir la correction peut être coûteuse si elle est longue et donc disqualifier le produit pour une utilisation commerciale.

3.2.4.1 Exagération du modèle pour illustrer le sur-apprentissage

L'ordre du polynôme pour ce set de test est fixé à $N=5$ avec une mémoire $M=5$, représentant 1 heure de retard sur les données. L'objectif est de donner la possibilité au modèle de converger au plus près du jeu de données proposé.

Lors des phases d'entraînement effectuées sur une journée, les erreurs maximales observées atteignent $\pm 1,9$ g/°C, tandis que la sensibilité moyenne est évaluée à 0,011 gramme par degré! Cependant, lorsqu'une correction basée sur ces résultats est appliquée à des données provenant de jours non inclus dans l'entraînement, l'erreur moyenne s'élève à 4,239 kg, avec une sensibilité de 758 g/°C. L'utilisation d'un modèle sans restriction sur l'ordre du polynôme permet d'améliorer les performances d'apprentissage sur l'ensemble de la période étudiée, aboutissant même à une amélioration des performances. Néanmoins, ce modèle présente des signes de sur-apprentissage (overfitting), entraînant une dégradation importante des performances lorsque l'extrapolation est appliquée à des périodes non incluses dans l'apprentissage.

La figure 3.11 illustre les résultats du sur-apprentissage sur les deux sondes d'une même balance. La journée utilisée comme période d'apprentissage est représentée en vert, où la dérive initiale (en bleu et rouge) est efficacement compensée à 0.4 grammes près. En revanche, il est observé que la correction appliquée aux autres périodes, indiquée en noir, aggrave considérablement les résultats, au point d'introduire une dérive thermique supplémentaire.

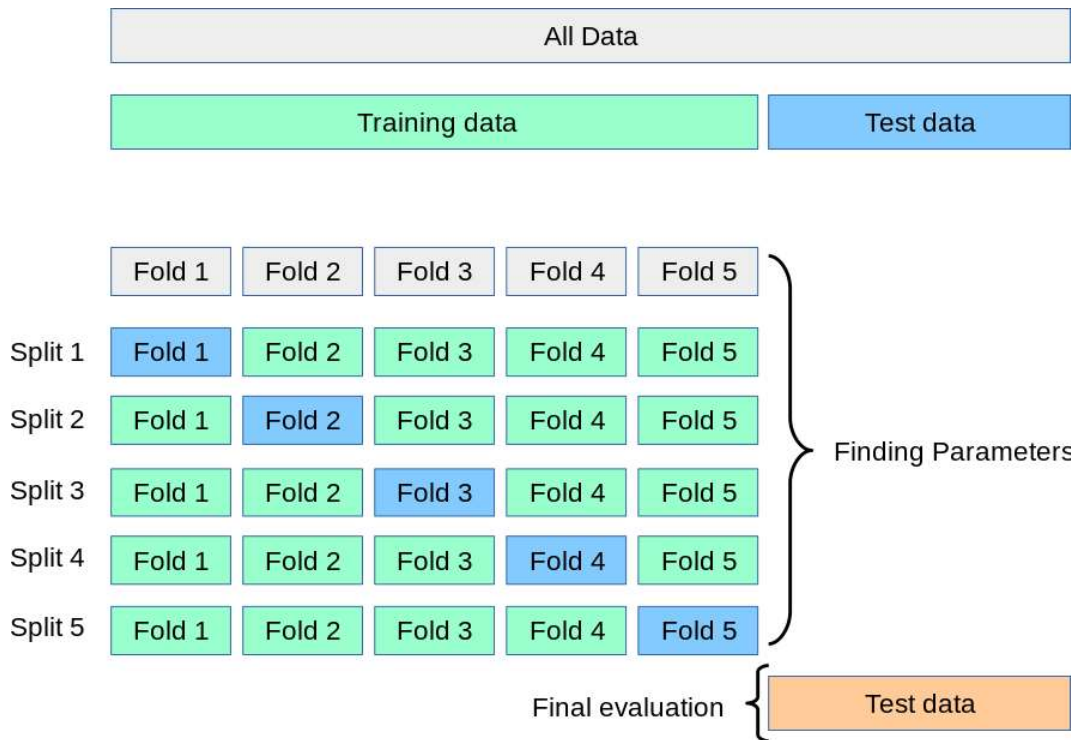


FIGURE 3.10 – Illustration de la technique de cross validation avec n-folds ou ici n=5

3.2.4.2 Limite de l'ordre du polynôme pour éviter le sur-apprentissage

Afin de prévenir le sur-apprentissage, il est nécessaire de limiter l'ordre du polynôme ainsi que la profondeur de mémoire utilisée. Dans le cas d'un polynôme de premier ordre basé sur une seule mesure de température, la correction de la mesure de masse s'effectue à l'aide d'un terme de la forme $c[k] = a \cdot T[k] + b$. Une fois les coefficients a et le biais b estimés par optimisation selon un critère H_1 , H_2 , ou H_∞ sur les données collectées durant une journée, l'estimation à l'instant k est corrigée selon l'expression suivante :

$$\text{masse}[k] = \text{ADC}[k] - c[k].$$

La figure 3.12 présente les résultats de cette méthode. Les performances sur le jour d'apprentissage (vert) sont moindres, mais une meilleure extrapolation sur les jours de tests (rouge et bleu) est réalisée. La sensibilité moyenne est réduite de 16 à 7 grammes par degrés grâce à cet algorithme.

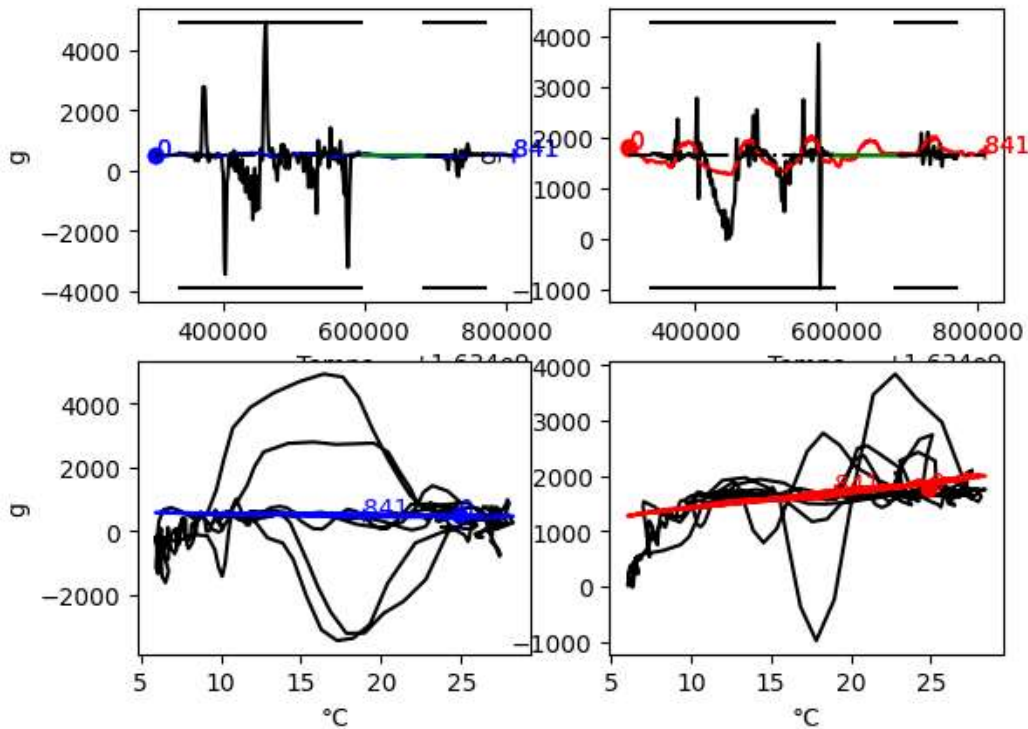


FIGURE 3.11 – Exemple de sur-apprentissage sur une ruche avec apprentissage sur une journée et extrapolation sur le reste de la période

3.2.5 Conclusion de la régression linéaire sur toutes les données

La meilleure solution se trouve sur un compromis entre l'ordre du polynôme et le degré de retard choisi. Sur les quelques essais réalisés, la meilleure méthode semble dépendre d'une sonde à l'autre, il faudrait augmenter le nombre de ruches observées et la quantité de données pour voir se décider une tendance. Cependant, en gardant à l'esprit que l'objectif est de réaliser un capteur low cost, cette méthode avec 5 jours d'observation des données de balances à vide augmente considérablement le coût de production. Il est alors nécessaire de trouver une autre méthode, plus proche des besoins des apiculteurs : l'indice de masse journalier (IMJ).

3.3 Compensation circadienne avec une mesure par jour IMJ

Le besoin apiculteur 1.2.2.4 étant d'avoir à disposition une mesure unique par jour à heure fixe pour éviter le surnombre de données, cela nous donne la possibilité de réduire le temps de réponse pour augmenter la fiabilité des mesures. Cela va nous permettre de supprimer les variations rapides de tempé-

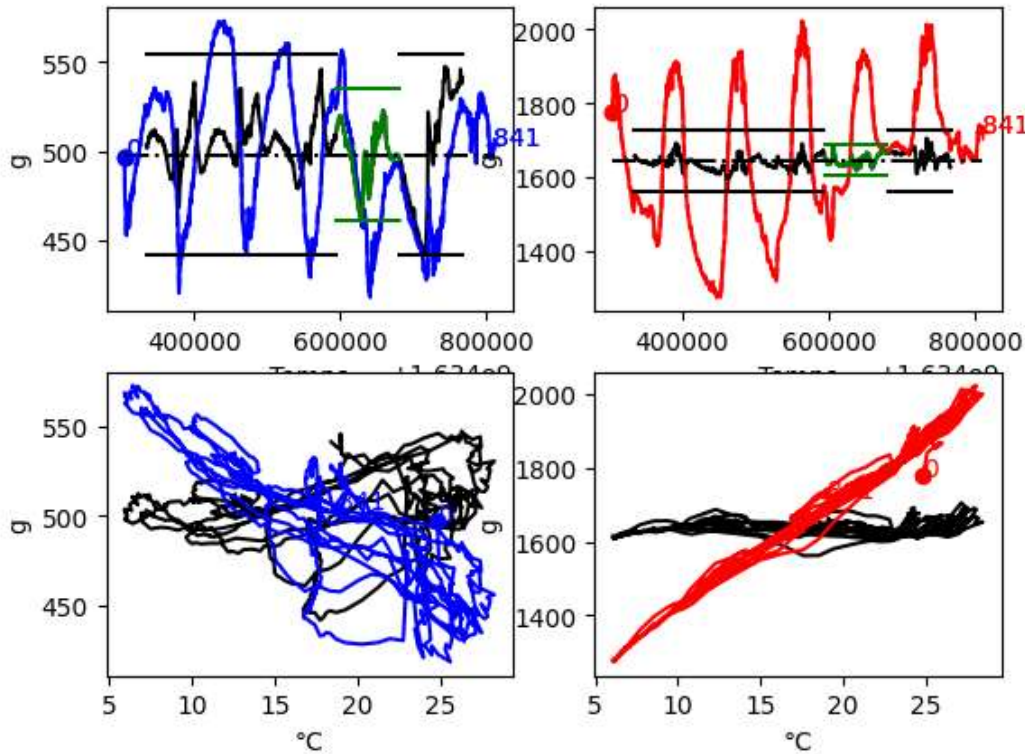


FIGURE 3.12 – Réduction du sur-apprentissage en utilisant un polynôme de degré 1 et sans retard.

rature qui posent le problème de déphasage et d'hystérésis pour se concentrer sur de l'information plus fiable.

3.3.1 Première visualisation de la dérive journalière

Le premier indice journalier est obtenu par sous-échantillonnage des données collectées entre 18h et 20h. La donnée journalière utile, notée **IMJ**, correspond à la masse mesurée en fin d'après-midi, au moment où l'activité des abeilles diminue avec le retour des butineuses à la ruche. Il est donc nécessaire d'effectuer un sous-échantillonnage quotidien à cette période.

Pour éliminer le bruit et éviter les phénomènes de repliement (aliasing), un filtre anti-repliement est appliqué au préalable. Un filtre simple, de réponse impulsionnelle rectangulaire d'une durée de deux heures (de 18h à 20h), est utilisé. Ce filtre à moyenne glissante, dont la réponse impulsionnelle h est définie par une fenêtre boxcar de deux heures, est exprimé de la manière suivante :

$$h : k \mapsto \begin{cases} 1 & \text{si } 0 \leq k < N_0 \\ 0 & \text{sinon} \end{cases}$$

où N_0 représente le nombre d'échantillons correspondant à la durée de moyennage. Dans ce cas, $N_0 = 6 \times 2 = 12$, car un échantillon est pris toutes

les 10 minutes.

La réponse y du filtre FIR (à réponse impulsionnelle finie) à une entrée x est donnée par la convolution suivante :

$$y[k] = \frac{1}{N_0} \sum_{n=0}^{N_0-1} x[k-n].$$

Cette moyenne glissante peut être calculée quotidiennement entre 18h et 20h, ou bien continuellement à l'aide d'un filtre IIR (à réponse impulsionnelle infinie) sous la forme récurrente suivante :

$$y[k] = x[k] - y[k - N_0].$$

Cette approche permet d'obtenir une estimation lissée des données tout en minimisant les effets du bruit, puis, échantillonnée à 20H. L'application de ce sous-échantillonnage donne les résultats présentés dans la figure 3.13. Les points bleus et croix oranges indiquent la valeur de l'IMJ calculée tous les jours à 20 heures. Les variations journalières et aberrations sont correctement filtrées avec cette méthode. On remarque que l'IMJ dégage une tendance à la hausse pour l'ADC1 tandis que l'ADC2 varie à l'opposé. Même si les variations circadiennes sont filtrées correctement, les dérives en température lentes ne sont pas rejetées. C'est cette dérive que nous souhaitons compenser.

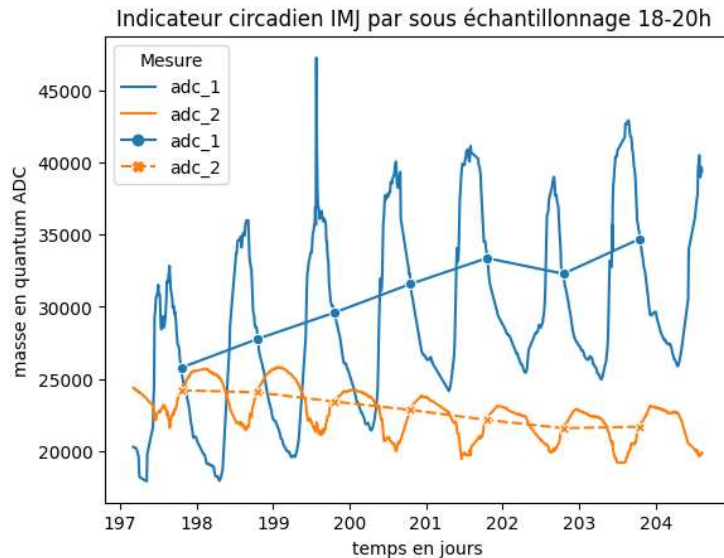


FIGURE 3.13 – Dérive Thermique des sondes de la balance CD6864F91CAE et extraction de la variation journalière de ces sondes pour créer un IMJ

La méthode de sous échantillonnage est pertinente sur les données à vides car il n'y a pas d'activité des abeilles. On est donc sûr, à heure fixe, que la masse ne varie pas. En revanche, lors de l'ajout d'une colonie d'abeilles sur la balance, des variations de masse peuvent encore se produire. Une méthode de

séparation temporelle est donc proposée afin de réaliser le sous échantillonnage journalier en valeur glissante 3.3.2.

3.3.2 Séparation temporelle par filtres complémentaires

La méthode de séparation temporelle du signal a pour objectif de séparer les variations rapides (saisonnalité circadienne) de la dérive thermique des variations lentes (tendance journalière). Les variations lentes de température sont prises sur une période d'une journée. Pour ce faire, la transformée de Fourier discrète (DCT) ainsi que des filtres à réponse impulsionnelle finie (FIR) sont utilisés. Le signal initial $s(t)$ est tout d'abord prolongé à l'aide de techniques d'interpolation afin de garantir une continuité dans les transformations et éviter les discontinuités qui introduiraient du bruit dans le signal (phénomènes de Gibbs). Ensuite, la transformée de Fourier est calculée sur le signal, et notée $\hat{s}(f)$.

Un filtre passe-bas est alors appliqué pour ne conserver que les composantes basses fréquences, correspondant aux variations thermiques lentes. Le filtre passe-bas est défini par la fonction de transfert $H(f)$, donnée par l'équation suivante :

$$H_{bas}(f) = \begin{cases} 1 & \text{si } f < f_c \\ 1 - \frac{f-f_c}{\theta \cdot f_c} & \text{si } f_c \leq f < f_c + \theta \cdot f_c \\ 0 & \text{sinon} \end{cases}$$

Le filtre complémentaire permettant d'extraire les hautes fréquence est $H_{haut}(f) = 1 - H_{bas}(f)$ La fréquence de coupure f_c est définie par $f_c = \frac{1}{6 \times 24}$, ce qui correspond à une période d'une journée (avec une fréquence d'échantillonnage des mesures de 10 minutes). Les composantes du signal ayant une fréquence supérieure à celle d'une journée sont filtrées et considérées comme des variations thermiques rapides, tandis que celles plus lentes représentent la dérive thermique lente, ciblée par l'indicateur IMJ.

Un paramètre important dans cette méthode est le chevauchement θ (overlap ou rolloff factor), qui contrôle le degré de recouvrement entre les deux composantes du signal séparé. Ce chevauchement, généralement fixé à 0.7, permet d'assurer une transition plus douce entre les composantes haute fréquence et basse fréquence, en évitant les artefacts brusques lors de la séparation.

Lors de la phase d'étalonnage, les filtres sont calculés directement en utilisant la transformée DCT hors-ligne. Après l'application du filtre passe-bas pour isoler les basses fréquences, le signal est reconstruit dans le domaine temporel à l'aide de la transformée de Fourier inverse. Toutefois, lors du calcul de la correction en temps réel, un filtre FIR est utilisé pour cette séparation. Il est défini par un nombre de taps $ntaps$, typiquement fixé à $ntaps = 1024 \times 4$, ce qui détermine la longueur du filtre et affecte la précision de la séparation des fréquences.

Cela permet d'obtenir deux composantes distinctes du signal : la composante basse fréquence $s_{bas}(t)$, qui représente la dérive thermique lente (ten-

dance journalière), et la composante haute fréquence $s_{haut}(t)$ (saisonnalité circadienne), qui contient les variations rapides de la température. Nous avons ainsi la décomposition :

$$s(t) = s_{bas}(t) + s_{haut}(t)$$

Qui est obtenue hors-ligne lors de la calibration dans le domaine fréquentiel :

$$\hat{S}(f) = H_{bas}(f) \cdot \hat{S}(f) + H_{haut}(f) \cdot \hat{S}(f)$$

Et est calculée en temps réel lors de la correction de mesure par un filtre FIR :

$$s[k] = h_{bas} \star s[k] + h_{haut} \star s[k]$$

Où \star est le produit de convolution discrète et h_{haut} la réponse impulsionnelle du filtre. Les résultats de séparation de la méthode sur une sonde sont présentés dans la figure 3.14 avec en bleu le signal d'origine. En rouge sont représentées les variations thermiques lentes, le filtre réalise bien un lissage des variations thermiques rapides tout en conservant la tendance de la dérive en température journalière. Les variations rapides sont quand à elles en vert et centrées en 0.

Cette méthode permet ainsi une analyse plus fine des variations lentes et rapides du signal que la méthode proposée en 3.3.1. Elle est la méthode utilisée pour l'indicateur IMJ dans la suite de l'étude.

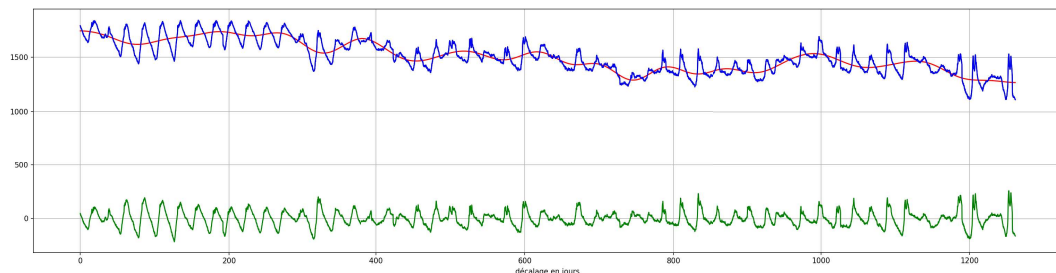


FIGURE 3.14 – Résultat de la séparation temporelle sur une sonde. Le signal d'origine est en bleu, en rouge sont les variations lentes et en vert les variations rapides qui ont été séparées l'une de l'autre

3.3.3 Corrections linéaires de l'IMJ

Maintenant que la tendance journalière est séparée de la saisonnalité circadienne, l'objectif est de corriger uniquement les variations lentes afin d'évaluer si cette approche améliore les performances en évitant l'impact des hystérésis thermique et la dynamique. La correction linéaire permet de modéliser la relation entre la composante lente de la température tmp_{bas} et les mesures analogiques adc_{bas} . Cette relation est décrite par l'équation suivante :

$$\hat{g}_{bas} = \beta_0 + \beta_1 \cdot tmp_{bas}$$

où :

- \hat{y}_{bas} est la valeur corrigée de adc_{bas} ,
- tmp_{bas} est la composante basse fréquence de la température,
- β_0 est l'ordonnée à l'origine,
- β_1 est le coefficient de pente, représentant la sensibilité de adc_{bas} à tmp_{bas} .

Le modèle est ajusté en minimisant l'erreur quadratique moyenne entre les valeurs observées $y_{bas,i}$ et les valeurs prédites $\hat{y}_{bas,i}$, donnée par :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{bas,i} - \hat{y}_{bas,i})^2$$

où n est le nombre d'observations.

En isolant et en compensant uniquement les variations lentes (composantes basses fréquences), il est possible de réduire les écarts dus aux phénomènes lents, comme les effets de la température. Cette approche permet de laisser intactes les composantes rapides du signal, qui sont susceptibles de contenir des informations pertinentes pour l'analyse des performances. La qualité globale des prédictions est ensuite évaluée en termes de réduction des erreurs, tout en maintenant les variations rapides non corrigées.

