UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
LAHCENE ZINNOUR

ÉVALUATION ET DÉVELOPPEMENT DE SERVICE (API) D'AIDE À LA DÉCISION CLINIQUE À L'AIDE DE RÉSEAUX DE NEURONES BAYÉSIENS

JANVIER 2023

Résumé

Les maladies cardiovasculaires deviennent l'un des problèmes de santé publique les plus importants dans la plupart des régions du monde, y compris la population canadienne. Une approche préventive éprouvée aux problèmes de santé cardiovasculaire est la pratique de l'activité physique régulière. Les exercices d'activité physique sont associés à une diminution de du risque de développer une maladie cardiovasculaire et de sa létalité. Pour aider à ancrer et à suivre les habitudes d'activité physique au sein de la population, un groupe de cinq personnes dont moi Lahcene Zinnour avec l'aide de l'étudiant Moulaye Abdallah Sadegh, et Myriam Pettigrew en tant que kinésiologue avec l'assistance du professeur Fadel Touré et de la professeure Julie Houle, nous avons développé un système d'aide à la décision clinique POD-iSanté (Plateforme Opérationnelle d'analyse de Données sur la pratique de l'activité physique appliqué à la santé) pour prescrire des exercices et proposer des recommandations aux patients. Notre système d'aide à la décision clinique utilisant l'apprentissage profond AP-SADC, un domaine de l'apprentissage automatique pour aider les cliniciens à prendre de meilleures décisions concernant l'activité physique des patients visant à améliorer leur santé. L'élément clé de notre AP-SADC proposé est l'estimation de l'incertitude qui l'accompagne dans la phase de traitement en raison de différents facteurs. Nous nous concentrons sur le AP-SADC et la façon bayésienne de penser pour la prise de décision, et de capture de l'incertitude. Différents modèles avec différentes architectures ont été proposés dans ce travail pour trouver le meilleur modèle possible. Dans ce travail, nous avons réalisé de multiples évaluations sur les différents modèles testés afin de mieux établir la performance et la faisabilité du POD-iSanté.

Mots-clés : Système d'aide à la décision clinique (SADC), Maladie cardiovasculaire, Réseaux de neurones bayésiens, Estimation de l'incertitude, Apprentissage profond, Prise de décision, Microservices.

Abstract

Cardiovascular disease is increasingly becoming one of the most important public health issues in most parts of the world, including the Canadian population. An approved solution to cardiovascular health problems is the regular physical activity. Physical activity exercises are associated with a decrease in cardiovascular mortality along with the risk of developing cardiovascular disease. To address this problem, a group of five people including me Lahcene Zinnour with the help of the student Moulaye Abdallah Sadegh, and Myriam Pettigrew as a kinesiologist with the assistance of professor Fadel Touré and professor Julie Houle, we developed a support system for the clinical decision have developed a clinical decision support system POD-iSanté (Operational Platform for Data Analysis on the practice of physical activity applied to health) to prescribe exercises and offers recommendations to patients. Our clinical decision support system using deep learning DL-CDSS, an area of machine learning to help clinicians make better decisions about patient physical activity to improve their health. The key element of our proposed DL-CDSS is the uncertainty quantification that accompanies it in the treatment phase due to varied factors. We focus on DL-CDSS and the Bayesian way of thinking for decision making and capturing uncertainty. Different models with different architectures are proposed to find the best possible model. In this work, multiples evaluations were performed on the different models proposed to better establish the performance and feasibility of the POD-iSanté.

Keywords: Clinical Decision Support System (CDSS), Cardiovascular disease, Bayesian neural networks, Uncertainty quantification, Deep learning, Decision making, Microservices.

Table des matières

Liste des tableaux

Liste des figures

A mes chers parents Fatma et Kouider, à ma sœur Faiza et à mes frères Mohamed, Larbi et Islam.

Remerciements

Ce mémoire est le résultat d'un travail de recherche de près de trois ans. En préambule, je veux adresser tous mes remerciements aux personnes avec lesquelles j'ai pu échanger et qui m'ont aidé pour la rédaction de ce mémoire.

En commençant par remercier tout d'abord Monsieur Fadel Toure, professeur à l'Université du Québec à Trois-Rivières et directeur de recherche de ce mémoire, pour son aide précieuse et pour le temps qu'il m'a consacré.

Merci à Madame Julie Houle et Myriam Pettigrew qui ont su me guider vers les bonnes références, et l'expérience que j'ai acquise avec elles dans le domaine médical.

Je remercie également Monsieur Fadel Touré et Madame Julie Houle pour leur contribution afin que je puisse recevoir une bourse Mitacs Accelerate.

Je remercie aussi mon ami Moulaye Abdallah Sadegh.

Je tiens également à remercier l'équipe de l'entreprise Intégration Santé pour leur aide durant le stage de quatre mois que j'ai passé avec eux.

Enfin, j'adresse mes plus sincères remerciements à ma famille : Mes parents, ma sœur et à mes frères.

Lahcene Zinnour

Chapitre 1 – Introduction

1.1. Apprentissage automatique

L'intelligence artificielle (IA) est un terme large qui fait référence à tout ce qui permet à un ordinateur d'agir d'une manière qui semble intelligente. L'intelligence artificielle peut être utilisée à de nombreuses fins différentes, notamment prendre des décisions, résoudre des problèmes ou surpasser les humains dans certaines tâches telles que la reconnaissance faciale et la détection d'objets, mais également utilisée dans le domaine du traitement du langage naturel et de la vision par ordinateur.

L'apprentissage automatique (en anglais: Machine Learning ML) est un processus de programmation d'un ordinateur pour prendre des décisions sans être explicitement programmé. Il existe de nombreuses méthodes d'apprentissage automatique, mais elles partagent toutes le même objectif : permettre à l'ordinateur d'apprendre à partir des données et d'améliorer ses performances au fil du temps. Certaines méthodes courantes d'apprentissage automatique comprennent l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

L'apprentissage supervisé est une méthode dans laquelle l'ordinateur reçoit un ensemble de données d'entraînement, ainsi que la bonne réponse. L'ordinateur utilise ensuite ces données pour apprendre à identifier correctement les modèles et à faire des prédictions. Les algorithmes d'apprentissage supervisé peuvent être utilisés pour des tâches telles que la classification (déterminer à quelle catégorie appartient un objet) ou la régression (prédire des valeurs numériques).

L'apprentissage non supervisé est une méthode dans laquelle l'ordinateur ne reçoit que des données d'entrée sans aucune réponse correspondante. Il doit ensuite trouver des modèles dans les données elles-mêmes afin de faire des prédictions. Ce type d'apprentissage peut être utilisé pour des tâches telles que le regroupement (regroupement d'objets en fonction de leurs

similitudes) ou la réduction de la dimensionnalité (réduction du nombre de dimensions dans les données d'entrée afin qu'il soit plus facile de travailler avec).

L'apprentissage en profondeur (AP) est un type d'apprentissage automatique. L'approche d'apprentissage en profondeur introduit des réseaux de neurones profonds complexes et multicouches pour résoudre des problèmes plus difficiles pour les algorithmes d'apprentissage automatique traditionnels, et peut utiliser des données beaucoup plus volumineux et complexes. Cela améliore l'apprentissage pour des tâches telles que la reconnaissance d'images et la reconnaissance vocale. L'apprentissage en profondeur a été à l'origine de certaines percées récentes dans le domaine de l'IA, telles que les voitures autonomes et la robotique.

Dans le secteur de la santé, l'apprentissage automatique peut être utilisé pour prédire les diagnostics des patients, identifier les facteurs de risque de maladies et trouver de nouveaux traitements pour les maladies. Pour le diagnostic, l'apprentissage automatique peut aider à identifier des modèles dans les données qui peuvent indiquer une maladie ou un état particulier. Cela peut conduire à des diagnostics plus précis et améliorer les résultats pour les patients. Par exemple, (Rajkomar, et al. 2018) à l'aide de données structurées du dossier de santé électroniques (DSE), ont développé des modèles d'apprentissage en profondeur pour prédire la mortalité hospitalière, le diagnostic et la réadmission du patient.

En termes de traitement, les algorithmes ML se sont révélés efficaces pour prédire comment les patients réagiront à certains médicaments. Ces informations peuvent ensuite être utilisées pour personnaliser les traitements pour chaque patient. Une étude a révélé qu'une application basée sur l'IA peut aider à choisir les thérapies initiales pour différents types de cancers (Zauderer, et al. 2014). De telles prédictions pourraient donner aux médecins suffisamment de temps pour intervenir avant qu'une hospitalisation ne se produise et prévenir d'éventuelles complications.

La technologie d'IA disponible aujourd'hui a une portée trop étroite pour remplacer entièrement le rôle d'un médecin dans les soins de santé. Il existe de nombreux aspects du diagnostic et du traitement des patients que la technologie actuelle de l'IA n'est pas encore en mesure de gérer. De plus, il existe de nombreux cas complexes où les médecins humains doivent

se fier à leur intuition et à leur expérience pour prendre la meilleure décision pour le patient. Bien que la technologie puisse être utile dans certains domaines de la santé, elle n'est pas encore en mesure de remplacer complètement le médecin.

Parce qu'il y a tellement d'enjeux en matière de santé humaine, nous ne pouvons pas nous fier uniquement aux simulations informatiques pour prendre des décisions sur les traitements qui fonctionnent le mieux.

L'utilisation de l'intelligence artificielle dans les soins de santé se développe rapidement alors que les prestataires cherchent à améliorer la qualité et l'efficacité des soins. Bien que l'utilisation de l'IA dans les soins de santé présente de nombreux avantages potentiels, notamment une meilleure prise de décision et de meilleurs résultats pour les patients, des résultats haute-fidélité et mesurés de manière fiable ne sont pas toujours réalisables, car les systèmes d'IA sont contraints d'apprendre à partir des données d'observation disponibles sur la santé.

L'un des défis liés au fait de s'appuyer sur des données d'observation est qu'elles peuvent être biaisées. Par exemple, si un hôpital enregistre uniquement des informations sur les patients qui se portent bien après le traitement, le système d'IA peut apprendre que le traitement X donne de bons résultats même si ce n'est pas le cas. Un autre défi est qu'il peut y avoir beaucoup de variations dans la façon dont les patients réagissent aux traitements en raison de différences individuelles telles que la génétique ou des facteurs liés au mode de vie. Cela signifie qu'un système d'IA peut ne pas toujours être en mesure de prédire avec précision comment un patient particulier réagira à un certain traitement.

Malgré ces défis, il existe encore un grand potentiel pour l'utilisation de l'IA dans les soins de santé. Grâce à une conception et une validation minutieuse par rapport à des ensembles de données cliniques, les systèmes d'IA se sont déjà révélés capables de surpasser les experts humains pour diagnostiquer les maladies et prédire les résultats des patients (Davenport & Kalakota, 2019).

1.2. Les systèmes d'aide à la décision clinique

Au cours des deux dernières décennies, le développement du système de surveillance des patients a considérablement augmenté, en particulier dans le domaine de la médecine générale. Ces systèmes sont conçus pour aider les professionnels de la santé à améliorer les soins aux patients en fournissant une vue complète de l'état et des tendances des patients au fil du temps. Il existe de nombreux types de systèmes de surveillance des patients disponibles sur le marché, tels que les moniteurs des signes vitaux, les systèmes d'aide à la décision clinique, la surveillance des alarmes intelligentes et d'autres systèmes de diagnostic assistés par ordinateur (Baig et al., 2017).

Les systèmes de surveillance des patients offrent un certain nombre d'avantages aux prestataires de soins de santé et aux patients. Pour les prestataires, les moniteurs de patient fournissent un accès instantané aux signes vitaux et à d'autres paramètres de santé, ce qui peut aider à identifier les changements dans l'état d'un patient qui peuvent nécessiter une intervention. De plus, en fournissant des données sur les tendances des patients au fil du temps, les moniteurs de patient peuvent aider les prestataires à optimiser les plans de soins et les protocoles de traitement. Pour les patients, l'accès en temps réel à leurs propres données de santé peut leur permettre de prendre des décisions plus éclairées concernant leurs soins. Les moniteurs patients incluent également souvent des fonctionnalités telles que des alarmes qui informent les soignants lorsque l'état d'un patient change afin que des mesures appropriées puissent être prises rapidement. Cela permet de s'assurer que les patients reçoivent les meilleurs soins possibles tout en minimisant le risque d'événements indésirables.

Les systèmes d'aide à la décision clinique (SADC) et les systèmes experts sont des programmes informatiques considérés comme les interventions les plus courantes qui aident les cliniciens à prendre de meilleures décisions. L'approche basée sur des règles utilisée par la plupart des systèmes experts peut être très longue et laborieuse pour les développeurs et les utilisateurs. Le nombre de règles peut être grand et il peut être difficile de déterminer quelle règle doit être appliquée dans une situation donnée. De plus, les règles peuvent ne pas couvrir tous les scénarios possibles, de sorte que le système peut ne pas fournir de conseils précis dans tous les cas. Une

autre limitation est que les systèmes experts ne peuvent prendre que les décisions pour lesquelles ils sont programmés. Ils ne peuvent pas apprendre et s'adapter comme les gens le peuvent. Enfin, les systèmes experts sont coûteux à développer et à entretenir.

D'autre part, les systèmes d'aide à la décision clinique fournissent des recommandations ou des alertes basées sur des directives fondées sur des preuves concernant les traitements les plus efficaces pour certaines conditions ou la manière dont les médicaments doivent être prescrits ensemble en toute sécurité. Le SADC peut aider les médecins à éviter de commettre des erreurs coûteuses ou de choisir des traitements qui pourraient ne pas fonctionner aussi bien pour leurs patients.

Les avantages des SADC basés sur l'intelligence artificielle (IA) incluent des diagnostics plus précis grâce à l'utilisation de grands ensembles de données, une meilleure compréhension des antécédents et de l'état de santé actuel d'un patient, interventions plus opportunes, et une meilleure communication entre les cliniciens. Cependant, ces systèmes présentent également des risques. Par exemple, si un système d'IA est utilisé pour recommander des traitements ou des médicaments aux patients, il peut ne pas être conscient de tous les facteurs individuels qui pourraient affecter la façon dont une personne particulière réagit à un certain médicament (comme l'âge ou le poids) ce qui pourrait entraîner des effets indésirables. Il est également possible que les systèmes d'aide à la décision basés sur l'IA commettent des erreurs dans leurs évaluations ou leurs recommandations, ce qui pourrait nuire aux patients. Une autre préoccupation est qu'à mesure que ces systèmes deviennent plus intelligents, ils peuvent mieux prédire le comportement humain, y compris les préférences personnelles et les vulnérabilités, qui pourraient ensuite être exploitées par des pirates ou d'autres personnes ayant des intentions malveillantes.

Toutes ces technologies jouent un rôle important pour aider les cliniciens à prendre des décisions éclairées rapidement et efficacement afin qu'ils puissent se concentrer sur la prestation de soins de qualité à leurs patients.

1.3. Les dossiers médicaux électroniques

En médecine, la quantité d'informations qui doit être traitée par un médecin pour prendre une décision clinique bien informée et optimale peut être énorme. Le clinicien doit peser les preuves issues de la recherche, les antécédents du patient et les symptômes actuels, ainsi que d'autres tests et résultats d'examens. Il est essentiel que les médecins se tiennent au courant des progrès médicaux en lisant des revues et en assistant à des conférences afin de pouvoir fournir à leurs patients les meilleurs soins possibles. Il est clair qu'être capable de traiter de grandes quantités d'informations rapidement et avec précision est essentiel pour une bonne pratique clinique en médecine aujourd'hui. Bien que cela puisse sembler intimidant au début, avec les bons outils et la bonne formation, cela est réalisable pour tous les professionnels de la santé. Heureusement, la technologie joue un rôle de plus en plus important pour aider les cliniciens à gérer cette mine de données. Les dossiers médicaux électroniques (DME) sont désormais courants dans la plupart des établissements médicaux et permettent aux médecins de suivre les antécédents des patients, de commander des tests de laboratoire ou de prescrire des médicaments par voie électronique. Cela permet non seulement de gagner du temps, mais aussi de réduire les erreurs puisque les ordonnances peuvent être vérifiées par rapport à l'historique des médicaments d'un patient avant d'être remplies.

La disponibilité des données de santé a considérablement augmenté ces dernières années avec l'adoption généralisée des appareils de surveillance médicale et le nombre croissant d'appareils connectés. Ce volume croissant de données offre d'énormes opportunités aux chercheurs, aux prestataires et aux patients d'améliorer la qualité des soins. Cependant, la gestion efficace de ces données est un défi qui doit être relevé pour réaliser tout son potentiel.

L'une des principales utilisations des données sur les soins de santé est l'amélioration des soins aux patients grâce à de meilleurs systèmes d'aide à la prise de décision. Par exemple, les DME et les SADC ont le potentiel de promouvoir une utilisation appropriée des antimicrobiens (Forrest, et al. 2014). En fournissant aux médecins des recommandations personnalisées basées sur le profil de séquence génomique spécifique de leur patient (plutôt qu'une approche unique),

ces outils ont le potentiel de réduire les effets secondaires de la chimiothérapie tout en maintenant, voire en améliorant les taux de guérison.

La mise en œuvre des DME et SADC a fourni aux scientifiques de riches dossiers longitudinaux, multidimensionnels et détaillés sur les données de santé d'un individu. L'utilisation de ces technologies a révolutionné la façon dont les chercheurs peuvent étudier le corps humain et la progression de la maladie. Les DME offrent une vue plus complète des antécédents médicaux d'un individu en incorporant des données provenant de plusieurs sources, notamment des hôpitaux, des cliniques, des pharmacies et des laboratoires. Cela permet une analyse plus précise des résultats des patients. En plus d'améliorer notre compréhension des maladies, les DME ont également le potentiel d'améliorer la qualité et la sécurité des soins aux patients. En réduisant les erreurs de médication et en promouvant des pratiques de soins fondées sur des données probantes, les DME peuvent aider à garantir que les patients reçoivent le meilleur traitement possible. Bien qu'il existe certaines préoccupations concernant la confidentialité et la sécurité liées aux systèmes de DME, ces problèmes peuvent être résolus grâce à une planification et une mise en œuvre minutieuses.

La pertinence de ce projet tient au fait que des études cliniques ont démontré l'efficacité de l'utilisation de moniteurs d'AP (podomètre, accéléromètre) combinés à des recommandations personnalisées afin de rehausser le niveau d'AP auprès de différentes populations atteintes de maladies chroniques (Baskerville et al., 2017; Hodkinson et al., 2019; Houle et al., 2012).

1.4. Incertitude dans la décision médicale

L'incertitude des décisions médicales est un problème courant auquel les médecins sont confrontés. Il peut être difficile de savoir quel est le meilleur plan d'action, surtout lorsqu'il n'y a pas de bonne ou de mauvaise réponse claire. Dans ces cas, il est important de peser toutes les options et de prendre la meilleure décision pour le patient.

De nombreux facteurs peuvent contribuer à l'incertitude des décisions médicales. Parfois, plusieurs traitements sont disponibles pour une condition particulière, et il peut être difficile de décider lequel est le plus approprié. De plus, de nouvelles recherches peuvent émerger après

qu’un médecin a posé son diagnostic initial, ce qui pourrait entraîner des changements dans les plans de traitement. Et enfin, les patients eux-mêmes peuvent ne pas toujours savoir ce qu’ils veulent ou ce dont ils ont besoin, ce qui rend difficile pour les médecins de déterminer le meilleur plan d’action.

Malgré les défis liés à la prise de décisions médicales dans des circonstances incertaines, il est important que les médecins fassent de leur mieux pour garantir des résultats positifs pour leurs patients. En examinant attentivement toutes les options et en tenant compte de la situation unique de chaque individu, les médecins peuvent prendre des décisions éclairées même lorsqu’ils sont confrontés à l’incertitude.

Le théorème du jury de Condorcet stipule que les décisions moyennes d’une foule d’experts impartiaux sont plus correctes que les décisions de n’importe quel individu. En utilisant l’apprentissage par réseau bayésien et la modélisation probabiliste, les chercheurs tentent d’éviter l’inconvénient de l’approche traditionnelle pour trouver des modèles (Chen, et al. 2017) (Klann, et al. 2014).

Dans le monde de la santé, il y a toujours de l’incertitude. Les médecins doivent constamment peser divers risques et avantages lorsqu’ils décident d’un traitement pour leurs patients. Dans certains cas, la décision peut être nette, mais dans de nombreux autres, il y a une place importante pour le débat. L’approche bayésienne peut aider à prendre ces décisions en incorporant de nouvelles informations dans un modèle existant afin de mettre à jour nos croyances sur la probabilité de résultats différents.

La théorie de la décision bayésienne peut aider les fournisseurs à prendre les décisions les plus éclairées possibles face à cette incertitude. L'inférence bayésienne fonctionne par traitement continu des évidences empiriques pour mettre à jour la probabilité qu'une hypothèse soit vraie (Pedersen & Ritter, 2022). Dans le contexte des soins de santé, cela signifie que nous devons continuellement réévaluer nos croyances sur les traitements les plus susceptibles d’aider les patients sur la base des dernières découvertes de la recherche. Cette approche peut nous aider à prendre de meilleures décisions même lorsque nous disposons d’informations limitées.

La première étape de l'utilisation de la théorie de la décision bayésienne consiste à établir une distribution de probabilité a priori. Cela peut être fait à l'aide de données historiques ou d'opinions d'experts. Une fois la probabilité a priori établie, de nouvelles informations peuvent être incorporées dans le calcul en ajustant les probabilités associées à chaque résultat. Cela permet des estimations actualisées du risque et de la récompense qui tiennent compte à la fois de l'expérience passée et des connaissances actuelles.

Une application de la théorie de la décision bayésienne dans les soins de santé consiste à aider les prestataires à choisir entre les options de traitement lorsqu'il existe plusieurs options viables. Dans de tels cas, l'analyse bayésienne peut être utilisée pour calculer la probabilité que chaque traitement réussisse, en tenant compte à la fois des données passées et des informations actuelles sur l'état du patient. Cela permet aux fournisseurs de soins de faire des choix éclairés en fonction de ce qui a les meilleures chances de succès, plutôt que de simplement choisir en fonction de préférences personnelles ou de suppositions. En fin de compte, la théorie de la décision bayésienne offre aux professionnels de la santé un moyen de prendre des décisions judicieuses malgré la gestion de grandes quantités d'incertitude.

1.5. Problématique, Objectifs et hypothèses

Les maladies cardiovasculaires (MC) deviennent de plus en plus l'un des problèmes de santé publique les plus urgents dans la plupart des régions du monde, y compris la population canadienne. Par exemple, en utilisant des données administratives sur la santé de dix provinces, une étude récente a estimé une prévalence de 26,5 % basée sur cinq conditions (maladie cardiovasculaire, maladies respiratoires, maladie mentale, hypertension, diabète). Une autre étude a rapporté une prévalence de 42,6 % pour la population nationale âgée de 18 ans et plus sur la base des données des dossiers médicaux électroniques (Houle and Gallani 2017).

Si les maladies cardiovasculaires ne peuvent pas être guéries, leur progression et leurs conséquences sur la qualité de vie peuvent néanmoins être diminuées grâce à l'adoption et au maintien de saines habitudes de vie, dont la pratique de l'activité physique (AP) (Lavie et al., 2015). Il est donc primordial que les professionnels de la santé soient en mesure d'évaluer et

d'intervenir afin de favoriser l'adhésion à l'AP auprès de personnes atteintes d'une MC (Houle and Gallani 2017).

Une application importante de la surveillance des patients est dans le domaine des soins d'activité physique AP. les soins d'AP utilisent divers types de moniteurs pour suivre la tension artérielle, la fréquence cardiaque, la fréquence respiratoire et d'autres signes vitaux d'un patient. Ces informations aident les cliniciens à s'assurer que le diagnostic se déroule comme prévu et que le patient ne subit aucun effet indésirable des traitements proposés.

Pour ce faire, nous menons un projet de recherche pour développer une plateforme numérique (Hybride de DME et SADC) permettant de soutenir l'intervention des professionnels de la santé, soit une application basée sur les moniteurs d'activité physique (podomètre) associé à l'intelligence artificielle. Nous avons intitulé cette application "Plateforme Opérationnelle d'analyse de Données sur la pratique de l'activité physique appliqué à la santé - projet POD-iSanté". Pour ce faire, nous avons besoin de recueillir des données à partir de podomètres et ce, dans le but d'entraîner le système d'intelligence artificielle pour qu'il puisse générer automatiquement des recommandations personnalisées adaptées à différents profils cliniques.

À notre connaissance, il n'existe aucune plateforme utilisant l'intelligence artificielle permettant de faire une évaluation de la pratique de l'AP, tenant compte à la fois de données objectives et subjectives du niveau d'AP, du niveau de motivation et des caractéristiques des individus. De plus, aucune plateforme ne permet de générer des recommandations basées sur la science à partir de ces données et ce, dans le but d'assister les intervenants dans la formulation de recommandations personnalisées auprès de personnes atteintes d'une maladie chronique. Nous souhaitons donc développer une plateforme numérique novatrice en la matière et ensuite évaluer l'efficacité de son utilisation par les professionnels de la santé ouvrant auprès de personnes atteintes de MC ou dans le but de prévenir ces maladies (Houle and Gallani 2017).

En résumé, dans ce mémoire, nous visons à proposer un système d'aide à la décision clinique utilisant l'apprentissage profond AP-SADC, un domaine de l'apprentissage automatique pour aider les cliniciens à prendre de meilleures décisions concernant l'activité physique des patients visant à améliorer leur santé. L'élément clé de notre AP-SADC proposé est l'estimation

des résultats du modèle d'apprentissage en profondeur qui l'accompagne dans la phase de traitement en raison de différents facteurs. Nous nous concentrons sur le AP-SADC et la façon bayésienne de penser pour la prise de décision, et de capturer l'incertitude. Nous décrivons les objectifs, les caractéristiques, le développement et l'évaluation de notre AP-SADC. Nous présentons les types de données et les sources que nous avons utilisées pour créer le noyau du AP-SADC. Nous introduisons également l'architecture Web sur laquelle nous avons basé notre AP-SADC et les différentes normes que nous avons utilisées. Nous présenterons les bases des réseaux de neurones artificiels. Nous expliquerons différentes méthodes et algorithmes utilisés pour nous aider à surmonter certaines des lacunes et des difficultés auxquelles nous avons été confrontés.

Chapitre 2 – Systèmes d'aide à la décision clinique

2.1. Introduction

Les SADC sont utilisées dans les soins de santé depuis de nombreuses années, mais ils sont devenus de plus en plus populaires ces dernières années en raison de l'essor des DME. Les DME permettent aux SADC d'accéder aux antécédents médicaux d'un patient et à d'autres données pertinentes, ce qui les aide à fournir des recommandations plus précises. Il existe de nombreux types de SADC, mais tous partagent le même objectif : aider les professionnels de la santé à prendre de meilleures décisions pour leurs patients. Certains SADC sont conçus spécifiquement pour certains types de traitements ou de maladies, tandis que d'autres sont des outils plus généraux qui peuvent être utilisés dans n'importe quelle situation.

Dans ce chapitre, nous discuterons des types de SADC et des sources de données pour améliorer la santé du patient en fonction de ses activités physiques, ainsi que du processus bayésien de prise de décision dans le domaine clinique.

2.2. Les systèmes d'aide à la décision clinique

Les systèmes d'aide à la décision clinique (SADC) sont des approches de systèmes électroniques conçues pour aider les cliniciens à prendre une meilleure décision visant principalement à améliorer les soins de santé et la sécurité du patient (Berner 2016). Il fonctionne en utilisant les données qui représentent l'état d'un patient qui ont été recueillies sur une période plus une connaissance clinique antérieure pour générer des évaluations ou des recommandations spécifiques au patient présentées aux cliniciens pour aider à la prise de décision (Bright, et al. 2012).

Habituellement, les suggestions du SADC sont combinées avec l'expertise et les connaissances du clinicien pour favoriser les meilleures décisions de santé du patient. L'évolution des systèmes matériels et logiciels au cours des dernières années a contribué à intégrer davantage le SADC dans le système de santé. Les applications Web sont considérées comme un moyen de mettre en œuvre une SADC à l'aide de DME pour des résultats précis. La flexibilité et l'agilité de ces applications en font une solution largement répandue. Des appareils tels que des

ordinateurs, des smartphones ou des tablettes peuvent être utilisés pour accéder de manière sécurisée au SADC de n'importe où à l'aide d'un navigateur Web (Sutton et al., 2020).

En général, les SADC sont classées en deux catégories principales, basées sur la connaissance ou non basées sur la connaissance. Les systèmes basés sur les connaissances suivent le paradigme des déclarations SI-ALORS qui sont créés à l'aide d'une expertise, de preuves et de spécifications relatives aux patients, ou à l'aide de ressources basées sur la littérature. Pour utiliser ces systèmes, une donnée initiale est nécessaire pour évaluer l'état d'un patient, en passant d'abord par un ensemble de règles prédéfinies se traduisant par un ou un ensemble de sorties et d'actions. Les systèmes non basés sur la connaissance englobent principalement les méthodes statistiques complexes. L'intelligence artificielle (IA) ou spécialement l'apprentissage automatique (ML) et les méthodes d'apprentissage en profondeur (AP) sont la solution idéale pour fournir une réponse précise et traiter des problèmes plus difficiles (Sutton et al., 2020).

Avec l'utilisation des données, les systèmes d'aide à la décision clinique d'apprentissage en profondeur AP-SADC offrent des résultats préférables lorsqu'il s'agit de problèmes liés aux Big Data ou d'un petit ensemble de données ou de problèmes qui ont une représentation complexe. Contrairement au premier type, nous essayons essentiellement de réduire l'influence du clinicien et de nous concentrer principalement sur les données cliniques, ce qui fait de ces dernières une partie importante d'un AP-SADC et présente la disponibilité des données comme l'un des premiers problèmes à rencontrer dans ces types de systèmes. Faire des recommandations à l'aide de AP-SADC est souvent ambigu et manque de justifications en raison que les réseaux de neurones artificiels (ANN) sont critiqués comme étant des boîtes noires, parce qu'il n'y a pas d'explication satisfaisante de leurs résultats qu'ils produisent. Nous pouvons avoir un modèle parfait mais il n'est pas nécessaire de comprendre pourquoi nous obtenons les résultats proposés par le modèle, de sorte que l'explicabilité est souvent inutile.

Une façon correcte d'utiliser le SADC est de le faire assister par le clinicien à prendre une décision, il ne doit donc pas être considéré comme un remplacement du rôle du clinicien dans le flux de travail clinique. Ils ont la caractéristique de boîtes noires pour le manque de raisonnement dans leurs résultats. Ainsi, le flux de travail d'un AP-SADC simple commence par recevoir un ensemble spécifié de données cliniques, il peut être sélectionné par le AP-SADC lui-même ou le clinicien, puis le AP-SADC produit des recommandations qui sont analysées et rationalisées

souvent par le clinicien pour aider à prendre une meilleure décision concernant la santé d'un patient (Mahadevaiah et al., 2020).

2.2.1. Types de systèmes d'aide à la décision clinique

2.2.1.1. Systèmes basés sur la connaissance

Les systèmes experts sont considérés comme le premier type de SADC moderne basé sur la connaissance. L'évolution de l'industrie du logiciel a conduit à une plus grande adoption du SADC lorsqu'il est utilisé dans le domaine de la médecine pour améliorer la qualité des soins de santé. Trois composants forment le SADC basé sur les connaissances, la base de connaissances, le moteur d'inférence et l'interface utilisateur. La base de connaissances comprend les connaissances de l'expert et les informations compilées à partir des expériences passées, et se présente souvent sous la forme de règles SI-ALORS. Le moteur d'inférence est considéré comme le mécanisme de raisonnement et contient les formules qui relient les données recueillies auprès des patients et les règles prédéfinies. Vient ensuite l'interface utilisateur pour faciliter le processus de saisie des données des patients dans le système par les utilisateurs et produire les résultats pour aider à la prise de décision (Berner 2016).