La méthode de validation croisée présentée en 3.10 est utilisée pour vérifier si cette correction des variations lentes améliore effectivement les performances du modèle, notamment en termes d'erreur quadratique moyenne (MSE).

3.3.4 Résultats de l'IMJ

La sensibilité des capteurs est présentée dans la table 3.3 en comparant la dérive thermique brute à celle compensée. Il est évident que les performances sont significativement améliorées passant d'une dérive moyenne par capteur de $\pm 15.13g/^{\circ}C$ avec un écart type de $\pm 11.21g/^{\circ}C$ à des valeurs proches de $\pm 2g/^{\circ}C$ sans dispersion. Les capteurs répondent de manière non homogène aux variations thermiques mais ont pour autant une dérive similaire une fois compensée. Les écarts extrêmes observés dans la sensibilité de base sont considérablement atténués, ce qui améliore la précision des mesures et réduit l'influence indésirable de la température. Les variations thermiques lentes d'une journée à l'autre ont une dynamique beaucoup plus faible que les variations à l'intérieure d'une même journée, ce qui fait de cet indicateur une référence fiable pour les mesures.

En appliquant la correction ainsi calculée aux données brutes, donc avec les variations rapides, on obtient les résultats présentés dans la courbe 3.15. Sur ce graphe, chaque couleur représente les données brutes d'une sonde et la valeur en pointillées noires est le résultat compensé. Les variations lentes de température sont effectivement compensées ce qui améliore l'horizontalité et donc la non variation de masse des capteurs. En revanche, sur la sonde rose, déjà très bien compensée, le résultat empire légèrement les données d'origine pour les caler sur les performances de la moyenne des sondes. La méthode de

3.3. Compensation circadienne avec une mesure par jour IMJ

TABLE 3.3 – Comparaison de la sensibilité avant (*Dérive thermique brute* ($g/^\circ C$)) et après correction (*Dérive thermique compensée* ($g/^\circ C$))

	Dérive brute ($\pm g/^\circ C$)	IMJ Dérive compensée ($\pm g/^\circ C$)
Mean	15,13	1,97
Std	11,21	0,12
Min	4,01	1,85
25%	6,15	1,91
50%	9,39	1,95
75%	26,66	1,97
Max	30,83	2,19

compensation IMJ est validée en fonctionnement avec les défauts de nécessiter une période de mesure à vide de 24 heures ainsi que d'introduire un déphasage dans la mesure donnée de cette même période de 24h. Une compensation des variations rapide pourrait être introduite pour encore améliorer le résultat.

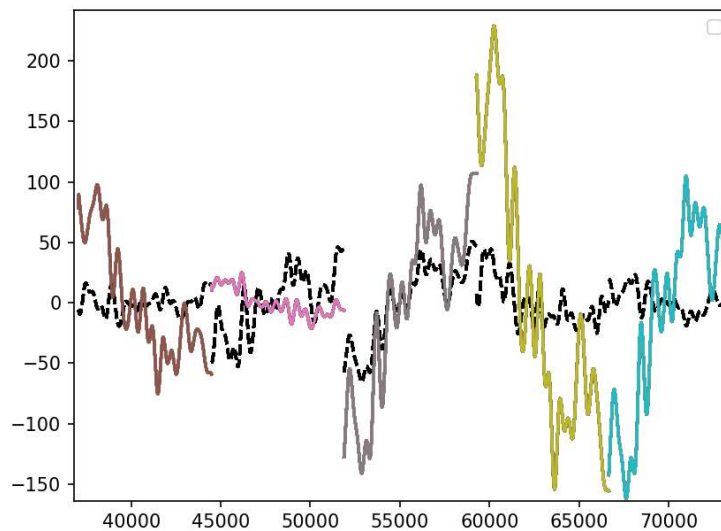


FIGURE 3.15 – Résultats de la correction IMJ en réinjectant les variations rapides dans le résultat. Les offsets des mesures ont été retirés pour plus de lisibilité

3.4 Discussion sur la possibilité d'embarquer le modèle dans un microcontrôleur

La correction thermique implémentée utilise un modèle polynomial avec retard, ce qui en fait une solution légère en termes de calcul. La complexité des opérations augmente de manière exponentielle avec l'ordre du polynôme choisi et le nombre de mesures de retard, mais l'analyse des résultats, présentée dans la section 3.2.4.1, met en évidence un problème de surapprentissage au-delà d'un certain seuil. Par conséquent, il est démontré qu'une correction polynomiale de faible ordre reste efficace tout en maintenant une charge de calcul raisonnable.

Sur un microcontrôleur de la série nRF52 cadencé à 64 MHz, équipé d'une unité de calcul en virgule flottante (FPU), les opérations matricielles nécessaires à la correction sont exécutées rapidement. Le temps d'exécution de ces calculs reste significativement plus court que celui requis pour transmettre les données via le protocole Bluetooth Low Energy (BLE). Cela permet de réaliser l'intégration de cette correction directement sur le capteur en améliorant les performances globale du système : qualité des mesures et consommation du système. Une fois les coefficients de correction déterminés, leur taille de stockage demeure négligeable par rapport au volume des données collectées, ce qui est compatible avec la mémoire disponible.

De plus, la capacité de stockage du microcontrôleur, qui permet de conserver jusqu'à trois jours de données continues (cf. chapitre 2), offre une flexibilité supplémentaire pour l'exploitation des données. Cette capacité permet d'implémenter directement l'Indice de Masse Journalier (IMJ) "in situ" en conservant les mesures sur une période glissante de 24 heures. Ainsi, toutes les informations nécessaires au calcul de l'IMJ sont disponibles localement, sans nécessiter de transmission externe ou de traitement supplémentaire sur un serveur distant. Cette approche réduit non seulement la consommation d'énergie due aux transmissions fréquentes, mais elle renforce également l'autonomie et l'efficacité du système embarqué.

En résumé, la solution de correction thermique proposée se distingue par sa simplicité, son faible coût de calcul et sa compatibilité avec une architecture de capteur embarqué à faible consommation d'énergie.

3.5 Conclusion

Les sondes utilisées dans ce travail présentent des dérives thermiques importantes en raison d'une mauvaise compensation matérielle. Conformément aux observations rapportées dans la littérature [37], ces dérives varient en amplitude et en polarité, rendant la correction uniforme complexe. Bien que l'électronique de la chaîne d'acquisition présente également une dérive, celle-ci reste relativement faible et proportionnelle aux variations de température, facilitant

ainsi sa compensation.

Une méthode d'extraction d'un Indice Massique Journalier (IMJ) est proposée pour rejeter les perturbations thermiques ainsi que les activités circadiennes de la ruche. Cet indicateur, restant toutefois sensible aux effets thermiques lents, nécessite une correction thermique supplémentaire, basée sur une phase de calibration d'une journée (un cycle thermique complet). Cette méthode, développée à partir d'outils algébriques simples, permet une implémentation sur un nœud de mesure, si nécessaire.

Le problème posé par le phénomène de sur-apprentissage sur une seule journée est contourné en proposant une méthode paramétrique de faible complexité, dont l'extrapolation sur de nouvelles données est évaluée par cross-validation de type "N-fold". La méthode d'apprentissage linéaire a ainsi démontré de bonnes propriétés d'extrapolation.

La sensibilité à la température des capteurs non corrigés a été évaluée, avec une dispersion en $g/^{\circ}C$ comprise entre 4 et 44, avec une moyenne de $14.96 g/^{\circ}C$. L'extraction d'une tendance lente journalière et l'application d'une correction linéaire permet de réduire cette sensibilité à $1.97 g/^{\circ}C$. Ces performances permettent d'améliorer le capteur en au-delà des spécifications du produit de $3g/^{\circ}C$ annoncés, mais n'atteignent pas les performances annoncées d'une sonde haut de gamme 2.6

Les travaux futurs consistent à déterminer la meilleure méthode pour bien caractériser la dérive thermique et la compenser, tout en minimisant le coût pour le produit final. Cela inclut la recherche de la période de temps optimale pour mesurer la balance à vide : suffisamment longue pour éviter le sur-apprentissage d'une journée, mais suffisamment courte pour limiter les opérations de manutention et les coûts engendrés.

Chapitre 4

Exploiter l’audio par l’IA

Contents

4.1	Importance du signal audio pour la santé de la ruche et applications actuelles	84
4.2	Revue de la littérature	86
4.2.1	Diversité des données audio utilisées dans la littérature scientifique	86
4.2.2	Labels utilisés pour la classification	89
4.2.3	Extraction de features pour la représentation temps/fréquence des fichiers audio	91
4.2.4	Modèles de classifications	93
4.2.5	Problème d’extrapolation vers une nouvelle ruche	94
4.3	Méthodologie	96
4.3.1	Unification des datasets publics : Construction du framework BeeTogether	96
4.3.1.1	Standardisation des données brutes	96
4.3.2	Extraction des features	98
4.3.2.1	Choix des features extraites pour application au modèle	99
4.3.3	Augmentation de données	103
4.3.4	Modèle d’apprentissage issu de l’état de l’art	104
4.3.5	Nouvelle méthode de classification inspirée du contrastive learning	106
4.3.6	Méthode de validation	107
4.3.6.1	Validation classique avec ensembles d’entraînement et de test :	107
4.3.6.2	Validation par extrapolation de type k-hive-fold :	107
4.3.7	Mesures de performance du modèle	107
4.4	Reproduction des performances de l’état de l’art et mise en lumière du problème d’extrapolation	108

4.4.1	Validation de l’approche CNN + Mel-Spec	108
4.4.2	Le label Abeille–PasAbeille est-il utile?	110
4.4.3	Sur la difficulté d’extrapoler la classification de la présence de la reine à des ruches non vues	111
4.4.3.1	Non-extrapolation des modèles classical et pairwise	111
4.4.3.2	La classification peut être axée sur de mau- vais paramètres	115
4.5	Extrapolation avec un modèle inspiré du contras- tative learning	119
4.5.1	Prospective sur une base de données non labellisée : piste de validations	121
4.6	Évaluation de la possibilité d’embarquer le modèle dans un microcontrôleur	124
4.7	Conclusion	126

En prolongeant l’idée de réduire le coût, la question posée est comment substituer la pesée qui impose une électronique compensée mais surtout une mécanique de support, en maintenant la pesée dans d’autres ruches? L’approche que nous allons envisager consisterait à équiper une ruche avec pesée et capteur audio pour apprendre et corrélérer en machine learning audio et variation de masse ; puis transférer cet apprentissage vers d’autres ruches seulement équipées du capteur audio.

L’idée sous-jacente est que l’audio est moins coûteux à développer qu’une balance tout en étant en capacité de déduire la production de miel corrélée à l’activité de la ruche. Cependant, l’analyse de la littérature ne montre pas de travaux réalisés dans cette direction et, plus préoccupant, révèle que les modèles d’IA appliqués aux sons des abeilles ne parviennent pas à s’extrapoler à de nouvelles ruches non vues lors de l’entraînement des modèles.

Ce chapitre se concentre sur le développement d’une méthode pour évaluer la faisabilité de l’extrapolation et se divise en plusieurs sections :

- La présentation des travaux existants dans la littérature pour justifier le choix des méthodes utilisées.
- Les défis liés au manque de données : présentation de l’unification des datasets publics pour un usage global.
- L’analyse des signaux audios pour comprendre la non extrapolation
- La présentation de la méthode inspirée du contrastive learning qui fonctionne.

4.1 Importance du signal audio pour la santé de la ruche et applications actuelles

En plus des phéromones, des contacts physiques et des interactions visuelles, le son joue un rôle essentiel dans la communication et l’organisation

sociale des abeilles. Les recherches de Wenner [56] ont été les premières à démontrer l'importance de la communication sonore chez les abeilles. D'autres études, comme celles de [57], ont fourni une analyse plus approfondie des sons spécifiques émis par les reines d'abeilles, tels que les signaux de "tooting" et de "quacking". La "danse frétilante", bien connue pour son rôle dans la communication de l'emplacement des sources de nourriture aux membres de la ruche, intègre à la fois des signaux sonores et vibratoires.

Des recherches supplémentaires menées par [58] ont approfondi ces perspectives en étudiant la communication acoustique chez les abeilles, en éclairant le rôle des signaux vibratoires émis non seulement par les reines, mais aussi par les ouvrières. Cette étude a révélé comment ces signaux sont transmis à travers la structure des rayons de la ruche et perçus par les abeilles, facilitant ainsi des interactions sociales cruciales telles que le partage de nourriture et la coordination des activités de la ruche.

Bien que les vibrations mécaniques semblent jouer un rôle dans la communication [59], et que des accéléromètres MEMS (systèmes microélectromécaniques) ou des capteurs piézoélectriques puissent être utilisés pour les mesurer [27, 60], très peu d'études ont été consacrées à ce sujet, et les vibrations n'ont pas été étudiées comme une caractéristique principale ou complémentaire [18].

Plus récemment, de nombreux articles scientifiques sont parus ayant pour objectif de diagnostiquer des éléments de santé de la ruche. Comprendre l'importance du son au sein d'une colonie d'abeilles a conduit à diverses initiatives visant à exploiter ces données, afin d'aider les apiculteurs à développer des méthodes pour surveiller la santé des colonies. Des travaux plus récents ont utilisé des techniques modernes de traitement numérique du signal et des algorithmes d'apprentissage automatique pour approfondir l'étude de ce système complexe de communication.

L'article de Ferrari *et al.* [61] vise à développer une méthode pour la détection précoce de la période d'essaimage dans les ruches. En enregistrant et en analysant les sons à l'intérieur des ruches, ainsi que les mesures de température et d'humidité, l'objectif est d'identifier des signatures acoustiques spécifiques et des schémas environnementaux associés à l'essaimage. Cette approche vise à permettre aux apiculteurs d'anticiper et de gérer les événements d'essaimage plus efficacement, réduisant ainsi l'impact économique de la perte de production de miel et améliorant les pratiques de gestion des ruches.

Des études récentes initiées par [62] et [63] ont déplacé l'accent vers l'analyse des sons émis par les abeilles en utilisant des techniques avancées d'extraction de caractéristiques telles que les coefficients de Cepstre de Fréquence Mel (MFCC). L'objectif était de classifier différents scénarios au sein de la ruche pour surveiller la santé de la colonie d'abeilles. Ces études se sont concentrées sur l'identification de la présence d'abeilles dans un fichier audio, menant au développement de méthodes innovantes pour la surveillance de la santé.

De plus, des revues complètes [64], [6] et [65] ont mis en évidence l'importance croissante de l'analyse sonore pour la surveillance non invasive des

ruches. Ces revues ont montré comment les sons au sein de la ruche pouvaient révéler des informations cruciales sur la santé de la colonie et faciliter la détection de variations à l'aide d'équipements simples, tels qu'un microphone et un système d'acquisition. L'évolution des méthodes d'analyse sonore, des premières observations à des algorithmes sophistiqués capables de classifier des états complexes de la ruche, démontre la reconnaissance croissante de l'acoustique comme paramètre essentiel pour la surveillance des ruches.

La métrique de santé analysée par plusieurs articles (par exemple, [62], [64], [40], [66], [67], [41], [43], [68]) est devenue la présence de la reine au sein de la colonie. Ce facteur critique met en lumière le rôle essentiel de la reine dans la stabilité et la productivité de la colonie.

En résumé, la communauté académique se concentre sur les deux indicateurs suivants à extraire de la classification audio :

- Bee–NoBee : qui indique si un bourdonnement significatif d'abeilles est enregistré et prépondérant par rapport aux interférences et aux bruits (voix humaine, animaux, voitures, ...)
- Queen–NoQueen : indique si une reine acceptée est présente dans la ruche.

Seul l'indicateur Queen–NoQueen est pertinent pour l'agriculture. Il est à noter qu'il manque des ensembles de données annotées empêchant l'étude d'indicateurs tels que la production de miel, l'infestation par le varroa et les attaques de frelons.

4.2 Revue de la littérature

4.2.1 Diversité des données audio utilisées dans la littérature scientifique

La table 4.1 présente les datasets de données audio utilisés par les principaux articles du domaine. La majeure partie des articles cités utilisent les quelques bases publiques à disposition qui sont labellisées. Dans la dernière année, quelques datasets privés ont commencé à faire leur apparition suite à l'engouement dans le domaine en 2018-2019 des suites des temps de conception et d'acquisition de ces données. Cela étant, peu de personnes utilisent des bases privées ou des bases non labellisées [69]

Comme souligné dans 4.1, trois datasets publics sont principalement utilisés dans la littérature pour les tâches de classification supervisée : NUHIVE, OSBP, et BUZZ. Ces ensembles de données sont détaillés ci-dessous en indiquant le nombre de ruches, les caractéristiques hardware et les labels disponibles. Il est à noter qu'en raison des fusions, croisements et extensions multiples des datasets, il est difficile de retracer la provenance des données de chaque ruche. La figure 4.1 illustre l'origine entremêlée des ensembles de données présentés.

TABLE 4.1 – Table de correspondance des citations et des datasets utilisés.

Date	Référence	NUHive	OSBP	BUZZ	SBCM	Dataset Privé
2018	Kulyukin[39]			•		
2018	Nolasco[63]	•	•			
2018	Cerjroski[62]					
2019	Nolasco[70]	•				
2019	Robles-Guerrero[40]					1
2020	Terenzi[64]	•				
2020	Zgank[71]		•			
2021	Kim[72]		•			
2021	Orlowska[66]	•	•			
2021	Zgank[73]		•			
2021	Kulyukin[74]			•		
2022	Quaderi[75]	•	•			
2022	Soares[67]	•				
2022	Kampelopoulos[41]					2
2023	Robles-Guerrero[42]					1
2023	Di[76]					•
2023	Truong[77]			•		
2023	Farina[78]					•
2023	Ruvinga[79]					•
2023	Phan[80]			•		
2023	Uthoff[69]					•
2023	Kanelis[68]					2
2023	Rustam[81]	•	•			
2023	Barbisan[43]				•	
2024	Bricout[38]	•	•		•	



FIGURE 4.1 – Visualisation de l'origine et de l'organisation des datasets publics

— Open Source Beehive Project (OSBH) — [82] : OSBH est un pro-

jet collaboratif rassemblant les fichiers audio fournis par des apiculteurs volontaires à travers le monde. De ce fait, il n’y a pas de spécification matérielle commune. La base OSBH est privée mais un sous ensemble nommé OSBP est accessible au public pour la recherche. Il peut être trouvé à l’adresse <https://zenodo.org/records/321345> (Consulté le 17/04/2024).

- **Projet NuHive (NUHIVE) — [83]** : Le projet NuHive fournit des enregistrements audio de deux ruches, avec et sans reine, accessibles à <https://zenodo.org/records/2667806> (Consulté le 18/07/2024). Les spécifications matérielles sont les suivantes : 32kHz, Stéréo, microphones ADMP401, carte son Behringer UCA222 [64]. Les mesures audio ont été effectuées sur seulement 4 jours mais en presque continu ce qui en fait une base de données avec un gros volume de données (Ruche1 le 12/06/2018 et le 31/05/2018 ; Ruche2 le 12/07/2017 et le 15/07/2017).
- **To Bee or Not to Bee (TBON) — [84]** : Ce jeu de données fusionne des données provenant des projets OSBP et NuHive, annotées avec la présence d’abeilles et de reine. Il est disponible à <https://www.kaggle.com/datasets/chrisfilo/to-bee-or-no-to-bee> (Consulté le 18/07/2024). Les données du dataset TBON sont composées à 65% de données issues du projet NUHIVE et qui sont donc redondantes. Les autres données sont issues de quatre ruches labellisées avec la présence ou non de la reine. Cependant ces quatre ruches ont un seul label, il n’est pas possible de travailler à séparer le label sur ces ruches en particulier.
- **To Bee or Not to Bee Processed (TBON processed) — [85]** : Une version améliorée de TBON, ayant les fichiers audio segmentés et annotés pour une analyse détaillée. Ce jeu de données peut être trouvé à <https://www.kaggle.com/datasets/yevheniiklymenko/bee-hive-buzz-anomalies> (Consulté le 18/07/2024). 4.2 illustre le contenu de la base de données TBON processed. Notez que la plupart des audios contenus proviennent des ruches 1_NUHIVE et 3_NUHIVE (65,78%) qui proviennent de la base de données NUHIVE. Ainsi, les données de ces ruches de la base TBON ont été exclues pour éviter les redondances. Les références à la base TBON dans la suite du document sont liées à la base processée et non brute.
- **Smart Bee Colony Monitor (SBCM) — [86]** : Le dataset SBCM comprend des enregistrements de ruches pendant un remérage artificiel offrant des informations sur le comportement des abeilles pendant ces événements, accessible à <https://www.kaggle.com/datasets/annajyang/bee-hive-sounds> (Consulté le 18/07/2024). Ici, le label de présence de la reine est enrichi avec les étiquettes Ancienne Reine, Nouvelle Reine, Présente et Rejetée, et Sans Reine. Les spécifications matérielles sont : Mono, 22,05kHz, 24 bits, Microphone référence INMP441. La durée des enregistrements est de 8 jours en juin 2022 pour deux ruches, et 1 mois en juillet 2022 pour les deux autres ruches.

Ruche ID	Reine	Sans reine	Total	(%)	Dataset
1	2,687	1,476	4,163	24.07%	NUHIVE
2	656	6,557	7,213	41.71%	NUHIVE
3	0	16	16	0.09%	OSBH
4	3,700	0	3,700	21.39%	OSBH
5	0	802	802	4.64%	OSBH
6	1,401	0	1,401	8.10%	OSBH
Total	8,444	8,851	17,295	100%	

TABLE 4.2 – Description du contenu audio de la base T BON. Le tableau est inspiré de [66]

- **Base BUZZ (BUZZ)** — [86] : La base BUZZ n'est plus disponible publiquement. Elle contient des enregistrements de colonies d'abeilles mélangés avec des sons de criquets et des bruits ambiants issus de l'environnement. Elle a été utilisée pour apprendre à différencier les sons des abeilles d'autres signaux.
- **Bee Audio Dataset (BAD)** — [87] : La base BAD est une collection de dix ruches enregistrées sur une période d'un mois, représentant une grande diversité de ruches. Cependant, ce jeu de données n'est pas labellisé, il ne peut donc être traité qu'avec des méthodes non supervisées. Ainsi, le dataset BAD n'est pas ajouté au dataset BT qui vise l'apprentissage supervisé. Ce jeu de données est accessible à <https://zenodo.org/records/7052981> (Consulté le 18/07/2024). La diversité des ruches de BAD est utilisée dans 4.5.1 comme données supplémentaires pour tester l'extrapolation sur de nouvelles ruches.
- **Bases Privées** : Certains pairs ont développé et travaillé sur leur propre jeu de données. Dans 4.1, un numéro est utilisé pour identifier les mêmes bases qui ont été utilisées plus d'une fois. Si la base n'a été utilisée que dans une seule publication, elle est représentée par un point.