La collecte des données peut être faite par l'utilisateur (le clinicien) ou à l'aide de systèmes de DME, ces derniers collectent les données après que le patient les a portées dans un délai spécifié et enregistrées automatiquement dans le système, ou manuellement par le clinicien.

L'entrée au SADC peut être les symptômes ou les données d'activité physique, déjà dans le système ou peuvent être saisies en place par le clinicien ou importées du DME. Après cela, avec l'aide de la base de connaissances, le moteur d'inférence propose un ensemble de diagnostics pour aider le clinicien à prendre une meilleure décision pour améliorer la santé du patient.

La sortie du SADC après son utilisation peut prendre la forme de recommandations ou d'alertes. Les recommandations peuvent être utilisées pour fournir une aide à la décision ou suggérer des traitements potentiels. Les alertes peuvent être utilisées pour empêcher un dysfonctionnement ou des erreurs en cours dans le système, comme il peut être utilisé comme hub de notifications. La figure suivante présente les différentes composantes d'un système basé sur la connaissance:

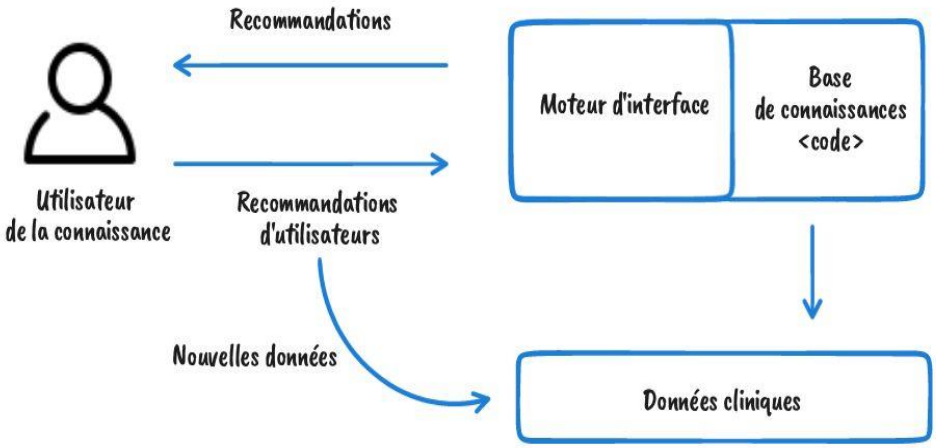


Figure 1 Système basé sur la connaissance.

2.2.1.2. Systèmes non basés sur les connaissances

Les SADC non basés sur les connaissances sont une forme de SADC qui exploite les données sur les connaissances comme celui que nous avons vu dans le système précédent en utilisant des méthodes d'apprentissage automatique. L'inférence statistique est l'élément central de ces méthodes qui ont la capacité d'apprendre des données cliniques passées et des expériences qui se sont produites pendant la phase de diagnostic (Berner 2016).

Les réseaux de neurones artificiels sont les méthodes les plus connues utilisées dans le domaine de l'apprentissage automatique. Fait de nœuds connectés appelés neurones qui tentent d'imiter les parties d'un neurone biologique telles que les synapses, les dendrites, les corps cellulaires et les axones à l'aide de modèles mathématiques simplifiés, un réseau de neurones artificiels possède une structure de trois parties; couche d'entrée, couches cachées et couche de sortie, avec un signal unidirectionnel entre les neurones. La couche d'entrée est la passerelle des données et la couche de sortie est chargée de fournir les résultats, tandis que les couches cachées traitent les entrées en essayant de représenter les données de la meilleure façon.

Nous remarquons que la base de connaissances et le réseau de neurones artificiels ont une tâche similaire en général, alors que la base de connaissances demande à un tiers d'acquérir les connaissances comme un clinicien, le réseau de neurones artificiels est responsable de

comprendre les données du patient (signes et symptômes) pour trouver les meilleurs schémas qui conduisent au diagnostic proposé.

L'entraînement d'un réseau de neurones artificiels est une opération itérative dans laquelle le réseau reçoit des données de manière incrémentielle et fait des prédictions qui peuvent être corrigées en utilisant la vraie sortie et en ajustant le poids dans la connexion entre les neurones.

Un avantage évident pour le SADC non basé sur les connaissances est qu'il élimine le besoin d'une base de connaissances et de l'expertise des cliniciens. Un autre avantage est la capacité de traiter une énorme quantité de données ou, dans certaines situations, le manque de données dont nous pouvons tirer parti de la qualité plutôt que de la quantité. Cela n'élimine pas certains inconvénients, comme l'interprétabilité des résultats compte tenu de la nature stochastique de l'ANN, ou encore le temps d'entraînement peu pratique qui peut consommer des heures, des jours ou des semaines dans certains cas. La figure suivante présente les différentes composantes d'un système non basé sur la connaissance:

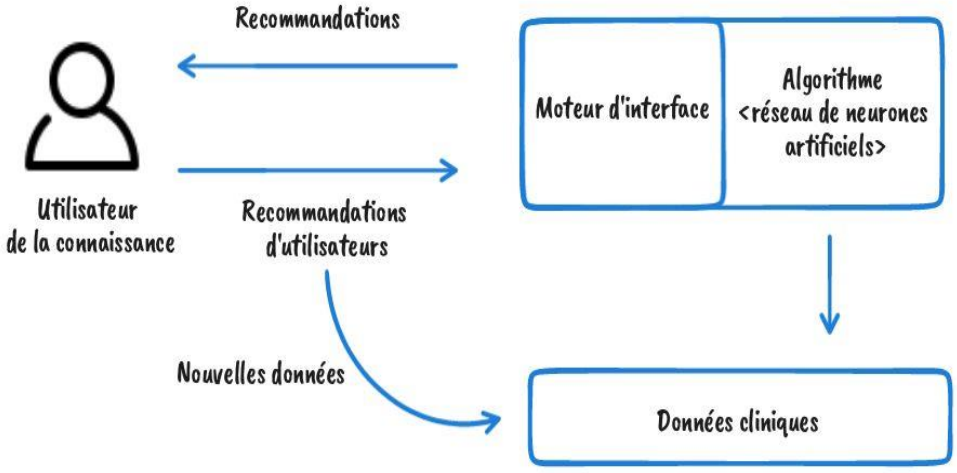


Figure 2 Système non basé sur les connaissances.

2.3. Activité physique

L'Activité Physique (AP) est définie comme tout mouvement corporel produit par les muscles squelettiques qui entraîne une dépense énergétique. L'AP peut être structurée ou

accessoire. L'AP est dite structurée si elle est planifiée à l'avance pour promouvoir la forme physique et la santé. Elle est dite accessoire lorsqu'elle n'est pas planifiée comme le fait d'effectuer notre travail quotidien ou de faire nos courses. L'AP a quatre dimensions (Strath et al., 2013):

- Mode ou type d'activité, elle définit l'activité pratiquée (ex : marche) ou peut également être utilisée pour définir les types d'activité dans le cadre physiologique et biomécanique (ex : activité aérobie versus anaérobie, résistance ou entraînement en force).
- Fréquence d'exécution d'une activité, c'est le nombre de sessions d'une activité spécifique par période.
- Durée de l'activité : C'est la durée en minute ou par heure dans tout autre intervalle de temps comme un jour ou une semaine.
- Intensité de la pratique d'une activité : c'est le coût métabolique de la pratique d'une activité, il peut être mesuré comme une quantité d'énergie nécessaire. L'intensité de l'AP peut être modérée ou vigoureuse.

Les AP sont classées en 4 domaines pour représenter leur lieu d'occurrence :

- Professionnel : tâches liées au travail comme marcher et soulever des objets.
- Domestique : activités liées à la maison comme les travaux ménagers et les soins infirmiers.
- Transport : activités associées au transport, comme se tenir debout dans un bus.
- Temps libre : temps libre ou récréatif comme les loisirs et les sports.

Il existe deux catégories de méthodes pour évaluer l'activité physique, les méthodes subjectives et les méthodes objectives. Les méthodes subjectives dépendent d'un individu en tant qu'acteur principal pour collecter des données et enregistrer des activités. Les méthodes objectives utilisent des appareils portables et des trackers pour collecter les données de l'activité.

2.4. Méthodologie

Pour que notre plateforme génère des recommandations personnalisées visant à augmenter le niveau d'activité physique, en tenant compte de la motivation de chacun, certaines données doivent d'abord être collectées au niveau de l'individu. Pour cela, lors de deux visites initiales l'intervenant tentera de recueillir les informations du patient.

Au départ, les participants ont reçu une lettre d'information pour les aider à comprendre exactement ce que leur éventuelle participation au projet implique afin qu'ils puissent prendre une décision éclairée à ce sujet.

Durant la première visite, le patient se présente à la clinique 15 minutes avant l'heure de son rendez-vous avec l'intervenant afin de compléter les informations initiales du profil (nom, prénom, adresse résidentielle, sexe, revenu familial, scolarité, milieu de vie, etc.) (Pettigrew 2021).

L'intervenant(infirmier/re) commence la rencontre, ouvre une session et complète les informations relatives à l'historique médical (antécédents, pharmacothérapie, etc.). Il collecte ensuite les données de l'examen clinique à l'aide d'un questionnaire (symptômes, tabagisme) et d'un examen physique (cardiovasculaire, anthropométrie).

L'intervenant répond aux questions et aux préoccupations du patient. Il effectue de l'enseignement et conseil par rapport à la thérapie pharmaco et le changement et l'adoption de saines habitudes de vie (nutrition, stress, intro activité physique) selon les besoins.

Après avoir introduit l'activité physique comme étant une composante essentielle au traitement, L'intervenant demande au patient de compléter le questionnaire BREQ (auto-administré) permettant de connaitre le type de motivation.

Si le patient présente une motivation extrinsèque ou intrinsèque envers la pratique de l'AP, l'intervenant lui présente un moniteur d'activité physique et lui propose de le porter pour au moins 2 semaines et ce, du matin au soir afin de connaitre son niveau d'activité physique quotidien. Il lui remet également un journal de bord afin de noter les informations pouvant être pertinentes à l'analyse des données (ex : pratique d'activités physiques pour lesquelles le podomètre n'est pas sensible comme le ski de fond ou le vélo par exemple, ou encore s'il existe de conditions qui modifient temporairement les habitudes comme un changement aigu de son état de santé ou encore un voyage). Finalement, il lui explique comment porter le moniteur et donne les conseils d'usage.

Le patient sera invité à revenir à la clinique pour un 2ème rendez-vous afin de connaitre les résultats (rapport du podomètre) et d'établir un plan d'action personnalisé.

Durant la deuxième visite, l'intervenant ouvre la session et vérifie s'il y a eu des changements au niveau de son état de santé ou de sa pharmacothérapie depuis la dernière rencontre et entre les données s'il y a lieu. Elle collecte ensuite les données de l'examen clinique à l'aide d'un questionnaire (symptômes, tabagisme) et d'un examen physique (cardiovasculaire, anthropométrie). L'examen clinique est fait à chaque visite car ces données peuvent varier et plus on a de données, meilleure sera l'évaluation.

L'intervenant récupère le moniteur d'activité physique et vérifie si le patient a des préoccupations particulières en lien avec le port du moniteur d'activité physique et consulte le journal de bord.

L'intervenant complète le questionnaire GPAQ (Ainsworth et al., 2015) avec le patient en lui précisant que ces informations aideront à compléter l'évaluation de son niveau d'activité physique.

L'intervenant propose au participant d'identifier les facilitants et les barrières à la pratique d'activité physique avec les questions à choix multiples proposées sur l'interface web. L'intervenant lui demande également le degré de confiance avec lequel il croit pouvoir intégrer l'activité physique dans son quotidien et le saisit dans le logiciel. Pendant le temps que le patient complète les informations, l'intervenant analyse les données du moniteur d'activité physique qui sont automatiquement enregistrés dans notre base de données.

Avec la totalité des informations rassemblées, l'application génère un rapport comprenant les résultats de l'examen clinique (visite 1 et 2), les données du moniteur d'activité physique et les données du questionnaire GPAQ. L'application génère également une liste de recommandations en lien avec la pratique d'activité physique qui devrait être atteinte par le patient d'ici la prochaine visite. L'intervenant s'assure que les recommandations fournies sont en lien avec les attentes du patient et sont atteignables selon lui. Il remet alors une copie du rapport et des recommandations au patient (format papier et/ou envoie par courriel) et dépose une copie de ce rapport dans le dossier du patient (DME).

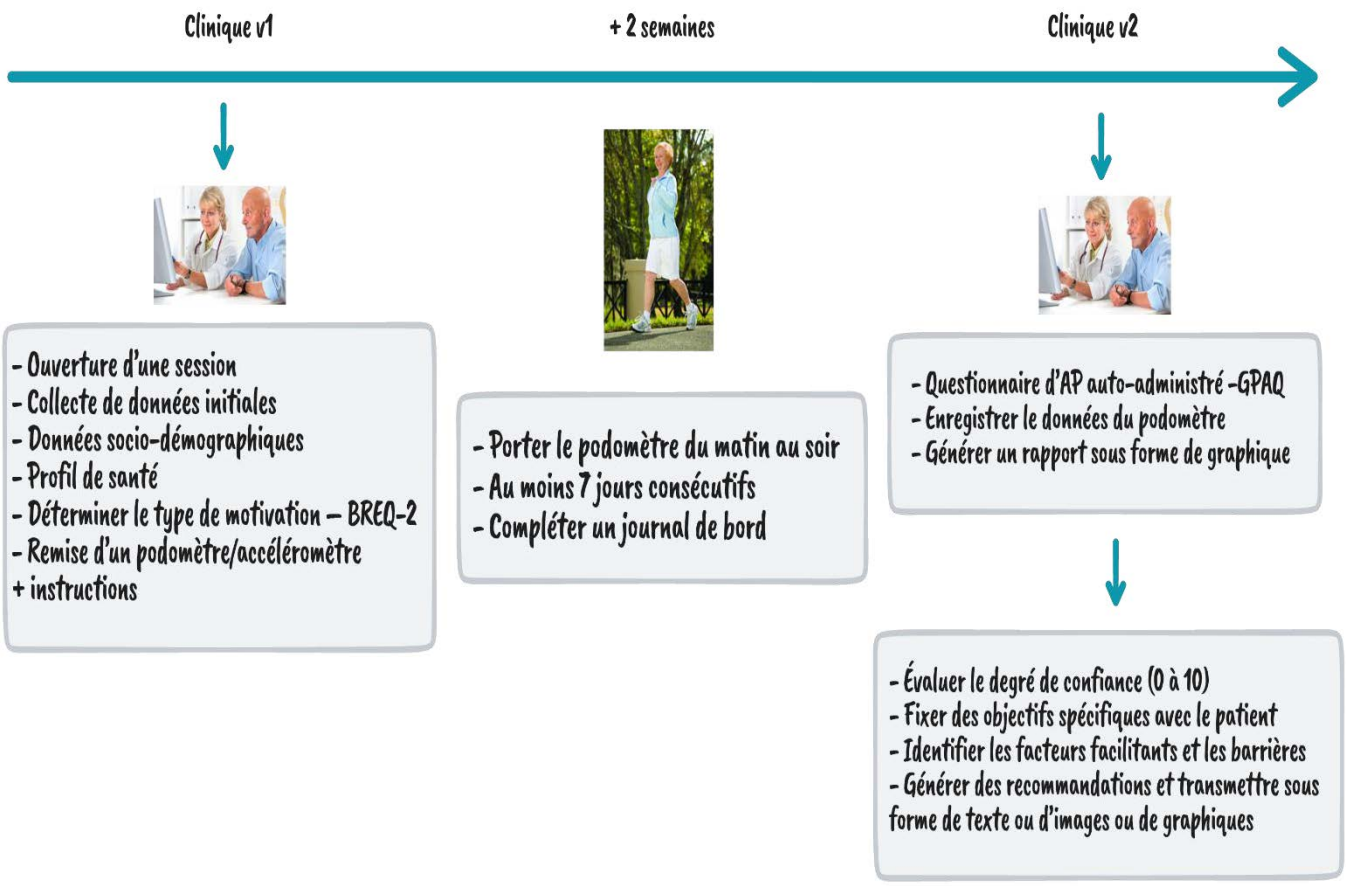


Figure 3 Aperçu général de l'utilisation de POD-iSanté.

2.4.1. Les critères

La population cible est composée d'hommes et de femmes âgées de plus de 30 ans atteints d'une maladie cardiovasculaire. Une méthode d'échantillonnage aléatoire simple sera utilisée. Les critères d'inclusion et d'exclusion pour le recrutement des patients sont présentés ci-dessous (Houle and Gallani 2017).

Les critères d'inclusion sont d'avoir:

- Un diagnostic de maladie cardiovasculaire et présenter au moins un facteur de risque modifiable (ex : hypertension artérielle, dyslipidémie, obésité, diabète, etc.),
- Une capacité à marcher au moins 10 minutes consécutives sans aide,
- Avoir une capacité de s'exprimer en français par écrit et oralement,
- Une capacité à naviguer sur Internet pour une utilisation de base,

- Accès à Internet soit à domicile ou ailleurs (ex : bibliothèque, café internet),
- Un niveau de littératie en santé permettant d'intégrer deux renseignements ou plus, de comparer et de distinguer des renseignements et d'interpréter des graphiques simples,
- L'intention d'adhérer à une pratique d'activité physique régulière.

Les critères d'exclusion sont de:

- Présenter une contre-indication à la pratique de l'activité physique selon les critères de l'*American College of Sport Medicine* (2013),
- Avoir subi des pontages coronariens au cours des derniers six mois,
- Souffrir d'un cancer ou de toute autre maladie invalidante,
- Ne pas avoir un environnement accessible permettant de marcher de façon sécuritaire,
- Être un membre de la famille ou avoir une relation client-professionnel avec un membre de l'équipe de recherche.

Les personnes sélectionnées qui répondent à ces critères seront invitées à participer au projet. Les candidats qui accepteront seront ensuite répartis aléatoirement entre le groupe expérimental et le groupe contrôle à l'aide d'une table de randomisation générée par ordinateur. Le groupe contrôle recevra un suivi usuel, tandis que le groupe expérimental recevra l'intervention expérimentale (ces interventions sont décrites ci-dessous). La taille d'échantillon visée est de 60 sujets au total.

Données (Visite 1)	
Données socio-démographiques	- Âge, Sexe - Statut civil - Lieu de résidence, emploi - Statut socio-économique, niveau de scolarité
Profil de santé	- ATCD médicaux, VO_2peak - Pression artérielle et FC de repos - Taille et poids, circonférence de taille - Bilan lipidique

	- Glycémie à jeun et HbA1c
Déterminer le type de motivation	- Questionnaire BREQ
Données (Visite 2)	
Niveau d'activité physique – Questionnaire GPAQ	- 16 questions dont : ▪ Activités au travail (6 questions) ▪ Déplacements (3 questions) ▪ Activités de loisirs (6 questions) ▪ Comportement sédentaire (1 question)
Niveau d'activité physique – Questionnaire GPAQ	- Nombre moyen de pas quotidiens - Nombre moyen de temps passé à intensité moyenne à vigoureuse - Nombre moyen de temps passé à activités sédentaires
Degré de confiance	- Quel est votre degré de confiance sur une échelle de 0 à 10 ▪ À augmenter son niveau d'AP ▪ À réduire la durée du temps sédentaire
Objectifs spécifiques	- Quel est votre objectif d'ici notre prochaine rencontre ?
Barrières et facteurs facilitant	- Qu'est-ce qui pourrait vous aider à atteindre votre objectif ? - Qu'est-ce qui pourrait vous empêcher d'atteindre votre objectif?

Tableau 1 Description des données (Pod-iSante).

2.4.2. Modèle

L'apprentissage supervisé est un type d'apprentissage automatique où un algorithme est "formé" avec un ensemble d'entrées et de sorties connues. L'algorithme utilise ensuite ces données d'apprentissage pour prédire la sortie de nouvelles entrées inconnues. Les algorithmes d'apprentissage supervisé sont divisés en deux catégories : la régression et la classification. Les algorithmes de régression sont utilisés pour prédire une valeur continue, telle que la tension artérielle d'un patient. Les algorithmes de classification sont utilisés pour prédire l'une de plusieurs valeurs discrètes, telles que le type d'activité physique. Dans notre projet, nous avons choisi le réseau de neurones comme algorithme d'apprentissage. Les réseaux de neurones sont

utilisés pour modéliser des modèles complexes de données et peuvent être entraînés pour reconnaître des modèles dans des images, du texte et d'autres types de données. Le processus d'entraînement d'un réseau de neurones produit un modèle qui représente au mieux les données.

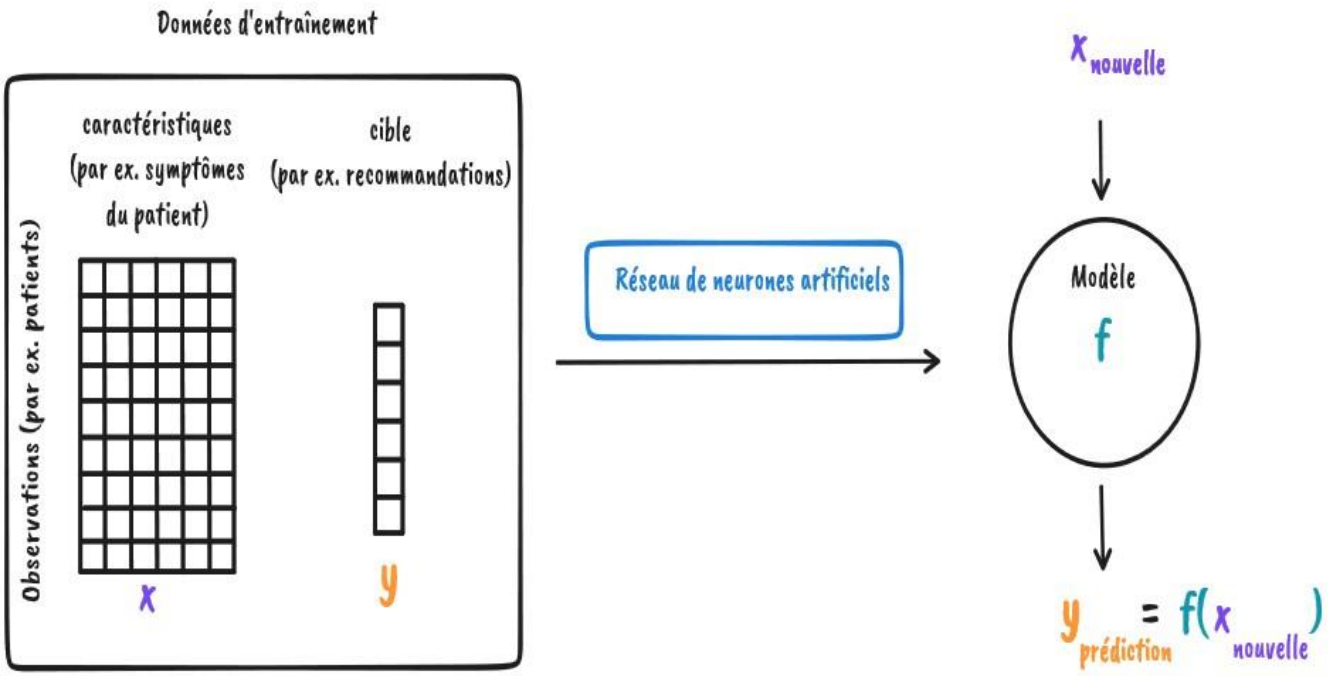


Figure 4 Le cœur d'une SADC, le modèle.

2.5. Source de données

Les données sont un élément essentiel pour prendre des décisions éclairées. Deux types de sources de données sont les sources objectives et les sources subjectives. La source objective des données est une information qui a été recueillie de manière systématique et impartiale, tandis que la source subjective des données reflète l'opinion ou le point de vue personnel de la personne qui les fournit.

2.5.1. Évaluations de l'AP avec des moniteurs portables

2.5.1.1. Podomètres

Les podomètres sont connus comme des dispositifs de comportement de marche en suivant le nombre de pas grâce à un capteur spécial. Normalement, les enregistrements de pas ont une durée de vie d'une journée à une semaine en fonction des caractéristiques du podomètre

comme la durée de vie de la batterie ou la taille de stockage. Ils sont capables de suivre le nombre de pas ou la distance totale, et également d'estimer le type d'intensité et le temps. Le prix joue un rôle majeur dans la précision et les caractéristiques d'un podomètre. Un podomètre peut être placé dans le poignet ou la hanche tandis que le résultat le plus précis est obtenu lorsqu'il est placé dans la cheville ou gardé dans la poche (Ainsworth et al., 2015).

2.5.1.2. Accéléromètres

Les capteurs à accéléromètres utilisés pour estimer l'activité physique fournissent une mesure des accélérations du corps pendant le mouvement et ont l'avantage de capturer la fréquence, la durée et l'intensité du mouvement physique de manière horodatée. La plupart des études d'étalonnage existantes reposent sur une métrique d'intensité sans unité ou sur des comptages, puis appliquent des seuils aux données résumées pour produire la durée et la fréquence de l'AP en intensités sédentaires, légères, modérées et vigoureuses (Ainsworth et al., 2015).

Les accélérations sont mesurées en fonction des plans des mouvements du corps. Les accéléromètres ont la capacité de stocker des données pendant plusieurs semaines. Principalement, les accéléromètres enregistrent l'accélération et la décélération du corps pendant une AP, puis nous utilisons l'unité de comptage pour décrire les données. L'unité de comptage dépend de l'appareil et non d'une définition normalisée en raison des différences entre les accéléromètres comme la précision et les algorithmes intégrés. La plupart des études d'étalonnage existantes reposent sur une métrique d'intensité sans unité ou sur des comptages, puis appliquent des seuils aux données résumées pour produire la durée et la fréquence de l'AP en intensités sédentaires, légères, modérées et vigoureuses. Les accéléromètres sont souvent portés à la hanche, au poignet ou à la cheville.

2.5.1.3. Moniteurs de fréquence cardiaque

Les moniteurs de fréquence cardiaque (MFC) enregistrent les signaux électriques du cœur provenant du stress cardiorespiratoire pendant une AP. MFC peut produire de fausses données dans une activité de faible intensité à la suite de l'intervention de différents facteurs comme une cause alimentaire ou un état émotionnel. D'autre part, MFC reçoit une augmentation significative des signaux lors d'une AP d'intensité modérée à vigoureuse. Le MFC peut être utilisé de manière

interchangeable avec les accéléromètres pour les mesures d'activités non ambulatoires comme le cyclisme et la natation (Ainsworth et al., 2015).

2.5.2. Évaluation de l'activité physique avec des outils d'auto-évaluation

2.5.2.1. Questionnaires globaux

Les questionnaires globaux sont des questionnaires courts comportant d'un à quatre items permettant de recueillir les informations essentielles sur le niveau d'AP d'une personne. Elle peut être spécifique à un ou à un ensemble de domaines. Il permet de classer les individus en évaluant l'adéquation de chacun à une activité physique particulière (Ainsworth et al., 2015).

2.5.2.2. Questionnaires de rappel court

Les questionnaires d'activité physique à rappel court exigent que l'individu réponde à 7 à 20 questions sur les activités physiques qui sont effectuées dans un intervalle de temps compris entre une semaine et un mois. Les questions peuvent porter soit sur la mesure de l'intensité de l'AP, soit sur les domaines de l'AP. Il peut également être utilisé pour identifier le changement de comportement en matière d'activité physique. Les questionnaires courts de rappel sont notés en fonction des réponses d'un individu. Un score numérique plus élevé indique un niveau élevé d'intensité de l'AP. Cela peut être fait par l'individu lui-même ou par un enquêteur comme un professionnel. Des exemples de questionnaires courts de rappel sont GPAQ et BREQ (Ainsworth et al., 2015).

2.5.2.3. Questionnaires quantitatifs de rappel des antécédents

Les questionnaires quantitatifs de rappel des antécédents sont administrés par l'intervieweur avec 60 questions ou plus. Ils sont utilisés pour rappeler l'intensité, la durée et la fréquence d'une AP effectuée sur une longue période spécifique comme un mois ou une année, ou la durée de vie de l'individu. Tout comme les questionnaires de rappel court, les questionnaires de rappel quantitatif des antécédents sont notés de la même manière et sont notés de manière similaire en faisant la moyenne des unités par jour ou par an. Leur principal avantage est la capacité d'aider à évaluer l'implication de l'AP sur la morbi-mortalité (Ainsworth et al., 2015).

2.6. Prise de décision dans la pratique médicale

2.6.1. Prise de décision médicale

La pratique de la médecine est complexe et en constante évolution. Afin de fournir les meilleurs soins possibles aux patients, les médecins doivent prendre des décisions fondées sur les données les plus récentes. Cela peut être difficile, car de nouvelles connaissances et informations sont constamment publiées et il peut être difficile de suivre toutes les dernières directives.

Une décision importante que les médecins doivent prendre est le choix de tests ou de procédures pour leurs patients. Il existe souvent plusieurs options disponibles, chacune avec ses propres risques et avantages, il peut donc être difficile de savoir quelle option est la meilleure dans une situation particulière. Les médecins doivent peser le pour et le contre de chaque option, puis décider ce qui est dans le meilleur intérêt de leur patient.

Une autre décision importante à laquelle les médecins sont confrontés est la prescription ou non de médicaments. Encore une fois, de nombreux facteurs doivent être pris en compte lors de la prise de telle décision, tels que les effets secondaires potentiels du médicament et la probabilité que le médicament aide à guérir ou à améliorer les symptômes. Les médecins doivent également tenir compte des allergies, des interactions médicamenteuses ou d'autres problèmes de santé que leur patient peut avoir avant de prescrire un médicament. La prise de décision dans la pratique médicale peut être difficile, mais elle est essentielle pour fournir des soins de qualité aux patients.

2.6.2. Incertitude dans la prise de décision médicale

L'incertitude a de nombreuses origines, car elle peut provenir de l'absence des connaissances requises ou du manque d'informations qualitatives. Elle peut également être le produit d'une mauvaise compréhension des données ou des rapports des patients (Dhawale et al., 2017).

Selon la théorie des jeux (Charles A Kamhoua et al., 2021), Il faut prendre en considération tous les résultats probabilistes possibles pour aider à prendre la décision optimale. Ainsi, toutes les parties du système sont considérées comme des acteurs importants pour décider du degré d'incertitude afin d'avoir une bonne perception de la décision que nous prenons. L'incertitude

empêche le clinicien de porter un bon jugement global sur un événement et de prendre les bonnes mesures.

Les incertitudes dans le domaine médical apparaissent généralement lors de la phase de diagnostic lorsqu'il s'agit de trouver les solutions idéales, ou dans la prédiction de l'efficacité du traitement. Un moyen efficace de réduire l'incertitude est d'exiger plus de données.

Habituellement, l'incertitude apparaît soit dans le processus de diagnostic pour déterminer l'état du patient et les symptômes ou l'état d'une maladie, soit peut survenir lors du traitement médical à la suite d'un diagnostic médical visant à améliorer un problème de santé, ou peut également survenir dans le processus de prédire l'efficacité d'un traitement ou l'effet secondaire d'un traitement.

Les sources d'incertitude des patients sont très larges et peuvent provenir de multiples raisons comme elles peuvent également être contrôlées et limitées. L'incapacité du patient à fournir des données historiques sur sa santé peut être l'une des sources d'incertitude, comme cela peut aussi être l'ambiguïté du traitement fourni et le manque d'information, ou l'absence de désir d'effectuer un traitement spécifique. D'autres facteurs peuvent accroître l'incertitude, tels que les antécédents médicaux du patient, son âge et son niveau d'instruction. En éduquant le patient et en communiquant et expliquant les méthodes et les techniques utilisées par le clinicien, nous visons à réduire l'incertitude du patient.

Les cliniciens peuvent également contribuer à augmenter le degré d'incertitude en omettant de fournir des explications claires au patient afin de recueillir des données ou afin d'obtenir le meilleur résultat du traitement proposé. Cela pourrait également provenir de limitations dans ses connaissances médicales personnelles dûe à la non mise à jour suivant le rythme des progrès médicaux et les derniers développements dans le domaine. Une autre source d'incertitude provenant des données sont des capteurs et des appareils qui enregistrent les différentes mesures de la santé des patients (Helou et al., 2020).

L'incertitude informationnelle provenant des données que nous avons collectées peut-être réduite en acquérant plus de données et en trouvant et en découvrant ses sources. Contrairement à l'incertitude informationnelle, il est difficile, voire impossible, de réduire l'incertitude intrinsèque. Deux types d'incertitude sont l'incertitude épistémique et l'incertitude

aléatoire. Le premier découle du manque de connaissance des informations qui affectent le résultat d'un événement, et le second représente le caractère aléatoire des données dont nous disposons (Vounatsos 2019). Les incertitudes épistémiques et aléatoires peuvent être capturées en utilisant la compréhension bayésienne de la théorie des probabilités.

2.6.3. Décision médicale bayésienne

Les statistiques bayésiennes sont basées sur le théorème de Bayes (Berrar, 2018), fruit des travaux de Thomas Bayes avec l'aide de Richard Price, généralisé plus tard par Pierre LaPlace. L'analyse bayésienne dans le domaine clinique présente de nombreux avantages pour améliorer la santé des patients. L'approche bayésienne et les avancées scientifiques dans le domaine de la santé suivent le même processus d'apprentissage à partir des expériences et d'apprentissage incrémental du passé qui expliquent la forte capacité du domaine de la santé à adopter les normes bayésiennes.

Selon le théorème de Bayes :

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)}$$

Où :

- $P(H|E)$ est la probabilité a posteriori qu'une hypothèse H soit vraie compte tenu de la preuve E.
- $P(E|H)$ est la probabilité de la preuve étant donné que l'hypothèse est vraie.
- $P(H)$ est la probabilité a priori.
- $P(E)$ est la probabilité de la preuve.

Tout d'abord, il y a un essai clinique suivi d'un ensemble d'analyses statistiques des résultats pour donner un résumé sur l'efficacité du traitement. Vient ensuite l'analyse bayésienne pour permettre d'évaluer l'effet du traitement après l'essai. L'analyse bayésienne aboutit à un jugement ou à une perception raisonnable concernant le traitement (connu sous le nom de probabilité a posteriori), également à l'appui du traitement sur la base des preuves et des données après l'essai (connu sous le nom de vraisemblance), ainsi qu'un point de vue global sur l'effet du traitement (connu sous le nom de probabilité a priori). La probabilité a posteriori est le résultat du théorème de Bayes en évaluant la vraisemblance et les nouvelles preuves issues de nouveaux essais par rapport aux connaissances antérieures et passées.

La prise de décision bayésienne est la décision qui a été prise en tenant compte de l'analyse bayésienne représentée dans le calcul de la probabilité a posteriori sur un ensemble fini de résultats possibles compte tenu de la probabilité a priori des informations passées et des nouvelles preuves (Gleason & Harris, 2019).

La probabilité a posteriori est la probabilité d'un événement tel qu'une douleur au genou gauche ou un patient en surpoids, étant donné que de nouvelles informations ou preuves ont été recueillies dans la phase de diagnostic, par exemple sur la probabilité de cet événement. La probabilité a posteriori est calculée en multipliant la probabilité a priori de cet événement (probabilité que l'événement soit vrai avant que la nouvelle information ou preuve ne soit obtenue et basée sur nos connaissances et nos convictions) et la fraction qui représente la probabilité de l'événement (représente la nouvelle information provenant de la nouvelle preuve) et la marginalisation (probabilité que la nouvelle information soit vraie). Cette formule peut s'écrire:

$$P(\text{Événement}|\ \text{Preuve}) = \frac{P(\text{Événement})P(\text{Preuve}|\text{Événement})}{P(\text{Preuve})}$$

Dans la prise de décision bayésienne pour les diagnostics médicaux, l'événement représente la présence réelle d'un cas de maladie chez un patient, et la preuve est le résultat d'un test de diagnostic pour que l'événement soit vrai. Ainsi lorsqu'un clinicien veut prendre une décision, il va essayer de trouver la meilleure estimation pour un paramètre inconnu tel que la probabilité de l'état de maladie puisse être vraie chez un patient. Sur la base de l'approche bayésienne, comme exigence, une hypothèse selon laquelle le clinicien a des connaissances préalables et des données historiques doit être faite pour déterminer la distribution de probabilité a priori. Après cela, le clinicien cherchera une meilleure compréhension du paramètre inconnu et une meilleure représentation en collectant plus de données et en les combinant avec la compréhension préalable. Si les nouvelles informations n'étaient pas précises et ambiguës, l'a priori pèsera plus dans la détermination de la probabilité a posteriori, sinon les nouvelles données pèseront plus.

Deux principales approches connues pour l'analyse statistique sont l'approche bayésienne et la plus traditionnelle, l'approche fréquentiste. L'interprétation fréquentiste de la probabilité

est basée sur des événements aléatoires répétés, et les prédictions sont faites sur les vérités sous-jacentes de l'expérience en utilisant uniquement les données de l'expérience en cours. De plus, l'incertitude est considérée comme le résultat d'erreurs commises lors de l'échantillonnage. En revanche, l'approche bayésienne tend à injecter de la subjectivité en prenant en considération l'opinion ou la croyance pour interpréter la probabilité (Spiegelhalter et al., 2003).

La distribution a priori peut être non informative lorsque nous ne disposons pas des informations ou connaissances requises. La distribution a priori peut être changée après avoir vu les données ou plus de données. La distribution a priori peut être définie après avoir réalisé l'expérience et vu les résultats (Spiegelhalter et al., 2003). Nous pouvons utiliser des méthodes d'élicitation principalement en interrogeant des experts pour décider des distributions a priori subjectives. Les résultats antérieurs et les preuves d'études similaires peuvent également être utilisés pour définir l'antériorité. L'a priori uniforme ou non informatif peut-être considéré comme un autre moyen de sélectionner l'a priori. De plus, les distributions a priori informatives comme les distributions a priori sceptiques peuvent être utilisés en présence d'une connaissance équitable basée sur des croyances externes à l'étude (Spiegelhalter et al., 2003).

Chapitre 3 – Apprentissage profond bayésien

3.1. Introduction

L'apprentissage supervisé représente un groupe de méthodes qui tentent de trouver la relation entre les attributs d'entrée (variables indépendantes) et un attribut cible (variable dépendante). La relation est représentée dans une structure appelée modèle. Habituellement, les modèles décrivent et expliquent les phénomènes, qui sont cachés dans l'ensemble de données et peuvent être utilisés pour prédire la valeur de l'attribut cible connaissant les valeurs des attributs d'entrée. Les méthodes supervisées peuvent être mises en œuvre dans une variété de domaines tels que le marketing, la finance et la fabrication.

3.2. Apprendre à partir des données

Le but de l'inférence statistique est d'aider à prendre une décision, par exemple si un médicament est donné à un patient ou non, ou quel type d'activités convient à une tranche d'âge, etc. Deux principales inférences statistiques dominent l'aspect de l'apprentissage à partir des données ou de l'inférence, l'inférence fréquentiste et bayésienne. Ils sont considérés comme distincts les uns des autres. Alors que le fréquentiste ne dépend que des données pour faire des inférences, l'approche bayésienne intègre les connaissances antérieures aux données pour faire des inférences.

3.2.1. Inférence bayésienne

L'approche bayésienne représente nos croyances et les met à jour à l'aide de nouvelles données. L'approche bayésienne est considérée comme une alternative à l'approche fréquentiste pour faire de l'inférence en nous permettant de déterminer la probabilité des paramètres du modèle compte tenu des données. La croyance est traitée comme une probabilité dans une approche bayésienne paramétrique.

Les croyances alignées sur les axiomes de probabilité sont positives et s'intègrent à 1. Soit θ le paramètre inconnu du modèle et on lui donne une probabilité a priori $P(\theta)$ représentant les croyances subjectives. Après avoir observé les données $X = \{ X_1,...,X_n \}$, on calcule la probabilité a posteriori $P(\theta \mid X)$ et on met à jour notre a priori en utilisant le théorème de Bayes:

$$P(\theta \mid X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

Où $P(X|\theta)$ est la fonction de vraisemblance et le P(X) est appelé l'évidence. Dans le cas de paramètres discrets ou continus, pour déterminer le dominant P(X) on calcule une sommation ou intégrale sur un grand nombre de paramètres, et c'est ce qui rend le dominant intraitable. On peut écrire :

$$P(\theta|X) \propto P(X|\theta)P(\theta)$$

On fait des prédictions en calculant la distribution prédictive a posteriori:

$$\mathrm{P}(X_{nouveau}|X) = \int P(X_{nouveau}|\theta)P(\theta|X)d\theta$$

Comme nous l'avons mentionné, l'approche bayésienne est considérée comme intensive en calcul par rapport à l'approche fréquentiste. Utiliser des a priori conjugués ou approximer la probabilité a posteriori à l'aide de méthodes d'échantillonnage ou les méthodes bayésiennes variationnelles, est considérée comme une bonne solution. La principale propriété distinctive d'une approche bayésienne est la marginalisation au lieu de l'optimisation.

3.2.2. Inférence fréquentiste

Contrairement à l'approche bayésienne, la fréquentiste dépend des données en ignorant les croyances subjectives antérieures. De plus, les méthodes fréquentistes supposent que les données observées sont échantillonnées à partir d'une certaine distribution. Nous appelons cette distribution de données la vraisemblance : $P(Données|\theta)$, où θ est traité comme étant constant et le but est de trouver le θ qui maximiserait la vraisemblance. La prise en compte d'un a priori uniforme ou non informatif dans le cadre bayésien conduit à la manière fréquentiste de traiter et de comprendre les données.

3.3. Base des réseaux de neurones

3.3.1. Réseau de neurones à propagation avant

Un réseau de neurones peut être représenté comme un modèle probabiliste $\mathrm{P}(Y|X, \theta)$. Pour un problème de classification, Y est un ensemble de classes et P est une distribution de probabilités discrète. Pour un problème de régression, Y est une valeur continue et P est une

distribution de probabilités continue. En tenant compte du fait que θ (les poids du réseau de neurones) sont les paramètres du modèle que nous voulons optimiser.

Lors de l'apprentissage étant donné les données d'apprentissage $\mathrm{D} = \{X_i, Y_i\}$, l'idée est d'ajuster une distribution de probabilité à un modèle avec des paramètres inconnus. La fonction de probabilité $P(D|\theta)$ est une méthode pour mesurer l'ajustement des données compte tenu des paramètres inconnus, des poids et des biais. Suivant l'approche fréquentiste, l'objectif est d'inférer sur les paramètres en maximisant la vraisemblance $P(D|\theta)$ en utilisant l'estimation du maximum de vraisemblance (MLE).

Dans le contexte des réseaux de neurones, l'optimisation signifie généralement minimiser une seule valeur ou une fonction, donc au lieu de maximiser la vraisemblance, nous minimisons la vraisemblance logarithmique négative. Le maximum de vraisemblance nous permet, sous certaines hypothèses, de dériver une fonction objective qui convient au problème que nous avons. Par exemple, la fonction objective d'entropie croisée donne de bons résultats compte tenu d'une distribution catégorielle. Un point négatif de l'utilisation du maximum de vraisemblance est le problème de surajustement et l'incapacité à traiter de nouvelles données.

De l'autre côté, l'approche bayésienne utilise la vraisemblance parallèlement à la distribution a priori $P(\theta)$ pour calculer la distribution a posteriori $P(\theta|D)$. L'estimateur du maximum a posteriori (MAP) est une méthode utilisée pour estimer un certain nombre de paramètres inconnus, comme les paramètres d'une densité de probabilité, reliés à un échantillon donné. La fonction objective dérivée du MAP est la même que le MLE, en ajoutant le logarithme de la probabilité a priori comme terme de régularisation. Contrairement au MLE, MAP a la capacité d'empêcher le surajustement.

Les réseaux de neurones à propagation avant tentent d'approximer une fonction f^*qui lie l'entrée x à une étiquette y donc $y = f^*(x)$, en considérant le mappage $y = f(x;\theta)$. Tout au long du temps d'entraînement, trouver la valeur optimale des paramètres du modèle w est le moyen d'obtenir la meilleure approximation de la fonction.

La rétropropagation explique la méthode de calcul des gradients tandis qu'un algorithme comme la descente de gradient effectue la partie apprentissage en calculant les dérivées de la fonction objectif.

Un réseau peut être représenté par une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Une couche est composée d'un groupe de nœuds d'un point de vue architecture et d'une fonction d'activation d'un point de vue mathématique pour introduire la non-linéarité dans le modèle pour une meilleure compréhension des données. La couche d'entrée est la première couche qui rencontre les données, après que les données sont traitées dans les couches cachées, et enfin la couche de sortie.

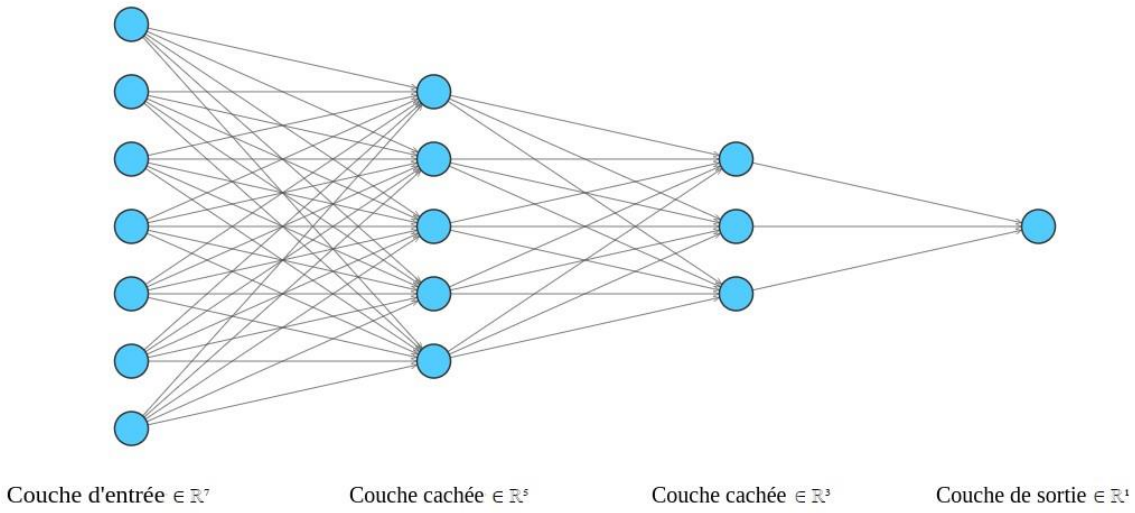


Figure 5 Exemple de réseau à quatre couches.

3.3.2. Entraînement de modèles profonds

Les réseaux de neurones peuvent être entraînés à l'aide d'un processus appelé rétropropagation. La rétropropagation est une technique d'entraînement des réseaux de neurones qui utilise l'algorithme de descente de gradient. La descente de gradient est un algorithme d'optimisation qui calcule la pente de la fonction d'erreur, ou la différence entre les valeurs prédites et les valeurs réelles des sorties du réseau neuronal. Le but de la descente de gradient est de trouver un minimum local sur cette fonction d'erreur, ce qui se traduira par de meilleures prédictions par le réseau de neurones. La rétropropagation fonctionne en propageant les erreurs vers l'arrière à travers les couches de neurones du réseau. Cela permet à chaque neurone d'apprendre comment sa sortie affecte les prédictions faites par les neurones en aval. En ajustant les poids sur les neurones individuels, la rétropropagation vise à améliorer la précision de la prédiction pour les modèles complexes de données.

3.3.3. Estimateurs ponctuels

L'estimation ponctuelle est la procédure d'utilisation des données pour estimer les meilleures valeurs individuelles d'une distribution de probabilité. Supposons que X est une variable aléatoire correspondant aux données observées et θ est un paramètre fixe à estimer. Une estimation ponctuelle pour un ensemble de données observé particulier (c'est-à-dire pour $X = x$) est alors $\hat{\theta}(x)$, qui est un estimateur d'une valeur unique fixe.

MLE maximise la vraisemblance $P(D|\theta)$ tandis que le MAP trouve la valeur qui maximise la probabilité a posteriori $P(\theta|D)$. Comme les deux méthodes vous donnent une seule valeur fixe, elles sont considérées comme des estimateurs ponctuels, contrairement à l'inférence bayésienne qui donne une distribution de probabilité complète.

Dans cette section, nous introduirons deux estimations populaires, l'estimation du maximum de vraisemblance et l'estimation maximale a posteriori (Brian J. Reich & Sujit K. Ghosh, 2019).

3.3.3.1. Estimation du maximum de vraisemblance

Pour obtenir le meilleur modèle, nous devons trouver les paramètres θ qui décrivent le résultat le plus similaire de $P(X;\theta)$ à $P(X)$, c'est-à-dire la fonction de vraisemblance. Notre objectif est de maximiser la fonction de vraisemblance, pour cela nous utilisons le principe d'estimation du maximum de vraisemblance MLE pour estimer les paramètres $\hat{\theta}$:

$$\hat{\theta}_{MLE} = argmax_{\theta}\, P(X;\theta)$$

Avec les données $X = \{ X_1, \dots, X_n \}$ est indépendant et identique distribué:

$$\hat{\theta}_{MLE} = argmax_{\theta} \prod_i P(x_i;\theta)$$

Un produit sur plusieurs probabilités n'est pas souhaitable, pour cela nous prenons le log de vraisemblance au lieu de la vraisemblance, et à partir de là nous devrions avoir une somme de log probabilités. Puisque $\log(x)$ est monotone, optimiser $f(x)$ est identique à optimiser $\log\big(f(x)\big)$. Ainsi, au lieu du MLE, nous prenons le log et minimisons la vraisemblance du log négatif (NLL) en utilisant un algorithme d'optimisation comme la descente de gradient.

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \sum_{i}^{m} \log P(x_i\,;\,\theta) \;\; = \arg\min_{\theta} - \sum_{i}^{m} \log P(x_i\,;\,\theta)$$

Dans le cas d'un problème de classification avec apprentissage supervisé :

$$\hat{\theta}_{MLE} = \arg\min_{\theta} P(Y|X\,;\,\theta) = \arg\min_{\theta} - \textstyle\sum_{i}^{m} \log P(\,y_i|\,x_i;\,\theta)$$

3.3.3.2. Estimation maximale a posteriori

Comme nous l'avons mentionné, l'inférence bayésienne calcule entièrement la distribution de probabilité a posteriori, donc la sortie n'est pas une valeur unique mais une fonction de densité de probabilité (lorsque θ est une variable continue) ou une fonction de masse de probabilité (lorsque θ est une variable discrète). La plupart du temps, la distribution a posteriori est insoluble et pour cela le MAP est considéré comme une bonne solution pour estimer le point de probabilité a posteriori maximal s'il n'y a pas besoin de la distribution a posteriori complète. Nous écrivons le MAP comme suit :

$$\hat{\theta}_{MAP} = \text{argmax}x_{\theta}\, \mathrm{P}(\theta|X) = \text{argmax}x_{\theta}\, \mathrm{P}(X|\theta)\mathrm{P}(\theta)$$

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{argmax}} \log \prod_{i}^{m} P(x_i\,|\,\theta) + \log P(\theta)$$

$$\hat{\theta}_{MAP} = argmin_{\theta} \; - \sum_{i}^{m} logP(x_i\,|\theta) + logP(\theta)$$

Dans le cas d'un problème de classification avec apprentissage supervisé :

$$\hat{\theta}_{MLE} = argmin_{\theta}\, P(Y|X,\theta) + logP(\theta) = argmin_{\theta} \; - \textstyle\sum_{i}^{m} logP(\,y_i|\,x_i;\theta) + logP(\theta)$$

En comparant les équations MLE et MAP, la seule chose qui diffère est l'inclusion d'a priori $P(\theta)$ dans le MAP, ou en d'autres termes le MLE est le MAP avec un a priori non informatif. La vraisemblance est maintenant pondérée avec un certain poids provenant de la probabilité a priori. Aussi, l'a priori peut donc être vu comme une régularisation de l'estimation du maximum de vraisemblance. La présence de la probabilité a priori dans le cadre des paramètres du problème, puis l'utilisation du MAP au lieu du MLE pourrait être un meilleur choix.

3.3.4. Fonction objective

Typiquement, le but dans les réseaux de neurones pendant le processus d'optimisation est de minimiser la valeur d'erreur en utilisant une fonction objective. La fonction objective est souvent appelée fonction de perte ou fonction de coût et la valeur calculée par celle-ci est simplement appelée perte ou coût. La fonction objective a des caractéristiques indésirables, comme la non-convexité et la non-linéarité avec une dimensionnalité et un bruit élevé, et est considérée comme complexe et coûteuse en calcul à évaluer.

L'optimisation consiste souvent à minimiser la fonction de coût pour trouver les poids et les biais du modèle qui décrivent le mieux nos données. Le choix de la fonction de coût dépend du type de la fonction d'activation dans la dernière couche de notre réseau. Dans les tâches de classification, nous utilisons généralement la fonction d'activation Softmax (problème de classification multi-classes) ou sigmoïde (problème de classification binaire ou multi-étiquettes) dans la dernière couche de l'architecture réseau. Une fonction de coût telle que l'entropie croisée est considérée comme un bon choix entre la distribution des données et la distribution prédite par le modèle.

Du point de vue d'un modèle probabiliste, MLE est utilisé pour déterminer la fonction de coût en fonction de certaines hypothèses que nous faisons en ce qui concerne les données et le problème que nous avons. Dans cette section, nous discuterons des fonctions de perte qui sont pertinentes pour les problèmes de classification.

3.3.4.1. Entropie croisée

Le maximum de vraisemblance conçoit la fonction de coût d'entropie croisée comme une solution pour mesurer l'erreur entre deux distributions de probabilité. Au cours de la phase d'apprentissage, le maximum de vraisemblance tente d'estimer les meilleurs paramètres du modèle qui minimisent la dissemblance entre la distribution de probabilité prédite du modèle et la distribution de probabilité des données. C'est ce que fait l'entropie croisée.

Connue également sous le nom de probabilité logarithmique négative, l'entropie croisée est souvent utilisée dans les modèles de réseaux neuronaux le long de la fonction d'activation sigmoïde ou du Softmax dans la dernière couche du réseau. En présence d'une variance entre la probabilité prédite et la probabilité réelle, la perte d'entropie croisée augmente. Il existe deux

variantes de l'entropie croisée en fonction du problème de classification que nous avons. De même, le choix de la fonction d'activation utilisée dans la dernière couche de l'architecture du réseau suit le choix de la fonction de perte. Avec une classification multi-classes, nous utilisons la fonction de perte d'entropie croisée catégorique avec l'activation Softmax, et dans la classification binaire ou la classification multi-étiquettes, l'entropie croisée binaire est utilisée avec l'activation sigmoïde.

Pour un cas discret, l'entropie croisée est calculée en utilisant les probabilités des événements à partir de la distribution des vraies étiquettes P et de la distribution de probabilité de prédiction Q , avec une variable X donnée comme suit :

$$H(P,Q) = -\sum_{x \in X} P(x) \log\big(Q(x)\big)$$

3.3.4.2. Divergence de Kullback-Leibler

La divergence de Kullback-Leibler KLD (ou divergence KL) est la mesure de la divergence entre deux distributions de probabilité, P la vraie distribution de probabilité et Q la distribution approximative. Pratiquement, la divergence KL et l'entropie croisée sont similaires. Une divergence égale à 0 entre deux distributions représente l'identité entre elles.

La plupart du temps, la divergence KL est utilisé dans les modèles essayant d'approximer une fonction complexe ou une distribution complète, comme nous le verrons dans la section du réseau de neurones bayésien lorsque nous essayons d'approximer la distribution a posteriori. De plus, il peut être utilisé pour la classification multi-classes. La divergence KL peut être calculée comme suit :

$$D_{KL}(P \parallel Q) = H(P,Q) - H(P)$$

Où ‖ est l'opérateur pour indiquer « divergence » ou la divergence de P par rapport à Q. L'entropie $H(X)$ dans un cas discret est représentée par :

$$H(X) = -\sum_{x \in X} P(x) \log P\,(x)$$

L'entropie $H(X)$ essaie de trouver la quantité d'informations dans une distribution de probabilité compte tenu des données, et les informations perdues lorsque les données changent. Intuitivement, c'est la valeur attendue de la probabilité des données dans une distribution.

3.3.5. Optimiseurs

L'optimisation est considérée comme le cœur de l'apprentissage automatique. L'optimisation en général fait référence à la tâche de trouver les points critiques dans une fonction donnée. Dans le domaine de l'apprentissage automatique, la fonction d'intérêt est la fonction de perte $J(\theta)$ paramétrée par les paramètres du modèle $\theta \in R^d$ (θ représente les poids et les biais du réseau de neurones). Habituellement, optimiser la fonction de perte revient à la minimiser en trouvant un point extremum local. Il existe un groupe d'algorithmes destinés à trouver l'optimum global, mais il suppose un ensemble d'hypothèses fortes sur la fonction comme sa convexité et sa régularité.

Nous aborderons dans ce qui suit certains algorithmes basés sur le gradient pour les tâches d'optimisation.

3.3.5.1. Descente de gradient

La descente de gradient est un algorithme d'optimisation itératif du premier ordre pour trouver un minimum local d'une fonction différentiable (Tolstikhin et al., 2018). Dans les réseaux de neurones artificiels, la descente de gradient permet de réduire l'erreur ou la perte et de mettre à jour les poids du modèle après avoir utilisé les données d'apprentissage pour faire des prédictions. Les mises à jour prennent la pente perpendiculaire au gradient de l'erreur, ce qui mène vers un minimum de la fonction, d'où le nom descente de gradient. Au début, la descente de gradient initialise les paramètres du modèle de manière aléatoire, puis utilise la fonction de coût pour calculer l'erreur en utilisant la prédiction du modèle. Le processus est répété et à chaque itération, les paramètres du modèle sont mis à jour.

Trois types de descente de gradient :

- Descente de gradient par lots : calcule l'erreur pour chaque échantillon et met à jour les paramètres du modèle une fois tous les échantillons évalués.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta)$$

Les $\boldsymbol{\theta}$ dans l'équation ci-dessus représentent les paramètres du modèle, et η représente le taux d'apprentissage qui détermine la taille du pas à chaque itération. La dernière partie est le gradient de la fonction de coût.

- Descente de gradient stochastique SGD: Contrairement au premier, SGD met à jour les paramètres du modèle en utilisant un seul échantillon $\left(x^{(i)}, y^{(i)}\right)$ à partir d'un ensemble de données à chaque itération.

$$\theta = \theta - \eta \cdot \nabla_\theta J\left(\theta; x^{(i)}, y^{(i)}\right)$$

- Descente de gradient par mini-lots : c'est l'implémentation la plus utilisée. Après avoir divisé l'ensemble de données en une taille plus petite appelée lot, la descente de gradient par mini-lots met à jour les paramètres du modèle à la fin de chaque lot.

$$\theta = \theta - \eta \cdot \nabla_\theta J\left(\theta; x^{(i:i+n)}, y^{(i:i+n)}\right)$$

Dans l'équation ci-dessus, le i représente le nombre de lots et le n représente la taille des échantillons dans le lot.

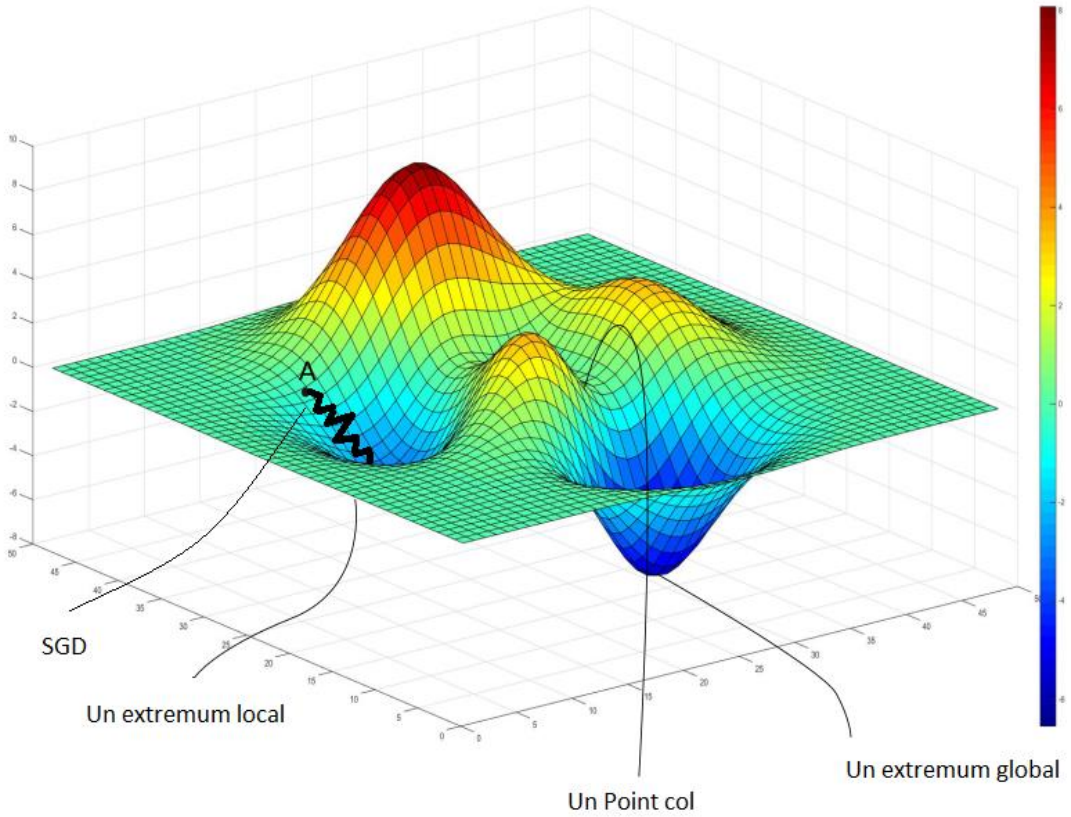


Figure 6 Une fonction de coût simplifiée (source: bdtechtalks).

3.3.5.2. Momentum

Le Momentum est une méthode qui permet d'accélérer SGD pour atteindre plus rapidement le minimum local (Ruder, 2016). Le Momentum est une moyenne mobile exponentielle des gradients passés paramétrés par γ (généralement égal à 0,9).

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

$$\theta = \theta - v_t$$

L'algorithme Momentum accumule une moyenne mobile décroissante de manière exponentielle des gradients passés et continue de se déplacer dans leur direction. L'effet de Momentum est illustré sur la figure.