La table 4.3 résume les caractéristiques de tous les ensembles de données sélectionnés pour être fusionnés dans le cadre proposé BT, avec le nombre correspondant de ruches, les étiquettes utilisées et la taille des fichiers audio bruts non compressés.

4.2.2 Labels utilisés pour la classification

Il existe une certaine diversité dans les labels fournis ou non avec chaque dataset. Cependant, tous ces labels n'ont pas été étudiés de manière égale dans la littérature. La table 4.4 résume le nombre d'articles dans lesquels chaque label est utilisée.

Le label le plus utilisé est celui de la présence de la reine dans la ruche, il

TABLE 4.3 – Caractéristiques des datasets publics indiquant le nombre de ruches ("# R"), les labels disponibles, et la quantité de données. Les bases de données BUZZ et privées non publiquement disponibles ont donc été exclues.

Dataset	# R	Labels	Volume
OSBH	4	Reine–SansReine, Essaimage, Varroa	0,8 Go
NUHIVE	2	Reine–SansReine	47,8 Go
TBON	6	Reine–SansReine, Abeille–PasAbeille	3,46 Go
TBON processed	6	Reine–SansReine, Abeille–PasAbeille	3,38 Go
SBCM	4	Reine–SansReine	21,7 Go
BAD	10	Pas de label	1,25 Go

TABLE 4.4 – Nombre d’articles par label étudiée.

Label	Nombre d’articles
Reine–SansReine	13
Essaimage	3
Abeille–Pas d’abeille	4
Son d’abeille, chant de criquet, bruit ambiant	3

est celui avec la plus grande base de données annotée. Il est aussi celui avec le plus de sens du point de vue d’un apiculteur. Bien que la détection d’essaimage soit également pertinente, sa nature sporadique la rend difficile à enregistrer et explique le manque de bases de données annotées sur ce sujet. De plus, un apiculteur cherche à éviter les essaimages donc avoir une alerte à ce sujet arrive trop tard, il faudrait travailler à son anticipation. Quant aux labels de présence d’abeilles ou de comparaison avec les sons de criquets, ils ne sont pas pertinents pour un apiculteur, car l’enregistrement effectué directement à l’intérieur d’une ruche garantit l’écoute exclusive des abeilles.

Les datasets OSBH et Nuhive ont été traitées et labélisées avec Abeille–Pas d’abeille en examinant les spectrogrammes, créant ainsi la base TBON [63]. Cela a offert une grande base de données annotée et a favorisé l’utilisation de l’apprentissage automatique sur cette étiquette spécifique. La séparation Abeille–Pas d’abeille représente essentiellement le rapport signal/bruit des abeilles, en analysant la quantité de bruit et d’interférence (bruit de la circulation, voix humaine, etc.) dans le signal par rapport à la présence d’abeilles.

Le label Abeille–Pas d’abeille n’a pas d’intérêt direct pour la surveillance de la santé, mais peut être utilisée pour sélectionner les échantillons audio non contaminés (bruit ambiant, parasite (voiture, avion, oiseau...)) avant d’apprendre d’autres labels plus pertinent. La section 4.4.2 discute l’intérêt de ce label et montre un faible impact sur la performance de la classification Reine–SansReine, par exemple. Par conséquent, l’étiquette Abeille–Pas d’abeille a été jugée inutile, et l’accent est mis sur le label Reine–SansReine pour tous les

fichiers du dataset BT.

Tous les ensembles de données rassemblés sont étiquetés avec le label de présence de reine ; cependant, la méthode par laquelle les étiquettes ont été établies diffère. Dans les bases de données OSBH, Nuhive et TBON, la source de l'étiquette est opportuniste : l'observation d'une colonie sans reine a conduit à l'enregistrement de la ruche et à la création de données labellisée sans reine. Les mêmes ruches ont ensuite été enregistrées avec une reine, créant des ensembles de données avec des fichiers avec et sans reine.

A l'inverse, la constitution du dataset SBCM suit une approche différente ; pendant quelques semaines, les ruches ont été enregistrées périodiquement. À mi-chemin de l'expérience, les ruches ont été délibérément orphelinées par l'apiculteur, créant artificiellement un état sans reine. Après quelques jours, une nouvelle reine est introduite dans les ruches, permettant la création du dataset incluant à la fois des états sans reine et avec reine.

4.2.3 Extraction de features pour la représentation temps/fréquence des fichiers audio

Peu de travaux utilisent sur le signal audio brut principalement car l'étude des séries temporelles en machine learning (ML) est moins courante et documentée que le traitement d'images.

La représentation temps/fréquence est principalement utilisée car elle conduit à une représentation 2D permettant l'utilisation de techniques de traitement d'images comme les réseaux de neurones convolutifs (CNN). Les principales représentations temps/fréquence sont les suivantes :

- MFCC : Mel-Frequencies Cepstrum Coefficient
- Mel-Spec : Mel-frequencies-Spectrogram
- HHT : Hilbert–Huang Transform
- STFT : Short-Time Fourier Transform

Pour effectuer la classification des enregistrements audio des ruches, il est essentiel de procéder à un pré-traitement des fichiers afin d'en extraire les informations pertinentes tout en réduisant leur taille. Diverses techniques d'extraction de features ont été explorées, parmi lesquelles les coefficients Mel-Frequencies Cepstrum Coefficient (MFCC), le Mel-Spectrogramme (Mel-Spec), la transformation de Hilbert-Huang (HHT) et le Short-Time Fourier Transform (STFT). Chacune de ces méthodes permet de convertir le signal brut en un format bi-dimensionnel temps/fréquence, adapté aux approches classiques de machine learning.

La majorité des études utilisent les MFCC, qui ont fait leurs preuves dans le domaine de la reconnaissance vocale. Cependant, l'application de cette technique aux enregistrements sonores des abeilles peut être discutée, car les MFCC sont conçus pour correspondre à la sensibilité auditive humaine, qui diffère probablement de la perception sonore des abeilles. Par conséquent, il est crucial

d'optimiser les paramètres des MFCC ou des Mel-Spectrogrammes, comme le soulignent [39, 63, 40], et [64], pour obtenir des performances optimales. Les features utilisées dans la littérature sont passées en revue dans 4.5.

TABLE 4.5 – Résumé des features utilisées dans la littérature. "MFCC" : Mel-frequency cepstrum ; "Mel-Spec" : Mel-spectrogram ; "HHT" : Hilbert–Huang transform ; "STFT" : Short-Time Fourier Transform.

Date	Référence	MFCC	Mel-Spec	HHT	STFT	Autres
2008	Ferrari [61]					Analyse fréquence et amplitude, température, humidité
2018	Kulyukin [39]	•	•		•	tonnetz pour les modèles ML
2018	Nolasco [63]	•	•			
2018	Cejrowski [62]					Coefficients LPC, température, humidité
2019	Nolasco [70]	•	•	•		
2019	Robles-Guerrero [40]	•				
2020	Terenzi [64]	•		•	•	Transformée en ondelettes avec et sans normalisation moyenne cepstrale, LPC
2020	Zgank [71]	•				Transformée constante-Q (CQT)
2021	Kim [72]	•	•			
2021	Orlowska [66]		•			
2021	Zgank [73]	•				
2021	Cejrowski [88]	•				ACI
2021	Kulyukin [74]	•	•			Chroma, coefficients de contraste spectral, coefficients tonnetz
2022	Quaderi [75]	•			•	Énergie RMSE, Centroïde spectral, Largeur de bande spectrale, Rolloff spectral
2022	Soares [67]	•				Temps, Fréquence, Chroma, Spectral, Taux de croisement à zéro
2022	Kampelopoulos [41]	•				
2023	Robles-Guerrero [42]	•				
2023	Di [76]	•				Intégration VGGish
2023	Truong [77]	•			•	
2023	Farina [78]	•			•	
2023	Ruvinga [79]	•			•	
2023	Phan [80]	•				

TABLE 4.5 – Résumé des features utilisées dans la littérature (suite)

Date	Référence	MFCC	Mel-Spec	HHT	STFT	Autres
2023	Uthoff [69]	•	•			Coefficients Chroma, coefficients de contraste spectral, coefficients tonnetz
2023	Kanelis [68]	•				Centroïde spectral, Taux de croisement à zéro, Chromagramme, Transformée Q-constante
2023	Rustam [81]	•				
2023	Barbisan [43]	•			•	
2024	Bricout [38]	•	•			

Les MFCC sont prédominantes par rapport à toutes les autres features, dans 21 sur 24 articles. Les Mel-Spec sont choisis dans cette étude comme détaillé dans 4.3.2.

Les familles des arbres de décisions (SVM, Boosts, forests, ...) et des classifications linéaires et régressions polynomiales sont largement représentées. Ayant à disposition un volume de données suffisant, ces techniques de machines learning peuvent être implémentées simplement.

Le deep learning, en particulier les architectures CNN, est principalement utilisé car il s'agit d'une méthode traditionnelle combinée avec les MFCC pour la reconnaissance vocale. Le CNN possède un espace paramétrique plus vaste que les techniques citées précédemment, ce qui le rend sujet au sur-apprentissage. Pour pallier ce problème, une cross-validation doit être implémentée pour confirmer la validité du modèle. De plus le sur-apprentissage peut provenir d'un manque de données. En outre, son implémentation au niveau du nœud est plus complexe que celle des techniques basées sur les arbres de décision.

4.2.4 Modèles de classifications

Différents modèles d'apprentissage sont présentés comme solutions au problème de classification, parmi lesquels les *Neural Network* (NN), les *Convolutional Neural Networks* (CNN), la *Support Vector Machine* (SVM), les *Random Forest* (RF), les *k-nearest neighbors* (KNN) et la *Logistic Regression* (LR).

De manière similaire, le tableau 4.6 récapitule les différentes études issues de la littérature ainsi que les modèles d'apprentissage correspondants.

Cependant, ces solutions offrent une grande précision et permettent de définir rapidement une limite de performance sans nécessiter un ajustement sophistiqué des paramètres de la méthode ni un long temps d'implémentation. Dans 4.3.4, nous présentons le modèle CNN utilisé.

TABLE 4.6 – Résumé des modèles utilisés dans diverses études. "CNN" : *Convolutional Neural Networks*; "SVM" : *Support Vector Machine*; "RF" : *Random Forest*; "KNN" : *k-nearest neighbors*; "LR" : *Logistic Regression*.

Date	Référence	CNN	SVM	RF	KNN	LR	Autres
2008	Ferrari [61]						
2018	Kulyukin [39]	•	•	•	•	•	
2018	Nolasco [63]	•	•				
2018	Cejrowski [62]		•				+C-classification and Gaussian-kernel
2019	Nolasco [70]	•	•				
2019	Robles-Guerrero [40]					•	Singular Value Decomposition (SVD)
2020	Terenzi [64]						
2020	Zgank [71]						HMM et GMM
2021	Kim [72]	•	•	•			XGBoost, VGG-13
2021	Orlowska [66]	•					
2021	Zgank [73]	•					
2021	Cejrowski [88]						
2021	Kulyukin [74]	•	•	•	•	•	
2022	Quaderi [75]	•	•	•			SNN, RNN, Decision Tree, Naïve Bayes
2022	Soares [67]		•	•			
2022	Kampelopoulos [41]						
2023	Robles-Guerrero [42]		•	•	•	•	NN
2023	Barbisan [43]		•				NN
2023	Di [76]		•	•	•		Decision Tree
2023	Ruinga [79]	•				•	LSTM, MLP
2023	Phan [80]		•	•	•	•	Decision Tree, Extra Trees, XGBoost
2023	Uthoff [69]	•	•	•	•	•	
2023	Kanelis [68]						
2023	Rustam [81]			•	•		
2024	Bricout [38]	•					Contrastive Learning

4.2.5 Problème d'extrapolation vers une nouvelle ruche

La majorité des publications existantes ont mis l'accent sur la performance de la classification des enregistrements audio en fonction des labels précédemment définis, adoptant ainsi une approche classique d'application de diverses techniques de machine learning (ML). Cependant, rares sont les articles qui ont abordé la problématique de l'extrapolation d'une ruche à une autre, en particulier vers des ruches non observées lors de l'entraînement du modèle. Il

s'agit d'une question cruciale, car elle détermine si le modèle entraîné obtenu peut être appliqué à la surveillance de nombreuses colonies d'abeilles dans un produit industriel. Seuls Nolasco [63, 70], Terenzi [64, 89] et Orłowska [66] ont étudié la capacité de leurs solutions à extrapoler vers de nouvelles ruches. La revue [3] et l'article [69] mentionnent également la problématique de la non-extrapolation entre ruches distinctes.

Nolasco et Terenzi ont travaillé sur le jeu de données NUHIVE [83], qui se compose uniquement de deux ruches labellisées "Reine" et "SansReine". Dans leurs travaux, ils ont exploré plusieurs features et modèles de classification, en isolant une ruche durant la phase d'apprentissage pour tester l'extrapolation. Bien que les modèles obtenus soient précis pour les ruches utilisées durant l'apprentissage, aucun d'entre eux ne parvient à extrapoler correctement à l'autre ruche du dataset.

Orłowska a mené ses recherches sur la base de données TBON [84], qui combine les jeux de données NUHIVE [83] et OSBH [82], représentant respectivement 65% et 55% du jeu de données TBON. Les résultats observés sont similaires : les modèles ne se généralisent pas aux ruches non observées.

Ce manque de généralisation peut s'expliquer par la faible diversité des ruches mesurées disponibles. Ainsi, bien que les modèles parviennent à différencier les labels pour chaque ruche individuellement, ils échouent lorsque confrontés à une nouvelle ruche non incluse durant la phase d'apprentissage.

L'augmentation de données (DA) peut aider à pallier ce manque de diversité en augmentant artificiellement la variété des données disponibles, améliorant ainsi le potentiel d'extrapolation. Plusieurs techniques d'augmentation sont introduites dans la littérature pour cette application :

- Ajout de bruit : [66] a augmenté artificiellement le nombre d'enregistrements audios dans le jeu de données d'apprentissage de 50% en ajoutant un bruit blanc gaussien aux données existantes, avec une variance ajustée pour obtenir un rapport signal-sur-bruit (SNR) de 30 dB.
- Découpe et décalage temporel des fichiers : [43] a augmenté le nombre d'échantillons d'apprentissage en segmentant les fichiers audios en morceaux plus petits (0,5, 1, 3, 5s) sans chevauchement.
- Décalage en temps et fréquence : [63] a enrichi les données en décalant périodiquement signaux audios dans le temps et fréquemment.

ces méthodes sont indépendantes d'une ruche à l'autre et n'augmentent pas la diversité de ruches dans le dataset. Le décalage en fréquence est déconseillé par Terenzi pour la DA, car la majorité des informations sont véhiculées par les fréquences du signal [89]. Bien que Terenzi suggère que le décalage en fréquence puisse être trompeur, Nolasco a utilisé cette technique. Cependant, aucune étude n'a encore mesuré l'impact réel de cette méthode de DA sur la performance de classification et l'extrapolation entre ruches.

4.3 Méthodologie

La figure 4.2 présente la méthode de travail proposée dans cette étude. Les différents éléments de la méthode sont présentés plus en détails dans cette section

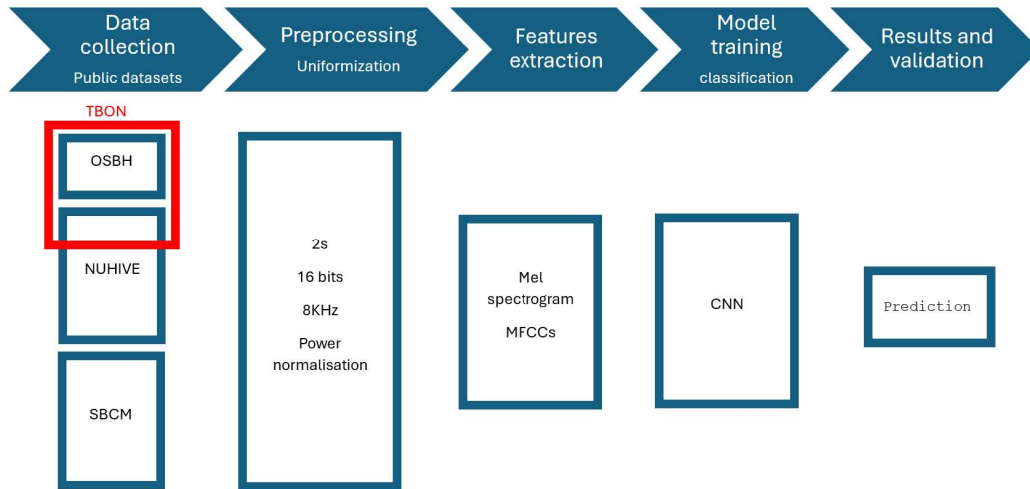


FIGURE 4.2 – Méthode de travail classique pour le machine learning.

4.3.1 Unification des datasets publics : Construction du framework BeeTogether

Tous les datasets audios publics labellisés sont rassemblés et standardisés pour pouvoir être utilisés pour les méthodes de machine learning. Les datasets utilisés sont le projet NUHive [83], OSBH [82], et SBCM [86]. La version du dataset OSBH utilisé est celle prétraitée par TBON. Les datasets nécessitent des étapes de prétraitement afin de permettre leur intégration cohérente dans le framework BT.

4.3.1.1 Standardisation des données brutes

Cette section détaille les étapes nécessaires pour assurer la compatibilité entre les jeux de données, en se concentrant sur la standardisation des échantillons de données :

- **Standardisation des labels** : L'étiquette commune "Reine/Sans-Reine", indiquant la présence de la reine abeille, est choisie comme critère principal d'analyse. Les labels spécifiques "Ancienne Reine", "Nouvelle Reine" et "Présente" du dataset SBCM sont regroupés sous l'étiquette "Reine", tandis que le label "Rejetée" est classée sous "SansReine".

- **Normalisation de la fréquence d'échantillonnage** : La majorité des fichiers audios ont été enregistrés à une fréquence d'échantillonnage de 44,1 kHz, sauf pour le dataset NUHive, enregistré à 32 kHz. Étant donné que les fréquences des sons des abeilles sont principalement comprises entre 20 et 2000 Hz, une fréquence d'échantillonnage de 4 kHz est suffisante selon le critère de Shannon-Nyquist. Cependant, pour préserver la richesse des données permettre une flexibilité future, une fréquence d'échantillonnage uniforme de 8 kHz est choisie pour ré-échantillonner tous les échantillons audios [39, 40, 41, 42, 43]. Le ré-échantillonnage est effectué en utilisant la fonction `resample` de la bibliothèque audio Librosa [90], qui applique un filtre anti-repliement, l'interpolation et la décimation sur le spectre.
- **Uniformisation de la profondeur de bits** : Tous les échantillons sont standardisés à une profondeur de 16 bits conformément aux normes industrielles pour la qualité audio. Bien que le jeu de données SBCM soit initialement en 24 bits, une réduction à 16 bits est effectuée pour garantir la cohérence entre les jeux de données. Des essais préliminaires sur la base SBCM ont révélé une différence de performance mineure entre les deux profondeurs de bits, justifiant ainsi la normalisation.
- **Normalisation** : Il est crucial d'uniformiser les niveaux de puissance entre les enregistrements afin de limiter la confusion des modèles d'apprentissage entre le label cible et la puissance moyenne du signal, forcément différente d'un dataset à l'autre, mais aussi d'une ruche à l'autre. Cette procédure ajuste l'amplitude du signal audio de sorte que sa puissance moyenne soit normalisée à une valeur cible, ici égale à un. La normalisation est appliquée aux données audios brutes avant la standardisation de la durée et l'extraction des features. Les signaux sont normalisés par la formule $\sqrt{N} \cdot \frac{s}{\|s\|_2}$, où N est le nombre d'échantillons et s_k est le k -ième échantillon. Cette approche garantit une puissance moyenne cohérente même avec des durées différentes.
- **Optimisation de la durée des fichiers** : La durée des fichiers est ajustée pour maximiser le nombre de segments utilisables tout en conservant suffisamment de contenu audio pertinent pour l'apprentissage. La littérature indique qu'une durée de 2 secondes est optimale [39, 40, 75, 80, 43]. Les fichiers ne pouvant être divisés en segments de 2 secondes sont tronqués pour garantir l'uniformité et éviter tout chevauchement entre les échantillons.

Le contenu de la nouvelle base de données BT, comprenant le nombre d'échantillons obtenus de chaque source après standardisation et la répartition des labels Reine–SansReine, est présenté dans la 4.7.

TABLE 4.7 – Quantité de données après standardisation, et distribution correspondante des labels.

Dataset	Nombre d'échantillons	Reine	SansReine
SBCM	213,000	184,620 (86.70%)	28,380 (13.30%)
NUHIVE	169,044	84,568 (50.02%)	84,476 (49.98%)
TBON	13,792	7,434 (53.92%)	6,358 (46.08%)
BAD*	40,000	-	-
Total	395,836	276,622 (69.88%)	119,214 (30.12%)

Note : *Base BAD traitée, non incluse dans le total car non labellisée.

4.3.2 Extraction des features

À partir de la base de données BT obtenue, des features sont extraites afin de pouvoir appliquer les techniques de machine learning (ML). Il ressort de 4.2 que la plupart des articles utilisent les MFCC. Le traitement détaillé des MFCC, illustré dans la 4.3 (extrait de [91]), peut être décomposé en deux étapes :

- Le calcul du Mel-Spec (4.3-d) donne une représentation temps/fréquence proche de la perception humaine du placement fréquentiel et de l'amplitude à ces fréquences.
- Les MFCC (4.3-e) visent à éliminer la corrélation sur l'axe en fréquence des coefficients Mel-Spec en utilisant une transformation en cosinus discrète (DCT).

La 4.3.2.1 détaille l'extraction des Mel-Spec en utilisant des paramètres inspirés des études antérieures et montre leur pertinence à travers le spectre moyen sur l'ensemble du jeu de données BT.

La représentation temps/fréquence du Mel-Spec accroît la quantité d'informations du signal en introduisant une forte corrélation entre ses coefficients. Cela pose des difficultés pour les techniques d'apprentissage non profondes, telles que la régression polynomiale, les SVM et les random forests (RF).

Les MFCC sont donc utilisés pour éliminer la redondance des fréquences et obtenir des features non corrélées entre elles de taille réduite, permettant de bonnes performances avec un effort de calcul moindre par rapport aux techniques d'apprentissage profond.

Les modèles d'apprentissage profond, tels que les CNN, offrent de bons résultats indépendamment de la corrélation des caractéristiques. Bien qu'ils puissent utiliser directement les caractéristiques Mel-Spec, la majorité des articles opte pour une caractéristique MFCC commune pour permettre une comparaison avec des algorithmes d'apprentissage non profond.

Cependant, la réduction des features de Mel-Spec à MFCC nécessite un ajustement supplémentaire des paramètres, et les algorithmes non profonds

requièrent également un réglage plus fin des hyper-paramètres que les CNN. Pour éviter des performances médiocres dues à un mauvais ajustement des paramètres et pour se concentrer sur la capacité de généralisation avec le jeu de données BT, l'ensemble complet des informations Mel-Spec est utilisé avec un CNN.

Le choix du CNN et les paramètres Mel-Spec sont validés dans la 4.3.2.1 en appliquant un modèle CNN simple pour classifier le label Reine–SansReine avec différentes valeurs pour la fréquence maximale conservée dans le Mel-Spec.

Les travaux futurs pourront ajouter les MFCC, calculés directement à partir du Mel-Spec, ainsi que d'autres techniques que le CNN pour atteindre des résultats comparables avec des caractéristiques plus compactes et des solutions à faible consommation pouvant être embarquées.

4.3.2.1 Choix des features extraites pour application au modèle

Nous décrivons d'abord comment les MFCC sont calculés à partir du Mel-Spectrogramme. Ensuite, la 4.3.2.1 présente et analyse quatre configurations différentes pour le placement des filtres en fréquence de Mel et conclut avec les paramètres finalement adoptés dans le dataset BT.

Calcul du Mel-Spectrogramme et des MFCC : La gamme de fréquences mélodiques, telle que définie dans [92], échantillonne les fréquences de manière à ce que les hauteurs perçues soient uniformément espacées selon la perception acoustique humaine. Toutefois, choisir cette échelle peut entraîner la perte de certaines fréquences importantes dans la communication sonore des abeilles, moins perceptibles pour l'oreille humaine. Ce choix, ainsi que l'utilisation des MFCC, peut refléter une dépendance à la trajectoire au sein de la communauté scientifique, comme le décrit [93].

Les sorties des filtres triangulaires chevauchants fournissent une moyenne de la puissance au fil du temps pour chaque fréquence mélodique. La perception de l'intensité sonore par l'humain varie selon la fréquence comme le montrent les mesures de Robinson et Dadson [94]. Cette courbe de sensibilité est représentée de manière approximative par une échelle logarithmique en décibels (dB), ce qui permet de générer un Mel-Spectrogramme (Mel-Spec) à échelle logarithmique (voir Figure 4.3d), plus aligné avec la perception humaine de la hauteur et de l'intensité.

Cependant, la représentation temporelle et fréquentielle par le Mel-Spec augmente la quantité d'informations en introduisant une forte corrélation entre les coefficients, ce qui pose des défis pour les techniques d'apprentissage automatique, en particulier les régressions, SVM et les arbres de décision.

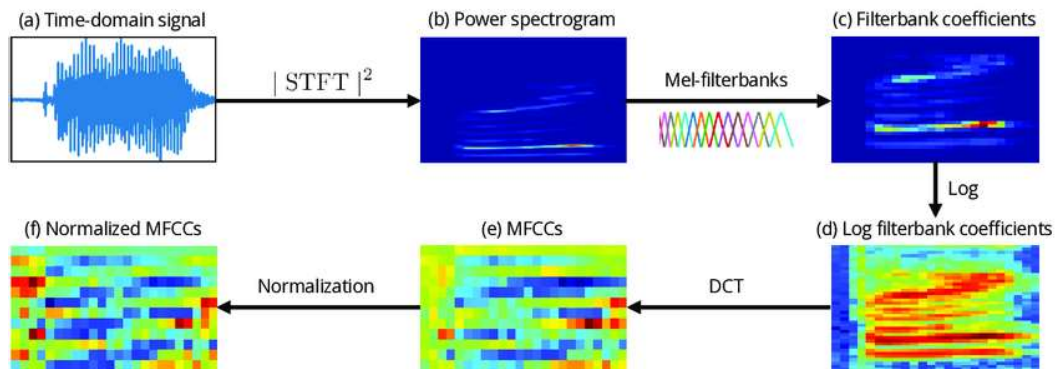


FIGURE 4.3 – (a) Signal temporel normalisé pour la puissance moyenne, la longueur et la profondeur de bits; (b) spectre de puissance perdant l’information de phase; (c) sortie de la banque de filtres mélodiques qui échantillonne les fréquences selon la sensibilité humaine; (d) amplitude logarithmique du Mel-Spec en décibels, proche de la perception humaine; (e) MFCC réduisant la corrélation fréquentielle grâce à la DCT; (f) MFCC normalisés atténuant l’effet de masquage du bruit.

La Transformée en Cosinus Discrète (DCT) est appliquée au Mel-Spec le long de l’axe fréquentiel pour chaque intervalle de temps, permettant d’obtenir les MFCC. Cette transformation réduit la redondance et améliore la performance des modèles de classification. Les MFCC produisent des caractéristiques de taille réduite et non corrélées, nécessaires pour les modèles tels que les SVM ou autres modèles non-profonds, permettant d’obtenir de bonnes performances avec une moindre charge de calcul comparée aux techniques de deep learning.

Étant donné que les MFCC sont dérivés du Mel-Spec avec des paramètres supplémentaires, et que l’utilisation de techniques comme les SVM ou les arbres de décision nécessite un ajustement fin des hyperparamètres, l’approche choisie consiste à utiliser directement les caractéristiques Mel-Spec avec un classificateur CNN. Cette méthode permet d’atteindre des performances similaires et d’évaluer la capacité de généralisation sans subir de baisse de performance liée à un mauvais ajustement des paramètres supplémentaires, bien qu’elle requière davantage de ressources en calcul et en taille de jeu de données.

Le choix du CNN et les paramètres du Mel-Spec sont discutés dans la section suivante.

Paramètres des fréquences mélodiques du Mel-spectrogramme : La transformation Mel-Spec est calculée à l’aide de la bibliothèque Python *librosa* [90] et de la fonction `MelSpectrogram`, comme suit :

```
spec = transforms.MelSpectrogram(sr, n_fft=n_fft,
                                 hop_length=hop_len,
                                 n_mels=n_mels)(sig)
spec = transforms.AmplitudeToDB(top_db=top_db)(spec)
```

Le choix des paramètres s'inspire de travaux antérieurs [39, 63, 40, 64], qui privilégient la précision des mesures plutôt que l'optimisation de la taille des caractéristiques.

— **n_mels = 64**

Le spectrogramme comporte 64 bandes mélodiques réparties de manière linéaire jusqu'à 1 kHz, puis logarithmique pour les fréquences supérieures, selon l'échelle de fréquences mélodiques définie dans [92].

— **n_fft = 1024**

Ce paramètre correspond à la taille de la fenêtre FFT (transformée de Fourier rapide) et à la longueur de la fenêtre temporelle glissante. Chaque segment audio est transformé en 1024 morceaux de fréquences avec une résolution d'environ 8 Hz, la fréquence d'échantillonnage utilisée étant de 8 kHz dans le jeu de données BT. La fenêtre temporelle correspondante est d'environ 125 ms.

— **hop_len = (win_length//2)**

La longueur de saut définit le nombre d'échantillons entre les trames successives et le chevauchement entre les trames du spectrogramme. Ici, 512 échantillons représentent environ 60 ms, ce qui est suffisant pour suivre le vibrato du bourdonnement d'une ruche.

— **top_db = 80**

Ce paramètre fixe le seuil pour le niveau minimum en décibels à afficher dans le spectrogramme. La conversion d'amplitude en décibels vise à normaliser la plage dynamique du spectrogramme, avec une valeur par défaut de 80 dB, ce qui signifie que tout signal en dessous de ce seuil est ramené à 80 dB.

— **fmax = 4000**

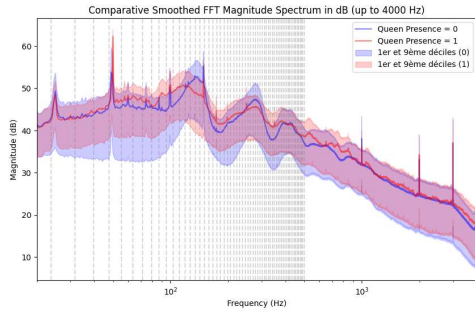
Ce paramètre indique la fréquence maximale (en Hz) à inclure dans le spectrogramme Mel. Le choix parmi 1, 2 et 4 kHz est discuté par la suite.

Une inspection visuelle des spectres des fichiers audio (4.5) révèle que la majorité des signaux significatifs se trouvent sous 500 Hz. Cela remet en question la pertinence des fréquences sélectionnées pour le Mel-Spec afin de représenter fidèlement les signaux nécessaires à la détection de la présence de la reine.

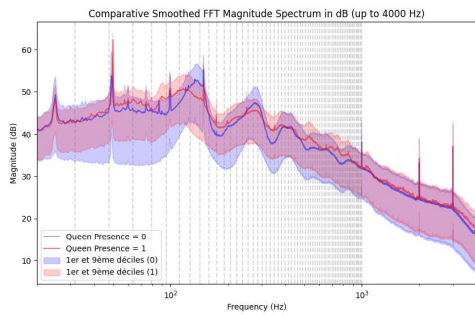
Les expériences ont consisté à conserver le même nombre de fréquences mélodiques tout en réduisant la fréquence maximale représentée. La Figure 4.8 montre le spectre moyen d'une ruche (Numéro 3 de NUHIVE) avec l'échelle Mel-Spec illustrée par des lignes pointillées verticales. La réduction de la fréquence maximale améliore la résolution des signaux à basse fréquence en réduisant l'intervalle entre deux fréquences. À 500 Hz, les pics de basse fréquence dans le spectre coïncident clairement avec les fréquences Mel, ce qui garantit une bonne représentation des informations pertinentes dans le spectrogramme Mel utilisé comme entrée du modèle. Cette optimisation se fait au détriment des fréquences élevées.

4.3. Méthodologie

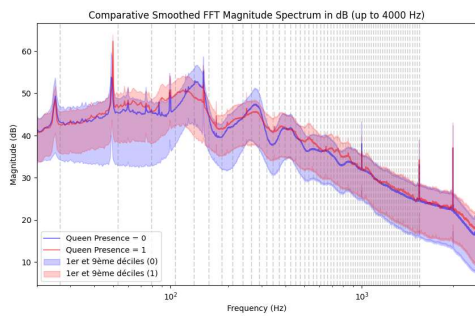
Spectre et échelle de Mel



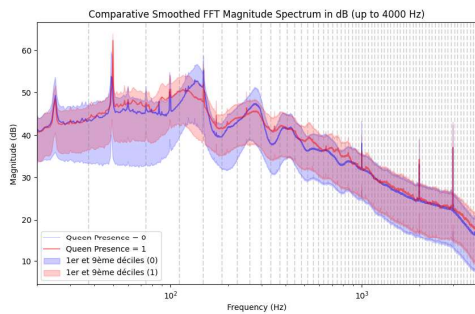
(a) Spectre et échelle de Mel pour Fmax à 500 Hz



(c) Spectre et échelle de Mel pour Fmax à 1 kHz

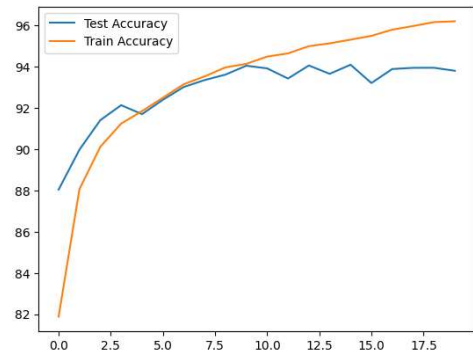


(e) Spectre et échelle de Mel pour Fmax à 2 kHz

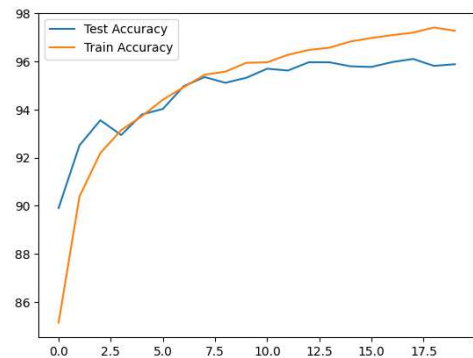


(g) Spectre et échelle de Mel pour Fmax à 4 kHz

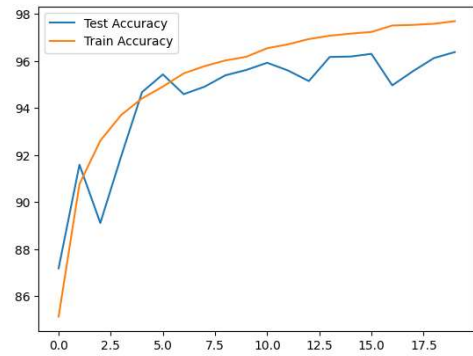
Résultats Mel Spec



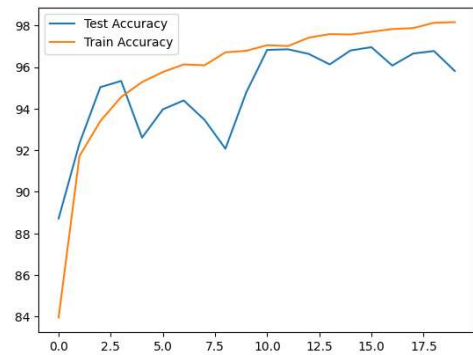
(b) Résultats Mel Spec 500 Hz



(d) Résultats Mel Spec 1000 Hz



(f) Résultats Mel Spec 2000 Hz



(h) Résultats Mel Spec 4000 Hz

TABLE 4.8 – Réglage de la fréquence maximale du Mel-Spec avec, par lignes, des paramètres de fréquence maximale de 0.5, 1, 2 et 4 kHz. À gauche, le spectre moyen sur l'ensemble des enregistrements audios d'une ruche (comme décrit dans la 4.5). L'échelle des fréquences de Mel (lignes en pointillés verticaux) est uniformément espacée sur l'échelle logarithmique après 1 kHz suivant la formule de la fréquence de Mel et linéairement espacée en dessous. À droite, les courbes d'apprentissage du `ClassiC` utilisant la caractéristique Mel-Spec correspondante.

L'inspection visuelle est complétée par un test de performance d'apprentissage avec les mêmes données et le CNN `ClassiC` décrit dans la Section 4.3.4. La Figure 4.8 présente les courbes d'apprentissage pour l'étiquette de la reine, comparées pour différents paramètres de caractéristiques. Bien que la réduction de la fréquence maximale à 500 Hz diminue les performances, rien n'indique la nécessité d'atteindre 4000 Hz. Les modèles à 2 kHz et 4 kHz obtiennent de meilleurs résultats globaux, justifiant une bande passante de Nyquist complète à 4 kHz.

4.3.3 Augmentation de données

Pour améliorer la capacité d'extrapolation des résultats des modèles d'apprentissage entre les ruches, la base de données BT est enrichie en utilisant une combinaison de techniques d'augmentation de données (DA) :

- **Décalage en fréquence** : Un décalage fréquentiel est appliqué aux signaux audios en utilisant plusieurs valeurs de transposition : $[-2, -1, 1, 2]$ demi-tons. Pour des fréquences initiales de 100 Hz et 300 Hz, les variations de fréquence correspondantes sont indiquées dans la 4.9.

TABLE 4.9 – Changements de fréquence résultant de la transposition de demi-tons pour des fréquences initiales de 100 Hz et 300 Hz.

Décalages (demi-tons)	Fréquence (100 Hz)	Fréquence (300 Hz)
-2	89,09 Hz	267,27 Hz
-1	94,39 Hz	283,18 Hz
1	105,95 Hz	317,84 Hz
2	112,25 Hz	336,75 Hz

Les fréquences de 100 Hz et 300 Hz sont choisies car elles correspondent aux plages où se concentre l'information utile dans les sons émis par les abeilles.

- **Ajout de bruit** : Un bruit gaussien est ajouté au Mel-Spec pour simuler des variations entre les enregistrements audios. Le facteur de bruit utilisé

dans ces techniques de DA est typiquement faible, compris entre 0,001 et 0,01. Un facteur de bruit de 0,005 a été utilisé, conformément à [95].

- **Augmentation des Mel-Spec avec masquage en temps et fréquences** : Cette technique renforce la robustesse du modèle face aux variations fréquentielles et temporelles [96].
 - **Masque en fréquence** : Jusqu’à 5% des fréquences du Mel-Spec peuvent être masquées pour chaque masque appliqué.
 - **Masquage temporel** : Un maximum de 15% du nombre total d’étapes temporelles peut être masqué pour chaque masque de fréquence et de temps.
- **Perturbation d’amplitude** : Des perturbations aléatoires sont appliquées à l’amplitude du Mel-Spec pour simuler des variations dans la force du signal. Les facteurs de perturbation varient généralement entre 0,05 et 0,2. Un facteur de perturbation de 0,1 a été sélectionné, ce qui permet des variations d’amplitude de $\pm 10\%$ [96].

Pour chaque échantillon audio, trois augmentations sont générées en appliquant successivement les méthodes présentées. Chaque fichier original est donc modifié et augmenté quatre fois.

La 4.10 présente le nombre d’échantillons résultants et la distribution des étiquettes Reine–SansReine après augmentation de données, sans décalage en fréquence.

TABLE 4.10 – Quantité de données après augmentation de la base de données BT sans transposition de tonalité, et distribution correspondante des labels.

Datasets	Total	Reine	SansReine
SBCM	852,000	738,480 (86.70%)	113,520 (13.30%)
NUHIVE	676,176	338,272 (50.02%)	337,904 (49.98%)
TBON	55,168	29,736 (53.92%)	25,432 (46.08%)
Total	1,583,344	1,106,488 (69.88%)	476,856 (30.12%)

La pertinence du décalage fréquentiel a été discutée par Terenzi [89], qui souligne que la majorité des informations utiles pour différencier le label Reine/SansReine se trouvent dans la fréquence des signaux. Pour tester spécifiquement cette technique d’augmentation de données, les signaux bruts transposés sont considérés comme de nouvelles ruches virtuelles. Quatre nouveaux fichiers sont ainsi générés, chacun correspondant à un décalage en fréquence différent.

La 4.11 présente le détail du dataset résultant après augmentation avec décalage en fréquence.

4.3.4 Modèle d’apprentissage issu de l’état de l’art

Avec la base de données BT construite et labellisée avec l’étiquette Reine–SansReine, différents modèles d’apprentissage peuvent être introduits pour ef-

TABLE 4.11 – Quantité de données après augmentation avec décalage en fréquence du dataset BT, et distribution correspondante des labels.

Dataset	Total	Reine	SansReine
SBCM	1,065,000	923,100 (86.70%)	141,900 (13.30%)
NUHIVE	845,220	422,840 (50.02%)	422,380 (49.98%)
TBON	68,960	37,170 (53.92%)	31,790 (46.08%)
Total	1,979,180	1,383,110 (69.88%)	596,070 (30.12%)

fectuer les tâches de classification. Cette section présente le modèle mis en œuvre, en se concentrant sur des approches capables de se généraliser à des ruches non observées lors de la phase d'entraînement.

En s'inspirant de la littérature, un classificateur basé sur un réseau de neurones convolutif (CNN) est sélectionné en raison de son efficacité pour identifier et catégoriser le label Reine–SansReine. Il est démontré au paragraphe 4.3.6.1 que ce modèle est comparable aux modèles les plus performants trouvés dans la littérature, même lorsqu'il est appliqué à l'ensemble de données hétérogènes BT.

L'architecture choisie pour le modèle CNN, décrite dans la 4.12, est composée de six couches de convolution et d'une couche d'adaptive average pooling. Chaque couche de convolution utilise la fonction d'activation ReLU et est suivie d'une étape de batch normalisation. Une couche linéaire de sortie, avec un neurone par label, termine le réseau pour produire le résultat de la classification.

Cette approche, appelée `classique`, consiste à utiliser le modèle CNN pour prédire l'étiquette Reine–SansReine en se basant sur les caractéristiques du Mel-Spec.

TABLE 4.12 – Architecture du modèle de classification audio. "Conv2d" : couche de convolution pour des entrée 2D ; "AdapAvgPool2d" : couche d'adaptive average pooling pour des entrées 2D. Les couches Conv2d utilisent ReLU et sont suivies d'une étape de batch normalisation.

Type	Taille du kernel	Stride	Padding	Canaux de Sortie
Conv2d	5×5	2×2	2×2	8
Conv2d	3×3	2×2	1×1	16
Conv2d	3×3	2×2	1×1	32
Conv2d	3×3	2×2	1×1	64
Conv2d	3×3	2×2	1×1	128
Conv2d	3×3	2×2	1×1	256
AdapAvgPool2d	-	-	-	1
Linéaire	-	-	-	2

4.3.5 Nouvelle méthode de classification inspirée du contrastive learning

Étant donné que [63] et [89] ont montré qu’il n’y a pas d’extrapolation entre les ruches lorsqu’un CNN est utilisé, une nouvelle approche de classification est proposée, inspirée du contrastive learning. Cette approche vise spécifiquement à apprendre les différences entre les échantillons de différents labels, plutôt que de se focaliser sur les caractéristiques spécifiques de ces classes distinctes (ce qui pourrait potentiellement conduire à un surapprentissage et à une faible capacité d’extrapolation).

Ces méthodes permettent une compréhension plus nuancée des données propres à chaque ruche en se concentrant sur les similitudes et les différences entre deux échantillons audios d’une même ruche, plutôt que de supposer qu’un classificateur puisse se généraliser à toutes les ruches. Cette approche garantit que les conditions et variations uniques de chaque ruche sont correctement prises en compte dans le processus d’évaluation, conduisant à des résultats plus précis et significatifs.

En pratique, la même architecture CNN que celle définie précédemment est conservée, mais les données d’entrée sont modifiées. Des paires d’échantillons de données sont générées à partir de fichiers audios provenant d’une même ruche, chaque échantillon étant utilisé une fois dans une paire tout en maintenant un nombre équilibré de chaque étiquette de **pairwise** comme défini dans le 4.13. Une nouvelle étiquette est associée à chaque paire d’échantillons. Deux ensembles d’étiquettes possibles sont appliqués, selon que l’on prédit chacune des étiquettes originales Reine-SansReine, ou si les échantillons appariés partagent la même étiquette d’origine.