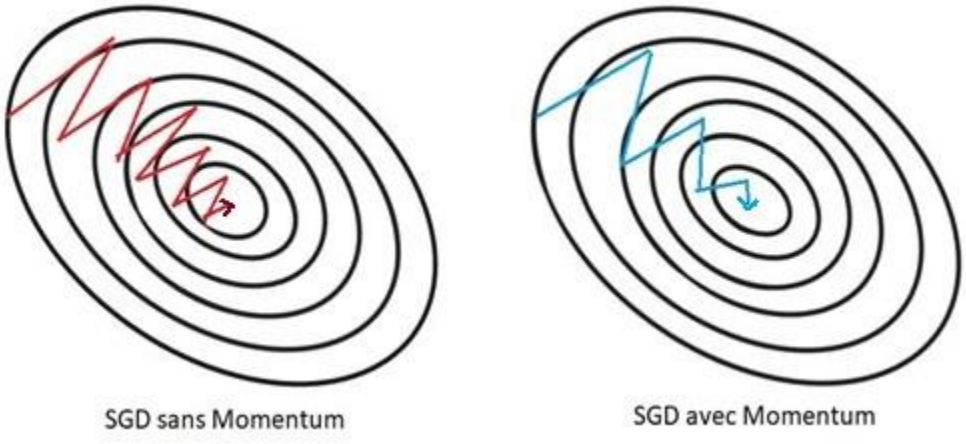


Figure 7 SGD avec et sans Momentum.

3.3.5.3. AdaGrad

Jusqu'à présent, nous utilisions un taux d'apprentissage fixe lors du calcul de la descente du gradient. Les gradients adaptatifs (AdaGrad) ajustent le taux d'apprentissage dans le temps et utilisent un taux d'apprentissage différent pour chaque paramètre θ_i à chaque pas de temps t (Ruder, 2016):

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i}$$

AdaGrad utilise le gradient passé pour mettre à jour le taux d'apprentissage :

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

Avec $G_t \in \mathbb{R}^{d\times d}$ est une matrice diagonale où chaque élément diagonal i, i est la somme des carrés des gradients par rapport à θ_i jusqu'au pas de temps t . Tandis que ϵ est un ajout pour éliminer la division par 0.

3.3.5.4. Adam

Si nous écrivons la moyenne en décroissance exponentielle des gradients passés au carré v_t et la moyenne en décroissance exponentielle des gradients passés m_t comme:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

L'adaptive Moment Estimation (en français: Estimation du moment adaptative Adam) met à jour les paramètres comme suit :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \ , \qquad \hat{m}_t = \frac{m_t}{1-\beta_1^t} \ , \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Où m_t et v_t sont respectivement des estimations du premier moment (la moyenne) et du deuxième moment (la variance non centrée) des gradients, et sont initialisés comme des vecteurs de 0. Les deux valeurs $\hat{m}_t$ et $\hat{v}_t$ sont une estimation de m_t et v_t respectivement, après que les auteurs de l'article d'Adam ont observé qu'ils sont biaisés vers la valeur 0. Les auteurs proposent des valeurs par défaut de 0,9 pourβ_1, 0,999 pour β_2 et 10^{-8} pour ϵ (Ruder, 2016).

3.3.6. Régularisation pour l'apprentissage en profondeur

Les méthodes de régularisation aident à réduire l'erreur de généralisation (la mesure de la précision avec laquelle un algorithme est capable de prédire les résultats à l'aide de nouvelles données) mais pas l'erreur d'apprentissage, ce qui signifie qu'elles aident le modèle à généraliser et à donner de bons résultats sur des données inconnues et en même temps à ne pas surajuster les données d'apprentissage. La régularisation est considérée comme une technique qui ajoute un terme de pénalité à la fonction de perte pour éviter le problème de surajustement.

Nous présentons ici deux techniques de régularisation de la fonction coût pour une meilleure généralisation.

3.3.6.1. Dropout

Le dropout est considéré comme une bonne solution pour le problème courant de surajustement et pour réduire l'erreur de généralisation. Cette technique fonctionne en abandonnant au hasard des unités dans des réseaux de neurones profonds pendant l'entraînement. La chute se fait temporairement le long de toutes les connexions entrantes et sortantes vers l'unité, comme indiqué sur la figure. Cette technique est utilisée uniquement pendant le temps d'entraînement. Au moment de l'entraînement, le dropout combine plusieurs modèles pour améliorer la généralisation. Les couches de dropout sont initialisées en tant que variables aléatoires distribuées de Bernoulli. Chaque unité se voit attribuer une probabilité d'être retenue qui est indépendante des autres unités. Les neurones sont soit abandonnés avec une probabilité p ou gardés avec une probabilité $1 - p$.

Au moment du test, le modèle est utilisé sans le dropout, et les poids du réseau sont réduits en le multipliant par la probabilité de rétention de l'unité abandonnée au moment d'entraînement.

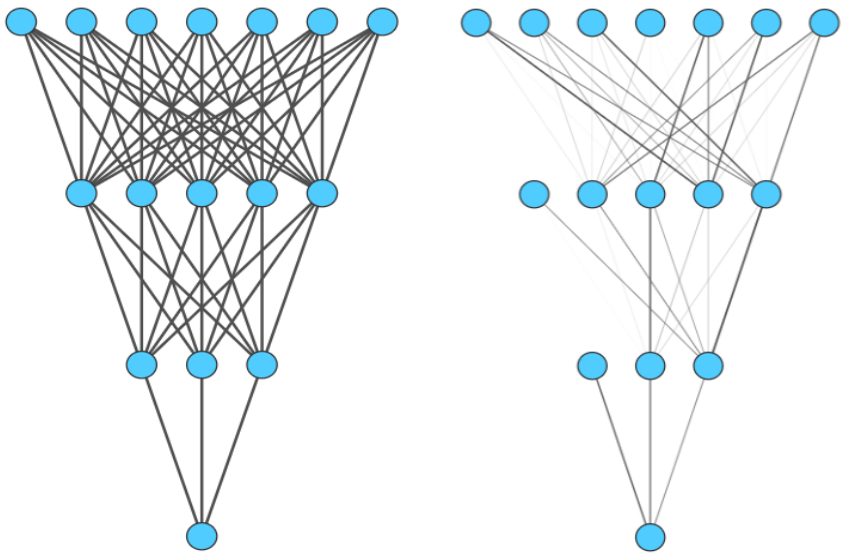


Figure 8 À gauche : un réseau de neurones standard. À droite : un réseau de neurones après l'application de la technique de dropout.

3.3.6.2. Pénalité de norme de paramètre

Une deuxième technique pour éviter le problème de surajustement consiste à pénaliser les grands paramètres du modèle en ajoutant une pénalité de norme de paramètre $\Omega(\theta)$ à la fonction de coût J :

$$J^*(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta)$$

Où $\alpha\epsilon[0, \infty)$ est considéré comme un hyperparamètre qui pondère l'effet de la pénalité sur la fonction de coût, plus la valeur de α est grande, plus l'effet de la régularisation sur la fonction de coût est important. Lorsque $\alpha = 0$, la pénalité est ignorée. Dans un contexte de réseau de neurones, nous régularisons uniquement les poids et ignorons les biais. La norme L_2 et la norme L_1 sont des cas particuliers de la norme L_p plus générale :

$$L_p = \left(\sum_{i=1}^{N} |x_i|^p\right)^{1/p} \text{ , pour p} \geq 1$$

3.4. Réseaux de neurones bayésiens

Les réseaux de neurones déterministes ont souvent tendance à être trop confiants en donnant faussement de mauvaises prédictions. Ils ne savent tout simplement pas quand ils sont censés être ignorants lorsqu'ils traitent de nouvelles données. L'approche bayésienne est un moyen de mettre à jour nos croyances antérieures sur les paramètres du modèle en utilisant de nouvelles informations observées. Lors de l'inférence bayésienne dans le domaine de l'apprentissage en profondeur, nous devons utiliser des réseaux de neurones stochastiques également appelés réseaux de neurones bayésiens BNN. Comme nous l'avons mentionné précédemment, l'idée d'un réseau de neurones déterministe consiste à effectuer une optimisation ponctuelle des paramètres du modèle à l'aide de méthodes telles que MLE. Contrairement aux réseaux de neurones déterministes, le problème de surajustement est traité dans les BNN.

Comme nous l'avons mentionné, le MLE et le MAP sont des estimateurs ponctuels, et pour calculer la distribution a posteriori complète sur les paramètres du modèle dans l'approche bayésienne, nous devons l'approximer, ce qui donne un moyen de mesurer l'incertitude des paramètres en la marginalisant à l'aide de la distribution prédictive a posteriori qui équivaut à

faire la moyenne des prédictions d'un ensemble de réseaux de neurones pondérés par les probabilités a posteriori de leurs paramètres .

Dans un monde parfait, nous voudrions peut-être avoir une distribution complète sur les paramètres du modèle. L'idée d'avoir une distribution sur les poids nous permettra d'avoir plusieurs valeurs pour les poids en échantillonnant à partir de la distribution de chaque poids, et pour cela nous obtiendrons un ensemble de prédictions plutôt qu'une. La stochasticité dans les paramètres du modèle nous permettra de mieux comprendre les résultats du modèle en introduisant la propriété d'incertitude.

Suivant l'approche bayésienne, en présence d'un grand ensemble de données, le calcul de la distribution a posteriori complète est très difficile, donc dans de nombreux cas nous visons à l'approximer en utilisant deux méthodes bien connues, les méthodes basées sur l'échantillonnage comme la chaîne de Markov Monte Carlo (MCMC) et l'inférence variationnelle (VI). Le choix de l'une des solutions compromet la précision du côté de MCMC et la vitesse du côté de VI. La MCMC a de nombreuses variantes pour estimer la distribution a posteriori complète, mais elles nécessitent du temps et des calculs excessifs. Contrairement à la MCMC, la VI approxime la distribution a posteriori complète au lieu de la calculer entièrement. L'objectif principal de la VI est de trouver une distribution qui est une approximation de la distribution a posteriori en utilisant la fonction objectif KLD qui minimise la divergence entre eux, aussi près que possible. La VI considère l'approximation comme un problème d'optimisation sur des espaces fonctionnels. De plus, le mode de la probabilité a posteriori est l'estimation maximale a posteriori (MAP) (Jospin et al., 2020).

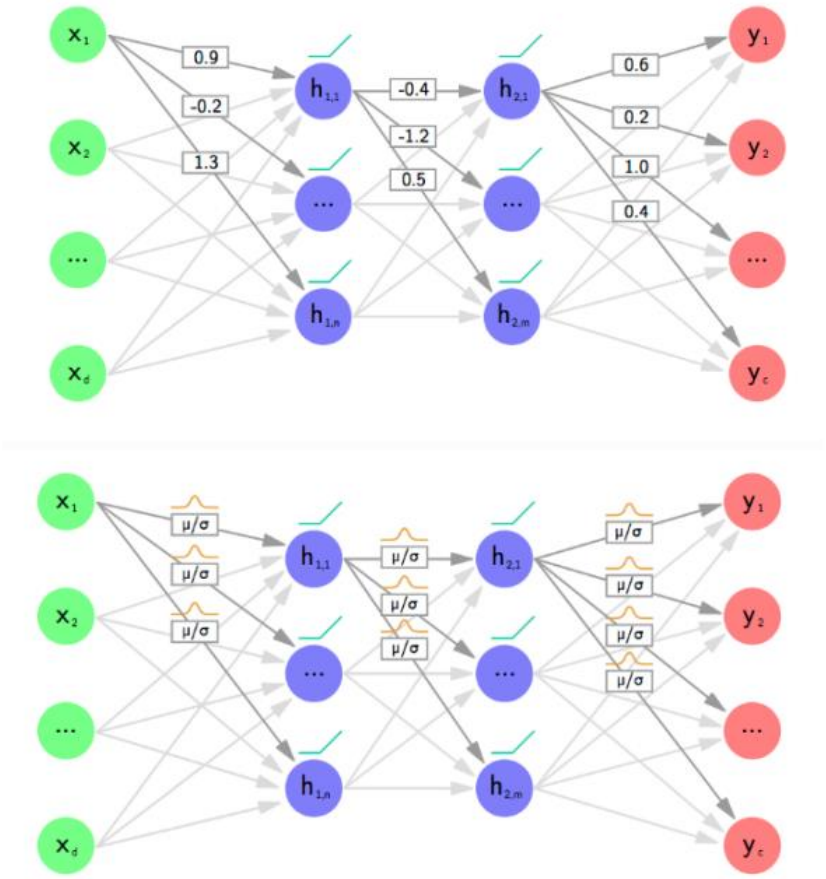


Figure 9 : En haut, un réseau de neurones standard où chaque poids est une valeur fixe. En bas, un réseau de neurones bayésien où chaque poids est une distribution, dans ce cas une distribution normale avec deux paramètres, la moyenne et la variance (source: Shridhar et al., 2019).

3.4.1. Inférence variationnelle

Trouver la distribution a posteriori est jugée trop compliquée et difficile, pour cela, l'inférence variationnelle donne comme approche la distribution a posteriori souhaitée comme le montre la figure 10:

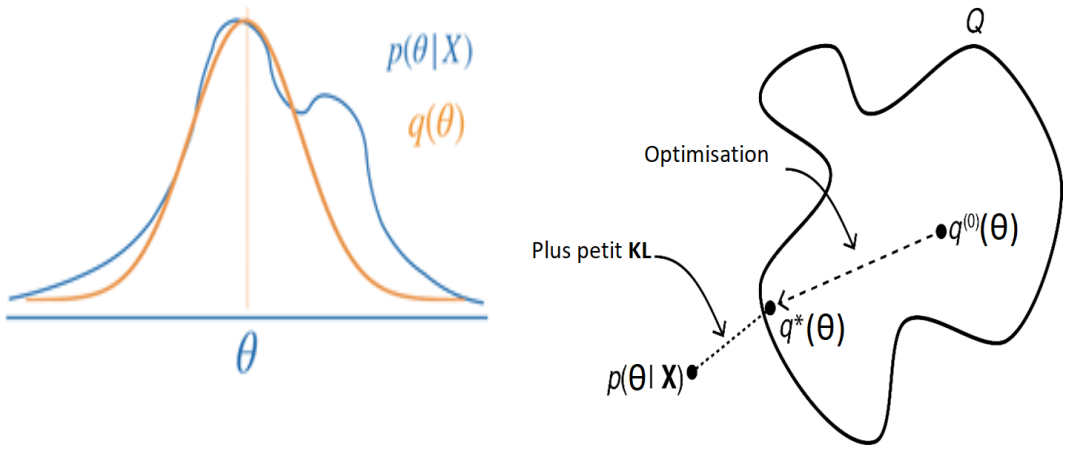


Figure 10. Approximation des distributions de probabilité (source: medium, gregorygundersen)

En suivant l'objectif d'optimisation, le but de la VI est d'utiliser la fonction KLD pour trouver une distribution optimale $q^*(\theta)$ et la plus proche de la distribution a posteriori $p(\theta|X)$, cette dernière appartient à un espace compliqué tandis que la première appartient à une classe de distributions traitables Q (Blei et al., 2016). Nous utiliserons ensuite q (plutôt que p) afin d'obtenir une solution approximative. Dans un BNN, les distributions dans Q sont définies sur les poids du réseau w. Par exemple, Q peut-être l'ensemble de toutes les distributions gaussiennes entièrement factorisées, auquel cas les paramètres variationnels θ sont un vecteur de moyennes et de variances. Ensuite, nous dérivons la fonction objectif finale pour l'utiliser dans un BNN :

$$\theta^* = \arg\min_{\theta} KL\big(q(w|\theta) \parallel p(w|D)\big)$$

$$\theta^* = \arg\min_{\theta} -KL[q(w|\theta) \parallel p(w)] + \mathbb{E}_{q(w|\theta)}[\log p\,(D|w)]$$

Pour simplifier on note:

$$F(D,\theta) = KL[q(w|\theta) \parallel p(w)] - \mathbb{E}_{q(w|\theta)}[\log p\,(D|w)]$$

La dernière formule est une fonction de coût connue sous le nom d'énergie libre variationnelle. L'énergie libre variationnelle négative s'appelle la limite inférieure de la preuve (ELBO).

3.4.2. Bayes par rétropropagation

Bayes par rétropropagation BBB (Bayes by Backprop) (Blundell et al., 2015) est une méthode d'inférence variationnelle pour estimer la distribution a posteriori sur les poids du

réseau de neurones, et considéré comme une solution au problème de la stochasticité des poids lorsque nous essayons d'appliquer un optimiseur comme l'algorithme de descente de gradient.

Comme nous l'avons mentionné précédemment, l'inférence bayésienne concerne la marginalisation au lieu de l'optimisation, et BBB vient faire passer le problème de la marginalisation sur les poids du réseau à l'optimisation des poids.

La distribution de probabilité a posteriori variationnelle dans BBB est une distribution gaussienne diagonale de paramètre $\theta = (\mu, \sigma)$, de paramètre moyen $\mu \in R^d$ et de paramètre écart type $\sigma \in R^d$, noté $\mathcal{N}(\theta|\mu, \sigma^2)$. La moyenne et l'écart type sont une variable latente dont les valeurs sont estimées lors de l'étape d'apprentissage, plus les poids w . Notez que d est la dimension des paramètres du réseau et que le deuxième paramètre de la distribution gaussienne est une matrice de covariance diagonale.

$$q(w|\theta) = \prod_i \mathcal{N}(w_i|\mu, \sigma^2)$$

Où le log a posteriori est :

$$\log q\,(w|\theta) = \sum_i log\mathcal{N}(w_i|\mu, \sigma^2)$$

Ensuite, en utilisant la méthode d'échantillonnage de Monte Carlo pour approximer la fonction de coût ELBO exact :

$$F(D, \theta) \approx \sum_{i=1}^{n} \log q\left(w^{(i)}\middle|\theta\right) - \log p\left(w^{(i)}\right) - \log p\left(D\middle|w^{(i)}\right)$$

Où $w^{(i)}$ désigne le nième échantillon de Monte Carlo tiré de la distribution a posteriori variationnel $q\left(w^{(i)}\middle|\theta\right)$.

Dans une configuration de réseau de neurones, nous utilisons des gradients pour mettre à jour les poids. Bayes par rétropropagation combine l'inférence variationnelle et astuce de reparamétrisation pour résoudre le problème de la stochasticité des poids lorsque nous échantillonnons à partir de la distribution variationnelle.

Pour utiliser l'optimisation par lots, nous utilisons la fonction de coût suivante :

$$\mathrm{F}(D_i, \theta) = \frac{N}{M}\mathrm{KL}[log\ q(\mathrm{w}|\theta) \parallel log\ p(w)] - E_{q(w|\theta)}[log\ p(D_i|w)]$$

$$\approx \frac{1}{N}\big(\log q\ (\mathrm{w}|\theta) - \log p\ (\mathrm{w})\big) - \frac{1}{M}\log p\ (D_i|\mathrm{w})$$

Où la première partie est l'entropie croisée et la deuxième partie est la Log-vraisemblance négative et M, N sont le nombre de lots et la taille des données d'entraînement.

Dans l'article de BBB, ils ont utilisé un mélange de deux densités gaussiennes comme l'a priori. Chaque densité a une moyenne nulle, mais des variances différentes.

3.5. Recherche automatique d'architecture neuronale

Dans les réseaux de neurones au moment de l'apprentissage, nous visons à trouver un modèle avec les meilleurs paramètres (ex. poids et biais) en utilisant une architecture avec les meilleurs hyperparamètres (ex. nombre de couches et de nœuds cachés, taux d'apprentissage). De même, en sélectionnant différents paramètres externes également appelés hyperparamètres dans une architecture de réseau de neurones, on contrôle les résultats et le comportement d'une technique ou d'une méthode utilisée lors de l'apprentissage, par exemple le choix de la fonction d'activation à chaque couche réseau, ou le taux d'apprentissage ou la probabilité d'abandon avant chaque couche.

Dans un cadre statistique pratique, nous faisons des hypothèses théoriques sur une fonction, comme les propriétés de lissage ou de convexité. Dans le monde réel de l'apprentissage automatique, lorsque nous avons une fonction complexe, ces hypothèses que nous faisons sont souvent violées. Souvent, il est coûteux et parfois impossible d'évaluer une fonction objective, ses dérivées et les propriétés de convexité sont inconnues. Dans une optimisation globale, la fonction est considérée comme une boîte noire comme dans notre problème de recherche des meilleurs hyperparamètres, et nous ne pouvons observer que la sortie compte tenu des entrées. Aussi, notons l'absence d'une fonction bien définie mathématiquement ainsi que ses dérivées ou encore le besoin de ressources mémoire conséquentes pour calculer les gradients.

Trouver et sélectionner manuellement les bons hyperparamètres nécessite beaucoup d'expérience et parfois de chance. La méthode est connue sous le nom d'essai et d'erreur. D'autres méthodes tentent de trouver automatiquement les hyperparamètres optimaux. La recherche automatique d'architecture neuronale artificielle (Neural Architecture Search NAS) est

un sous-domaine d'AutoML et l'une des méthodes qui tentent d'automatiser le processus d'ingénierie d'architecture et le domaine de l'apprentissage automatique. Nous pouvons représenter le NAS à travers trois composantes : l'espace de recherche, la méthode d'optimisation et la méthode d'évaluation (Kyriakides & Margaritis, 2020). L'optimisation des hyperparamètres est considérée comme un sous-domaine de la recherche sur l'architecture neuronale artificielle, et l'espace de recherche des meilleurs paramètres d'architecture de réseau neuronal est généralement plus grand que dans l'optimisation des hyperparamètres.

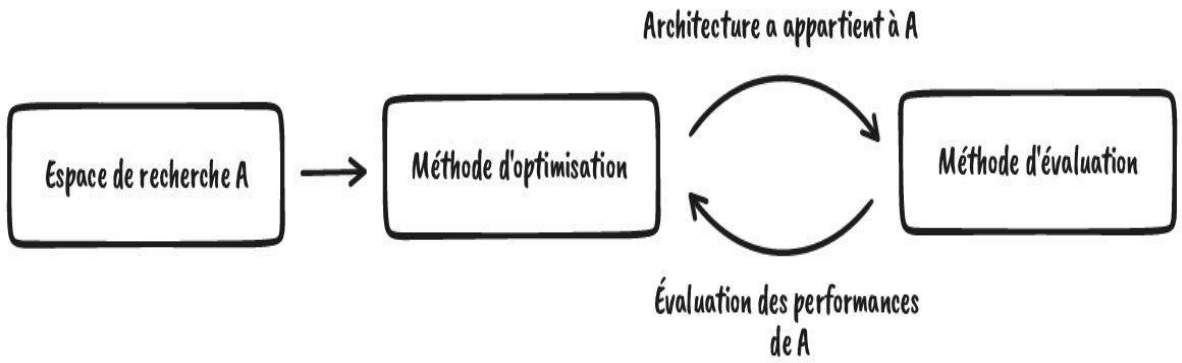


Figure 11. Trois composants principaux des modèles de recherche automatique d'architecture neuronale (Image inspirée par (Elsken et al., 2018)).

L'espace de recherche définit la taille et la limite que nous sommes autorisés à rechercher pour la meilleure architecture. Cela nécessite des connaissances préalables qui peuvent présenter un certain biais humain dans le processus. Le choix de l'espace de recherche est crucial pour minimiser la complexité de la recherche et donc les besoins de calcul (Elsken et al., 2018).

La méthode d'optimisation détaille la façon dont nous explorons l'espace de recherche. Elle utilise le compromis exploration-exploitation pour observer la recherche spatiale et découvrir les configurations optimales globales de l'architecture.

Enfin, la méthode d'évaluation permet d'évaluer et de comparer les premiers résultats de la méthode d'optimisation utilisée pour trouver les meilleures configurations d'architecture. Elle est considérée comme une méthode coûteuse et empêche parfois de trouver les meilleures configurations d'architecture en limitant la découverte de sous-espaces supplémentaires dans

l'espace de recherche. Les meilleures méthodes sont donc celles qui réduisent le coût d'estimation de la méthode d'optimisation, et permettent d'aller aux limites dans l'espace de recherche.

3.5.1. Optimisation bayésienne

Certaines méthodes connues pour automatiser le choix des meilleurs hyperparamètres sont la recherche par grille ou la recherche aléatoire. Comme il s'agit de stratégies de recherche séquentielles, elles n'utilisent pas toutes les connaissances des évaluations passées lorsqu'elles évaluent certains hyperparamètres pour décider des meilleurs points suivants à évaluer.

Parfois, utiliser un algorithme basé sur le gradient comme la descente de gradient pour optimiser une fonction n'effectue pas le travail en raison du fait qu'il est difficile de calculer la dérivée de la fonction, pour cela nous utilisons une méthode non basée sur le gradient pour optimiser une fonction de boîte noire telle que la méthode d'optimisation bayésienne.

L'optimisation bayésienne (OB) est une optimisation sans dérivée, elle utilise tout l'état précédent de la fonction f pour sélectionner le point suivant à évaluer plutôt que d'utiliser uniquement le dernier ou les quelques derniers états comme la descente de gradient (Frazier, 2018). L'OB est un algorithme itératif comme la descente de gradient. Nous partons d'un point initial, passant par une séquence de paires actions-états, pour arriver à l'état d'intérêt souhaitable. De plus, nous ne pouvons pas supposer la concavité ou la linéarité de la fonction f d'où le nom de la boîte noire. OB est une méthode d'optimisation globale. Nous utilisons l'OB lorsque nous n'avons pas l'expression sous forme fermée de la fonction objectif f, mais nous pouvons obtenir des observations $f(x)$ aux points échantillonnés x.

L'OB utilise l'approche bayésienne pour trouver un optimum global en définissant un a priori sur la fonction objective f représentant notre croyance antérieure sur l'espace des fonctions et en le combinant avec la fonction de vraisemblance pour obtenir une fonction a posteriori. Deux éléments définissent l'OB, soit un modèle de substitution probabiliste et une fonction d'acquisition. Le rôle d'un modèle de substitution probabiliste est de modéliser l'espace des fonctions que l'on souhaite évaluer, et d'aider à approximer la fonction objective. Pour sélectionner le prochain point échantillonné dans la fonction objectif, nous utilisons la fonction d'acquisition. Par rapport à la véritable fonction objectif de la boîte noire f, la fonction de substitution f^* est considérée comme peu coûteuse à évaluer et à optimiser.

3.5.1.1. Processus gaussiens

L'optimisation bayésienne est un groupe de méthodes d'optimisation basées sur l'apprentissage automatique essayant de résoudre le problème suivant :

$$\max_{x \in \mathcal{A}} f(x)$$

Dans notre contexte, l'entrée $x \in \mathbb{R}^d$ sont les hyperparamètres, et d est le nombre d'hyperparamètres typiquement inférieur à 20, et $\mathcal{A}$ est l'ensemble des échantillons. Nous avons choisi de modéliser la fonction objective continue f comme un modèle de processus gaussien. Lors de l'évaluation de la fonction f, le bruit est supposé indépendant et gaussien avec une variance constante (Frazier, 2018).

Le modèle paramétrique est une famille de distributions de probabilité qui a un nombre fini de paramètres. D'autre part, le nombre de paramètres dépend des données dans un modèle non paramétrique. Les processus gaussiens sont considérés comme une méthode non paramétrique inférant une distribution sur une fonction directement plutôt que sur ses paramètres.

Les processus gaussiens sont considérés comme une extension de la distribution gaussienne multivariée à un processus stochastique de dimension infinie, et une jointure finie de ces dimensions est une distribution gaussienne. Le $\mathcal{GP}$ est une distribution gaussienne sur des fonctions, spécifiée par une fonction moyenne m et fonction de covariance k (Brochu et al., 2010).

$$\mathrm{f}(x) \sim \mathcal{GP}\big(m(x), k(x, x_*)\big)$$

Pour obtenir la distribution prédictive a posteriori $P(f_*|x_*, D)$, nous utilisons l'approche bayésienne en utilisant une distribution a priori de processus gaussiens sur les fonctions $P(f|x) \sim \mathcal{N}(\mu, \Sigma)$ et les données d'apprentissage D pour modéliser la distribution conjointe de $f = f(X)$ (X est le vecteur d'observations d'entraînement) et $f_* = f(x_*)$ (prédiction à l'entrée du test) (Weinberger 2018).

Dans la figure suivante, nous remarquons à chaque point x_i que nous avons une distribution normale avec moyenne et variance.

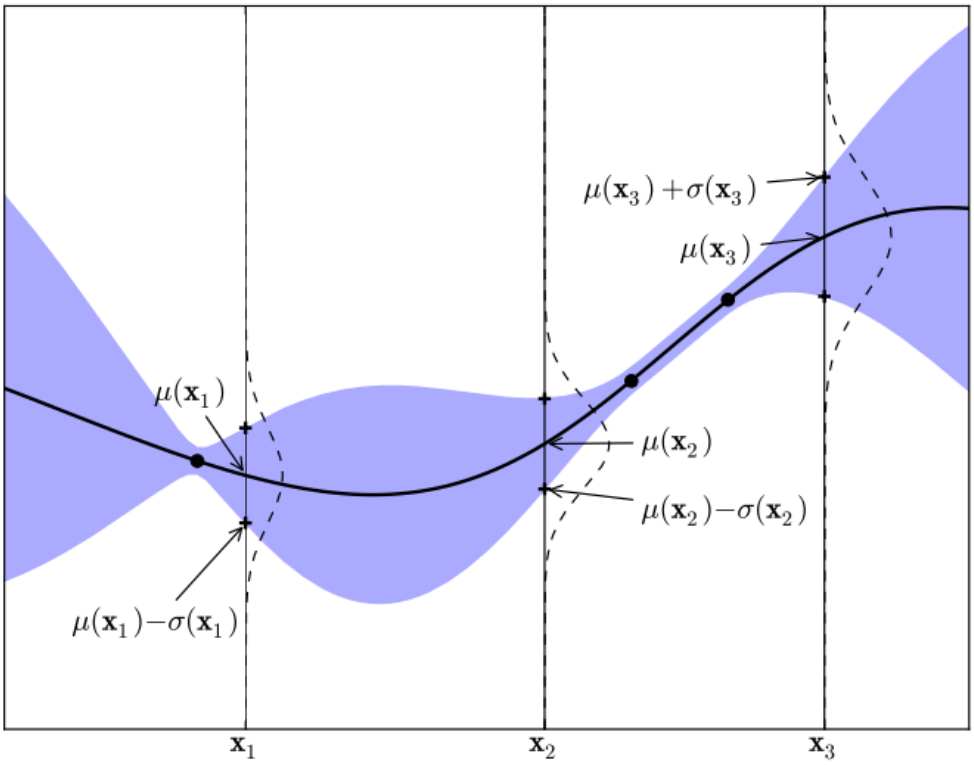


Figure 12. Processus gaussien 1D simple avec trois observations (source : (Brochu et al., 2010)).

Le vecteur moyen est calculé en évaluant la fonction moyenne μ à chaque x_i, où la matrice de covariance Σ (ou σ) est calculée en évaluant une fonction de covariance k à chaque paire de points (x_i, x_j). Le choix de la fonction de covariance k est basé sur la grande corrélation positive entre les deux points d'entrée les plus proches (x_i, x_j) qui se traduit par des valeurs de fonction similaires. Il convient également de mentionner que la matrice de covariance résultante est une matrice semi-définie positive.

3.5.1.2. Fonction d'acquisition

Pour choisir le meilleur échantillon suivant $x_{t+1} \in \mathcal{A}$, l'OB utilise une fonction d'acquisition. La fonction d'acquisition oscille entre l'exploration (où la fonction objective est très incertaine) et l'exploitation (zone dans laquelle la fonction objective a été échantillonnée et a donné de bons résultats) dans l'espace de recherche pour sélectionner les prochains hyperparamètres de la fonction f à évaluer.