TABLE 4.13 – Création des paires et les labels associés.

	SR/SR	SR/R	R/SR	R/R
Labels pairwise	0	1	2	3
Labels pairwiseXNOR	0	1	1	0

Note : R : Reine, SR : SansReine

Ainsi, deux approches de classification distinctes sont obtenues en fonction des labels utilisées :

- **pairwise** : prédiction de chaque label de la paire,
- **pairwiseXNOR** : prédiction de l’identité des deux labels de la paire.

Dans une approche classique de contrastive learning, telle que l’utilisation de réseaux de neurones siamois [97], l’objectif principal est d’apprendre un espace de représentations où la distance entre les paires similaires est minimisée, et celle entre les paires dissemblables est maximisée. Cela différencie explicitement les classes en augmentant le contraste entre elles. En revanche, l’approche

proposée considère implicitement le contraste entre les classes en apprenant à reconnaître quand les paires d'échantillons ont des labels distincts. Ainsi, bien que cette méthode soit inspirée du *contrastive learning*, elle traite le problème comme une de classification simple avec des labels modifiées.

4.3.6 Méthode de validation

Deux méthodes de validation sont utilisées en fonction de la question à laquelle répondre.

4.3.6.1 Validation classique avec ensembles d'entraînement et de test :

La première question consiste à déterminer si le modèle CNN sélectionné atteint de bonnes performances (c'est-à-dire comparables à celles de la littérature) sur le jeu de données BT. Pour ce faire, un processus de validation par ensembles d'entraînement et de test est employé, avec une répartition aléatoire de 80% des données pour l'entraînement et 20% pour le test. Cette méthode de validation est utilisée au paragraphe 4.3.6.1 pour les trois méthodes de classification.

4.3.6.2 Validation par extrapolation de type *k-hive-fold* :

La seconde question, qui est également la principale interrogation de ce travail, est de savoir si un modèle peut être généralisé (ou extrapolé) à des ruches différentes. Pour évaluer les modèles dans les Sections 4.4.3 et 4.5, une variante de la validation croisée classique *k-fold* est utilisée, où les ensembles de train et tests ne sont pas définis aléatoirement mais correspondent aux ruches. Les étapes suivantes résument ce processus :

1. Ajout d'une colonne au dataset, nommée *groupID*, pour identifier, pour chaque fichier audio, le numéro de la ruche et son dataset d'origine.
2. Pour chaque fold une ruche est exclue du jeu de données, et le label de présence de la reine est appris à partir des données des autres ruches. La ruche exclue est retirée de la phase d'apprentissage afin qu'elle ne soit jamais vue par le modèle.
3. Une fois le modèle entraîné, il est testé sur les données de la ruche exclue afin d'évaluer si les résultats du modèle s'extrapolent sur cette ruche.
4. Ce processus est répété pour l'ensemble des ruches du dataset BT.

4.3.7 Mesures de performance du modèle

Pour évaluer les modèles, la précision (accuracy : nombre d'étiquettes correctement prédites) et la matrice de confusion sont utilisées.

La mesure de précision permet de valider la performance globale du modèle, fournissant une évaluation générale de sa capacité à prédire correctement les résultats. Dans le cas d'une classification binaire, la formule de précision est la suivante :

$$\text{Précision} = \frac{VP + VN}{VP + VN + FP + FN}$$

où VP est le nombre de vrais positifs, VN est le nombre de vrais négatifs, FP est le nombre de faux positifs, et FN est le nombre de faux négatifs.

La matrice de confusion est particulièrement utile dans ce contexte, car le jeu de données est déséquilibré. Elle permet de visualiser et de quantifier le nombre de faux positifs et de faux négatifs, offrant ainsi une compréhension claire de leur impact sur la performance du modèle. La matrice de confusion est présentée comme suit :

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Cette matrice est exprimée en pourcentages afin de faciliter une compréhension rapide et intuitive des performances du modèle pour les différentes classes.

4.4 Reproduction des performances de l'état de l'art et mise en lumière du problème d'extrapolation

4.4.1 Validation de l'approche CNN + Mel-Spec

Dans cette section, l'approche de validation du paragraphe 4.3.6.1 est appliquée, en utilisant 80% de la base de données BT comme ensemble d'entraînement et 20% comme ensemble de test. Les données originales de la base de données BT, sans augmentation de données, sont utilisées, comme décrit dans le 4.7. L'entraînement du CNN est effectué sur 100 époques afin d'évaluer un éventuel surapprentissage. Les résultats ne montrent aucun signe de surapprentissage, et la convergence est atteinte en 20 époques.

Le tableau 4.14 présente la précision et la matrice de confusion obtenues sur l'ensemble de test pour chacune des trois méthodes de classification introduites dans le 4.3.4, à savoir les classificateurs `classical`, `pairwise`, et `pairwiseXNOR`.

Pour rappel, ces classificateurs reposent tous sur le même modèle CNN. L'approche `classical` prédit directement les labels cibles 0=Sans-Reine et 1=Reine sur un échantillon unique. Les approches `pairwise` et `pairwiseXNOR` sont appliquées à des paires issues de deux échantillons audios provenant de la même ruche. Ces paires sont formées par concaténation des deux Mel-Spectrogrammes de chaque fichier, chaque fichier

TABLE 4.14 – Précision et matrice de confusion pour les modèles `classical`, `pairwise`, and `pairwiseXNOR` décrits en 4.3.4. La matrice de confusion est normalisée par le nombre d'éléments dans chaque classe du label.

classification	Accuracy (%)	Confusion matrix (%)			
classical	99.2	98.2	1.8		
		0.4	99.6		
pairwise	98.5	98.9	0.0	0.0	1.1
		0.2	99.5	0.3	0.0
		0.3	0.6	99.1	0.1
		3.6	0.0	0.0	96.4
pairwiseXNOR	99.9	100.0	0.0		
		0.1	99.9		

étant utilisé une seule fois dans le processus de création des paires. Les classificateurs `PairC` et `XNORC` prédisent respectivement les labels correspondants de `pairwise` (c'est-à-dire 0=SansReine-SansReine; 1=SansReine-Reine; 2=Reine-SansReine; 3=Reine-Reine) et de `pairwiseXNOR` (0=identique/1=différent), comme décrit dans le 4.13.

Pour chaque approche, les performances sont excellentes en termes de précision lorsque l'apprentissage est réalisé sur l'ensemble de la base de données sans distinctions sur l'origine des données, avec respectivement plus de 98%, 96%, et 99,9% d'échantillons correctement classifiés pour chaque étiquette séparée. Cette comparaison valide les choix de normalisation des données et des méthodes de classification, car les résultats sont en étroite correspondance avec ceux précédemment établis dans la littérature.

En raison du déséquilibre des données entre les labels Reine et SansReine, la matrice de confusion montre un taux d'erreur plus élevé de 98,2% pour la détection du label SansReine, comparé à 99,6% pour le label Reine, lorsque l'approche `classical` est utilisée. Globalement, les approches `classical` et `pairwise` apparaissent très comparables.

Pour l'approche `pairwise`, il est observé que la précision est plus élevée avec 99,5% et 99,1% lorsque les labels sont différents, comparativement à 98,9% et 96,4% pour des labels identiques. Ces résultats suggèrent que détecter un changement d'état entre deux audios est une tâche plus facile que de prédire les étiquettes réelles. Ce constat est confirmé par les résultats de précision quasi-parfaite obtenus avec l'approche `pairwiseXNOR`, indiquant une capacité de comparaison optimale de l'état de la ruche entre deux cadres temporels distincts.

4.4.2 Le label Abeille–PasAbeille est-il utile ?

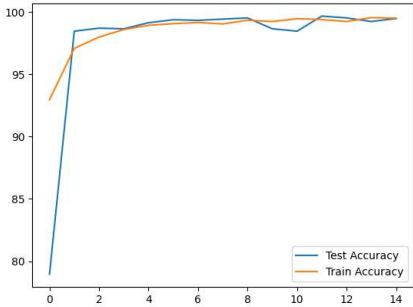
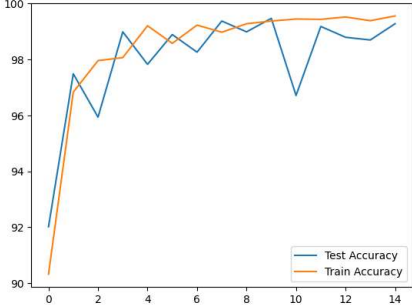
Le label Abeille–PasAbeille est l'un des premiers à avoir été introduit dans les datasets audios des ruches. Il n'a pas d'intérêt direct pour le suivi de la santé des colonies, sauf pour indiquer si une ruche est morte ou vivante. Malgré son nom, cette étiquette sert principalement à détecter si les échantillons audios sont contaminés par du bruit ambiant ou des interférences externes.

Cette étiquette pourrait être intégrée dans un schéma de surveillance en deux étapes :

- Utiliser l'étiquette Abeille–PasAbeille pour exclure les échantillons audios perturbés et ne conserver que ceux contenant le bourdonnement des abeilles avec un faible niveau de bruit ambiant.
- Appliquer la classification de l'étiquette d'intérêt sur les échantillons non contaminés sélectionnés.

La colonne "Abeille" de la 4.15 présente les performances de l'apprentissage de l'étiquette Reine–SansReine lorsque la base de données a été nettoyée de tous les fichiers étiquetés "PasAbeille", comme si la première étape avait été appliquée de manière idéale. Ces résultats sont comparés avec ceux de la colonne "PasAbeille" de la 4.15, qui représente un apprentissage direct en une seule étape de l'étiquette Reine–SansReine sur un ensemble de données partiellement contaminé par des fichiers "PasAbeille". Ces résultats sont obtenus en utilisant les paramètres d'apprentissage automatique et la méthode de travail décrits dans la 4.3.

TABLE 4.15 – Constitution de l'ensemble de données, matrice de confusion de la phase de test et courbe d'apprentissage train/test du label Reine/SansReine. La colonne Abeille correspond à un ensemble de données propres, la colonne PasAbeille correspond à un ensemble de données à moitié contaminé.

Abeille	PasAbeille
Tous les fichiers étiquetés Abeille avec un équilibre Reine (5473) / SansReine (4873)	Fichiers équilibrés Abeille (50%) et PasAbeille (50%) avec un équilibre Reine (1961) / SansReine (1497)
$\begin{pmatrix} 99.39 & 0.61 \\ 0.83 & 99.17 \end{pmatrix}$	$\begin{pmatrix} 99.48 & 0.52 \\ 0.54 & 99.46 \end{pmatrix}$
	

L'analyse de la matrice de confusion révèle que l'étiquette de présence d'abeilles n'est pas pertinente pour l'apprentissage de la présence de la reine. Dans l'implémentation de [63], cette étiquette sert essentiellement d'indicateur de bruit, précisant si un fichier est exempt de bruit. Toutefois, ce bruit n'obscurcit pas nécessairement l'information utile contenue dans le fichier, ce qui permet d'utiliser les fichiers étiquetés "PasAbeille" durant la phase d'apprentissage. En réalité, l'inclusion de ces fichiers peut même agir comme un processus d'augmentation de données, similaire au masquage temporel ou fréquentiel, et ainsi renforcer le modèle. Il existe cependant une différence notable en termes de vitesse et de stabilité de l'apprentissage : l'étiquette "Abeille" accélère l'apprentissage, tandis que l'inclusion des fichiers "PasAbeille" le prolonge.

Pour l'étude en cours, il a été décidé d'ignorer l'étiquette Abeille–PasAbeille et d'utiliser tous les fichiers sans distinction. Il est proposé de réinterpréter cette étiquette comme une indication de la présence d'abeilles, en se concentrant sur la question de savoir si le bruit dans le fichier audio obscurcit ou non toutes les informations. Étant donné que tous les ensembles de données utilisés dans cette étude ont été enregistrés à l'intérieur des ruches, la probabilité que le bruit obscurcisse complètement l'information est minimale.

4.4.3 Sur la difficulté d'extrapoler la classification de la présence de la reine à des ruches non vues

Cette section examine le potentiel d'extrapolation entre ruches pour la classification de la présence de la reine à l'aide des approches `classical` et `pairwise`. Pour ce faire, l'approche de validation `k-hive-fold` du paragraphe 4.3.6.2 est appliquée, c'est-à-dire une validation `k-fold` où les folds correspondent aux ruches distinctes dans le dataset BT. L'entraînement du CNN est réalisé sur 20 époques, ce qui est suffisant pour obtenir une bonne convergence.

4.4.3.1 Non-extrapolation des modèles `classical` et `pairwise`

Non extrapolation sans augmentation des données : Les approches `classical` et `pairwise` sont appliquées aux données originales, sans augmentation de données (DA), comme indiqué dans la 4.7. Le 4.16 présente la précision et la matrice de confusion obtenues lorsque chaque ruche est successivement considérée comme ensemble de test (exclue de l'ensemble d'entraînement).

4.4. Reproduction des performances de l'état de l'art et mise en lumière du problème d'extrapolation

TABLE 4.16 – Précision et matrice de confusion obtenues pour chaque ruche lors de l'application de la validation k-hive-fold sur la base de données BT sans DA. "Ruche" : identifiant de la ruche utilisée comme ensemble de test ; "(% BT)" : pourcentage de la base de données complète BT correspondant à la ruche actuelle ; "Acc" : Précision pour l'ensemble du jeu de données ; "Mat. Conf." : matrice de confusion normalisée par le nombre d'éléments réels dans chaque classe.

Ruche	% BT	classical		pairwise					
		Préc. (%)	Mat. Conf. (%)	Préc. (%)	Mat. Conf. (%)				
1_NUHIVE	22,94	54,2	49	51	59,1	52	0	0	48
			41	60	3	61	34	2	
			2	37	58	2			
3_NUHIVE	21,42	84,3	78	22	59,4	35	0	0	65
			9	91	1	57	23	19	
			2	20	60	18			
1_SBCM	21,82	80,2	4	96	41,1	6	0	0	94
			10	90	0	22	53	25	
			0	34	43	23			
3_SBCM	8,73	67,6	42	59	43,1	19	0	0	81
			28	72	3	41	23	34	
			2	38	29	32			
4_SBCM	19,59	34,1	83	17	46,2	16	0	0	84
			73	27	76	0	0	24	
			13	44	40	3			
5_SBCM	4,47	75,4	7	93	44,1	14	49	35	2
			1	99	71	0	0	29	
			0	24	48	27			
3_TBON	0,002	0,0	0	100	0,0	0	100		
			NA	NA	NA	NA			
			NA	NA					
4_TBON	0,51	77,4	NA	NA	61,8	NA	NA		
			23	77	38	62			
			NA	NA					
5_TBON	0,1	26,7	27	73	40,4	40	60		
			NA	NA	NA	NA			
			NA	NA					
6_TBON	0,41	72,5	NA	NA	83,5	NA	NA		
			27	73	16	84			
			NA	NA					

Les résultats obtenus pour les deux approches de classification sur toutes les ruches ne donnent pas de bonnes précisions. En fin de compte, les approches

`classical` et `pairwise` présentent des comportements similaires en ne parvenant pas à extrapoler les prédictions des étiquettes Reine ou SansReine. Les prédictions sont soit aléatoires (précisions proches de 50%), soit constantes pour l'un des deux labels sur l'ensemble du jeu de données. Ces dernières peuvent produire des précisions extrêmes pour les ruches TBON 3 à 6, où une seule étiquette est présente dans les ensembles de données correspondants (bon pour les ruches 4 et 6, mauvais pour les ruches 3 et 5).

Cependant, l'examen de la structure de la matrice de confusion pour le `PairC` montre que les zéros sur la première et la dernière ligne indiquent que les étiquettes 1 et 2 ne sont pas confondues avec les étiquettes 0 et 3, suggérant une capacité à détecter un changement d'étiquettes dans les paires. Cela motive l'utilisation de l'approche `pairwiseXNOR`.

Non extrapolation avec augmentation de données :

TABLE 4.17 – Précision et matrice de confusion obtenues pour la base de données BT enrichie avec DA sans et avec décalage fréquentiel. Notations similaires à celles de la 4.16.

Ruche	% BT	Sans DA			DA - sans décalage fréquentiel			DA - décalage fréquentiel		
		Pr. (%)	Mat.	Conf. (%)	Pr. (%)	Mat.	Conf. (%)	Pr. (%)	Mat.	Conf. (%)
1_NUHIVE	22,94	54,2	49 41	51 60	45,4	61 70	39 30	56,6	39 26	61 74
3_NUHIVE	21,42	84,3	78 9	22 91	56,1	35 23	65 77	63,3	43 16	57 84
1_SBCM	21,82	80,2	4 10	96 90	47,7	8 12	92 88	49,6	7 8	93 92
3_SBCM	8,73	67,6	42 28	59 72	50,6	18 17	82 83	49,5	10 12	90 88
4_SBCM	19,59	34,1	83 73	17 27	52,8	30 25	70 75	55,4	49 38	51 62
5_SBCM	4,47	75,4	7 1	93 99	50,0	12 12	88 88	54,3	7 4	93 96
3_TBON	0,002	0,0	0 -	100 -	7,14	7 -	93 -	33,3	33 -	67 -
4_TBON	0,51	77,4	- 23	- 77	69,1	- 31	- 69	86,4	- 14	- 86
5_TBON	0,10	26,7	27 -	73 -	39,9	40 -	60 -	33,1	33 -	67 -
6_TBON	0,41	72,5	- 27	- 73	92,9	- 7	- 93	96,1	- 4	- 96

Pour améliorer la capacité d'extrapolation, l'augmentation de données (DA) est envisagée. L'attention est portée sur le classificateur `classical`, étant

donné que les deux approches précédentes ont des comportements similaires en termes d'extrapolation pour la prédiction des étiquettes Reine–SansReine. La DA avec et sans décalage en fréquence est distinguée, comme détaillé dans le 4.3.3. Le contenu des ensembles de données correspondants est donné dans les tableaux 4.10 et 4.11.

La précision et la matrice de confusion obtenues pour chaque ruche avec les ensembles de données augmentés sont présentées dans le 4.17.

Les résultats restent médiocres avec l'utilisation de la DA, que ce soit avec ou sans décalage fréquentiel, qui ne modifie pas les résultats, bien que l'information pour la prédiction de la présence de la reine soit supposée se trouver dans la fréquence du signal [89].

L'utilisation d'un nombre limité de ruches différentes (seulement dix) dans le processus d'apprentissage pourrait entraîner un surapprentissage du CNN sur les ruches connues, ce qui explique ces dysfonctionnements.

L'évolution de la précision sur les ensembles d'entraînement et de test pour le k-hive-fold excluant les données 1_NUHIVE est illustrée dans la 4.4.

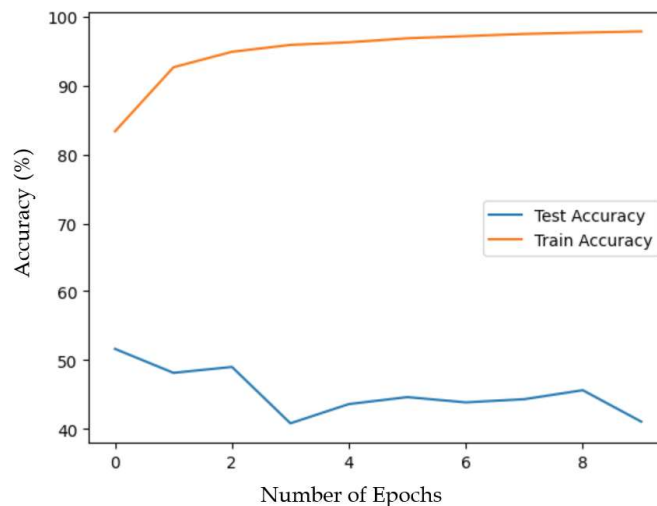


FIGURE 4.4 – Évolution de la précision sur les ensembles d'entraînement et de test pour le k-hive-fold excluant les données 1_NUHIVE.

Il est observé que la précision sur l'ensemble de tests ne dépasse jamais 50%, avec des résultats similaires pour toutes les ruches. Cela rend peu probable que le surapprentissage soit la cause principale de la faible capacité de généralisation.

D'autres tests visant à éviter le surapprentissage, incluant la réduction des caractéristiques par la sélection de moins de fréquences dans le Mel-Spec et la simplification de la structure du modèle CNN, n'ont pas conduit à des conclusions qualitativement différentes.

Les approches `classical` et `pairwise` ne parviennent pas à extrapoler la classification du label Reine et SansReine à des ruches non observées pendant

la phase d'apprentissage. Ce résultat est conforme aux quelques observations de non-extrapolation déjà publiées dans la littérature [89, 69].

En particulier, l'approche `pairwise` ne réussit pas l'extrapolation entre ruches. Bien que cette méthode tente toujours de prédire des étiquettes exactes à partir des signaux audios, même lorsqu'elle utilise des échantillons appariés, elle se rapproche du comportement du classificateur `classical`. Toutefois, l'analyse de la matrice de confusion de `pairwise` suggère qu'un changement d'étiquette dans une paire peut être détecté avec efficacité.

4.4.3.2 La classification peut être axée sur de mauvais paramètres

Cette section vise à proposer des explications au problème d'extrapolation. Pour cela, deux aspects des données sont explorés : les fréquences et les similitudes des signaux audios en fonction de la ruche et des labels.

Analyse des fréquences des fichiers audios : En observant le comportement de la colonie, un apiculteur est capable de détecter la présence de problèmes liés à la reine. L'activité d'une colonie d'abeilles est différent lorsqu'elle est orpheline. Étant donné que [89] a souligné que la majorité de l'information utile se trouve dans la fréquence des signaux, l'impact de la présence des abeilles est étudié dans le domaine des fréquences.

Des inspections manuelles des Mel-Spec pour divers fichiers audios semblent indiquer une différence dans les basses fréquences selon la présence ou l'absence de la reine, en particulier un décalage de fréquences des pics des signaux.

Pour analyser ce phénomène, la Transformée de Fourier Rapide (FFT) est calculée pour chaque signal afin de les convertir du domaine temporel au domaine fréquentiel, facilitant ainsi l'analyse des composants fréquentiels de chaque signal. La 4.5 présente les spectres de fréquence moyens des enregistrements audios pour chaque ruche et chaque catégorie d'étiquettes, permettant d'observer les différences de fréquences de pic et de réponses en amplitude.

La réponse en fréquence semble varier davantage en fonction du label "présence de la reine" dans les datasets NUHIVE et TBON que dans l'ensemble de données SBCM.

Dans le cas de l'ensemble de données NUHIVE, ces différences peuvent s'expliquer par le fait que les deux ruches ont été échantillonnées sur des périodes d'activité saisonnière significativement différentes. Par conséquent, les variations de la population de la colonie, de l'emplacement de la ruche et des ressources disponibles (par exemple, miel, pollen) ont probablement aussi influencé la réponse en fréquence.

L'ensemble de données TBON représente environ 1% des données totales, ce qui explique la forte variance de la courbe moyenne par rapport aux deux autres ensembles de données, qui sont 40 fois plus importants. De plus, les ruches avec des reines étaient différentes de celles sans reines ; le spectre des deux ruches avec présence de reines est montré sur la figure. Les grandes différences

4.4. Reproduction des performances de l'état de l'art et mise en lumière du problème d'extrapolation

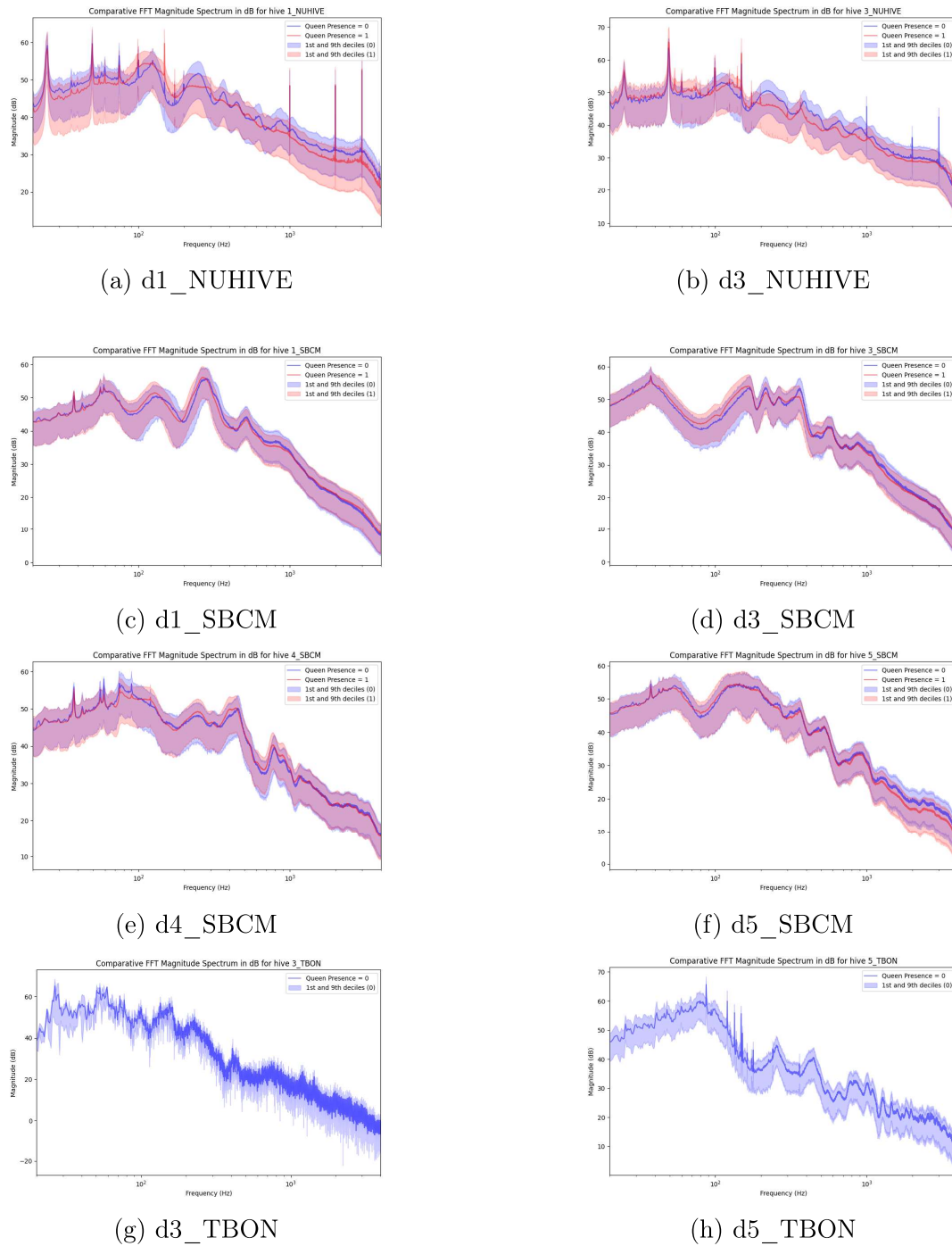


FIGURE 4.5 – Spectre de magnitude FFT comparatif en dB des signaux normalisés provenant de différentes ruches. Les courbes bleue et rouge représentent respectivement les enregistrements audio étiquetés comme PasReine et Reine. Les lignes correspondent à la valeur moyenne, et la zone englobe les données entre 8 déciles : entre 10% et 90%.

observées dans le spectre ne sont donc pas dues aux différences de label, mais peuvent s'expliquer par des variations spectrales d'une ruche à l'autre ou d'une saison à l'autre.

En revanche, l'ensemble de données SBCM a suivi quatre ruches au cours de la même période de l'année (été 2022) avec un échantillonnage consécutif. Cette cohérence dans les paramètres et les changements minimes dans les conditions des ruches ont conduit à une réponse en fréquence similaire dans tous les échantillons.

Ces observations suggèrent que la fréquence peut varier davantage en fonction de l'activité, de la saison et de la ruche, plutôt que de la présence de la reine.

Apprentissage non supervisé pour analyser les similarités entre les signaux audios : Pour explorer plus en profondeur les composants des ensembles de données, une méthode d'apprentissage automatique non supervisée, *t-Distributed Stochastic Neighbor Embedding* (t-SNE), a été utilisée. Cette méthode est une technique de réduction de dimension permettant de visualiser des données de haute dimension en les projetant dans un espace de dimension inférieure, généralement de deux ou trois dimensions. Le t-SNE fonctionne en convertissant les distances euclidiennes de haute dimension entre les points de données en probabilités conditionnelles qui représentent les similarités, et il cherche à minimiser la divergence entre ces similarités dans les espaces de haute et de basse dimension. Cette approche préserve efficacement les structures locales, permettant des visualisations significatives des ensembles de données. Le t-SNE a été appliqué directement sur les Mel-Spec dans le domaine fréquentiel. Un travail similaire utilisant cette méthode a été présenté par [62] sur des données audios d'abeilles avec les mêmes labels.

Pour des représentations simplifiées, l'analyse se concentre sur les bases de données NUHIVE et SBCM, qui représentent la grande majorité des données dans l'ensemble BT. Le t-SNE est ensuite utilisé pour visualiser la similarité des données provenant de différentes bases de données, puis de différentes ruches au sein de ces bases de données.

La 4.6 montre le contenu du dataset en 2D obtenu avec les t-SNE. Les données sont colorées en jaune et en bleu respectivement pour les ensembles de données NUHIVE et SBCM. Une séparation claire est observée basée sur le dataset d'origine des fichiers audios.

Ensuite, chaque dataset est séparé dans la 4.7 pour montrer la représentation 2D obtenue avec le t-SNE pour chaque base de données, avec une coloration :

- en fonction du numéro de la ruche,
- en fonction du label Reine–SansReine.

Avec la méthode t-SNE, il n'est pas possible de séparer efficacement l'étiquette de présence de la reine. Cependant, une séparation claire est observée

4.4. Reproduction des performances de l'état de l'art et mise en lumière du problème d'extrapolation

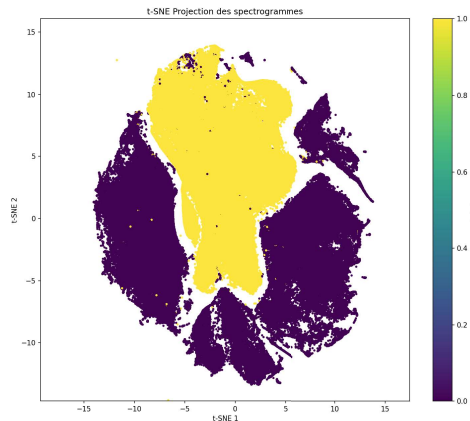


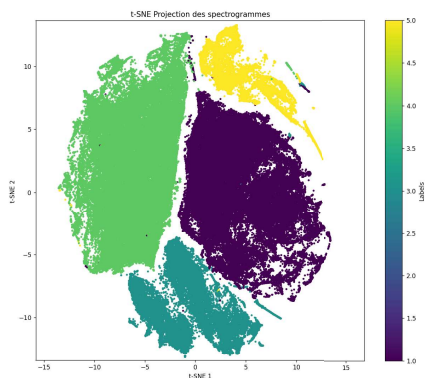
FIGURE 4.6 – Visualisation des ensembles de données NUHIVE (jaune) et SBCM (bleu) en 2D à l'aide de la réduction de dimension t-SNE.

entre les différentes ruches. Cela suggère qu'il est relativement facile d'apprendre les différences entre les ruches à l'aide de méthodes de régression, mais qu'il n'existe pas de moyen simple de discriminer la présence de la reine à l'intérieur de la ruche.

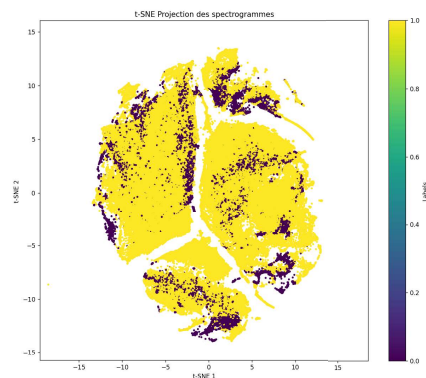
Dans l'expérience SBCM, des étiquettes Reine et SansReine ont été obtenues dans des conditions similaires et sur un intervalle de temps proche, où le t-SNE 4.7-b montre des signaux Reine/SansReine très proches. En revanche, dans le cas de NUHIVE, les différentes étiquettes Reine ont été obtenues dans des conditions différentes au sein de la même ruche, où le t-SNE 4.7-d semble un peu plus séparable. Cela indique que les conditions saisonnières peuvent jouer un facteur important dans les ensembles de données TBON et NUHIVE.

Bien que l'analyse en fréquence montre clairement une différence entre les enregistrements de certaines ruches, les données correspondant aux différents labels Reine–SansReine restent très similaires. Ainsi, il est probable que l'entraînement de modèles d'apprentissage automatique utilisant des transformations dans le domaine fréquentiel n'apprenne pas réellement la présence de la reine à l'intérieur de la ruche, mais plutôt des informations corrélées à d'autres facteurs, notamment :

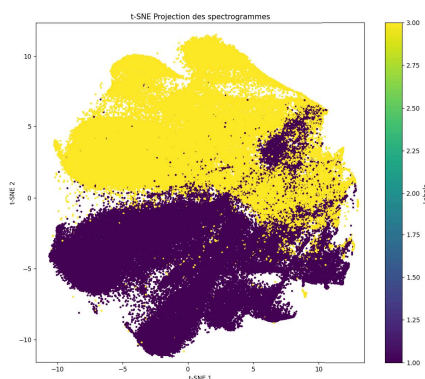
- L'appartenance à une ruche, chaque colonie étant différente d'une autre,
- Les variations saisonnières et d'activité conduisant à des comportements différents des abeilles,
- Les différences de position et de population de la colonie dans la ruche,
- Les différences dans le matériel utilisé pour les enregistrements,
- Les variations de la position du microphone à l'intérieur de la ruche.



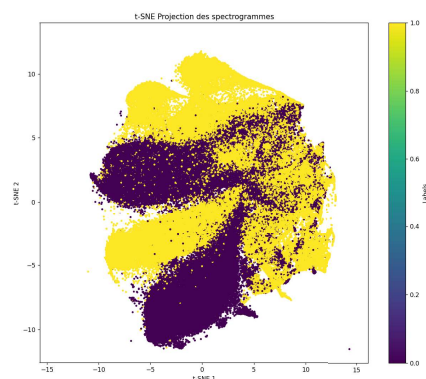
(a) t-SNE par numéro de ruche (SBCM)



(b) t-SNE par présence de la reine (SBCM)



(c) t-SNE par numéro de ruche (NUHIVE)



(d) t-SNE par présence de la reine (NUHIVE)

FIGURE 4.7 – Graphiques t-SNE des bases SBCM et NUHIVE selon la séparation par numéro de ruche et présence de la reine

4.5 Extrapolation avec un modèle inspiré du contrastive learning

Cette section s'appuie sur deux résultats précédents :

- La classification `pairwise` semble pouvoir extrapoler lorsqu'elle se limite à prédire si les labels des paires d'audios sont identiques ou non.
- L'extrapolation de la classification de la présence de la reine est complexe, possiblement en raison d'informations extérieures intégrées dans le signal audio qui agissent comme des facteurs cachés.

Pour résoudre ce problème, l'approche `pairwiseXNOR` est proposée, qui ne vise plus à prédire le label exact, mais à déterminer si deux signaux audios appartiennent ou non au même label.

Cette approche est appliquée sur les données originales, sans augmentation de données (DA), comme indiqué dans la 4.7. De nouveau, l’approche de validation k-hive-fold décrite au paragraphe 4.3.6.2 est utilisée. L’entraînement du CNN est effectué sur 20 époques, ce qui a été jugé suffisant pour atteindre la convergence. La 4.18 présente la précision et la matrice de confusion obtenues en utilisant les configurations et conventions similaires à celles de la 4.16.

TABLE 4.18 – The XNORC accuracy and confusion matrix obtained for the BT database without DA. Similar notations than for 4.16

Hive	% BT	XNORC accuracy %	XNORC confusion Matrix (%)	
1_NUHIVE	22.94	99.98	100.00 0.04	0.00 99.96
3_NUHIVE	21.42	99.78	100.00 0.44	0.00 99.56
1_SBCM	21.82	98,98	100.00 2.04	0.00 97.96
3_SBCM	8.73	99.83	100.00 0.33	0.00 99.67
4_SBCM	19.59	99.34	100.00 1.32	0.00 99.68
5_SBCM	4.47	99.98	99.96 0.18	0.04 99.82
3_TBON	0.002	100	100.00 NA	0.00 NA
4_TBON	0.51	100	NA 100.00	NA 0.00
5_TBON	0.1	100	100.00 NA	0.00 NA
6_TBON	0.41	100	NA 100.00	NA 0.00

L’approche `pairwiseXNOR` démontre une capacité à se généraliser aux ruches non vues avec une précision quasi-parfaite pour toutes les ruches.

Cette méthode permet l’application du modèle à des données des ruches non vues pendant la phase d’apprentissage. Cependant, comme le modèle repose sur la discrimination des différences entre deux fichiers audios, un premier échantillon labellisé est nécessaire dans la paire pour déterminer la présence de la reine. Il s’agit de la principale limitation de cette approche. Il est donc

nécessaire que l'apiculteur vérifie la présence de la reine au moment de l'introduction du capteur audio dans la ruche. Une fois le fichier initial labellisé, les enregistrements audios ultérieurs peuvent être comparés à cet enregistrement de référence.

De plus, étant donné que la ruche est un système évolutif avec les saisons, des conditions météorologiques et des ressources disponibles, il est essentiel de mettre à jour le fichier de référence au fil du temps. Cet ajustement régulier garantit que le fichier de référence reste cohérent avec les conditions actuelles de la ruche, minimisant ainsi le risque de divergences importantes qui pourraient conduire à des faux positifs ou négatifs en raison de variations significatives dans le système.

4.5.1 Prospective sur une base de données non labellisée : piste de validations

Après avoir établi le potentiel d'extrapolation entre les ruches de l'approche XNORC, une évaluation prospective de son utilité est menée en comparant les approches **ClassiC** et XNORC lorsqu'elles sont appliquées aux fichiers audios contenus dans l'ensemble des données BAD [87] (détaillé en 4.2.1). Cet ensemble de données BAD n'est pas labellisé, mais il propose 10 ruches inconnues, avec une configuration matérielle inconnue, enregistrant plus de 100 fichiers audios de 8 secondes pour chaque colonie, échantillonnée sur une période de 20 jours. Ces fichiers audios sont formatés et inclus dans le dataset BT sans label (le volume représente environ 400 fichiers audios de 2 secondes pour chacune des dix ruches).

La continuité temporelle et de configuration offerte par ces mesures suggère que les étiquettes elles-mêmes pourraient être continues : la reine peut être continuellement présente dans la plupart des ruches enregistrées, ou une reine peut être absente pendant plusieurs jours, avant qu'un remérage naturel ne se produise.

La 4.8 montre le statut de la reine prédit par l'approche **ClassiC** (entraînée sur l'ensemble de données étiqueté complet BT) appliquée à 4 ruches sélectionnées dans l'ensemble de données BAD (ruches 24, 26, 28 et 30). L'approche **ClassiC** ne parvient pas à se généraliser à ces 4 ruches inconnues, affichant presque un changement continu des étiquettes de reine, ce qui est très peu probable. En tenant compte de l'hypothèse de continuité des étiquettes, cela confirme la mauvaise précision, proche de 50%, précédemment constatée avec la technique Hive-fold pour ce classificateur.

Des paires d'échantillons audios sont créées pour alimenter le modèle XNORC (entraîné sur l'ensemble du dataset BT entièrement labellisé), qui est conçu pour détecter un changement de label plutôt qu'une étiquette absolue Reine/SansReine. Chaque échantillon audio de rang k est associé au cinquième échantillon suivant, de rang $k+5$, afin de minimiser la différence de temps entre les paires dans cet ensemble de données spécifique. Un fichier audio de 8,2 se-

Classic

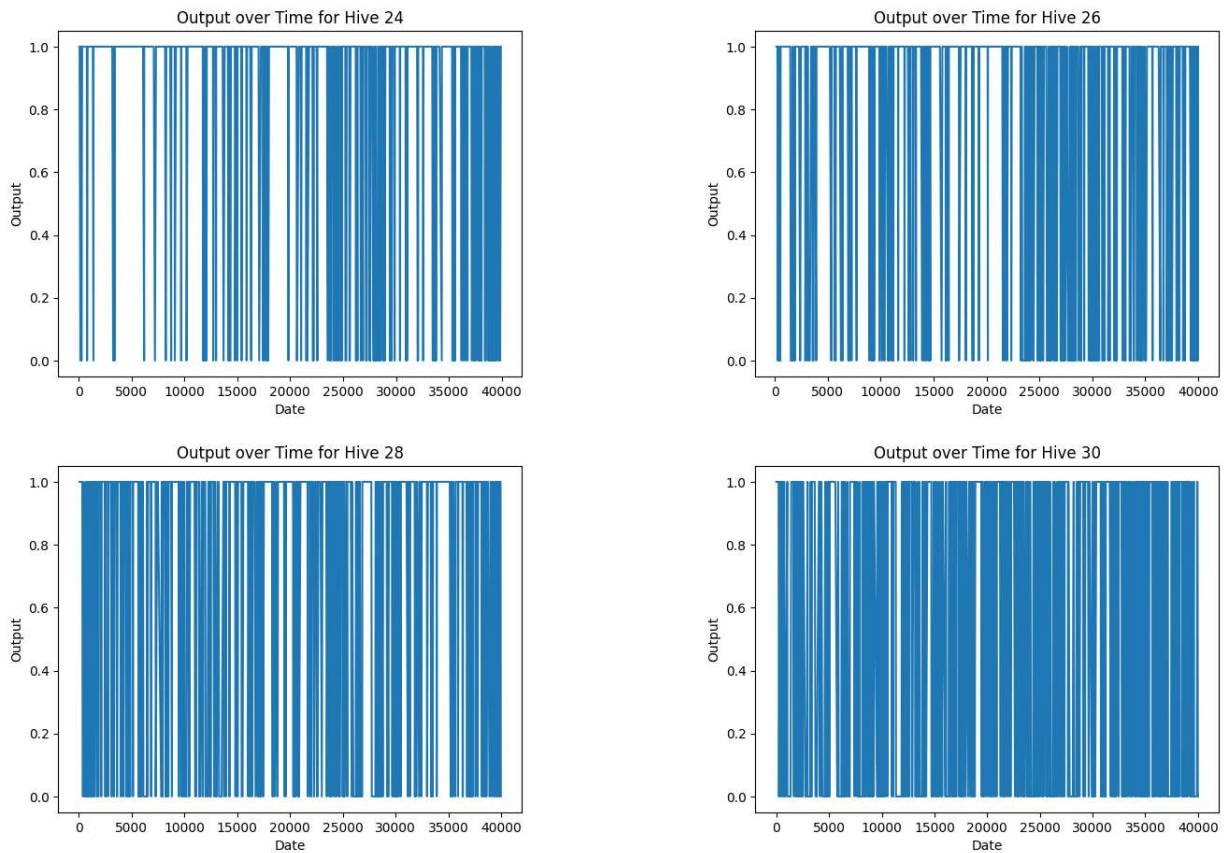


FIGURE 4.8 – Statut Reine/SansReine donn e par l’approche Classic dans le dataset non labellis e BAD (ruches 24, 26, 28 et 30). Le temps est en secondes   partir de la premi re mesure, la sortie est 1 pour le label "Reine Pr sente".

condes de l’ensemble de donn es BAD est converti en 4 fichiers audios standardis s de 2 secondes chacun pour l’ensemble de donn es BT. L’exp rience BAD g n re des s ries de 4  chantillons audios cons cutifs de 2 secondes, chaque s rie  tant espac e d’au moins une heure (et jusqu’  un maximum de 12 heures) en raison de la collecte manuelle des donn es dans l’ensemble de donn es BAD.

Le changement de statut de la pr sence de reine est illustr  dans la 4.9 en utilisant les m mes  chantillons et ruches que ceux de la 4.8. En supposant que la reine soit toujours pr sente, le pire sc nario observ  sur l’ensemble de donn es BAD montre 20 fausses d tections sur un mois, ce qui correspond   un taux de pr cision global de 99,51%. Cette pr cision est coh rente avec le score d’extrapolation obtenu dans la 4.5 en utilisant la validation par k-hive-fold.