L'idée est d'explorer des zones spatiales à partir desquelles nous n'avons pas échantillonné, ou nous avons échantillonné quelques points et nous sommes susceptibles de trouver l'optimum global de la fonction ou d'exploiter les zones spatiales à partir desquelles nous avons déjà échantillonné, et nous avons de bons résultats et échantillonnons à nouveau depuis. Idéalement, pour obtenir les meilleurs résultats, nous souhaitons combiner les approches d'exploration et d'exploitation.

En général, une acquisition élevée correspond à des valeurs potentiellement élevées de la fonction objectif du fait que l'incertitude est grande ou que la prédiction est élevée ou les deux. En maximisant la fonction d'acquisition, on peut sélectionner le point suivant de la fonction objectif. Pour sélectionner les hyperparamètres suivants du réseau, la fonction d'acquisition utilise la moyenne et l'écart type de la distribution prédictive du modèle de processus gaussien à x_t, et nous échantillonnons f à $\arg max_{x_t} \mathrm{u}(x|D_{1:t-1})$ où u représente la fonction d'acquisition et $D_{1:t-1} = (x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$ sont les échantillons $t-1$ tirés de f jusqu'à présent (Brochu et al., 2010).

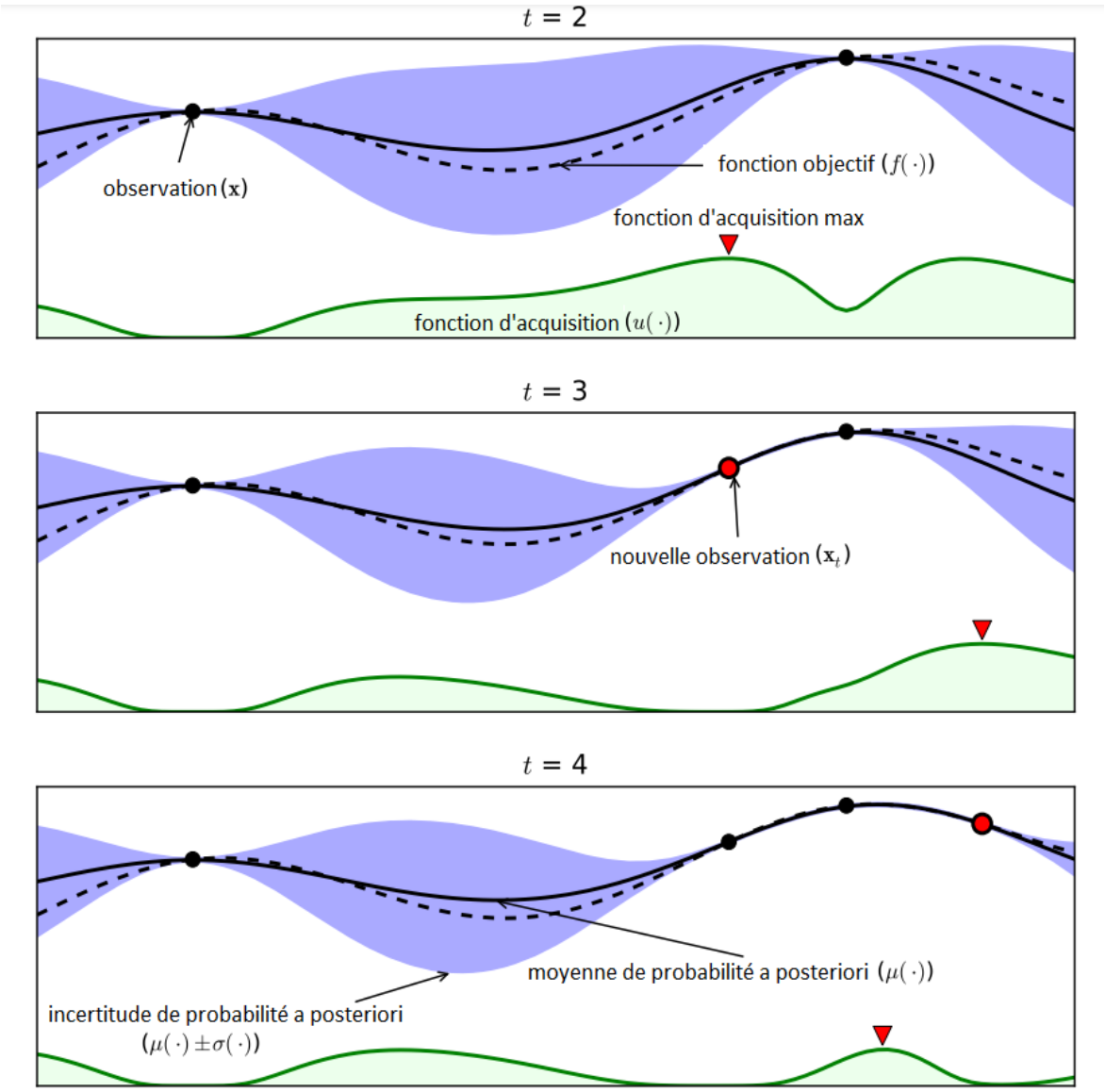


Figure 13. Exemple d'optimisation bayésienne (la source d'image: (Brochu et al., 2010) avec traduction française).

3.6. Incertitude

Dans la modélisation bayésienne, l'incertitude prédictive peut être classée en deux types; incertitude aléatoire et épistémique. Après avoir utilisé un ensemble spécifique d'attributs de données (c'est-à-dire des colonnes ou des propriétés) sélectionnées à partir des données du patient, le modèle donne une prédiction représentant une sélection d'un groupe de classes pour différentes recommandations et objectifs en tant que diagnostic pour le patient. Chaque classe a une vraisemblance comprise entre 0 et 1 pour expliquer la probabilité d'un ensemble d'attributs qui conduisent le modèle à une classe spécifique avec une probabilité faible ou élevée. Parfois, le

modèle commet une erreur de jugement ou des prédictions peu claires, comme donner une probabilité de 0,5 ou une prédiction trop confiante. Cette variabilité inhérente est appelée incertitude aléatoire. De plus, en raison de nouvelles données ou fonctionnalités, nous pouvons être confrontés à des prédictions plus ambiguës qui peuvent être capturées en introduisant l'incertitude épistémique pour aider le clinicien à évaluer les résultats.

En général, les modèles d'un réseau de neurones artificiels standard ont tendance à être trop confiants dans leurs résultats, ce qui conduit à ne pas généraliser lorsqu'ils voient de nouvelles données par exemple. La capture de l'incertitude dans les résultats n'est pas considérée comme l'une des caractéristiques de ces modèles.

L'estimation bayésienne des paramètres se présente sous la forme d'une distribution complète sur les poids, ce qui se traduit par un ensemble de prédictions donné comme la capacité de déterminer à quel point nous sommes certains d'une prédiction.

L'approche bayésienne fournit un moyen de raisonner sur la confiance de notre estimation et de saisir l'incertitude qui s'accompagne souvent d'un prix sur la performance. La source d'incertitude peut être :

- Les conditions et données initiales.
- Type de modèle, paramètres et hyperparamètres.
- Configurations et mesures des capteurs.

Nous présenterons ensuite les deux catégories d'incertitude, l'incertitude aléatoire et l'incertitude épistémique.

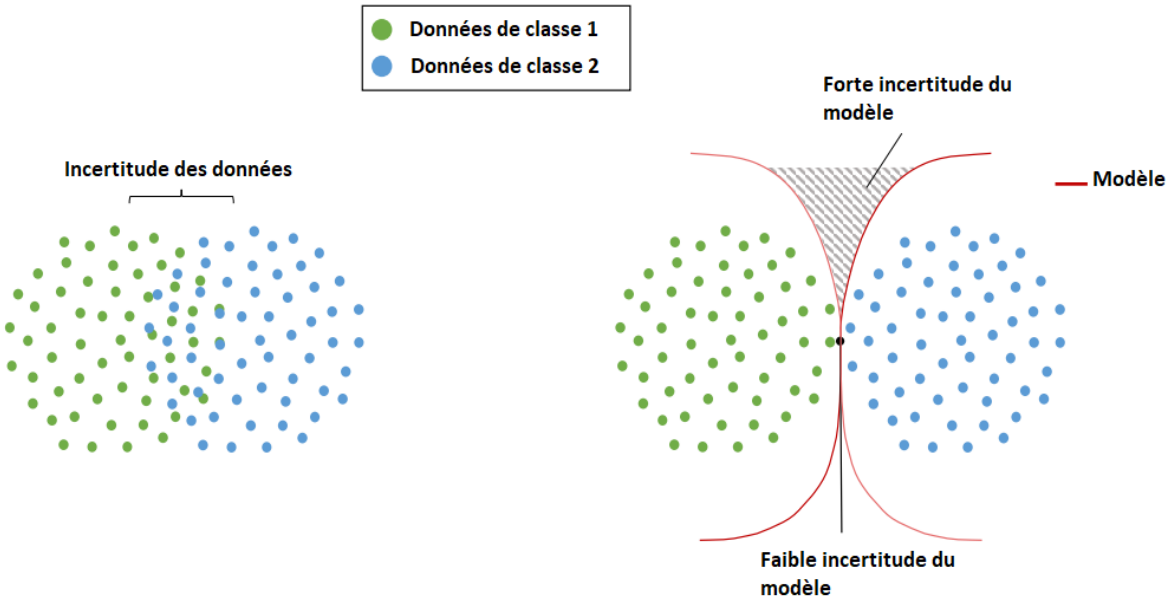


Figure 14 Visualisation des données, du modèle et de l'incertitude pour les modèles de classification (source: Gawlikowski et al., 2021).

3.6.1. Incertitude aléatoire

L'incertitude aléatoire également appelée incertitude des données, est connue comme la partie irréductible de l'incertitude. Rien ne peut le changer car c'est l'une des caractéristiques d'un processus du monde réel. Lorsque le processus de génération de données ne parvient pas à capturer la totalité ou la plupart des informations d'un événement, l'incertitude aléatoire vient à l'image. En conséquence, il y aura un élément de stochasticité dans la dépendance entrée/sortie. L'incertitude aléatoire est considérée comme une propriété inhérente à la distribution des données plutôt qu'au modèle, et générer plus de données ne réduit pas l'incertitude aléatoire.

Pour décrire l'incertitude aléatoire, nous pouvons la représenter en utilisant les moments d'une fonction de distribution de probabilité prédictive.

3.6.2. Incertitude épistémique

Le deuxième type d'incertitude est l'incertitude épistémique connue également sous le nom d'incertitude du modèle. Elle est liée à notre connaissance ou ignorance concernant le processus générant les données, et se produit lorsque le modèle ne parvient pas à expliquer au mieux les données. Elle augmente dans la zone où il y a peu ou pas d'échantillons de données, et

peut être réduite en collectant plus de données. L'incertitude épistémique est vraiment importante à modéliser pour :

- Applications critiques, car l'incertitude épistémique est nécessaire pour comprendre des exemples différents des données initiales.
- Petits ensembles de données.

Dans un réseau de neurones bayésien, les poids sont représentés par une distribution de probabilité et pour capturer l'incertitude épistémique, nous utilisons les poids du modèle. L'incertitude épistémique est capturée par la distribution a posteriori variationnelle. En revenant à l'algorithme de Bayes par rétropropagation, la minimisation de l'énergie libre variationnelle garantit que nous réduisons l'incertitude épistémique. Nous pouvons également utiliser l'écart type comme mesure de l'incertitude épistémique.

Chapitre 4 – Microservices

4.1. Introduction

Le terme « Microservices » est un terme moderne utilisé pour décrire un schéma traditionnel de « séparation des préoccupations » au sein d'un projet distribué. Les microservices sont une idée qui suit une ancienne philosophie appelée philosophie Unix. Doug McIlroy, l'inventeur des tubes Unix et l'un des fondateurs de la tradition Unix a résumé la philosophie Unix comme suit ; « Écrivez des programmes qui font une seule chose et faites-le bien. Écrivez des programmes pour travailler ensemble. Écrivez des programmes pour gérer les flux de texte, car il s'agit d'une interface » (Raymond 2003). En d'autres termes, l'approche des microservices explique l'importance de la modularité dans le développement logiciel par rapport au monolithisme en décomposant les systèmes métier monolithiques en petits services atomiques et déployables indépendamment qui représentent chacun une capacité métier.

Concevoir une bonne architecture à l'aide de l'approche des microservices est assez difficile pour plusieurs raisons telles que le choix du bon contexte et de la bonne limite pour chaque service, les communications entre les services et la gestion des données (théorème CAP), et d'autres que nous aborderons dans les prochaines étapes.

4.2. Du monolithe aux microservices

Pour mieux expliquer l'approche de style Microservices pour concevoir (développer) une application, il est préférable de la comparer à l'approche de style monolithe. Dans l'approche monolithique, l'application est construite comme une seule unité. Les applications d'entreprise sont souvent divisées horizontalement en trois parties principales ; base de données (couche de persistance), côté serveur (logique métier) et côté client (couche de présentation), et verticalement dans différents domaines métier. Le côté serveur gère les requêtes http du côté client pour récupérer ou mettre à jour les données stockées dans la base de données et exécuter la logique du domaine. Une petite modification d'une application monolithique nécessite qu'elle soit reconstruite et déployée, ce qui entraîne des difficultés à maintenir une bonne structure.

Ces raisons ont conduit au style architectural des microservices en découpant la grande unité en plusieurs petites unités ou, comme on dit, en microservices. Les microservices sont une

alternative aux monolithes, où différentes parties du système sont implémentées en tant que services séparés qui communiquent entre eux à l'aide d'API. Cette approche présente plusieurs avantages :

- Les services peuvent être développés et déployés indépendamment, ce qui facilite leur maintenance et leur mise à l'échelle.
- Les services peuvent être écrits dans différents langages ou Frameworks, ce qui permet plus de flexibilité et d'innovation.
- L'architecture globale est modulaire, ce qui facilite l'adaptation aux évolutions des besoins.

4.3. Architecture des microservices

L'architecture des microservices nous permet d'éviter les applications monolithiques, et fournit un couplage faible entre les processus collaboratifs qui s'exécutent indépendamment dans différents environnements.

L'architecture de microservices est une approche du développement logiciel qui encourage la division des applications volumineuses en éléments plus petits et plus faciles à gérer. Cela peut être fait pour diverses raisons, notamment une organisation, une évolutivité et une maintenabilité améliorées.

L'architecture des microservices a gagné en popularité au cours des dernières années car elle permet une livraison et un déploiement continus. En décomposant une application en petits services, chacun pouvant être développé, testé et déployé indépendamment. Aussi, il permet aux organisations de déployer des modifications plus fréquemment et avec moins de risques. Cela rend l'architecture de microservices bien adapté aux environnements DevOps modernes où l'agilité est essentielle.

Les microservices offrent plusieurs avantages par rapport aux applications monolithiques traditionnelles :

- Chaque service peut être développé et déployé indépendamment, sans affecter le reste du système.
- Les services peuvent être mis à l'échelle horizontalement, ce qui signifie qu'ils peuvent être ajoutés ou supprimés du système selon les besoins pour répondre à la demande.
- La complexité globale du système est réduite, car il n'y a pas de point de défaillance unique (ou de goulot d'étranglement).

4.3.1. L'écosystème de microservices

Plusieurs facteurs doivent être pris en compte lors de la création d'un écosystème de microservices. Le premier est la façon dont nous allons stocker les données pour s'assurer que les services sont faiblement couplés. Deuxièmement, la communication entre les services. Les services doivent pouvoir communiquer entre eux afin de partager des données et de collaborer sur des tâches. Nous devons également réfléchir à la manière dont les services seront déployés et gérés. Les services doivent pouvoir évoluer à la hausse ou à la baisse selon les besoins, et nous avons besoin d'outils pour les gérer efficacement. Enfin, nous devons réfléchir aux implications pour la sécurité de l'interaction de plusieurs services entre eux. Nous aurons besoin de mécanismes pour protéger les données et garantir que seuls les utilisateurs autorisés peuvent accéder au système.

4.3.2. Conception de services

La conception de microservices est une approche de l'architecture logicielle qui permet le développement de grandes applications complexes en tant qu'ensembles de petits services. Chaque service peut être développé, testé et déployé indépendamment des autres, ce qui permet des délais d'exécution plus rapides et une meilleure résilience.

Une application basée sur des microservices se compose généralement de plusieurs centaines de services ou plus. Ce nombre élevé peut sembler décourageant, mais il est important de se rappeler que chaque service est relativement petit et facile à comprendre et à gérer par lui-même. De plus, étant donné que les services individuels peuvent être mis à jour sans affecter le reste de l'application, les développeurs peuvent expérimenter de nouvelles idées rapidement et facilement. Cela fait de la conception de microservices une solution idéale pour des équipes de développement agiles.

Lors de la conception d'une application basée sur des microservices, il y a plusieurs choses à garder à l'esprit:

- Chaque service doit avoir une responsabilité unique.
- Les services doivent communiquer entre eux à l'aide d'API bien définies, cela permet de s'assurer qu'ils restent faiblement couplés et faciles à mettre à niveau.
- Un mécanisme pour traiter les demandes perdues entre les services et les erreurs de réseau.

4.4. Les patrons d'architecture de microservices

The Gang of Four sont les quatre auteurs du livre "Design Patterns: Elements of Reusable Object-Oriented Software" (Gamma, et al. 1994), il définissent le patron de conception comme quatre éléments essentiels :

- Le nom du patron de conception.
- Le problème qui décrit quand appliquer le patron de conception.
- La solution qui décrit les éléments qui composent la conception, leurs relations, leurs responsabilités et leurs collaborations.
- Les conséquences sont les résultats et les compromis de l'application du patron de conception.

Les patrons architecturaux sont similaires aux patrons de conception de logiciels mais ont une portée plus large. Les patrons architecturaux abordent divers problèmes de génie logiciel, tels que les limitations de performances du matériel informatique, la haute disponibilité et la minimisation d'un risque commercial. Certains modèles architecturaux ont été implémentés dans des structures logicielles comme la structure logicielle Spring.

Les patrons d'architecture de microservice nous aident à concevoir une application à l'aide de l'architecture de microservice. Ensuite, nous présenterons certains des patrons d'architecture de microservices.

4.4.1. Patrons de décomposition

Le premier groupe de patron relève du parapluie des patrons de décomposition. Décomposer par sous-domaine et décomposer par fonctionnalité métier sont les deux principaux modèles de définition de l'architecture de microservice d'une application.

À côté de cela, nous pouvons suivre certains principes et directives de la conception orientée objet lors de l'application du modèle d'architecture de microservice, tels que le principe de responsabilité unique pour définir les responsabilités d'une classe et le principe de fermeture commun pour organiser les classes en packages.

4.4.1.1. Décomposer par capacité métier

La décomposition par capacité métier est l'une des stratégies pour concevoir un système basé sur l'architecture des microservices. Concept issu de la modélisation de l'architecture d'entreprise, une capacité métier est quelque chose qu'une entreprise fait pour générer de la

valeur. L'ensemble des capacités d'une entreprise donnée dépend du type d'entreprise. Par exemple, les capacités d'une compagnie d'assurance incluent généralement la souscription, la gestion des réclamations, la facturation, la conformité, etc. Les fonctionnalités d'une boutique en ligne incluent la gestion des commandes, la gestion des stocks, l'expédition, etc. (Richardson 2018). Les capacités métier sont ce qu'une entreprise fait et peut faire et se concentre sur le « quoi » et non sur le comment. Les capacités métier sont logiques, stables et fondamentales. Elles résument ce qu'une entreprise fait et peut faire à un niveau d'abstraction utile et pratique (Capstera 2022). Une capacité métier représente ce qu'une entreprise fait dans un domaine particulier pour remplir ses objectifs et ses responsabilités. Chaque microservice expose une API que les développeurs peuvent découvrir et utiliser en libre-service. Une fois les capacités métiers identifiées, nous pouvons définir un service pour chaque capacité ou groupe de capacités associées (Dehghani 2018).

4.4.1.2. Décomposer par sous-domaine

La conception pilotée par le domaine est une approche de développement logiciel qui aide les développeurs à se concentrer sur le domaine métier et ses complexités. Pour ce faire, il divise le système global en morceaux plus petits et plus gérables appelés sous-domaines (Vernon 2013). Chaque sous-domaine peut être décomposé en ses propres éléments gérables, jusqu'à ce que vous atteigniez le niveau des classes ou des objets individuels. Cela permet une compréhension plus granulaire du domaine métier, ce qui facilite l'écriture de code qui le reflète avec précision. Il existe trois types de sous-domaines :

- Domaines principaux : Un domaine principal est ce qui rend une organisation spéciale et différente des autres organisations, et c'est la partie la plus précieuse de l'application.
- Sous-domaines de support : Un sous-domaine spécialisé, est nécessaire pour des besoins spécifiques pour que l'organisation réussisse.
- Sous-domaines génériques : ils n'ajoutent rien de spécial à l'entreprise, mais sont nécessaires à la solution commerciale globale.

4.4.1.3. Décomposer par transactions

L'architecture des microservices est décomposée en transactions. Chaque transaction est atomique, cohérente, isolée et durable (ACID). Cela garantit que les données sont toujours dans un état cohérent et qu'aucune corruption de données ne se produit. Les transactions sont également isolées les unes des autres, ce qui signifie qu'une transaction n'affecte pas le résultat

d’une autre transaction. Cela permet de s’assurer que le système reste fiable et disponible à tout moment. Enfin, les transactions sont durables, ce qui signifie qu’une fois qu’elles ont été validées dans la base de données, elles ne peuvent pas être annulées ou perdues.

4.4.2. Passerelle d’API

Passerelle d’API est un patron de conception qui fournit un point d’entrée centralisé pour les API. Il agit comme une façade aux services sous-jacents et cache la complexité du système aux consommateurs. Cela permet de simplifier la gestion des API et d’améliorer les performances en mettant en cache les données fréquemment consultées.

Passerelle d’API peut être implémenté à l’aide de diverses technologies, telles que des serveurs Web, des files d’attente de messages ou des bus de services. Il se compose généralement de deux composants : un proxy d’API et un registre de service principal (Fowler 2002). Le proxy intercepte toutes les demandes des clients et les achemine vers les services principaux appropriés. Le registre stocke des informations sur tous les services principaux et leurs points de terminaison associés.

Comme on peut le voir sur la figure 15, le client envoie une requête pour obtenir une ressource du microservice 01. La requête est d'abord passée par la passerelle d’API pour la valider et la renvoyer au microservice 01, qui envoie la réponse à la passerelle d’API pour le livrer au client.

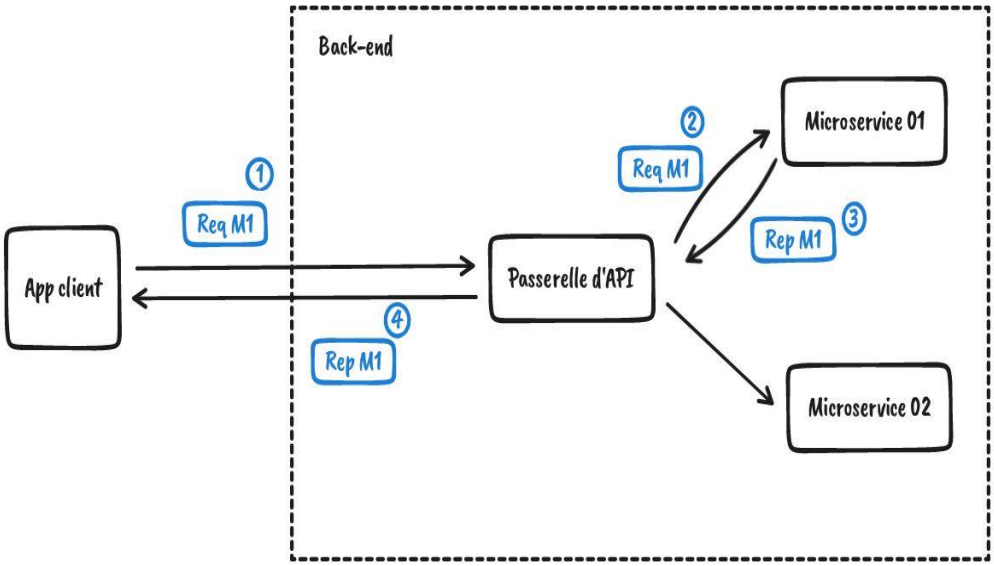


Figure 15 Passerelle d'API.

4.4.3. Patrons de communication

La communication diffère de l'application monolithique à l'application basée sur les microservices sous plusieurs aspects, alors que la première approche communique dans un système de processus unique via des appels de fonctions et de méthodes au niveau du langage, une application basée sur les microservices pourrait s'exécuter sur une grande variété de serveurs, d'hôtes et de processus qui représentent certains défis pour un système distribué solide. Par conséquent, les services doivent interagir à l'aide d'un protocole de communication inter-processus (IPC) tel que HTTP (HyperText Transfer Protocol), AMQP (Advanced Message Queuing Protocol) ou un protocole binaire comme TCP (Transmission Control Protocol), selon la nature de chacun des services. Les services peuvent utiliser un format binaire tel qu'Avro ou un format textuel lisible par l'homme tel que JSON ou XML (Anil 2022).

Ensuite, nous aborderons quatre modèles qui relèvent du modèle de communication et qui peuvent être utilisés dans une architecture de microservices. Après cela, nous présenterons deux autres modèles utilisés comme solution à certains problèmes liés aux styles de communication synchrones.

4.4.3.1. Requête-Réponse synchrone

Dans ce mécanisme, un service appelle une autre API de service via un protocole tel que HTTP ou gRPC. C'est ce qu'on appelle un modèle de messagerie synchrone car le demandeur attend une réponse du fournisseur avant de poursuivre le traitement. La connexion entre la requête et le fournisseur est considérée comme une connexion avec état puisque les deux points de terminaison conservent des informations sur la conversation elle-même tout au long de sa vie. La figure 16 montre d'abord la requête du service A à un autre service B qui répond après un intervalle de temps (Hohpe and Woolf 2003).

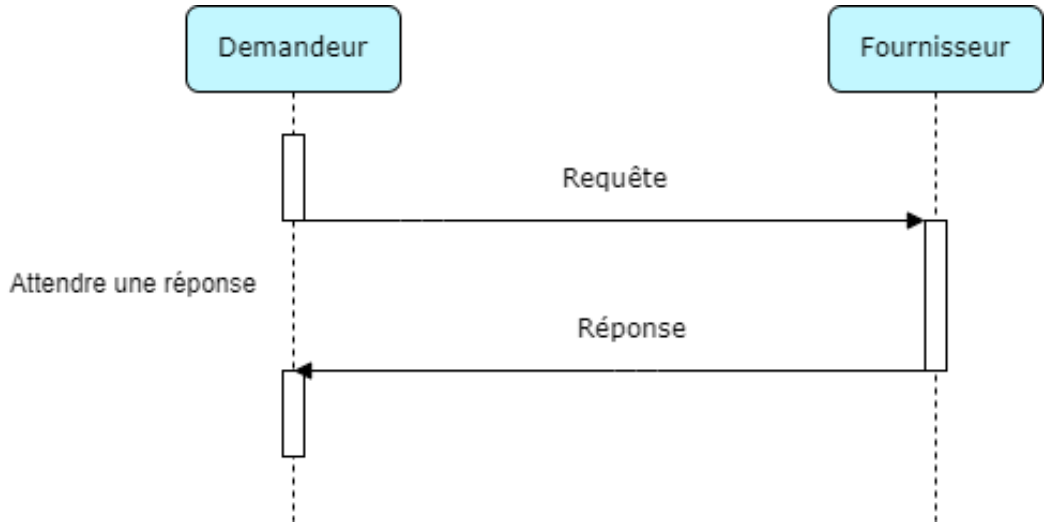


Figure 16 Requête-Réponse synchrone.

4.4.3.2. Requête-Réponse asynchrone

Dans ce mécanisme, le demandeur n'attend pas la réponse du fournisseur et il ne sauvegarde pas l'état de la conversation entre les deux, ce qui en fait un modèle sans état du point de vue du fournisseur (voir la figure 17). Ce modèle réduit le couplage par rapport au modèle précédent. Ordinairement, une infrastructure de messagerie basée sur un patron de messagerie est utilisée à côté de ce pattern pour traiter la file d'attente des messages dans la partie fournisseur (Hohpe and Woolf 2003).

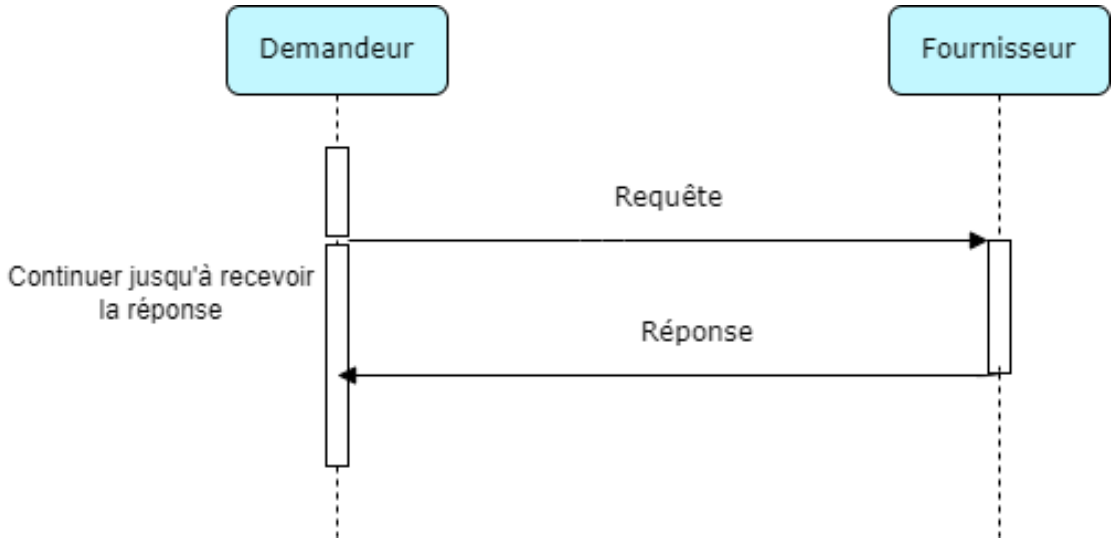


Figure 17 Requête-Réponse asynchrone.

4.4.3.3. Requête-Réponse avec nouvelle tentative

Un échec peut se produire après qu'un service a tenté d'accéder à une ressource à partir d'un autre service, ce qui peut affecter la stabilité du système. Les pannes incluent la perte momentanée de la connexion réseau aux composants et aux services, l'indisponibilité temporaire d'un service. Une solution à ce problème consiste à réessayer l'opération ou la transaction qui a échoué. Trois stratégies pour gérer ces échecs, en annulant la demande, en réessayant la demande infructueuse ou en réessayant la demande infructueuse après un certain temps. La troisième stratégie est considérée comme une bonne solution à la plupart des cas d'échecs.