Ces r sultats renforcent tout d’abord la confiance dans l’utilisation de l’approche contrastive pairwiseXNOR pour obtenir une bonne capacit  de g n ralisation. Deuxi mement, en utilisant l’hypoth se de continuit  de l’ tat de

XNORC

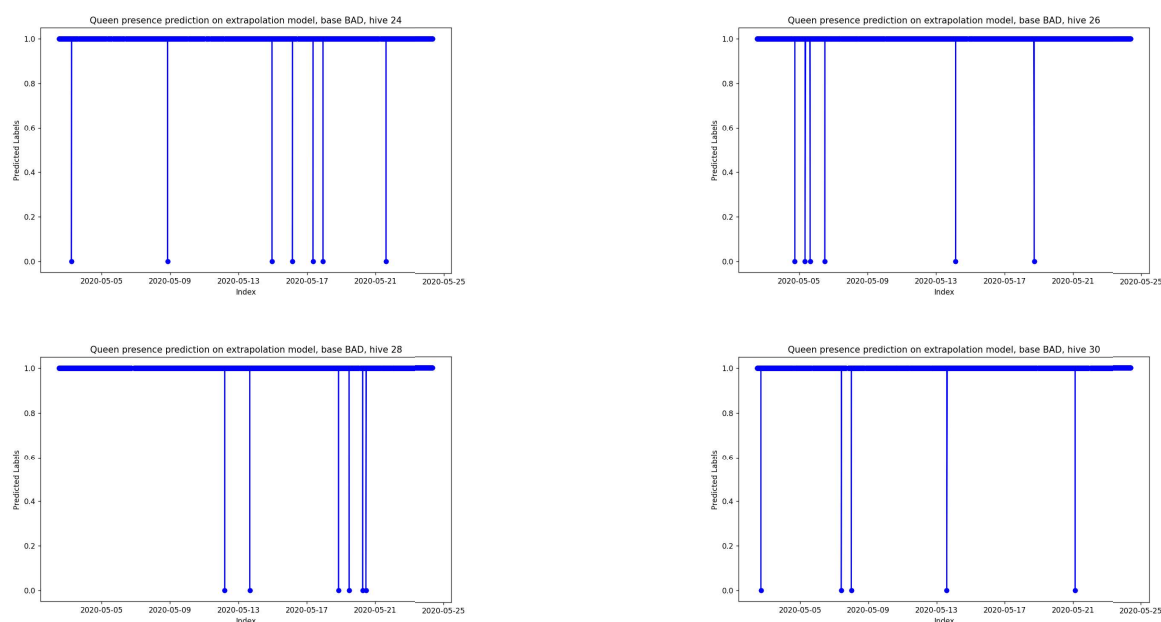


FIGURE 4.9 – Changements de labels détectés par XNORC appliqué par paires sur les mêmes ruches et échantillons que ceux de la 4.8. L'abscisse correspond à la date du second élément dans la paire d'échantillons audios. La sortie est 1 pour "mêmes labels dans la paire" et 0 pour "label changé".

présence de la reine et la précision élevée obtenue, il est possible de proposer un processus de filtrage temporel afin de produire une prédiction absolue de l'état Reine/PasReine.

La sortie "Changement d'étiquette de reine" de XNORC peut être filtrée dans le temps en s'appuyant sur ces hypothèses. Pour l'échantillonnage du jeu de données BAD (avec des rafales de 4 échantillons consécutifs, chaque rafale étant séparée d'au moins une heure), un changement de statut de la reine peut être conclu si, et seulement si, les 4 paires d'échantillons d'une rafale sont détectées comme indiquant un "changement de reine". Par exemple, la matrice de confusion de la 4.18 peut être utilisée pour établir une probabilité de 2% de détection erronée de changement dans un échantillon, ce qui correspond à une précision de 0,16 fausse détection par million pour une rafale.

En supposant que la reine soit présente au moment de l'introduction du capteur audio, le modèle peut prédire la présence continue de la reine pendant toute la période enregistrée en intégrant l'indicateur de changement de reine filtré.

Étant donné que le remérage n'est pas immédiat, il est nécessaire de prévoir une période de transition durant laquelle les "changements de reine" doivent être ignorés et considérés comme des faux négatifs.

Une évaluation du modèle sur les données produites par les capteurs audio décrits au chapitre 2 montre des résultats similaires à ceux de la base BAD. L'absence de labels Reine–SansReine ne permet pas non plus de conclure, mais n'infirmes pas la validité du modèle.

4.6 Évaluation de la possibilité d'embarquer le modèle dans un microcontrôleur

La méthode ayant été développée et testée sur une machine ayant de grosses capacités de calcul et sans se préoccuper des contraintes liées à l'embarqué, il est essentiel d'évaluer si les résultats obtenus seront facilement embarquables dans un système autonome. Dans cette section, une estimation du temps de calcul est réalisée afin de déterminer la quantité de ressources nécessaires sur le microcontrôleur et si l'application est, en définitive, possible.

La cible choisie est un microcontrôleur de la famille des nRF52 comme décrit dans le chapitre 2 pour les capteurs. Cela induit de fonctionner sur un Cortex ARM M4, cadencé à 64MHz et ayant une unité de calcul flottant (FPU). Avec ces caractéristiques, on souhaite connaître la taille de mémoire, Flash et RAM nécessaire pour accueillir et exécuter le modèle ainsi que les temps de calculs estimés. À partir de la "consommation en mode run" du microcontrôleur, il est ensuite possible de quantifier grossièrement l'énergie consommée pour l'approche développée.

Afin d'obtenir un résultat il faut réaliser deux étapes :

- Extraire le Mel-Spectrogramme du fichier Audio
- Soumettre la feature extraite au modèle pour obtenir un résultat

Extraction du Mel-Spectrogramme : La librairie CMSIS propose des fonctions pour le calcul des Mel-Spectrogrammes et des coefficients MFCC. Dans notre cas, les paramètres des Mel-Spectrogrammes définis en 4.3.2.1 sont utilisés pour estimer rapidement le temps de calcul nécessaire.

La transformation de Fourier de courte durée (STFT) est réalisée avec une taille de fenêtre de 1024 points (`n_fft = 1024`) et un pas de 60 ms (`hop_len`). Pour chaque segment d'audio, une STFT est effectuée, suivie de l'application d'une banque de filtres Mel de taille 64 (`n_mels = 64`). Les fonctions FFT de la librairie CMSIS permettent de réduire significativement le temps de calcul grâce à des optimisations spécifiques aux microcontrôleurs embarqués. Sur un microcontrôleur cadencé à 64 MHz, avec FPU, le calcul d'une FFT de 1024 points prend entre 200 et 400 microsecondes par segment. En tenant compte des multiplications nécessaires pour l'application des filtres Mel, le temps total par segment est estimé à environ 500 microsecondes. Avec un pas de 60 ms, 16,67 segments sont calculés par seconde, ce qui correspond à un temps de calcul total d'environ 8,3 ms pour une seconde d'audio, et donc en élargissant la marge 20ms pour 2 secondes d'enregistrement audio.

Évaluation de l'échantillon par le modèle : Afin de déterminer les performances du modèle sur une cible embarquée, l'outil <https://studio.edgeimpulse.com> est utilisé. Il s'agit d'un service qui, à partir du modèle enregistré évalue les ressources en mémoire nécessaires et le temps de calcul pour évaluer un échantillon par le modèle. L'évaluation est réalisée pour une cible spécifique et prend donc en compte les spécifications matérielles de la cible. Les résultats obtenus pour le modèle sont présentés en figure 4.10.



FIGURE 4.10 – Temps de calcul et ressources nécessaires pour exécuter le modèle sur une cible nRF52

Discussion sur la faisabilité du edge computing : En l'état, le modèle ne peut pas être embarqué sur une cible nRF52 en raison de la quantité de mémoire Flash requise, qui dépasse les capacités disponibles. Cependant, il est important de noter que ce modèle a été développé sans tenir compte des contraintes matérielles pour l'embarqué. Il est donc possible de réduire son empreinte mémoire et le temps de calcul en optimisant la taille du modèle dès le départ et en intégrant des optimisations spécifiques lors de l'implémentation de la solution. Les leviers d'optimisation envisageables incluent la technologie du modèle, sa taille ainsi que la dimension des features. En ajustant ces paramètres tout en surveillant les performances, il devient envisageable de proposer une solution répondant aux limitations matérielles.

Il pourrait également être nécessaire de modifier la technologie de modélisation pour opter pour des solutions plus compactes, ce qui mérite une étude approfondie. De plus, les outils de développement utilisés, tels que PyTorch, ne sont pas spécialement conçus pour les implémentations embarquées. Il sera donc crucial d'explorer des outils dédiés aux modèles embarqués (TensorFlow Lite, CMSIS-NN (Cortex Microcontroller Software Interface Standard - Neural Network, Keras to C++...) [11, 12].

Un autre aspect à considérer est la consommation énergétique entre la transmission des données et le calcul local, suivi de l'envoi du seul résultat. Actuellement, la transmission d'un échantillon audio nécessite 2 secondes et consomme 2,55 mA. Avec une consommation de run mode du microcontrôleur de 4 mA, le temps total d'exécution du modèle doit être inférieur à une seconde pour que cette approche soit plus efficace en termes de consommation énergétique.

La question peut ensuite se déplacer sur l'exécution du modèle sur le nœud supérieur : la gateway. Dans cette perspective, il serait pertinent de concevoir une gateway équipée d'un microcontrôleur dédié à l'exécution du modèle. Une telle approche permettrait d'éviter la transmission des données au serveur, poste le plus coûteux en énergie pour la gateway, tout en réduisant la

consommation de données mobiles. En utilisant un microcontrôleur Cortex M7 cadencé à 280 MHz, le temps d'exécution du modèle est réduit à 34 ms par échantillon évalué, pour une taille mémoire similaire. Un STM32H7A3ZIT6 avec 2Mo de Flash et 1376 ko serait compatible avec ce type d'application.

La stratégie consistant à séparer le microcontrôleur applicatif de celui dédié au calcul repose sur l'idée de minimiser les modifications du travail existant en ajoutant simplement une nouvelle fonction. Cela permet de limiter le re-développement tout en optimisant la consommation énergétique, puisqu'il est possible de désactiver complètement l'alimentation de cette fonction quand elle n'est pas utilisée.

Avec cette approche, il est possible de calculer le Mel-Spectrogramme directement sur la capteur audio, ce qui réduit le volume de données transféré à la gateway ; qui évalue ensuite les échantillons reçus sur le modèle déployé en embarqué. La prise de décision de transmission d'une alerte à l'apiculteur peut ainsi se faire en local du rucher, réduisant la consommation globale de la solution, le volume de données à stocker.

4.7 Conclusion

La première étape de ce chapitre réside dans la fusion et la standardisation de l'ensemble des datasets audio publics sur les ruches, créant ainsi un jeu de données suffisant pour l'exploitation par des techniques d'IA. Ce dataset, publié sous le nom de *BeeTogether* [98] sur Kaggle, constitue la base des analyses et de l'entraînement des modèles d'apprentissages. À partir du dataset *BeeTogether* les performances des méthodes IA existantes pour détecter la présence de la reine dans les ruches sont reproduites en utilisant les Mel-Spectrogrammes et un réseau de neurones convolutifs (CNN) simple [69, 89], et le problème d'extrapolation des modèles est mis en évidence.

Pour comprendre pourquoi, les fichiers audio sont analysés par des techniques de visualisation spectrale ainsi que d'apprentissage non supervisé, mettant en évidence que la colonie d'origine des données et la période de la saison influencent davantage les caractéristiques audio que le label Reine–SansReine. Les modèles actuels, basés sur des datasets uniques et peu variés, semblent capturer des facteurs cachés de présence de la reine, ce qui explique leur faible généralisation à d'autres ruches ou saisons. Des techniques d'augmentation des données avec et sans décalage en fréquence sont mises en œuvre pour augmenter la diversité des données mais sans plus de résultats.

Une méthode basée sur le *contrastive learning* est introduite, utilisant la comparaison de paires de fichiers audio sans modifier les caractéristiques du modèle CNN de référence. Les performances obtenues pour l'extrapolation du modèle surpassent largement les résultats précédents [70, 69], passant de 80% de succès à 99.9% avec la méthode proposée. La méthode est ensuite appliquée à une base de données non labellisée [87] ainsi qu'aux données collectées par nos capteurs. Les résultats obtenus sont cohérents avec ceux précédemment

observés, démontrant la capacité du modèle à s'extrapoler sur de nouvelles ruches et différents hardwares de mesure. Toutefois, l'absence de labels *Reine-SansReine* dans ces données empêche de conclure avec certitude.

Enfin, la solution est évaluée pour déterminer la possibilité d'embarquer le modèle au plus près de la mesure, soit directement sur le capteur audio, soit sur la gateway. Les résultats montrent qu'à ce stade, le modèle peut déjà être embarqué dans un microcontrôleur, bien que sa taille reste encore légèrement importante. Des travaux d'optimisation seront donc envisagés pour produire un système pleinement exploitable sur la solution matérielle actuelle.

En conclusion, les résultats d'extrapolation de la méthode de contrastive learning montrent une bonne capacité de généralisation, s'étendant non seulement à de nouvelles ruches, mais aussi à de nouveaux matériels et méthodes de mesure non rencontrés lors de la phase d'apprentissage. Ces résultats confirment la cohérence de développer une nouvelle méthode pour corrélérer audio et masse sur les données collectées par l'architecture présentée et déployée en 2, ainsi que la possibilité de placer l'intelligence directement dans le capteur, en faisant une solution optimisée.

4.7. Conclusion

Conclusion générale et perspectives

Conclusion

Dans cette thèse, nous avons démontré que l'approche consistant à concevoir des capteurs de faible performance, et donc à faible coût, puis de les améliorer par des solutions logicielles est cohérente et permet de proposer des solutions de métrologies pertinentes. Les données collectées peuvent ensuite être exploitées par des méthodes IA pour en extraire la valeur. Les solutions logicielles peuvent ensuite être embarquées au plus près de la mesure, réduisant ainsi le volume de données à transmettre, stocker puis traiter.

Ces solutions sont possibles grâce aux architectures électroniques récentes avec des microcontrôleurs qui offrent une grande capacité de calcul tout en restant extrêmement économes en énergie. De plus la réduction des coûts et la miniaturisation des mémoires sur ces architectures permettent d'implanter les solutions logicielles calculées à partir de grands volumes de données sur des machines alimentées sur secteur. Ces corrections et exploitations logicielles ont pour objectif d'extraire l'information utile des données pour en déduire des interprétations plus haut niveau. Cela réduit le volume de données à transmettre, et augmente la durée d'autonomie des produits sur batterie si les exécutions des programmes coûtent moins d'énergie que la transmission des données brutes, même compressées. Des capteurs conçus avec cette approche sont donc également intéressants du point de vue écologique, car pour un déploiement matériel similaire à un capteur sans intelligence, ils sont moins consommateurs en énergie et en architecture de stockage et analyse des données.

L'application de cette méthode à l'apiculture est évaluée dans un premier temps sur le capteur de référence : une balance pour ruche. Les choix technologiques et de qualité de la chaîne d'acquisition, ADCs et Jauges de contraintes *low cost* entraînent une dérive thermique des mesures importante. Ce défaut est compensé par une approche logicielle déterministe simple, la régression linéaire. Le modèle prend en compte le déphasage entre la température réelle et la dérive des capteurs. Cette amélioration logicielle permet une utilisation cohérente de la balance en apiculture et plusieurs évènements sur les ruches ont pu être constatés (miellée, essaimage, force d'une colonie...) sur les don-

nées rassemblées lors de la phase de déploiement de la solution dans un rucher. Pour répondre aux demandes des apiculteurs de limiter la quantité de mesures journalières pour en simplifier l'interprétation et faciliter la prise de décision, un indicateur de masse journalier est implémenté, permettant, grâce au sous-échantillonnage, de proposer une solution améliorant encore la stabilité thermique des mesures de la balance.

Dans un second temps, l'exploitation d'un capteur audio est mise en place grâce à de l'intelligence artificielle. Les objectifs initiaux visant à corrélérer la masse avec l'activité sonore se sont heurtés, à la lecture de la littérature, à un problème de généralisation des modèles aux données de ruches non observées pendant l'entraînement des modèles. Par conséquent, les travaux se sont concentrés sur les raisons de cette non-extrapolation, afin de valider la pertinence d'utiliser la métrique audio en apiculture. Pour évaluer ce problème, l'ensemble des données audio publiques labellisées selon la présence de la reine dans une colonie sont rassemblées en un dataset, ce qui maximise la diversité des ruches analysées. Un modèle de réseau de neurones convolutif est développé, reproduisant les meilleures performances de classification de la littérature, et illustrant bien la non-généralisation. La solution finalement trouvée après analyse des fichiers audio est la comparaison de deux fichiers audio issus d'une même ruche, où le label de l'un des fichiers est déjà connu. Ainsi, au lieu de classifier directement un fichier, le modèle prédit la différence. Cette solution permet d'envisager une application capable d'être déployée à grande échelle et sur des ruches différentes. Cette approche IA est également embarquable *in situ*, au plus près du capteur, augmentant sa valeur ajoutée.

Afin de minimiser le coût de la solution dans le déploiement multi capteurs sur un rucher, le maximum de fonctions électroniques est factorisé dans une gateway. Celle-ci collecte, stocke et transmet les données des capteurs. La gateway fonctionne avec deux liens radio, un lien courte portée pour communiquer en local du rucher avec les capteurs, et un longue portée pour la transmission à la base de donnée. Ce système réduit les fonctions des capteurs à l'essentiel : la mesure et un lien radio local de courte portée peu énergivore, ici le Bluetooth Low Energy. En plus de son aspect économique, cette approche permet d'optimiser la couverture réseau en positionnant la gateway dans une zone favorable, par exemple en hauteur, même dans des environnements reculés tels que les zones montagneuses ou rurales.

Perspectives

Cette thèse valide la logique de réalisation de capteurs à basse performance, basse consommation, autonomes en énergie, améliorés et exploités par le logiciel embarqué. Ces travaux sont à poursuivre dans leur réalisation et ouvrent des perspectives.

Les études ont démontré la faisabilité d'embarquer les solutions directement dans les capteurs, en termes de quantité de mémoire et de puissance de calcul.

Le déploiement de ces solutions doit s'accompagner d'une étude comparative au niveau énergétique entre la transmission des données et le calcul des résultats à partir des mesures. Une optimisation des méthodes pour réduire leurs besoins tout en conservant les meilleures performances est également un objectif futur.

Un déploiement à grande échelle de la balance et des capteurs audio pour collecter un volume de données conséquent permettrait de développer de nouvelles solutions, tant pour l'amélioration des capteurs que pour l'exploitation des données. Augmenter le volume de données de masse et les associer à des observations d'apiculteurs sur le terrain ouvre la possibilité d'identifier des patterns précédant un essaimage, de traiter une maladie à temps, d'identifier la présence d'un prédateur, et ainsi d'agir au mieux pour la santé de l'abeille en améliorant les pratiques apicoles. Avec un volume de données conséquent, la méthode de calibration des capteurs sera également à l'étude pour proposer les meilleures performances.

De même, pour les données audio, l'identification de ces événements est un objectif, car cette métrique est moins coûteuse et contraignante que l'utilisation d'une balance. La difficulté réside dans l'attribution de labels pertinents aux échantillons sonores car, à la différence des données de poids, il n'est pas possible de déduire des informations fiables par une analyse opérateur a posteriori. Une piste est donc d'utiliser, dans un premier temps, une paire de capteurs audio-balance pour avoir comme référence les données de masse afin de labelliser les échantillons audios. Des modèles visant à l'identification des événements apicoles et leur anticipation seront développés à partir des données audio exclusivement.

Cependant, conscient de la difficulté de classifier efficacement les données audio, une méthode proposée est de déployer sur un rucher une paire de capteurs audio-masse dans une ruche, et uniquement des capteurs audios dans les autres ruches du rucher. Le travail consiste alors à identifier non pas un label spécifique, mais une dérive de comportement d'une ruche par rapport à la paire étalon et/ou à l'ensemble des autres données audio du rucher. Cette méthode, plus proche du système de comparaison de paires présenté au chapitre 4, est la plus prometteuse.

Bibliographie

- [1] A. M. Klein *et al.*, “Importance of pollinators in changing landscapes for world crops,” *Proceedings of the Royal Society B : Biological Sciences*, vol. 274, no. 1608, pp. 303–313, 2007.
- [2] S. G. Potts *et al.*, “Safeguarding pollinators and their values to human well-being,” *Nature*, vol. 540, no. 7632, pp. 220–229, 2016.
- [3] M. Abdollahi, P. Giovenazzo, and T. H. Falk, “Automated beehive acoustics monitoring : A comprehensive review of the literature and recommendations for future work,” 4 2022.
- [4] A. Zaman, A. Zaman, A. Dorin, and A. Dorin, “A framework for better sensor-based beehive health monitoring,” *Cold Spring Harbor Laboratory - bioRxiv*, 2022.
- [5] M. Z. Sharif, N. Di, and B. Yu, “Honeybee (apis spp.) (hymenoptera : Apidae) colony monitoring using acoustic signals from the beehive : An assessment by global experts and our feedback,” *Agriculture (Switzerland)*, vol. 13, 4 2023.
- [6] A. Bansal and N. K. Garg, “Environmental sound classification : A descriptive review of the literature,” 11 2022.
- [7] W. Shi, W. Shi, J. Cao, J. Cao, Q. Zhang, Q. Zhang, Q. Zhang, Y. Li, Y. Li, L. Xu, and L. Xu, “Edge computing : Vision and challenges,” *IEEE Internet of Things Journal*, 2016.
- [8] M. Satyanarayanan and M. Satyanarayanan, “The emergence of edge computing,” *IEEE Computer*, 2017.
- [9] D. Sehrawat, D. Sehrawat, N. S. Gill, and N. S. Gill, “Smart sensors : Analysis of different types of iot sensors,” *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019.
- [10] S. Gervais-Ducouret and S. Gervais-Ducouret, “Next smart sensors generation,” *2011 IEEE Sensors Applications Symposium*, 2011.
- [11] I. L. Orășan, I. L. Orășan, C. Seiculescu, C. Seiculescu, C. D. Căleanu, and C. D. Căleanu, “A brief review of deep neural network implementations for arm cortex-m processor,” *Electronics*, 2022.
- [12] F. Khan, F. H. Khan, F. H. Khan, M. A. Pasha, M. A. Pasha, S. Masud, and S. Masud, “Advancements in microprocessor architecture for ubiqui-

- tous ai-an overview on history, evolution, and upcoming challenges in ai implementation.” *Micromachines*, 2021.
- [13] J. Patnaik, J. Patnaik, J. Patnaik, J. Patnaik, B. Bhowmick, B. Bhowmick, B. Bhowmick, and B. Bhowmick, “Appropriate technology and management for sustainability,” *IEEE International Conference on Industrial Engineering and Engineering Management*, 2018.
- [14] P. Söderholm and P. Söderholm, “The green economy transition : the challenges of technological change for sustainability,” *Sustainable Earth*, 2020.
- [15] B. Nicollet, *Comment débuter en apiculture ?* Paris : Rustica, 2020.
- [16] J. Riondet, *Le rucher durable : Guide pratique de l’apiculteur d’aujourd’hui*. Paris : Ulmer, 2023.
- [17] H. Clément, *Le traité Rustica de l’apiculture*. Paris : Rustica Éditions, 2018.
- [18] M. Alleri, S. Amoroso, P. Catania, G. L. Verde, S. Orlando, E. Ragusa, M. Sinacori, M. Vallone, and A. J. Vella, “Recent developments on precision beekeeping : A systematic literature review,” *null*, 2023.
- [19] S. Cecchi, S. Cecchi, S. Spinsante, S. Spinsante, A. Terenzi, A. Terenzi, S. Orcioni, and S. Orcioni, “A smart sensor-based measurement system for advanced bee hive monitoring.” *Sensors*, 2020.
- [20] A. Dsouza, P. B. Aditya, and S. Hegde, “Hivelink, an iot based smart bee hive monitoring system,” *arXiv.org*, 2023.
- [21] D. Cota, J. Martins, H. S. Mamede, and F. Branco, “Bhivesense : An integrated information system architecture for sustainable remote monitoring and management of apiaries based on iot and microservices,” *null*, 2023.
- [22] J. C. Navarro, F. Lima, M. Porto, and L. Steinfeld, “Beesmart : a real-time remote monitoring and control system for beekeeping,” *null*, 2022.
- [23] N. Andrijević, V. Urošević, B. Arsić, D. Herceg, D. Herceg, and B. Savić, “Iot monitoring and prediction modeling of honeybee activity with alarm,” *Electronics*, 2022.
- [24] P. Catania, P. Catania, M. Vallone, and M. Vallone, “Application of a precision apiculture system to monitor honey daily production.” *Sensors*, 2020.
- [25] W. Hong, W. Hong, W. Hong, B. Xu, B. Xu, B. Xu, X. Chi, X. Chi, X. Cui, X. Cui, Y. Yinfa, Y. Yan, Y. Yan, T. Li, T. Li, and T. Li, “Long-term and extensive monitoring for bee colonies based on internet of things,” *IEEE Internet of Things Journal*, 2020.
- [26] A. Szczurek, M. Maciejewska, and P. Batog, “Monitoring system enhancing the potential of urban beekeeping,” *Applied Sciences*, 2023.
- [27] I. Rigakis, I. Potamitis, N.-A. Tatlas, G. Psirofonis, E. Tzagaraki, and E. Alissandrakis, “A low-cost, low-power, multisensory device and multi-variable time series prediction for beehive health monitoring,” *null*, 2023.

-
- [28] ADA AURA, “Fiches techniques - les balances,” 2019, consulté le 30 septembre 2024. [Online]. Available : <https://www.ada-aura.org/wp-content/uploads/2019/03/fiches-techniques-balances.pdf>
- [29] H. Human, R. Brodschneider, V. Dietemann, G. Dively, J. Ellis, E. Forsgren, I. Fries, F. Hatjina, F.-L. Hu, R. Jaffé, A. Jensen, A. Köhler, J. Magyar, A. Özkyrym, C. Pirk, R. Rose, U. Strauss, G. Retschnig, D. Tarpy, and H.-Q. Zheng, “miscellaneous standard methods for apis mellifera research,” *Journal of Apicultural Research*, vol. 52, 09 2013.
- [30] D. W. Fitzgerald, D. W. Fitzgerald, F. E. Murphy, F. E. Murphy, W. G. Wright, W. M. D. Wright, P. M. Whelan, P. M. Whelan, E. Popovici, E. M. Popovici, and E. Popovici, “Design and development of a smart weighing scale for beehive monitoring,” *Irish Signals and Systems Conference*, 2015.
- [31] P. Bratek, P. Bratek, P. Dziurdzia, and P. Dziurdzia, “Energy-efficient wireless weight sensor for remote beehive monitoring.” *Sensors*, 2021.
- [32] A. A. Fröhlich, L. P. Horstmann, and J. L. C. Hoffmann, “A secure iiot gateway architecture based on trusted execution environments,” *Journal of Network and Systems Management*, 2023.
- [33] D. Bienhaus, D. Bienhaus, A. Ebner, A. Ebner, L. Jäger, L. Jäger, R. Rieke, R. Rieke, C. Krauß, and C. Krauß, “Secure gate : Secure gateways and wireless sensors as enablers for sustainability in production plants,” *Simulation Modelling Practice and Theory*, 2021.
- [34] G. C. Blog, “Tested by nordic : Bluetooth long range,” 2020, accessed : 2024-09-04. [Online]. Available : <https://blog.nordicsemi.com/getconnected/tested-by-nordic-bluetooth-long-range>
- [35] H. Karvonen, H. Karvonen, C. Pomalaza-Ráez, C. Pomalaza-Raez, K. Mikhaylov, K. Mikhaylov, M. Hämäläinen, M. Hämäläinen, J. Iinatti, and J. Iinatti, “Experimental performance evaluation of ble 4 versus ble 5 in indoors and outdoors scenarios,” *Advances in Body Area Networks I*, 2019.
- [36] Qoitech, “Qoitech - power optimization tools for developers,” 2024, consulté le 30 septembre 2024. [Online]. Available : <https://www.qoitech.com/>
- [37] B. He, B. He, G. Lü, G. Lu, K. Chu, K. Chu, G. Ma, and G. Ma, “Developing of high accuracy and low capacity strain gage based load cell for electronic scale,” *International Conference on Electronic Measurement and Instruments*, 2009.
- [38] A. Bricout, P. Leleux, P. Acco, C. Escriba, J.-Y. Fourniols, G. Soto-Romero, and R. Floquet, “Bee together : Joining bee audio datasets for hive extrapolation in ai-based monitoring,” *Sensors*, vol. 24, no. 18, 2024. [Online]. Available : <https://www.mdpi.com/1424-8220/24/18/6067>
- [39] V. Kulyukin, S. Mukherjee, and P. Amlathe, “Toward audio beehive monitoring : Deep learning vs. standard machine learning in classifying beehive audio samples,” *Applied Sciences (Switzerland)*, vol. 8, 9 2018.

- [40] A. Robles-Guerrero, T. Saucedo-Anaya, E. González-Ramírez, and J. I. De la Rosa-Vargas, “Analysis of a multiclass classification problem by Lasso Logistic Regression and Singular Value Decomposition to identify sound patterns in queenless bee colonies,” *Computers and Electronics in Agriculture*, vol. 159, no. January, pp. 69–74, 2019. [Online]. Available : <https://doi.org/10.1016/j.compag.2019.02.024>
- [41] D. Kampelopoulos, I. Sofianidis, C. Tananaki, K. Tsiapali, S. Nikolaidis, and K. Siozios, “Analyzing the beehive’s sound to monitor the presence of the queen bee,” in *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*, 2022, pp. 1–4.
- [42] A. Robles-Guerrero, T. Saucedo-Anaya, C. A. Guerrero-Mendez, S. Gómez-Jiménez, and D. J. Navarro-Solís, “Comparative Study of Machine Learning Models for Bee Colony Acoustic Pattern Classification on Low Computational Resources,” *Sensors*, vol. 23, no. 1, jan 2023.
- [43] L. Barbisan, G. Turvani, and R. Fabrizio, “Audio-based identification of queen bee presence inside beehives,” in *2023 IEEE Conference on Agri-Food Electronics (CAFE)*, 2023, pp. 70–74.
- [44] F. Ojeda, D. Méndez, A. Fajardo, and F. Ellinger, “On wireless sensor network models : A cross-layer systematic review,” *J. Sens. Actuator Networks*, 2023.
- [45] S. Qazi, S. Qazi, B. A. Khawaja, B. A. Khawaja, Q. U. Farooq, and Q. U. Farooq, “Tot-equipped and ai-enabled next generation smart agriculture : A critical review, current challenges and future trends,” *IEEE Access*, 2022.
- [46] R. Wu, H. Li, and L. Gao, “Research on temperature drift mechanism and compensation method of silicon piezoresistive pressure sensors,” *AIP Advances*, 2023.
- [47] L. Piao, L. Piao, Y. Hu, Y. Hu, X. Chang, and X. Chang, “Software compensation technology of the piezoelectric fluidic gyroscope based on dsp,” *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2017.
- [48] L. Wu, L. Wu, G. Zhao, G. Zhao, J. Yin, J. Ying, Y. Jing, Z. Feng, and Z. Feng, “A thermal drift compensation method for precision sensors considering historical temperature state,” *IEEE Transactions on Industrial Electronics*, 2021.
- [49] C. C. Sheng, C. C. Sheng, M. T. Hua, and M. T. Hua, “The research of temperature compensation technology of high-temperature pressure sensor,” *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, 2011.
- [50] J.-L. L. Goër and J. Avril, “Conception mécanique des capteurs,” *Techniques de l’Ingénieur*, 2024. [Online]. Available : <https://www.techniques-ingenieur.fr/base-documentaire/>

- mesures-analyses-th1/mesures-de-longueurs-et-d-angles-42408210/
capteurs-a-jauges-extensometriques-r1860/conception-mecanique-des-capteurs-r1860niv1000
html
- [51] HBM, “Compensation de température des jauges de contrainte,” *HBM*, 2023. [Online]. Available : <https://www.hbm.com/fr/6725/article-compensation-de-temperature-des-jauges/>
- [52] A. Socha and J. Izydorczyk, “Strain gauge calibration for high speed weight-in-motion station,” *Italian National Conference on Sensors*, 2024.
- [53] O. S. Al-Dahiree, O. S. Al-Dahiree, M. O. Tokhi, M. O. Tokhi, N. H. Hadi, N. H. Hadi, N. R. Hmoad, N. R. Hmoad, R. A. R. Ghazilla, R. A. R. Ghazilla, H. J. Yap, H. J. Yap, E. A. Albaadani, and E. A. Albaadani, “Design and shape optimization of strain gauge load cell for axial force measurement for test benches,” *Sensors*, 2022.
- [54] B. Iñigo, N. Colinas-Armijo, L. N. L. D. Lacalle, and G. Aguirre, “Digital twin for volumetric thermal error compensation of large machine tools,” *Italian National Conference on Sensors*, 2024.
- [55] L. Liu and L. Liu, “Sensitivity temperature drift error compensation of piezoelectric ceramic sensor based on big data,” *International Conference Computing Methodologies and Communication*, 2022.
- [56] A. M. Wenner, “Sound communication in honeybees,” *null*, 1964.
- [57] A. Michelsen, A. Michelsen, W. H. Kirchner, W. H. Kirchner, B. Andersen, B. B. Andersen, M. Lindauer, M. Lindauer, and M. Lindauer, “The tooting and quacking vibration signals of honeybee queens : a quantitative analysis,” *Journal of Comparative Physiology A-neuroethology Sensory Neural and Behavioral Physiology*, 1986.
- [58] W. H. Kirchner, “Acoustical communication in honeybees,” *Apidologie*, 1993.
- [59] A. Michelsen, A. Michelsen, W. H. Kirchner, W. H. Kirchner, M. Lindauer, M. Lindauer, and M. Lindauer, “Sound and vibrational signals in the dance language of the honeybee, *apis mellifera*,” *Behavioral Ecology and Sociobiology*, 1986.
- [60] M. Bencsik, M. Bencsik, J. Bencsik, J. Bencsik, M. Baxter, M. Baxter, A. Lucian, A. Lucian, J. Romieu, J. Romieu, M. Millet, and M. Millet, “Original paper : Identification of the honey bee swarming process by analysing the time course of hive vibrations,” *Computers and Electronics in Agriculture*, 2011.
- [61] S. Ferrari, S. Ferrari, S. Ferrari, S. Ferrari, M. Silva, M. Silva, M. Guarino, M. Guarino, D. Berckmans, and D. Berckmans, “Monitoring of swarming sounds in bee hives for early detection of the swarming period,” *Computers and Electronics in Agriculture*, 2008.
- [62] T. Cejrowski, J. Szymański, H. Mora, and D. Gil, “Detection of the bee queen presence using sound analysis,” *Lecture Notes in Computer Science*

- (including subseries *Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*), vol. 10752 LNAI, pp. 297–306, 2018.
- [63] I. Nolasco and E. Benetos, “To bee or not to bee : Investigating machine learning approaches for beehive sound recognition,” *Detection and Classification of Acoustic Scenes and Events 2018*, 11 2018. [Online]. Available : <http://arxiv.org/abs/1811.06016>
- [64] A. Terenzi, S. Cecchi, and S. Spinsante, “On the importance of the sound emitted by honey bee hives,” *Veterinary Sciences*, vol. 7, pp. 1–14, 2020.
- [65] H. Hadjur, D. Ammar, and L. Lefèvre, “Toward an intelligent and efficient beehive : A survey of precision beekeeping systems and services,” *Computers and Electronics in Agriculture*, vol. 192, 2022.
- [66] A. Orłowska, D. Fourer, J.-P. Gavini, and D. Cassou-Ribehart, “Honey bee queen presence detection from audio field recordings using summarized spectrogram and convolutional neural networks,” in *21st International Conference on Intelligent Systems Design and Applications (ISDA 2021)*, ser. Lecture Notes in Networks and Systems, vol. 418, Seattle, WA, (World Wide Web), United States, Dec 2021, pp. 83–92. [Online]. Available : <https://hal.science/hal-03439646>
- [67] B. S. Soares, J. S. Luz, V. F. de Macêdo, R. R. V. e. Silva, F. H. D. de Araújo, and D. M. V. Magalhães, “Mfcc-based descriptor for bee queen presence detection,” *Expert Systems with Applications*, vol. 201, 9 2022.
- [68] D. Kanelis, V. Liolios, F. Papadopoulou, M.-A. Rodopoulou, D. Kampelopoulos, K. Siozios, and C. Tananaki, “Decoding the behavior of a queenless colony using sound signals,” *Biology*, vol. 12, p. 1392, 2023.
- [69] C. Uthoff, M. N. Homsy, and M. von Bergen, “Acoustic and vibration monitoring of honeybee colonies for beekeeping-relevant aspects of presence of queen bee and swarming,” *Computers and Electronics in Agriculture*, vol. 205, p. 107589, 2023. [Online]. Available : <https://doi.org/10.1016/j.compag.2022.107589>
- [70] I. Nolasco, A. Terenzi, S. Cecchi, S. Orcioni, H. L. Bear, and E. Benetos, “Audio-based identification of beehive states,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, pp. 8256–8260, 2019.
- [71] A. Zgank, “Bee swarm activity acoustic classification for an iot-based farm service,” *Sensors (Switzerland)*, vol. 20, 1 2020.
- [72] J. Kim, J. Oh, and T. Y. Heo, “Acoustic scene classification and visualization of beehive sounds using machine learning algorithms and grad-cam,” *Mathematical Problems in Engineering*, vol. 2021, 2021.
- [73] A. Zgank, “Iot-based bee swarm activity acoustic classification using deep neural networks,” *Sensors (Switzerland)*, vol. 21, pp. 1–14, 2 2021.
- [74] V. Kulyukin, “Audio, image, video, and weather datasets for continuous electronic beehive monitoring,” *Applied Sciences (Switzerland)*, vol. 11, 2021.

- [75] S. J. S. Quaderi, S. A. Labonno, S. Mostafa, and S. Akhter, “Identify the beehive sound using deep learning,” *International Journal of Computer Science and Information Technology*, vol. 14, pp. 13–29, 8 2022.
- [76] N. Di, M. Z. Sharif, Z. Hu, R. Xue, and B. Yu, “Applicability of vggish embedding in bee colony monitoring : comparison with mfcc in colony sound classification,” *PeerJ*, vol. 11, 1 2023.
- [77] T. H. Truong, H. D. Nguyen, T. Q. A. Mai, H. L. Nguyen, T. N. M. Dang, and T. T. H. Phan, “A deep learning-based approach for bee sound identification,” *Ecological Informatics*, vol. 78, p. 102274, 2023. [Online]. Available : <https://doi.org/10.1016/j.ecoinf.2023.102274>
- [78] A. Farina, “Discovering ecoacoustic codes in beehives : First evidence and perspectives,” *BioSystems*, vol. 234, p. 105041, 2023. [Online]. Available : <https://doi.org/10.1016/j.biosystems.2023.105041>
- [79] S. Ruvinga, G. Hunter, O. Duran, and J. C. Nebel, “Identifying queenlessness in honeybee hives from audio signals using machine learning,” *Electronics (Switzerland)*, vol. 12, 4 2023.
- [80] T. T. H. Phan, D. Nguyen-Doan, D. Nguyen-Huu, H. Nguyen-Van, and T. Pham-Hong, “Investigation on new mel frequency cepstral coefficients features and hyper-parameters tuning technique for bee sound recognition,” *Soft Computing*, vol. 27, pp. 5873–5892, 2023. [Online]. Available : <https://doi.org/10.1007/s00500-022-07596-6>
- [81] F. Rustam, M. Z. Sharif, W. Aljedaani, E. Lee, and I. Ashraf, “Bee detection in bee hives using selective features from acoustic data,” *Multimedia Tools and Applications*, 2023.
- [82] J. A. Calvo, “Open source Behive Project,” OSBH Zenodo dataset, 2 2017. [Online]. Available : <https://zenodo.org/records/321345>
- [83] I. Nolasco, A. T. S. Cecchi, S. Orcioni, and H. L. B. E. Benetos, “Audio-Based identification of Beehive states : The dataset,” NuHive Zenodo dataset, 2 2019. [Online]. Available : <https://zenodo.org/records/2667806>
- [84] “To bee or not to bee,” TBON Kaggle dataset, 2018. [Online]. Available : <https://www.kaggle.com/datasets/chrisfilo/to-bee-or-no-to-bee>
- [85] “Beehive buzz anomalies,” TBON processed Kaggle dataset, 2018. [Online]. Available : <https://www.kaggle.com/datasets/yevheniiklymenko/beehive-buzz-anomalies>
- [86] A. Yang, “Smart bee colony monitor : Clips of beehive sounds,” SBCM Kaggle Dataset, 2022. [Online]. Available : <https://www.kaggle.com/dsv/4451415>
- [87] Naturami and D. T. Varkonyi, “beehive audio recordings,” BAD Zenodo dataset, 9 2022. [Online]. Available : <https://zenodo.org/records/7052981>
- [88] T. Cejrowski and J. Szymański, “Buzz-based honeybee colony fingerprint,” *Computers and Electronics in Agriculture*, vol. 191, 2021.

- [89] A. Terenzi, N. Ortolani, I. Nolasco, E. Benetos, and S. Cecchi, "Comparison of feature extraction methods for sound-based classification of honey bee activity," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 112–122, 2022.
- [90] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, B. van Niekirk, D. Lee, F. Cwitkowitz, F. Zalkow, O. Nieto, D. Ellis, J. Mason, K. Lee, B. Steers, E. Halvachs, C. Thom e, F. Robert-St oter, R. Bittner, Z. Wei, A. Weiss, E. Battenberg, K. Choi, R. Yamamoto, C. Carr, A. Metsai, S. Sullivan, P. Friesch, A. Krishnakumar, S. Hidaka, S. Kowalik, F. Keller, D. Mazur, A. Chabot-Leclerc, C. Hawthorne, C. Ramaprasad, M. Keum, J. Gomez, W. Monroe, V. A. Morozov, K. Eliasi, nullmightybofo, P. Biberstein, N. D. Sergin, R. Hennequin, R. Naktinis, beantowel, T. Kim, J. P.  asen, J. Lim, A. Malins, D. Here n u, S. van der Struijk, L. Nickel, J. Wu, Z. Wang, T. Gates, M. Vollrath, A. Sarroff, Xiao-Ming, A. Porter, S. Kranzler, Voodoohop, M. D. Gangi, H. Jinoz, C. Guerrero, A. Mazhar, toddrme2178, Z. Baratz, A. Kostin, X. Zhuang, C. T. Lo, P. Campr, E. Semeniuc, M. Biswal, S. Moura, P. Brossier, H. Lee, and W. Pimenta, "librosa/librosa : 0.10.2.post1," 5 2024. [Online]. Available : <https://doi.org/10.5281/zenodo.11192913>
- [91] P. Champion, "Anonymizing speech : Evaluating and designing speaker anonymization techniques," Ph.D. dissertation, Universit e de Lorraine 2023, 2023, th ese de doctorat dirig ee par Jouv et, DenisLarcher, Anthony et Ouni, Slim. [Online]. Available : <http://www.theses.fr/2023LORR0101/document>
- [92] S. Umesh, L. Cohen, and D. Nelson, "Fitting the mel scale," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 1, 1999, pp. 217–220 vol.1.
- [93] J. Mahoney and D. Schensul, "454 Historical Context and Path Dependence," in *The Oxford Handbook of Contextual Political Analysis*. Oxford University Press, 03 2006. [Online]. Available : <https://doi.org/10.1093/oxfordhb/9780199270439.003.0024>
- [94] D. W. Robinson and R. S. Dadson, "A re-determination of the equal-loudness relations for pure tones," *British Journal of Applied Physics*, vol. 7, no. 5, p. 166, may 1956. [Online]. Available : <https://dx.doi.org/10.1088/0508-3443/7/5/302>
- [95] G. Kim, G. Kim, D. K. Han, D. K. Han, D. K. Han, H. Ko, and H. Ko, "Specmix : A mixed sample data augmentation method for training with time-frequency domain features," *Interspeech*, 2021.
- [96] T.-Y. Hu, T.-Y. Hu, T.-Y. Hu, T.-Y. Hu, A. Shrivastava, A. Shrivastava, J.-H. R. Chang, R. Chang, J.-H. R. Chang, J.-H. R. Chang, H. S. Koppula, H. Koppula, S. Braun, S. Braun, S. Braun, S. Braun, K. Hwang, K. Hwang,

- O. Kalinli, O. Kalini, O. Kalinli, O. Kalinli, O. Tuzel, O. Tuzel, O. Tuzel, and O. Tuzel, "Sapaugment : Learning a sample adaptive policy for data augmentation," *arXiv : Learning*, 2020.
- [97] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, 1993.
- [98] A. Bricout, "Bee Together," BT Kaggle dataset, 8 2024. [Online]. Available : <https://www.kaggle.com/datasets/augustin23/beetogether/data>