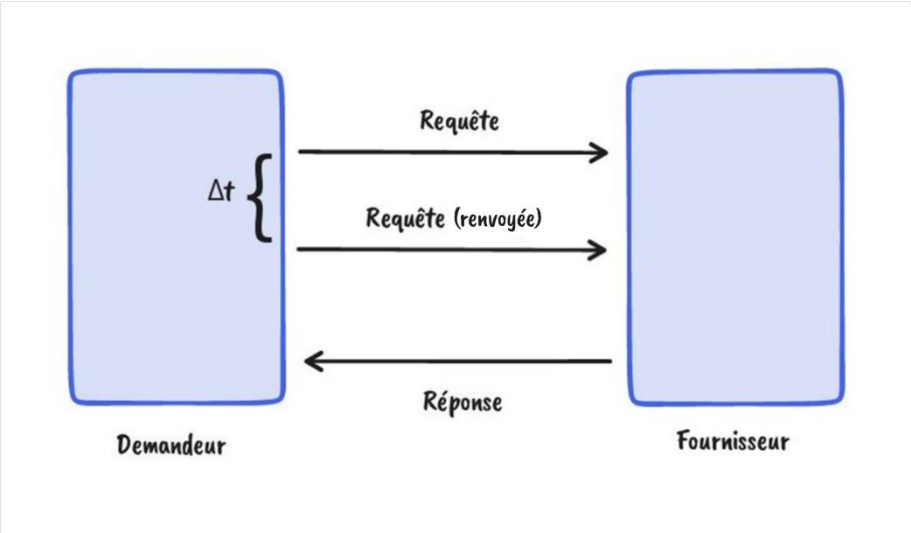


Figure 18 Requête-Réponse avec nouvelle tentative.

4.4.3.4. Publier-S'abonner

Publier-S'abonner est un modèle de messagerie dans lequel les composants, ou "éditeurs", publient des messages dans un "sujet" central et d'autres composants, ou "abonnés", s'abonnent à un ou plusieurs sujets et reçoivent les messages publiés selon le sujet. Cela permet un couplage faible entre les éditeurs et les abonnés. Lorsqu'un éditeur envoie un message, tous les abonnés le reçoivent. Ce modèle peut être implémenté à l'aide de diverses technologies, telles que les files d'attente et les agents de messages.

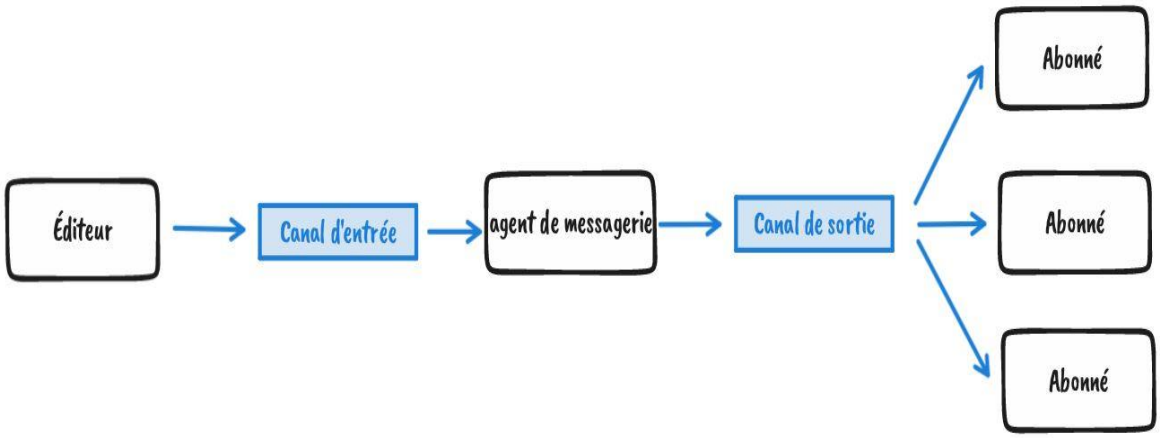


Figure 19 Publier-S'abonner.

4.4.3.5. Disjoncteur

Dans un système distribué tel qu'une application basée sur des microservices, une défaillance partielle est possible et un problème inévitable dans certaines circonstances doit être résolu. Une panne partielle doit être gérée de manière à diminuer le risque qu'elle se produise, et pour cela le disjoncteur vient comme une solution pour réduire l'impact d'une panne partielle dans un seul composant sur les autres microservices, et sur les performances du système global. L'idée de base d'un disjoncteur est de suivre le nombre de requêtes réussies et échouées, et si le taux d'erreur dépasse un certain seuil, le disjoncteur se déclenche de sorte que les tentatives ultérieures échouent immédiatement (Richardson 2018). Un grand nombre de requêtes qui échouent suggèrent que le service n'est pas disponible. Après un délai d'attente, le client doit réessayer et, en cas de succès, le disjoncteur se ferme.

Selon la figure 20, le disjoncteur a les trois états:

- Fermé : Lorsque tout est normal, le disjoncteur reste à l'état fermé et tous les appels transitent vers les services. Lorsque le nombre de pannes dépasse un seuil prédéterminé, le disjoncteur se déclenche et passe à l'état ouvert.
- Ouvert : Le disjoncteur renvoie une erreur pour les appels sans exécuter la fonction.
- Semi-ouvert : après une période de temporisation, le circuit passe à un état semi-ouvert pour tester si le problème sous-jacent existe toujours. Si un seul appel échoue dans cet état, le disjoncteur est à nouveau passe à l'état ouvert. S'il réussit, le disjoncteur revient à l'état fermé normal.

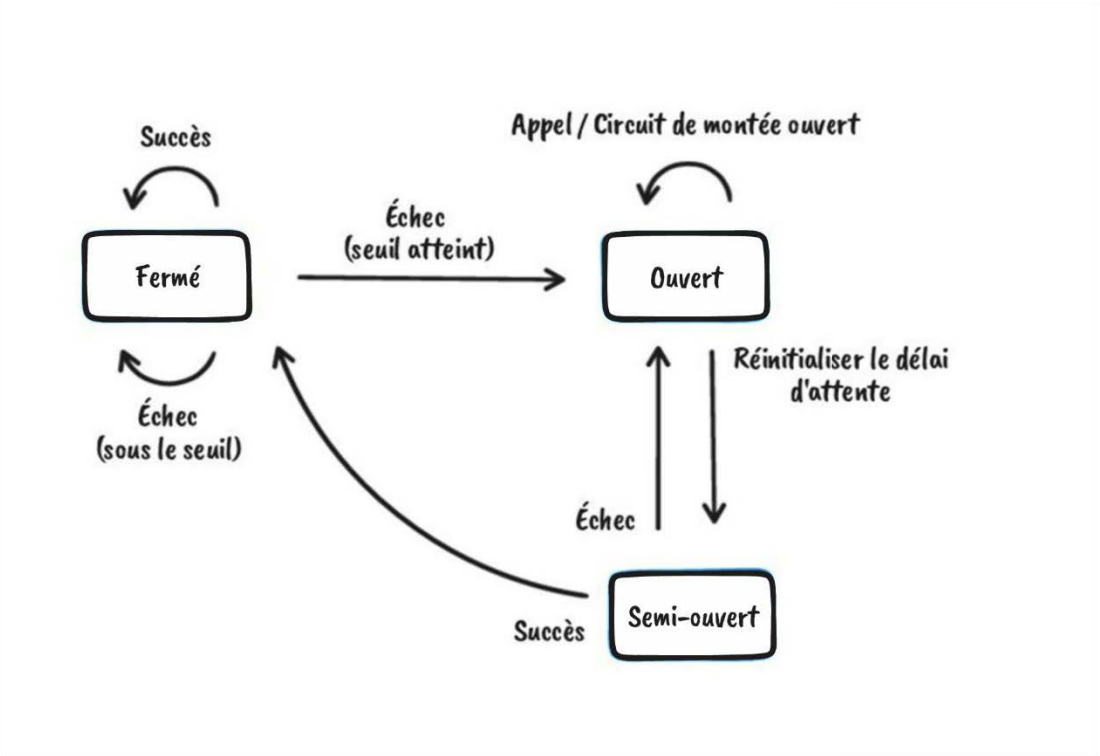


Figure 20 Disjoncteur.

4.4.3.6. Patron de découverte de service

La découverte de service est un modèle utilisé dans les systèmes distribués pour rechercher et se connecter à des services. Lorsqu'un client a besoin d'utiliser un service, il interroge le système de découverte de service pour obtenir des informations sur les services disponibles. Le système de découverte de service fournit ensuite au client une liste d'instances de service et leurs emplacements. Cela facilite l'ajout de nouveaux services et facilite la mise à l'échelle de l'architecture en ajoutant plus de serveurs pour des services individuels. La découverte de service permet également la tolérance aux pannes ; si un service tombe en panne,

les autres peuvent continuer à fonctionner. Deux types de modèle de découverte de service, le premier est le modèle de découverte côté client et le second est le modèle de découverte côté serveur.

Il existe plusieurs mécanismes de découverte de service différents, mais le plus courant est appelé registre. Un registre contient des informations sur tous les services du système, y compris leur emplacement et la manière dont ils sont accessibles. Lorsqu'un service a besoin de trouver un autre service, il interroge le registre pour obtenir des informations sur les options disponibles. Cela permet aux services de communiquer entre eux même s'ils ne savent pas où ils se trouvent ou comment se joindre directement.

4.4.3.7. Sidecar

Le modèle de service sidecar est un modèle de conception qui peut être utilisé pour améliorer la modularité et la résilience d'une application. Le modèle sidecar Service implique la création d'un service qui s'exécute parallèlement à l'application principale et qui fournit des fonctionnalités supplémentaires. Le modèle sidecar Service peut être requise par l'application principale, ou elle peut être fournie pour la commodité des utilisateurs finaux. Le modèle sidecar Service peut être implémenté à l'aide de diverses technologies, notamment les servlets Java, les modules Node.js et les scripts Python (voir la figure 21). Il peut également être mis en œuvre à l'aide de technologies basées sur des conteneurs telles que Docker ou Kubernetes.

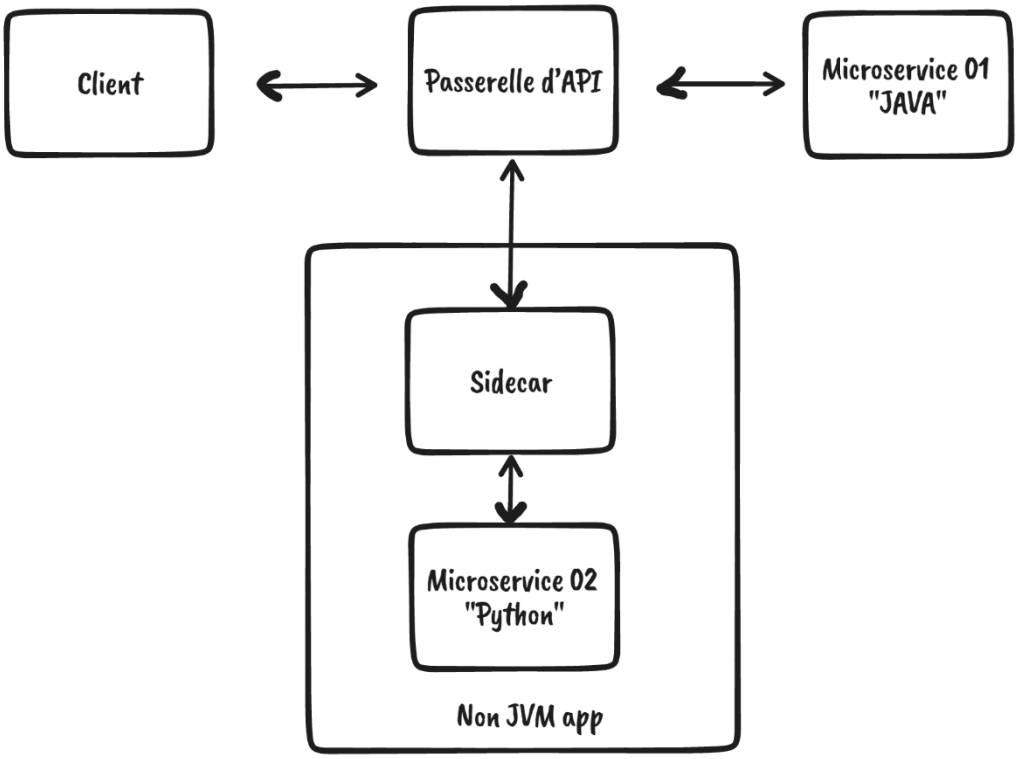


Figure 21 Sidecar pour l'intégration d'applications non JVM.

4.4.4. Patrons de gestion des données

Lorsque nous traitons des systèmes distribués et des aspects de l'architecture des microservices, le comportement du flux de données et de la sauvegarde des données évolue vers une perspective distribuée qui soulève de multiples exigences. Les patrons de gestion des données apportent des solutions à ces défis.

4.4.4.1. Base de données par service

Dans le modèle de base de données par service, chaque microservice possède une base de données distincte. Ce modèle est considéré comme le plus utilisé en matière de gestion des données dans l'architecture des microservices. Les transactions de chaque service se font dans le cadre de sa propre base de données. Les microservices communiquent entre eux pour accéder aux ressources de données via des appels d'API afin de mettre en œuvre leur logique métier. La communication entre les microservices lorsque nous appliquons ce modèle peut conduire à une topologie de type maillage, ce qui est une mauvaise pratique dans l'architecture des microservices qui préconise le principe de couplage faible, ainsi que le déploiement et la mise à l'échelle indépendants des microservices. C'est pourquoi il est très important de concevoir un

bon contexte de délimitation et de choisir la bonne approche de décomposition pour chaque microservice.

4.4.4.2. Base de données partagée

Le deuxième pattern est le pattern de base de données partagée. Comme son nom l'indique, il s'agit essentiellement de partager une base de données entre plusieurs services, ce qui permet d'appliquer les principes ACID (atomicité, cohérence, isolation, durabilité) dans les transactions de base de données pour faire respecter la validité et la cohérence des données même en cas de panne du système, de panne de courant, etc.

4.4.4.3. Saga

L'un des principaux défis est de maintenir la cohérence des données entre les services. Le pattern Saga est une solution à ce problème en implémentant des transactions qui couvrent plusieurs services.

Par définition, une Saga est une séquence de transactions locales chargées de mettre à jour les données au sein d'un service unique et d'une ressource de données (par exemple une base de données relationnelle ou un agent de messages). Un service publie un message asynchrone ou un événement après chaque transaction effectuée. Si une transaction échoue, la Saga annulera toutes les modifications précédentes et annulera toutes les transactions locales qui ont été validées. Pour ce faire, saga utilisera un autre type de transactions locales appelées « transactions de composition ». Ainsi, pour chaque transaction échouée, nous avons une transaction de compensation correspondante (Richardson 2018).

Deux approches sont à l'origine de la logique du pattern Saga et chargées de coordonner les transactions locales :

- Les sagas basées sur la chorégraphie.
- Sagas basées sur l'orchestration.

4.4.4.3.1. Saga basée sur la chorégraphie

Dans les sagas basées sur la chorégraphie, un agent d'événements ou de messages contrôle le message échangé entre les microservices dans un style de communication asynchrone. Pour cela, les microservices doivent s'abonner à ce canal et publier des événements

pour déclencher une transaction locale dans un autre service, également pour exécuter des transactions locales en fonction des événements dans l'agent (voir la figure 22).

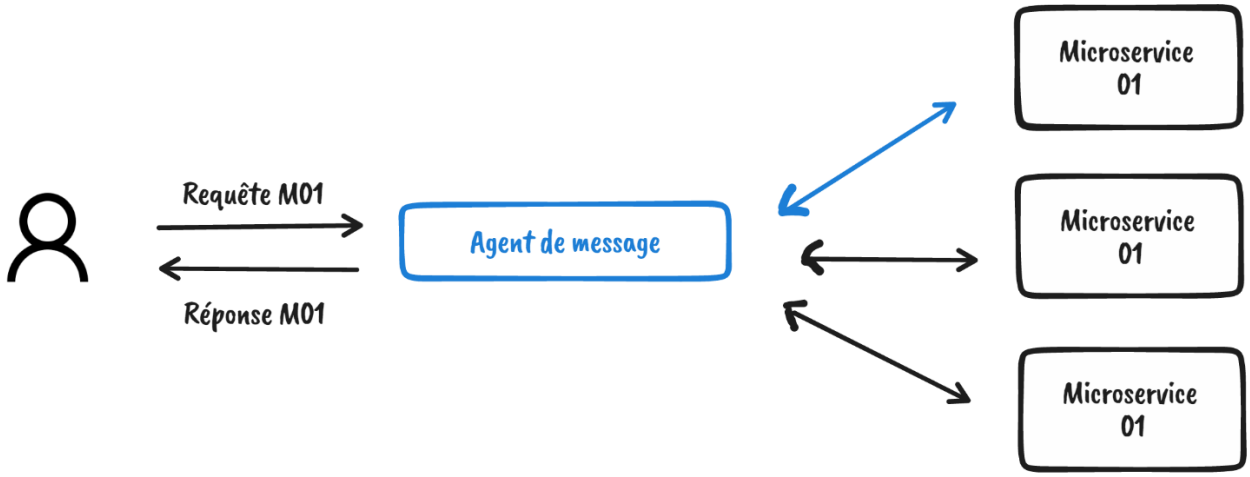


Figure 22 Saga basée sur la chorégraphie.

4.4.4.3.2. Saga basée sur l'orchestration

Dans les sagas basées sur l'orchestration, un service centralisé gère toutes les communications entre les microservices en décidant des transactions locales à exécuter par les participants à la saga dans d'autres microservices en fonction des événements (voir la figure 23). Le service d'orchestration est également chargé de gérer tout échec des transactions locales avec des transactions de compensation.

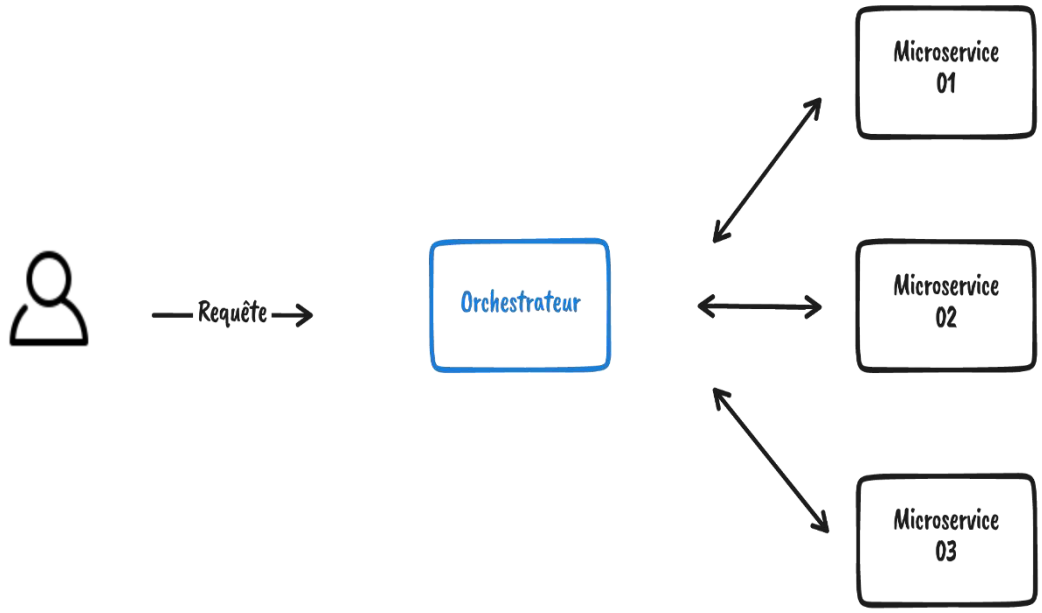


Figure 23 Saga basée sur l'orchestration.

4.4.4.4. Ségrégation des responsabilités entre commande et requête

La mise en œuvre de requêtes qui récupèrent des données à partir de plusieurs microservices était l'un des défis liés à la base de données par modèle de service. Dans un système de gestion de base de données relationnelle, le même système est responsable des demandes et des commandes comme moyen de modifier les données. Au contraire, un service basé sur la ségrégation des responsabilités entre commande et requête (Command Query Responsibility Segregation CQRS) sépare le côté commandes du côté requêtes, où commandes sont utilisées pour mettre à jour les données et les requêtes pour lire les données.

La ségrégation des responsabilités entre commande et requête, comme son nom l'indique, concerne la ségrégation ou la séparation des activités. Une partie distincte du service gère le modèle de domaine côté commandes en traitant et en effectuant les opérations CRUD. L'autre partie est responsable du modèle de domaine côté requêtes en accédant à des vues prédéfinies pour récupérer des données spécifiques. Le côté commandes publie des événements chaque fois que les données changent à l'aide du pattern Event Sourcing pour faire face à certaines limitations fournies avec l'approche CRUD. Les vues côté requête CQRS sont le résultat de l'abonnement à un canal d'événements publiés par un ou plusieurs services via la partie côté commandes, et il renvoie les dernières mises à jour et modifications. Une bonne pratique consiste à implémenter la vue côté requête CQRS en tant que service autonome (Richardson 2018).

4.4.4.5. Event Sourcing

Comme nous l'avons mentionné dans la section Modèle CQRS, nous pouvons utiliser le pattern Event Sourcing pour publier des événements chaque fois que les données changent. L'Event Sourcing appartient à la fois à la section de la gestion des données et des modèles de communication. L'Event Sourcing garantit que les actions qui conduisent l'état des données sont stockées sous la forme d'une séquence ou d'un flux d'événements plutôt que d'un objet de domaine, ces événements étant considérés comme des objets immuables. Tout d'abord, l'application stocke les événements dans un magasin d'événements, puis publie ces événements pour qu'ils soient consommés par d'autres consommateurs de services. Le magasin d'événements se comporte comme un agent de messages en stockant et en traitant les événements et en fournissant une API aux consommateurs (Fowler 2002).

4.5. Sécurité

Jusqu'à présent, nous avons discuté de l'architecture des microservices avec les avantages et les défis qui l'accompagnent, et quelques patrons de conception pour résoudre ces défis, nous allons maintenant aborder l'aspect sécurité. La question que nous devons nous poser est de savoir comment nous sécurisons une application avec un architecture de microservices, devons-nous sécuriser uniquement l'API Gateway ou chaque microservice, et comment nous sécurisons les communications entre les différentes parties de l'application et les microservices. La plupart de ces problèmes se concentrent sur deux aspects de sécurité essentiels, les aspects d'authentification et d'autorisation. Pour cela, nous présenterons quelques techniques et approches comme solutions à ces questions.

4.5.1. Authentification

Dans une architecture basée sur des microservices, un microservice individuel chargé de maintenir l'authentification est considéré comme un moyen pratique et une meilleure solution pour valider et vérifier les informations d'identité. Le processus centralisé d'authentification peut être géré sur la base de deux méthodes, une authentification basée sur une session ou une authentification basée sur un jeton, les deux méthodes pouvant être utilisées ensemble.

4.5.1.1. Authentification basée sur la session

Authentification basée sur la session également appelée authentification avec état, est une méthode permettant de vérifier les informations d'identification de l'utilisateur pour accéder à une ressource côté serveur de l'application. L'idée est de stocker la plupart des informations d'identification de l'utilisateur dans une session dans le serveur (voir la figure 24). Ainsi, lorsqu'un client envoie une demande pour une ressource, le serveur doit vérifier les informations d'identification de l'expéditeur dans la session. Dans une architecture de microservices, une bonne pratique consiste à vérifier les informations d'identification au niveau de la passerelle API. Les sessions peuvent être stockées sous forme de cookies ou de base de données en mémoire. L'authentification basée sur la session est simple et facile à mettre en œuvre, mais l'un de ses inconvénients est le manque d'évolutivité.

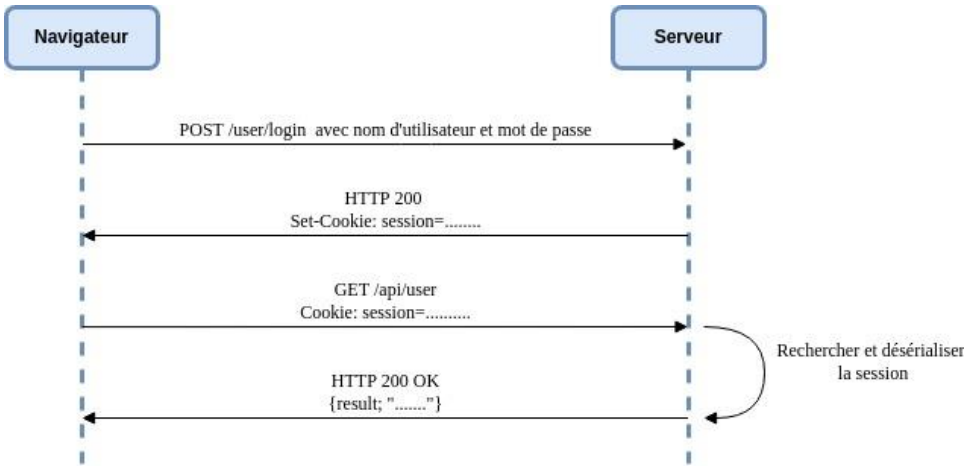


Figure 24 Authentification traditionnelle basée sur les cookies.

4.5.1.2. Authentification basée sur les jetons

Dans une authentification basée sur les jetons, aucune session n'est requise, les informations d'identification de l'utilisateur sont échangées contre un jeton généré par le serveur et conservé côté client, ce que nous appelons une authentification sans état (voir la figure 25). Chaque requête du client contient un jeton dans l'en-tête permettant au serveur de vérifier l'authenticité de la requête. JSON Web Token (JWT) est une norme conçue pour implémenter cette méthode. Comme nous le mentionnons dans la méthode d'authentification basée sur la

session, une bonne pratique consiste à vérifier le JWT dans la passerelle API avant de transmettre la demande au microservice spécifique.

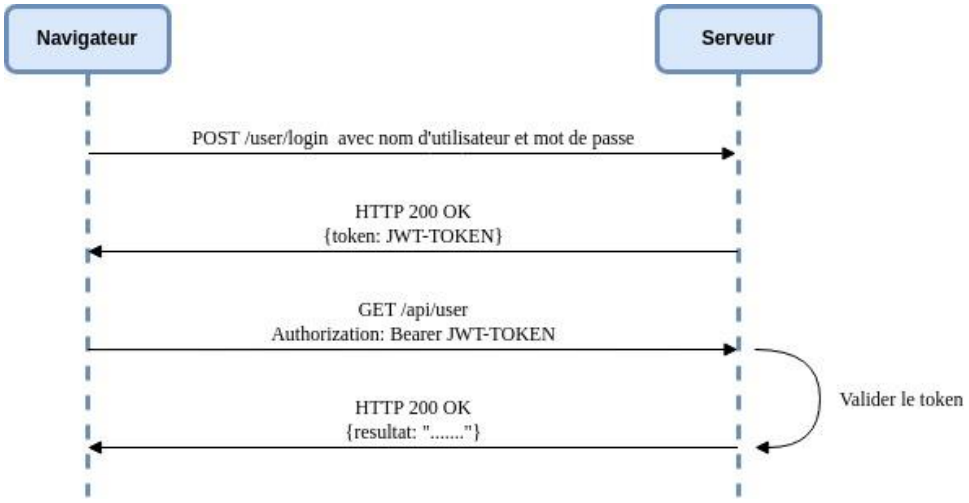


Figure 25 Authentification basée sur des jetons.

4.5.2. Autorisation

L'autorisation est une méthode qui renforce l'aspect sécurité en plus de l'authentification. Une application met en œuvre un mécanisme d'autorisation pour vérifier que le client a la permission d'effectuer l'opération demandée. L'idée principale est de restreindre ou d'autoriser certains clients (par exemple, navigateur, application mobile) ou utilisateurs (par exemple, administrateur) à accéder à une ressource. La passerelle API est l'une des parties de l'application pour implémenter l'autorisation, comme cela peut être dans un microservice individuel.

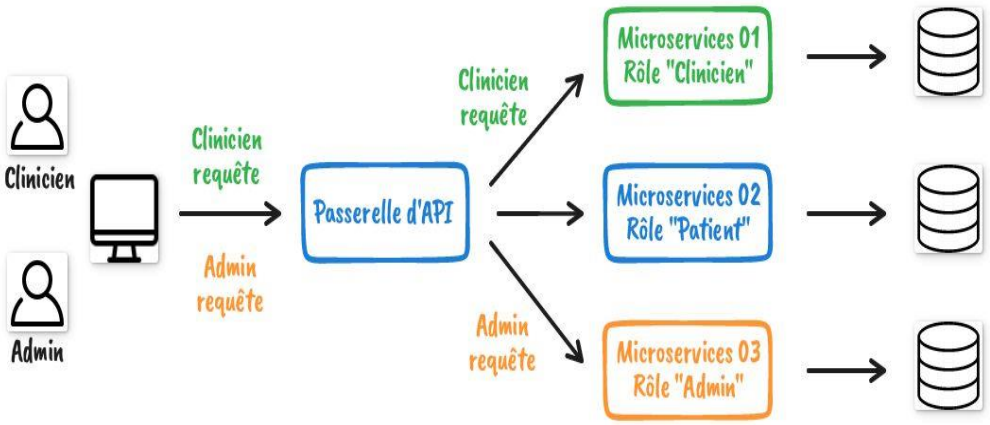


Figure 26 Autorisation exemple.

Chapitre 5 – Conception de POD-iSanté et évaluation de modèles bayésiens

5.1. Introduction

Dans ce chapitre, nous présenterons notre application POD-iSanté basée sur AP-SADC, qui inclut la modélisation et l'architecture, nous expliquerons les façons dont nous avons décidé d'implémenter les patrons de microservices. Ensuite, nous plongerons dans l'implémentation du réseau de neurones bayésien, et nous comparerons différents modèles et évaluerons différents aspects. Il convient de mentionner que l'approche consiste à disposer des données initiales des patients pour créer notre première version du modèle, puis après un certain temps, nous aurions plus de données qui permettraient de créer une deuxième version de notre modèle pour améliorer notre système. En l'absence de données initiales des patients, nous montrerons ensuite une simulation de la manière dont nous aurions créé la première version du modèle.

5.2. Architecture

Nous avons développé notre application côté serveur en mettant en œuvre les normes d'architecture de microservices à l'aide du Spring cloud framework. Nous avons déjà discuté de certains patrons de microservices, nous allons maintenant expliquer comment nous avons mis en œuvre ces modèles. En commençant par la gestion des données, nous avons choisi d'utiliser le patron de base de données par service, pour cela nous avons créé quatre services avec quatre bases de données : Utilisateur, Patient, Fitbit et IA.

5.2.1. Utilisateur

Le service utilisateur gère les informations d'identification, l'authentification et les autorisations de l'utilisateur. Quatre rôles principaux sont l'administrateur, le clinicien, le patient et un chercheur, chacun avec un ensemble spécifique d'autorisations pour accéder aux ressources et aux données.

5.2.2. Patient

Le service patient gère les informations personnelles du patient et son dossier médical ainsi que ses cliniciens et ses rendez-vous. Le lien entre le patient et son dossier médical a été crypté pour plus de sécurité.

5.2.3. Fitbit

Le service Fitbit est chargé de contrôler le flux de données provenant des podomètres et de le stocker. Il gère également la relation entre notre système et l'API Fitbit, comme l'autorisation et la gestion des appareils.

Pour assurer une livraison et un déploiement continus dans notre projet, nous avons choisi de décomposer notre application en quatre services en suivant la décomposition par modèle de capacité métier (voir la figure 27). Chaque service est petit et facile à tester et à entretenir, et ses parties appartiennent à la même logique métier.

Ensuite, nous avons utilisé Spring cloud pour implémenter le reste de notre application comme :

- Patron de découverte de service: Eureka Server et Eureka Clients.
- Disjoncteur: Hystrix.
- Passerelle d'API et routage: Zuul.
- Agent de messagerie: Spring AMQP et RabbitMQ.
- Sidecar: Spring Sidecar.
- Réessayer les demandes ayant échoué: Spring Retry.
- Authentification/Autorisation et sécurité: Spring Oauth2 avec JWT.
- Application Client: Angular.

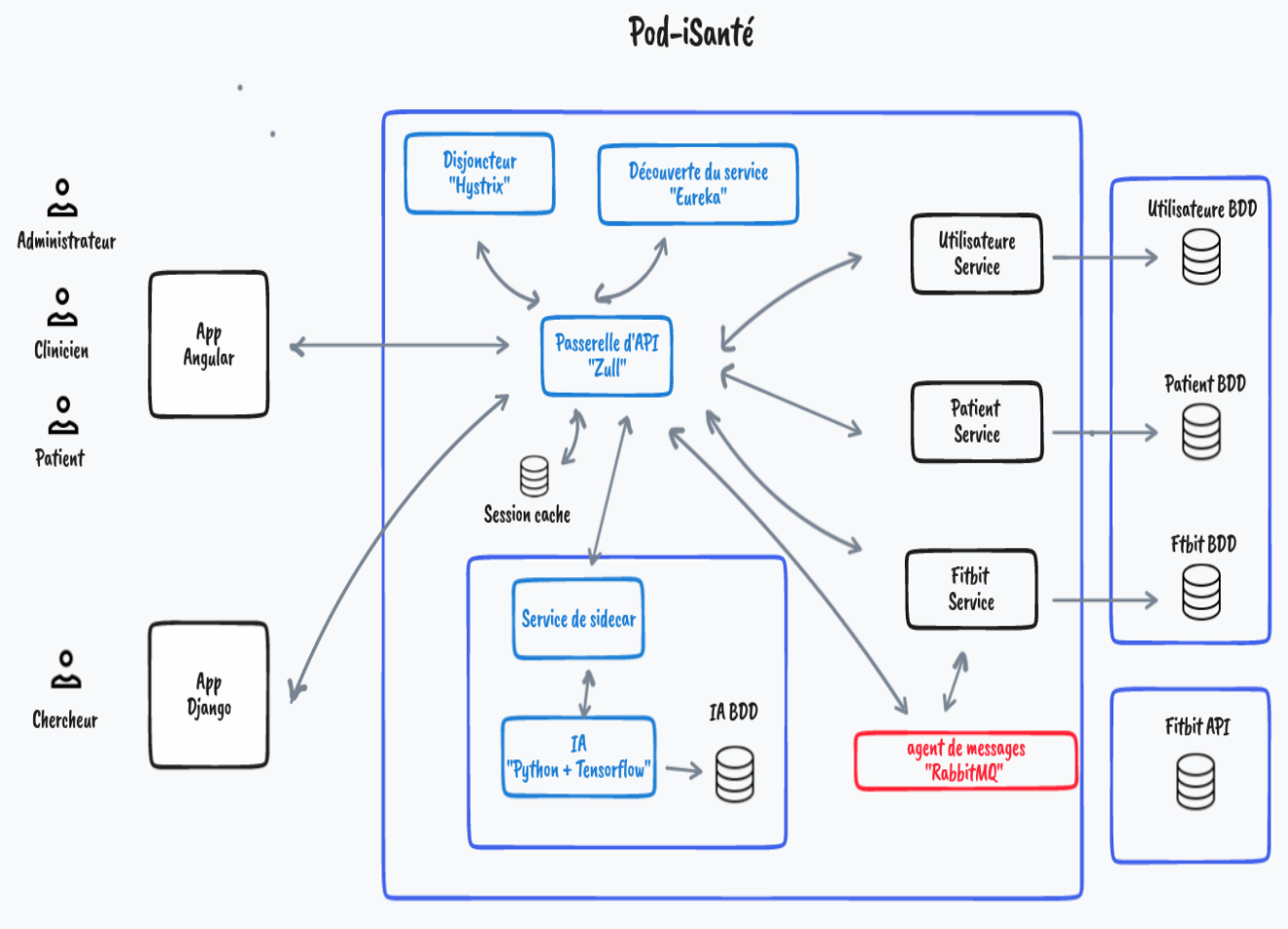


Figure 27 Architecture microservices du POD-iSante.

5.3. Analyse de POD-iSanté

Ensuite, nous utiliserons deux types de diagrammes UML pour mieux comprendre l'application. Tout d'abord, nous avons utilisé un diagramme de cas d'utilisation qui regroupe quatre acteurs principaux et leurs actions envers le système, plus un autre diagramme pour l'affectation d'un podomètre, et un autre pour autoriser un podomètre. Ensuite, à l'aide du diagramme de séquence, nous décrivons une ressource principale des données dans notre système.

5.3.1. Diagramme de cas d'utilisation

Ensuite, nous avons trois diagrammes de cas d'utilisation, le premier regroupe tous les acteurs et leurs actions envers le système (voir la figure 28).

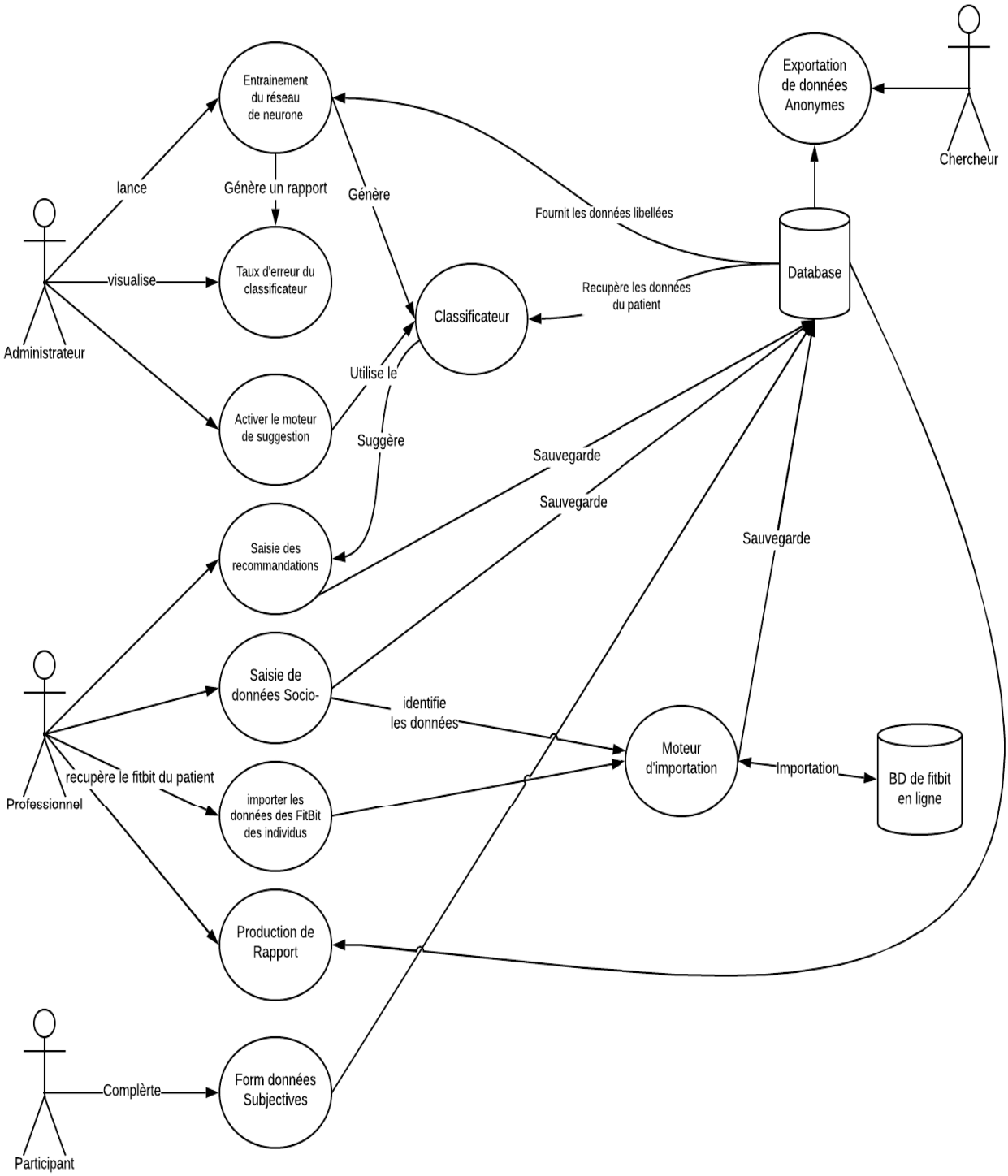


Figure 28 Diagramme de cas d'utilisation global.

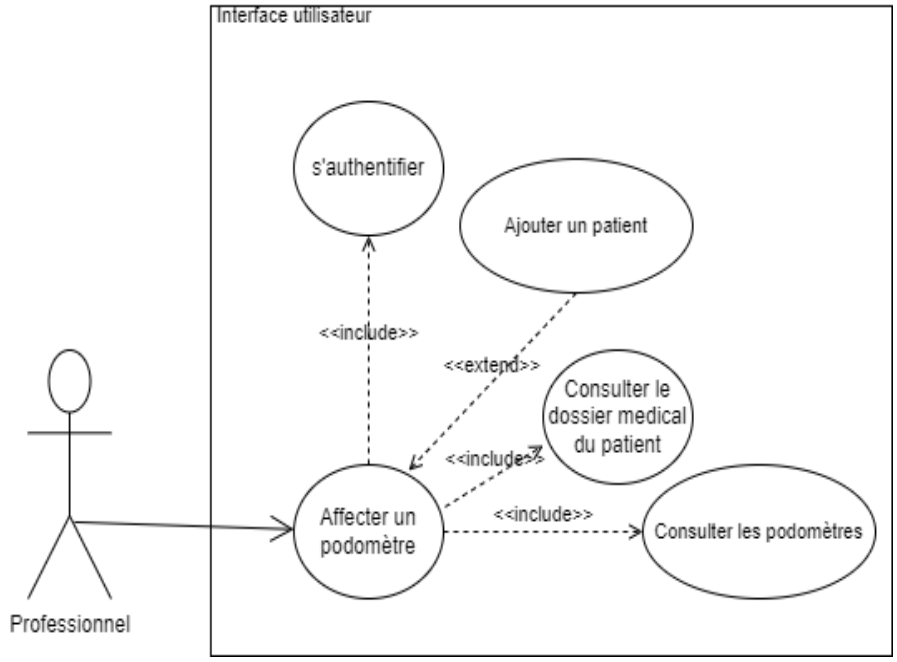


Figure 29 Affecter un podomètre.

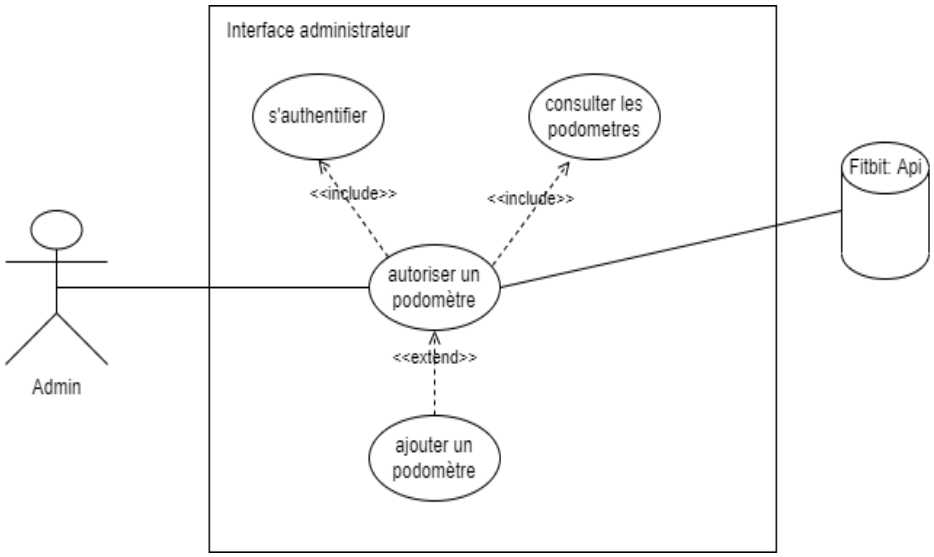


Figure 30 Autoriser un podomètre.

5.3.2. Diagramme de séquence

Ensuite, nous avons un diagramme de séquence qui explique comment les données du podomètre ont été récupérées à partir de l'API Fitbit après que le patient a synchronisé ses données.

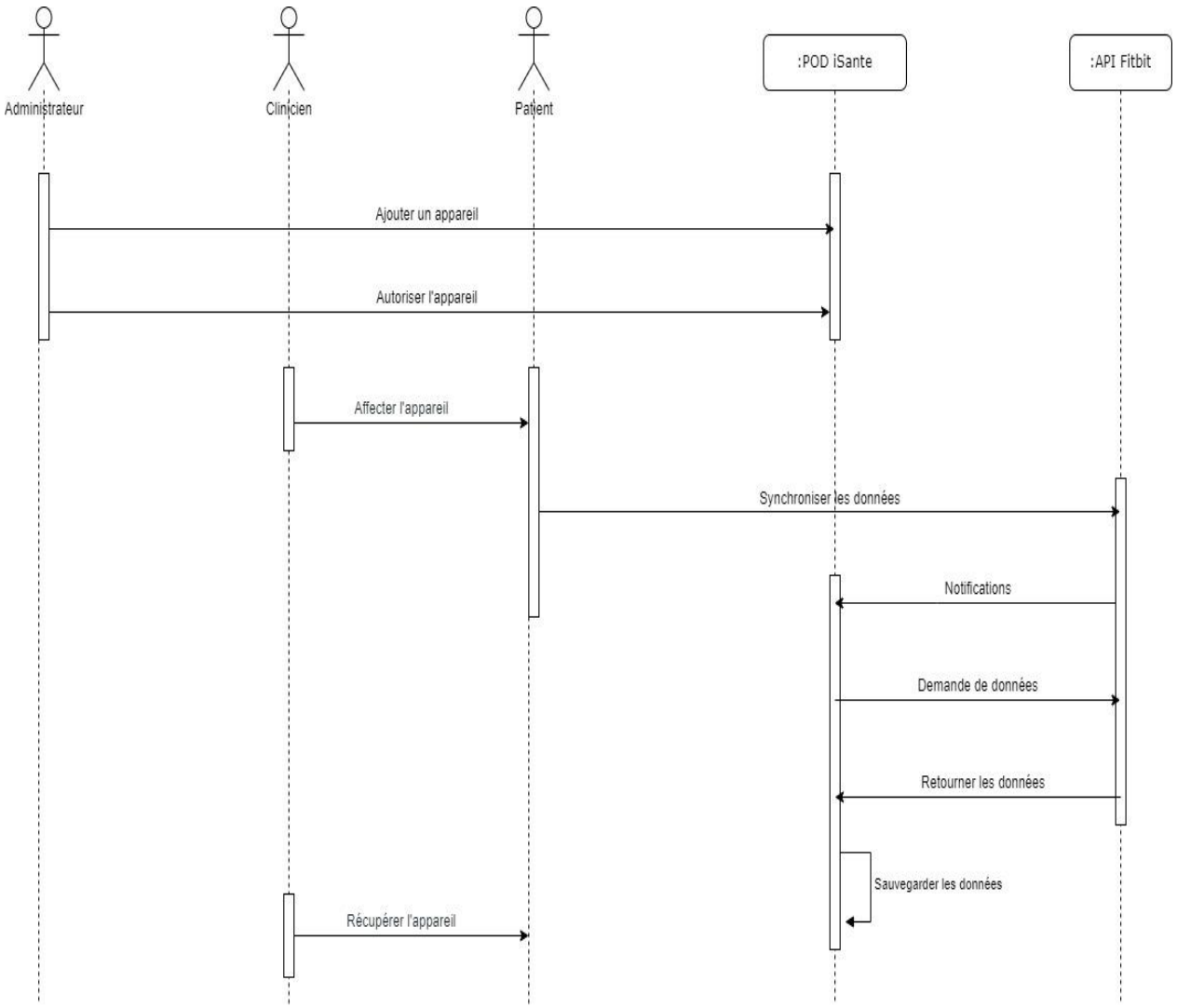


Figure 31 Gestion des données Fitbit.

5.4. Les données

Parce que nous n'avons pas obtenu les données des patients, nous avons décidé d'utiliser le même type de données tabulaires en utilisant les données d'une compétition de Kaggle « Forest Cover Type Prediction » (Newman and Asuncion 2007) pour simuler le même flux de travail que nous aurions fait avec les données des patients. Les échantillons de cet ensemble de données correspondent à des parcelles de forêt de 30 × 30 m aux États-Unis, collectées dans le but de prédire le type de couverture de chaque parcelle, c'est-à-dire l'espèce d'arbre dominante. Il existe sept types de couverture, ce qui en fait un problème de classification multiclasse. Chaque échantillon comporte 54 attributs. Certaines des caractéristiques sont des indicateurs booléens, tandis que d'autres sont des mesures discrètes ou continues (Forest covertypes 2016). Pour le

processus d'apprentissage, nous divisons les données en données d'apprentissage avec 80 % du total des données et 20 % pour les données de validation.

Il s'agit donc de classer l'échantillon dans l'une des 7 classes représentant les types de couverture. Ceci est important pour les gestionnaires des ressources naturelles pour prendre une décision sur les stratégies de gestion des écosystèmes en fonction du type de terrain, mais souvent ils n'ont pas ces informations sur les terres voisines en dehors de leur juridiction. Par conséquent, ils obtiendraient les informations à l'aide de modèles prédictifs (Edinburgh n.d.).

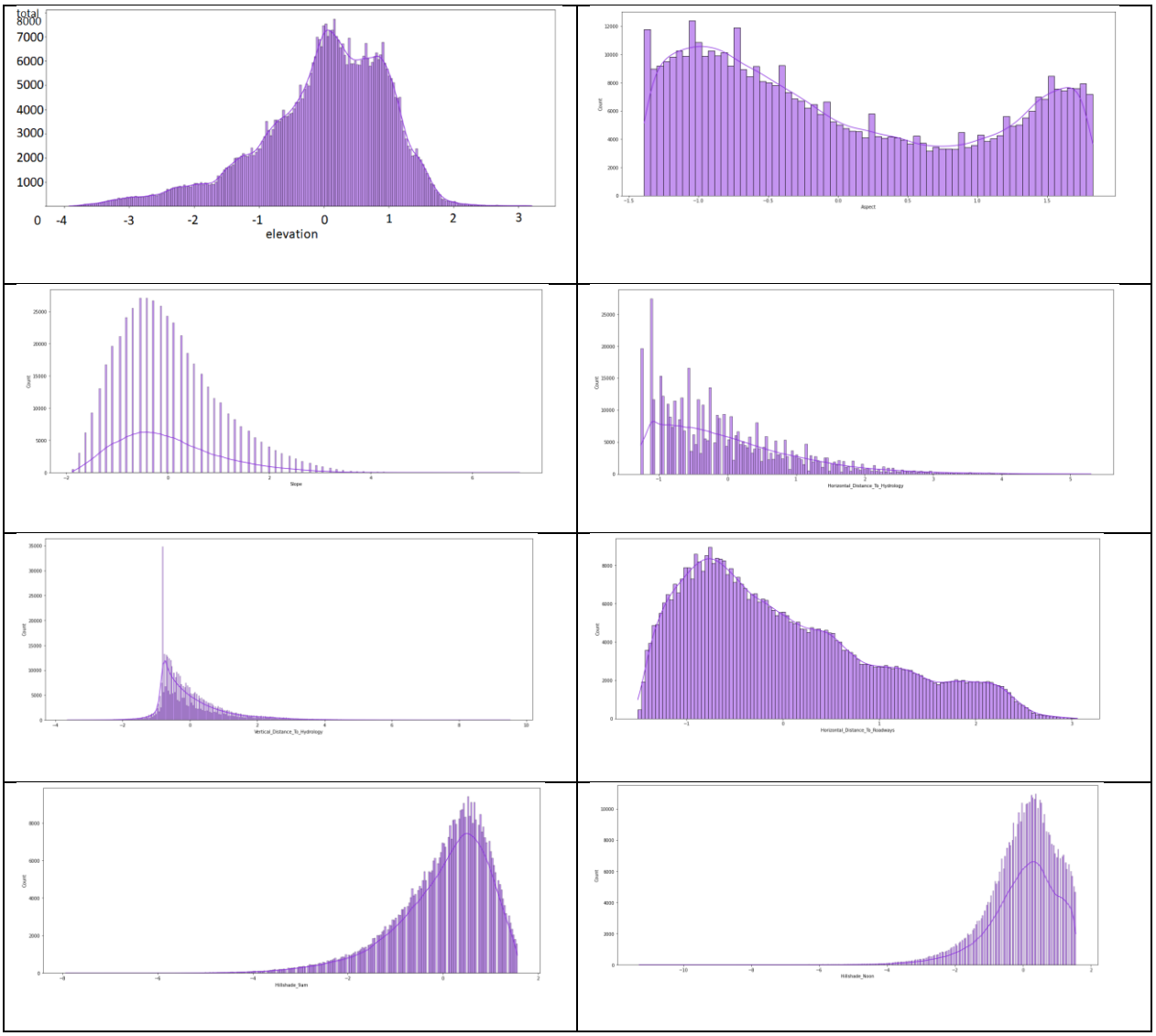
Les classes sont assez déséquilibrées, mais il n'y a pas de valeurs manquantes. Les attributs sont un mélange de valeurs numériques et catégorielles. Pour cela, un simple prétraitement des variables nominales aux variables binaires a été effectué. Le nombre d'instances (une instance est une seule ligne dans une table de données) est de 581012, tandis que le nombre d'attributs est de 54.

Nom	Description
Elevation	Altitude en mètres
Aspect	Aspect en degrés d'azimut
Slope	Pente en degrés
Horizontal_Distance_To_Hydrology	Distance horizontale par rapport aux plans d'eau de surface les plus proches
Vertical_Distance_To_Hydrology	Distance verticale par rapport aux plans d'eau de surface les plus proches
Horizontal_Distance_To_Roadways	Distance horizontale à la chaussée la plus proche
Hillshade_9am	Indice d'ombrage à 9h, solstice d'été
Hillshade_Noon	Indice d'ombrage à midi, solstice d'été
Hillshade_3pm	Indice d'ombrage à 15h, solstice d'été

Horizontal_Distance_To_Fire_Points	Distance horizontale jusqu'aux points d'allumage des feux de forêt les plus proches
Wilderness_Area (4 colonnes binaires)	Désignation de zone sauvage
Soil_Type (40 colonnes binaires)	Désignation du type de sol
Cover_Type (7 types)	Désignation du type de couvert forestier

Tableau 2 Description de l'ensemble de données (Forest Cover Type).

Ensuite, nous avons les distributions des attributs numériques.



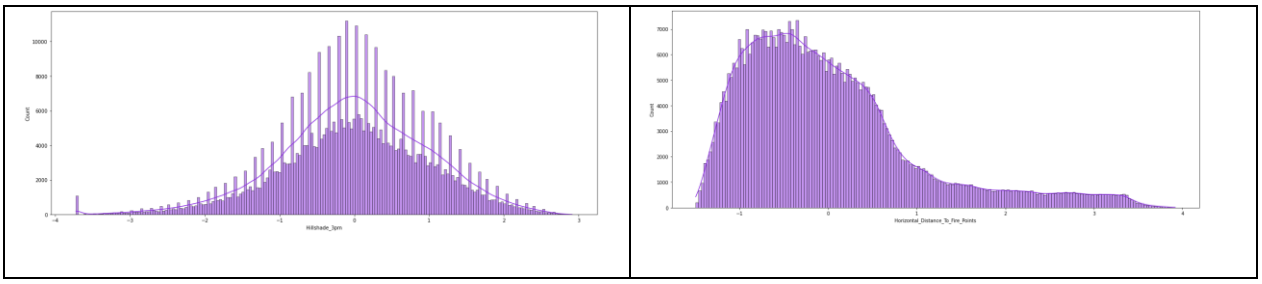


Tableau 3 Distribution des attributs continus des données d'entraînement.

Nous pouvons observer dans la matrice de corrélation qu'il y a peu ou pas de corrélation entre les attributs continus et le type de couverture.

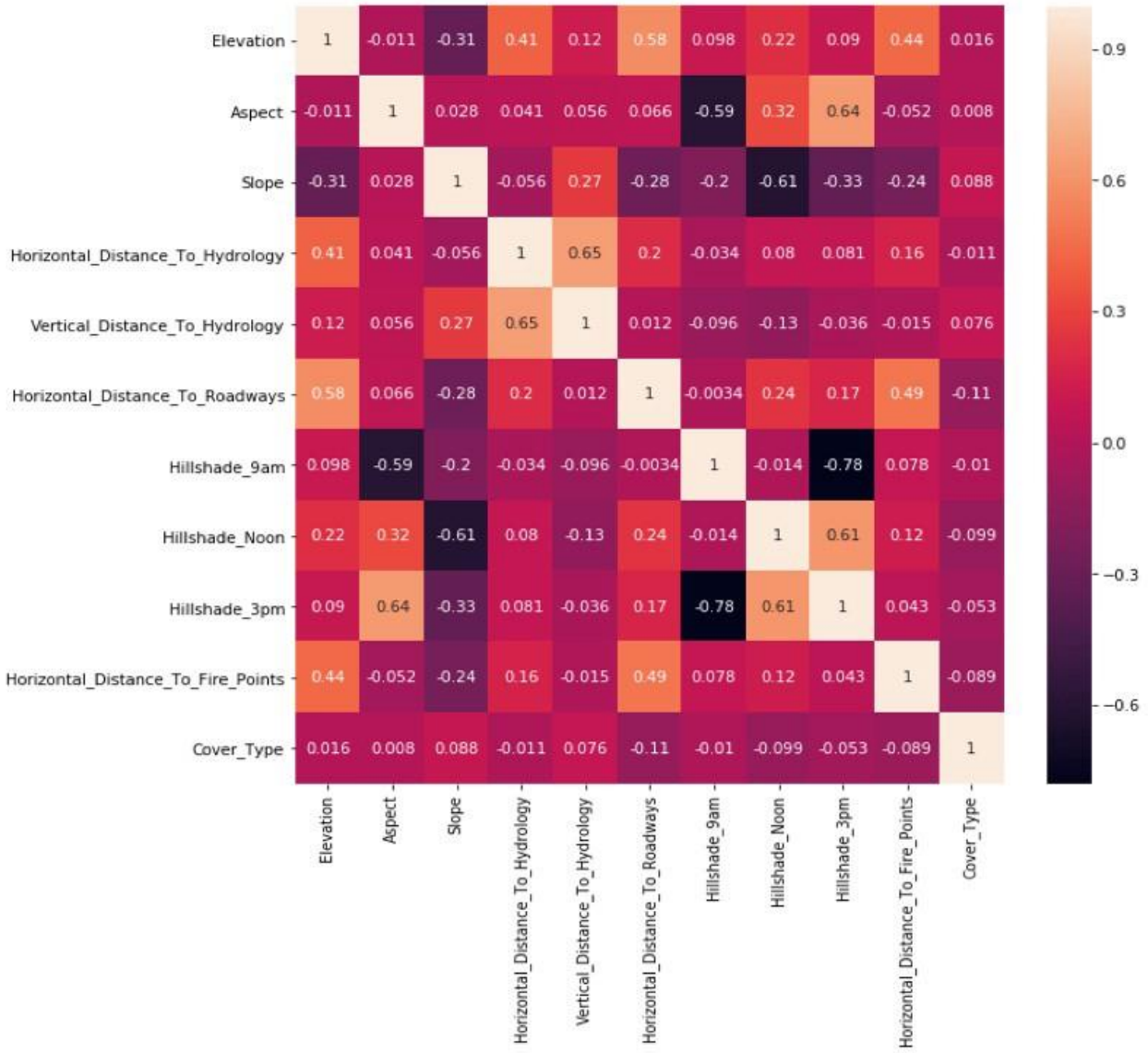


Figure 32 Matrice de corrélation.

Ensuite, nous avons les données distribuées par le type de couverture « Cover_Type », comme nous le voyons dans la figure 33.

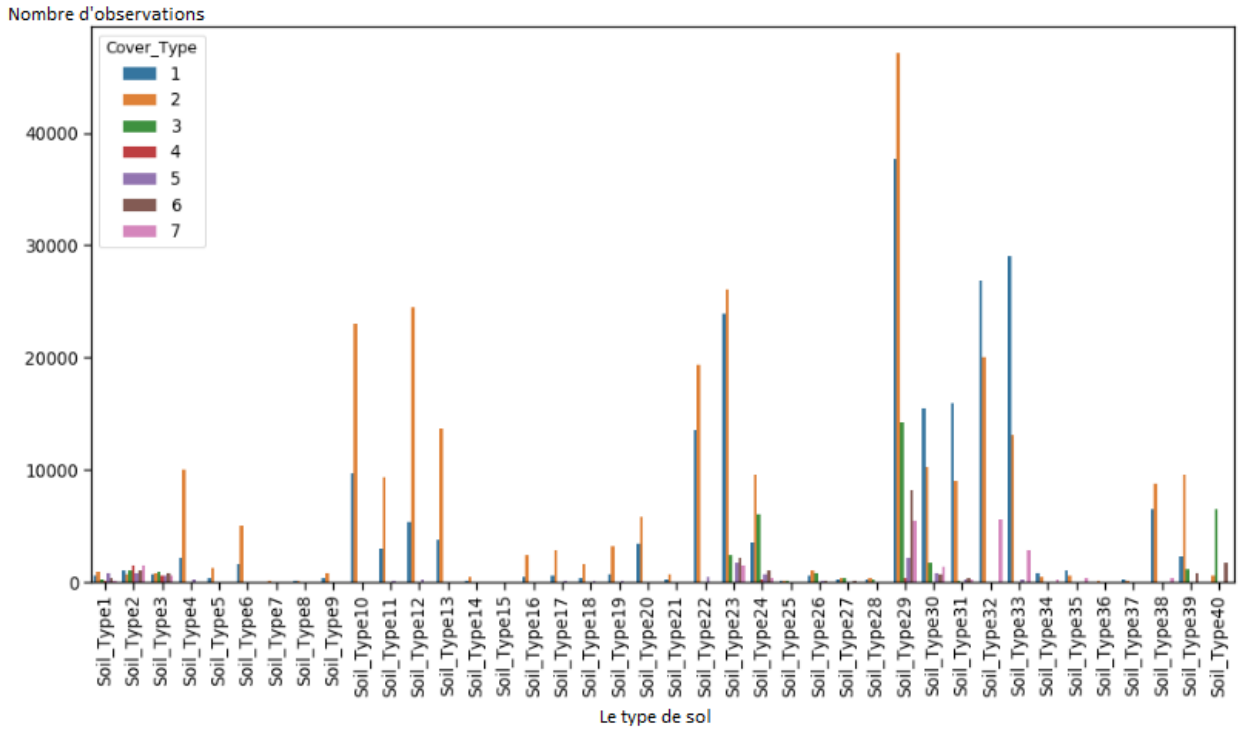


Figure 33 Nombre d'observations par type de couvert et de sol.

5.5. La sélection de l'a priori

La sélection de l'a priori est sans doute l'une des questions les plus délicates de l'inférence bayésienne. Comme nous l'avons mentionné précédemment, il peut être très difficile de trouver le bon a priori, pour cela nous avons choisi de tester 5 a priori différents, chacun d'eux est un a priori conjuguée à la distribution a posteriori.

Nous pouvons classer les a priori en deux catégories :

- **Les a priori informatifs** qui représentent nos solides connaissances a priori.
- **Les a priori non informatifs** qui représentent notre ignorance complète ou une vague connaissance d'une variable. Typiquement, dans ce cas, nous voulons choisir un a priori conjugué.

Dans nos cinq a priori, ce sont tous des a priori conjuguées, ce qui signifie qu'ils sont dans la même famille de distribution de probabilité avec la distribution a posteriori. Les deux premiers

sont des a priori non informatifs et les trois derniers sont des a priori informatifs. Il convient de mentionner que l'empirique de Bayes et la diagonale normale sont des méthodes empiriques de Bayes où la distribution a priori est estimée à partir des données. Cette approche contraste avec les méthodes bayésiennes standard, où la distribution a priori est fixée avant que les données ne soient observées.

5.5.1. Mélange de distributions gaussiennes SMP

Le premier a priori que nous avons sélectionné a été proposé dans l'article BBB (Bayes by Backprop), et c'est un mélange de deux distributions gaussiennes (en anglais: Scale mixture prior). Cet a priori est le mélange d'échelle (variance) de deux densités gaussiennes, chaque densité est de moyenne nulle et de variances différentes (voir la figure 34). Cet a priori s'écrit comme suit:

$$P(w) = \prod_j \pi \mathcal{N}\left(w_j|0,\sigma_1^2\right) + (1-\pi)\mathcal{N}(w_j|0,\sigma_2^2)$$

Où w_j est le $j^{\text{ième}}$poids du réseau, $\mathcal{N}\left(w_j\middle|0,\sigma_2^2\right)$ est la densité gaussienne évaluée en x de moyenne μ et de variance σ^2 et σ_1^2 et σ_2^2 sont les variances des distributions de mélange. De plus, la première gaussienne de mélange a une variance plus grande que la seconde $\sigma_1^2 > \sigma_2^2$, fournissant une queue plus lourde dans la densité a priori qu'une a priori gaussienne simple comme nous le voyons dans la figure 35. En plus de cela, le deuxième mélange gaussien a une petite variance $\sigma_2^2 \ll 1$ qui conduit à ce que les poids se concentrent a priori étroitement autour de zéro (Blundell et al., 2015). Il convient de mentionner qu'un mélange de deux a priori conjugués est également conjugué.

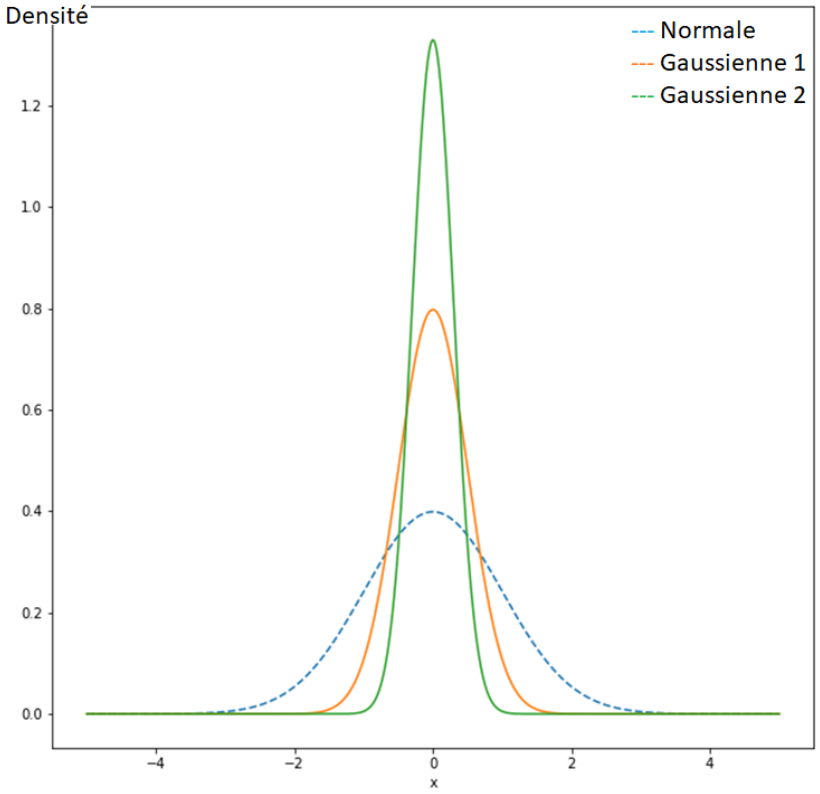


Figure 34 Exemple de Spike-and-slab de deux distributions normales.

5.5.2. Spike and Slab SSP

L'a priori précédent s'inspirait des distributions Spike and Slab. Lors de l'utilisation de l'a priori Spike-and-slab, les paramètres sont partagés entre tous les poids lors de l'optimisation par descente de gradient stochastique. Les paramètres partagés ressemblent à l'utilisation de méthodes de maximum de vraisemblance de type II (alias Bayes empiriques) pour apprendre ces paramètres à partir des données.

L'a priori Spike and Slab est un mélange de deux distributions : une distribution avec un pic autour de zéro pour les petits poids (Spike) et une distribution vague pour les grands poids (Slab) (van Erp, 2020).

Dans notre cas, nous avons utilisé une distribution Spike and Slab d'un mélange de deux gaussiennes. Le premier a un écart type de 1 et le second un écart type de 10 (voir la figure 35). Nous pouvons observer à partir de la figure 36 que le Spike est très pointu avec une faible variance, et que le Slab est très large avec une forte variance, ce qui permet aux valeurs éloignées de 0 d'être plus probables et aide le modèle à explorer un espace de poids plus grand.

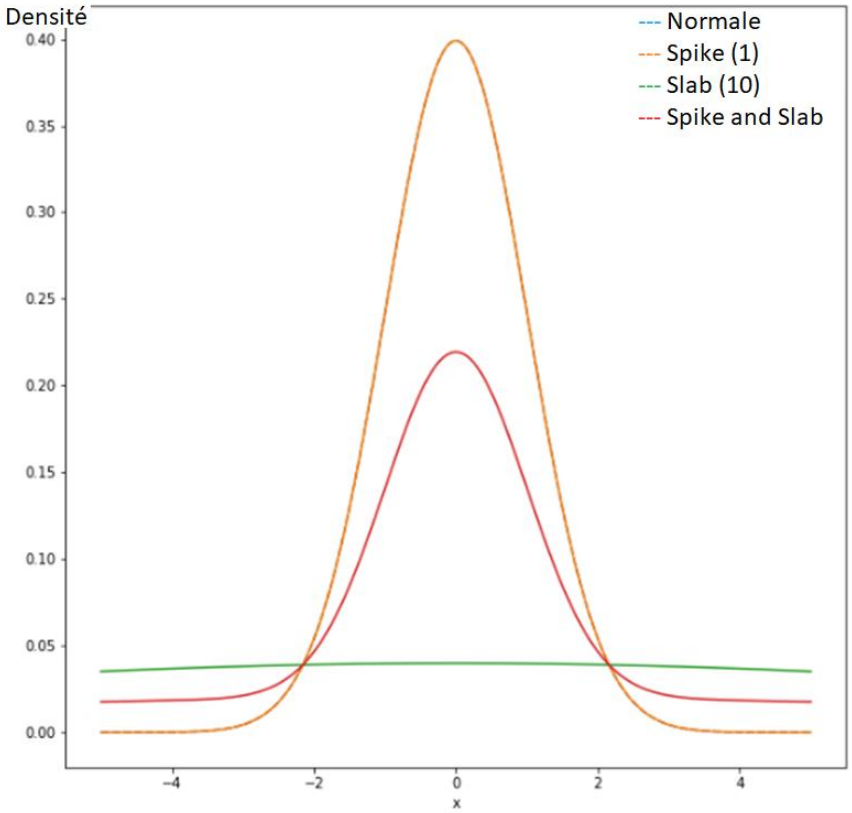


Figure 35 Exemple de Spike and Slab de deux distributions normales.

5.5.3. Bayes empirique EBP

Bayes empirique estime l'a priori à partir de l'ensemble des données en maximisant la vraisemblance marginale et en calculant les estimations ponctuelles des hyperparamètres paramétrant notre $P(\theta)$ à l'aide du maximum de vraisemblance de type II (à ne pas confondre avec le MLE standard où nous maximisons $P(X|\theta)$). Donc, nous formons et optimisons à la fois $P(\theta)$ et $Q(\theta)$.

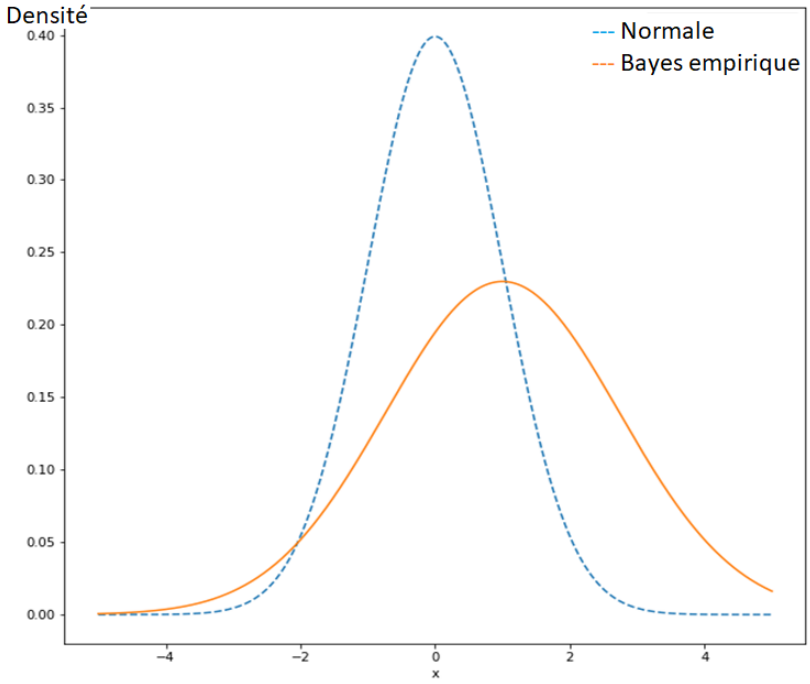


Figure 36 Un exemple de distribution normale.

5.5.4. Normale isotrope INP

Une normale isotrope est une distribution normale univariée de variance égale à 1 (ou dans le cas de distribution normale multivariée où la matrice de covariance est représentée par la matrice diagonale $\sum = \sigma^2 I$). De plus, nous avons initialisé la moyenne à zéro plutôt que de l'optimiser, et nous pouvons remarquer que sa fonction de densité de probabilité ressemble à la fonction de densité de probabilité unitaire normale, comme nous le voyons dans la figure 37.

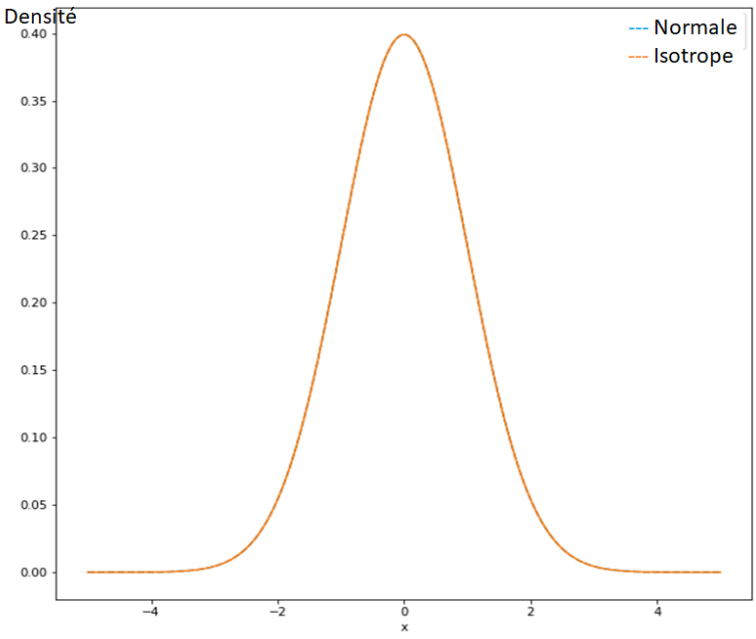


Figure 37 Normale isotrope.

5.5.5. Normale diagonale DTP

L'a priori normal diagonal (en anglais : Diagonal normal prior) est le même que l'a priori isotrope, avec une moyenne entraînable. Nous avons donc choisi d'optimiser la moyenne lorsque nous entraînons le modèle, ce qui rend le modèle plus biaisé vers les données que l'a priori.

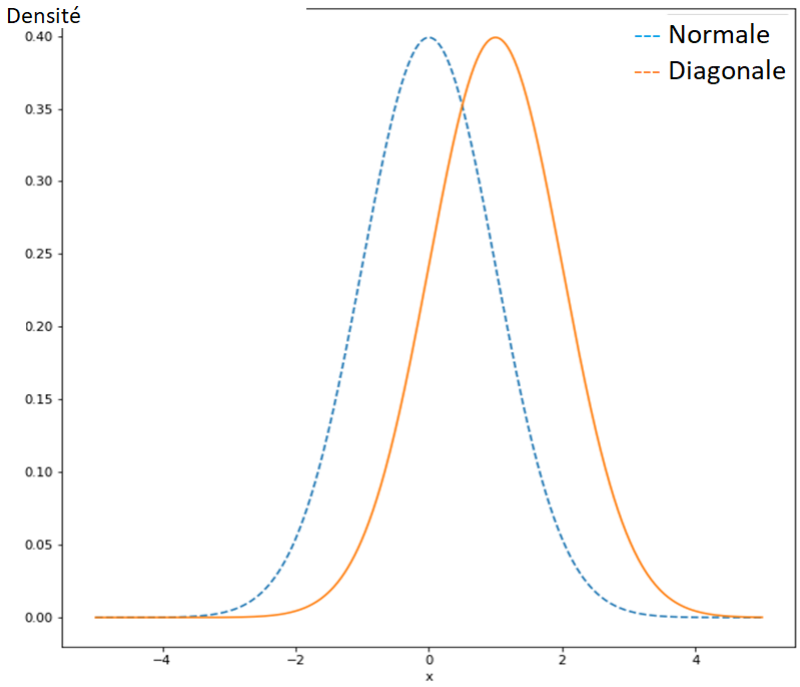


Figure 38 Normale diagonale.

5.6. Architecture et hyperparamètres du réseau

Comme nous en avons discuté précédemment, nous utiliserons l'optimisation bayésienne comme algorithme pour trouver les hyperparamètres dans les architectures de notre réseau de neurones (voir la figure 39). Nous avons spécifié le nombre d'itérations pour exécuter la boucle d'échantillonnage à partir du modèle GP. Pour le réseau de neurones bayésien, nous avons réduit de moitié le nombre d'itérations, la raison en est que le réseau de neurones bayésien a généralement plus de paramètres à optimiser que les standards.

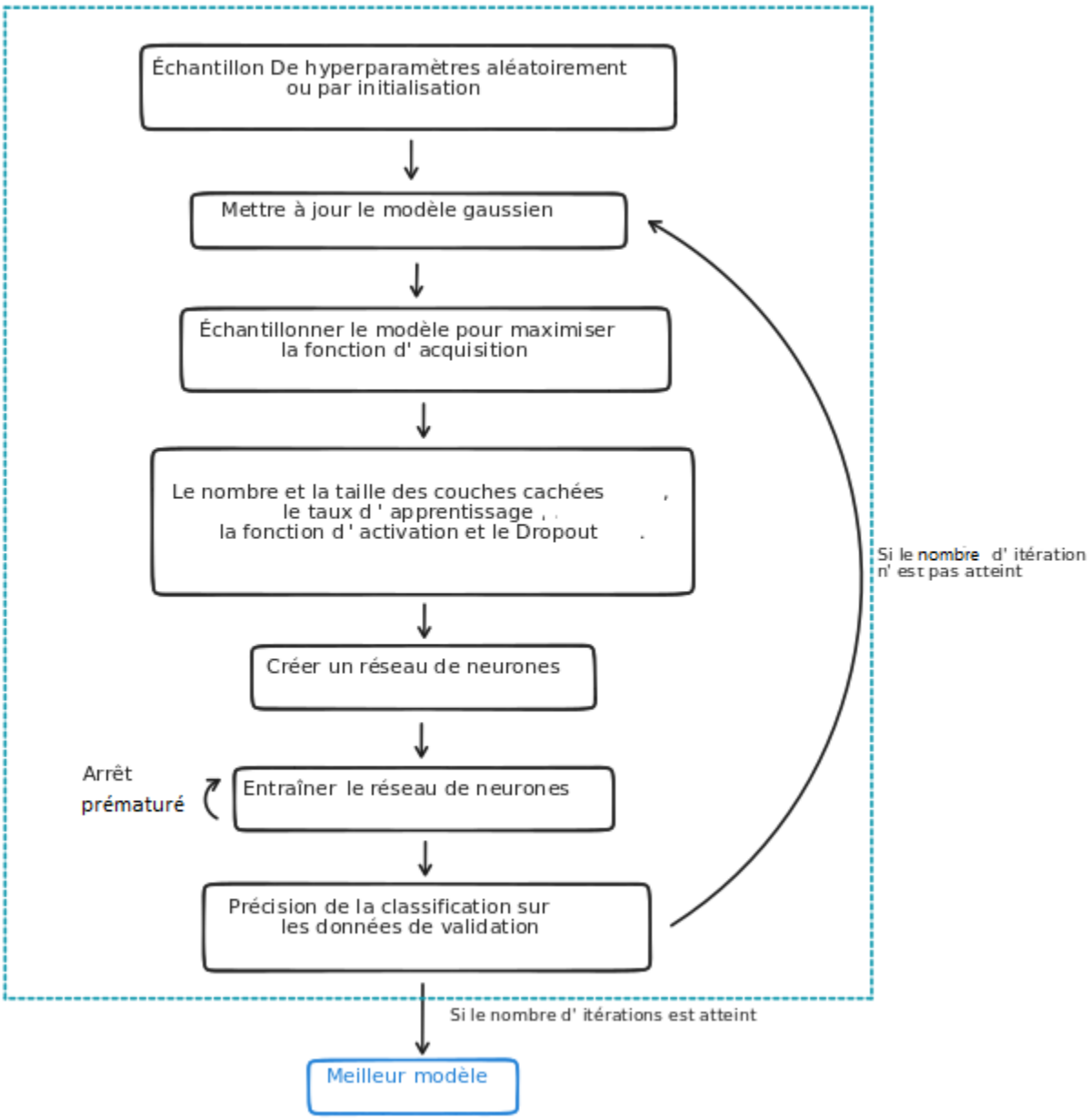


Figure 39 Workflow d'optimisation bayésienne.

Pour nos expériences, nous avons créé deux modèles différents. Le premier modèle pour deux réseaux de neurones standard, le premier utilise le SGD comme optimiseur et dans le second nous avons ajouté le dropout technique après chaque couche cachée. Ensuite, nous avons créé un deuxième modèle pour créer un réseau de neurones bayésien. Nous avons utilisé ce modèle pour créer cinq architectures avec différentes distributions de probabilité a priori.

5.6.1. Optimisation bayésienne

Pour trouver les meilleurs hyperparamètres pour nos architectures, nous avons utilisé *Scikit-optimize*. Scikit-optimize est une bibliothèque simple et efficace pour minimiser les

fonctions de boîte noire coûteuses et bruyantes. Ensuite, nous expliquerons l'implémentation *Scikit-optimize* de l'optimisation bayésienne.

5.6.1.1. Processus gaussiens

La bibliothèque propose une implémentation des processus gaussiens *gp_minimize*. Comme nous en avons discuté précédemment, BO a besoin d'un modèle probabiliste pour la fonction objectif et dans notre cas, nous avons choisi un modèle de processus gaussien, d'autre part, il a besoin d'une fonction d'acquisition. Pour cela, *Scikit-optimize* propose une implémentation pour ces deux éléments enveloppés dans le package *gp_minimize*. Les deux fonctions principales de *gp_minimize* sont *GaussianProcessRegressor* et *Optimiser*. Nous avons besoin de *GaussianProcessRegressor* principalement pour initialiser la fonction de covariance k que nous avons choisi. Dans nos expérimentations nous avons opté pour la fonction de covariance Matern :

$$k_{Matern}(x_i, x_j) = \frac{1}{\Gamma(v)2^{v-1}}\left(\frac{\sqrt{2v}}{l}d(x_i, x_j)\right)^{v} K_v\left(\frac{\sqrt{2v}}{l}d(x_i, x_j)\right)$$

La valeur par défaut du paramètre v est 1,5, qui contrôle le lissage de la fonction résultante. Plus v est petit, moins la fonction approchée est lisse. De plus, $d(\cdot,\cdot)$ est la distance euclidienne, $K_v(\cdot)$ est une fonction de Bessel modifiée et $\Gamma(\cdot)$ est la fonction gamma (scikit-learn, Matern n.d.).

Pour intégrer le bruit, on ajoute à la fonction de covariance Matern la valeur de la fonction de covariance White :

$$k_{\text{White}}(x_1, x_2) = \begin{cases} niveau\ de\ bruit, & si\ x_i == x_j \\ 0, & sinon \end{cases}$$

5.6.1.2. Fonctions d'acquisition

Il existe 3 fonctions d'acquisition implémentées dans *Scikit-optimize*, la fonction d'amélioration attendue (en anglais : Expected improvement EI), la fonction de limite de confiance inférieure (en anglais : Lower confidence bound LCB) et la fonction de probabilité d'amélioration (en anglais : Probability of improvement PI). Pour maximiser la probabilité de trouver les meilleurs hyperparamètres d'échantillonnage suivants, nous utiliserons *gp_hedge* qui choisit de manière probabiliste l'une des trois fonctions d'acquisition précédentes à chaque

itération. Chaque fonction d'acquisition est optimisée indépendamment, puis les meilleurs points sont choisis (scikit-optimize n.d.).

Limite de confiance inférieure (LCB)

Le LCB est défini comme :

$$LCB(x) = \mu_{GP}(x) + \kappa\sigma_{GP}(x)$$

Le paramètre positif k fait un compromis entre l'exploitation dans les régions où $\mu(x^*)$ est grand et l'exploration dans les régions où l'incertitude $\sigma(x^*)$ est grande.

Probabilité d'amélioration (PI)

Cette fonction est basée sur l'amélioration de la variable aléatoire (Noè & Husmeier, 2018). L'idée ici est d'essayer de trouver le meilleur point probable x à évaluer sans évaluer$f(x)$. Donc, pour cela, nous lui attribuons $I(x)$:

$$I(x) = \max\left(f_{min} - f(x), 0\right)$$

Où f_{min} est la meilleure valeur de fonction actuelle à l'itération n. $I(x)$ attribue une récompense d'amélioration de $f_{min} - f(x)$ si $f(x) < f_{min}$, et zéro sinon. Ensuite, le point x avec la probabilité la plus élevée d'amélioration par rapport à la meilleure valeur de fonction actuelle f_{min} est sélectionné, ce qui correspond à la maximisation de la probabilité $P(I(x) > 0)$, puis la fonction d'acquisition de probabilité d'amélioration (PI) est définie comme :

$$PI(x) = P(I(x) > 0) = P(f(x) < f_{min}) = P\left(\frac{f(x) - \hat{f}(x)}{s(x)} < \frac{f_{min} - \hat{f}(x)}{s(x)}\right)$$
$$= \Phi\left(\frac{f_{min} - \hat{f}(x)}{s(x)}\right)$$

Où $\hat{f}(x)$ et $s(x)$ sont la moyenne et la covariance du GP, et Φ est la fonction de distribution cumulative.

Amélioration attendue (EI)

Le PI ne prend en considération que la probabilité d'amélioration sans l'ampleur de l'amélioration. L'amélioration attendue (EI) est une fonction d'acquisition alternative qui rend compte de l'ampleur de l'amélioration et est obtenue en faisant la moyenne sur l'utilité $u(x) =$

$f_{min} - f(x)$ quand $f(x) < f_{min}$ et 0 sinon, d'où $u(x) = I(x)$. La fonction d'acquisition EI correspond à l'espérance de la variable aléatoire $I(x)$ et est égale à (Noè & Husmeier, 2018):

$$EI(x) = \mathbb{E}(I(x))=(f_{min} - \hat{f}(x))\Phi(u) + s(x)\phi(u)$$

Où ϕ est la fonction de densité de probabilité.

5.6.2. Arrêt prématuré

L'arrêt prématuré est une technique utilisée dans l'apprentissage automatique pour éviter le surajustement. Ceci est accompli en mettant fin à un processus d'apprentissage lorsque l'erreur prédite sur les données retenues ne s'améliore plus. Cela peut être déterminé soit par un nombre fixe d'époques, soit par une certaine mesure de réduction d'erreur. Les avantages d'un arrêt prémature sont qu'il peut améliorer la précision et accélérer le temps d'entraînement. Cela évite également de gaspiller des ressources pour former davantage un modèle qui fonctionne déjà aussi bien que possible compte tenu des données disponibles.

Cela peut être fait manuellement ou automatiquement en surveillant les performances du modèle sur les données de validation ou en utilisant un seuil pour la quantité d'erreur pouvant être tolérée.

Le tableau 4 représente la valeur de chaque paramètre de la fonction de rappel tensorflow « EarlyStopping » :

Monitor	Min_delta	Patience	Baseline	Mode
'val_accuracy'	0.0005	5	0.30	'max'

Tableau 4 Initialisation de la fonction EarlyStopping.

La figure 40 est un exemple d'utilisation de l'arrêt prématuré. On peut observer que l'utilisation de l'arrêt prématuré permet de réduire les coûts et de minimiser le temps d'entraînement en arrêtant le cas où les résultats sont plus mauvais que le meilleur connu.

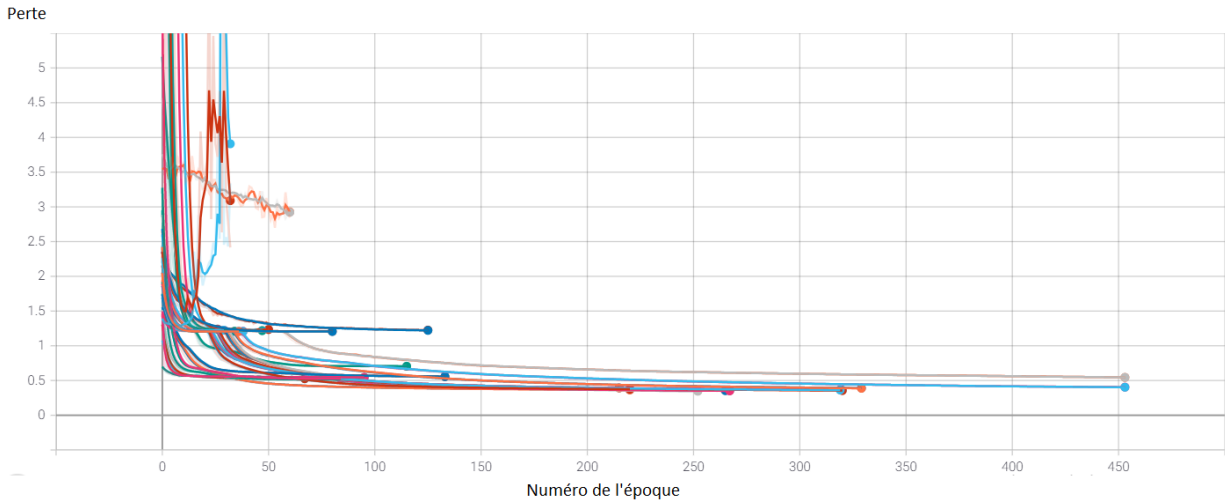


Figure 40 Mécanisme d'arrêt lors de l'utilisation de l'optimisation bayésienne pour trouver les meilleurs hyperparamètres du premier réseau de neurones simple.

5.6.3. Modèles d'optimisation bayésiens

Pour trouver les meilleurs hyperparamètres, nous avons initialisé un espace de recherche utilisé par l'optimisation bayésienne comme un espace des meilleurs hyperparamètres possibles. Nous avons sélectionné cinq hyperparamètres dans le tableau suivant chacun avec son intervalle de l'espace de recherche.

Nœuds	Couches	Fonction d'activation	Taux d'apprentissage	Dropout
[1, 700]	[1, 40]	[relu, sigmoid]	[1e-6, 1e-1]	[0.1, 0.5]

Tableau 5 Espace de recherche.

Le tableau suivant présente chaque nom de modèle avec son acronyme.

Modèle	Acronyme
SGD	SGD
SGD + Dropout	SGD + Dropout
Mélange de distributions gaussiennes (Scale mixture prior)	SMP
Spike and Slab (Spike and Slab prior)	SSP

Bayes empirique (Empirical bayes prior)	EBP
Normale isotrope (Isotropic normal prior)	INP
Normale diagonale (Diagonal trainable prior)	DTP

Tableau 6 Noms et acronymes des modèles.

Ci-dessous, nous pouvons trouver les résultats des opérations d'optimisation bayésienne exécutées pour trouver les meilleurs hyperparamètres pour chacun des sept modèles. Les deux premiers sont un réseau de neurones standard, tandis que les cinq derniers sont des réseaux de neurones bayésiens avec des a priori différents. Tous les modèles bayésiens sont optimisés à l'aide de l'optimiseur Adam.

	Préc entr Perte entr Préc val Perte val	Taux d'apprentissage	Nœuds par couche	Nombre de couches	Fonction d'activation	Temps	GP iters	Dropout
SGD	0.941 0.146 0.941 0.155	2.3e-04	8	700	relu	2h 31min 48s	100	/
SGD + Dropout	0.908 0.223 0.927 0.179	4.4e-04	9	692	relu	2h 18min 35s	100	0.1
SMP	0.512 0.47	3.4e-03	1	1	sigmoid	2h 8min 58s	50	/

	0.511 0.472							
SSP	0.865 0.392 0.861 0.406	1.1e-02	2	700	sigmoid	1h 51m in 3s	50	/
EBP	0.784 0.364 0.78 0.368	7.8e-03	1	700	sigmoid	1h 18m in 57s	50	/
INP	0.675 0.673 0.681 0.664	1.0e-01	1	700	sigmoid	1h 22m in 15s	50	/
DTP	0.718 0.514 0.723 0.509	4.7e-02	2	235	sigmoid	1h 35min 51s	50	/

Tableau 7 Résultats de l'optimisation bayésienne.

5.7. Vérification et évaluation des modèles

5.7.1. Courbes de précision et de perte

Nous avons testé 5 a priori différents avec 5 modèles différents pour trouver le meilleur modèle. En utilisant la précision de la validation des données et la perte de validation, deux a priori en particulier arrivent en tête et produisent de bons résultats. Le premier et le meilleur modèle était le modèle SSP et après lui le modèle l'EBP avant. Les trois autres a priori donnent de mauvais résultats par rapport au modèle SSP et au modèle EBP.

Dans les réseaux de neurones standard, le modèle SGD + Dropout a les meilleurs résultats en ce qui concerne les données de validation par rapport au modèle SGD. Dans les résultats des modèles bayésiens, le modèle SSP est le meilleur en termes de précision à la fois dans les données d'entraînement et de validation. Alors que le modèle EBP donne un meilleur résultat dans la valeur de perte à la fois dans les données d'entraînement et de validation.

Modèle	Précision d'entraînement	Perte d'entraînement	Précision de la validation	Perte de validation
SGD	**0.991**	**0.0237**	0.96	0.172
SGD + Dropout	0.974	0.0629	**0.969**	**0.0865**
SMP	0.57	0.732	0.728	0.572
SSP	**0.866**	0.408	**0.861**	0.42
EBP	0.784	**0.36**	0.782	**0.366**
INP	0.631	0.777	0.655	0.746
DTP	0.677	0.612	0.675	0.608

Tableau 8 Performances des 7 modèles.

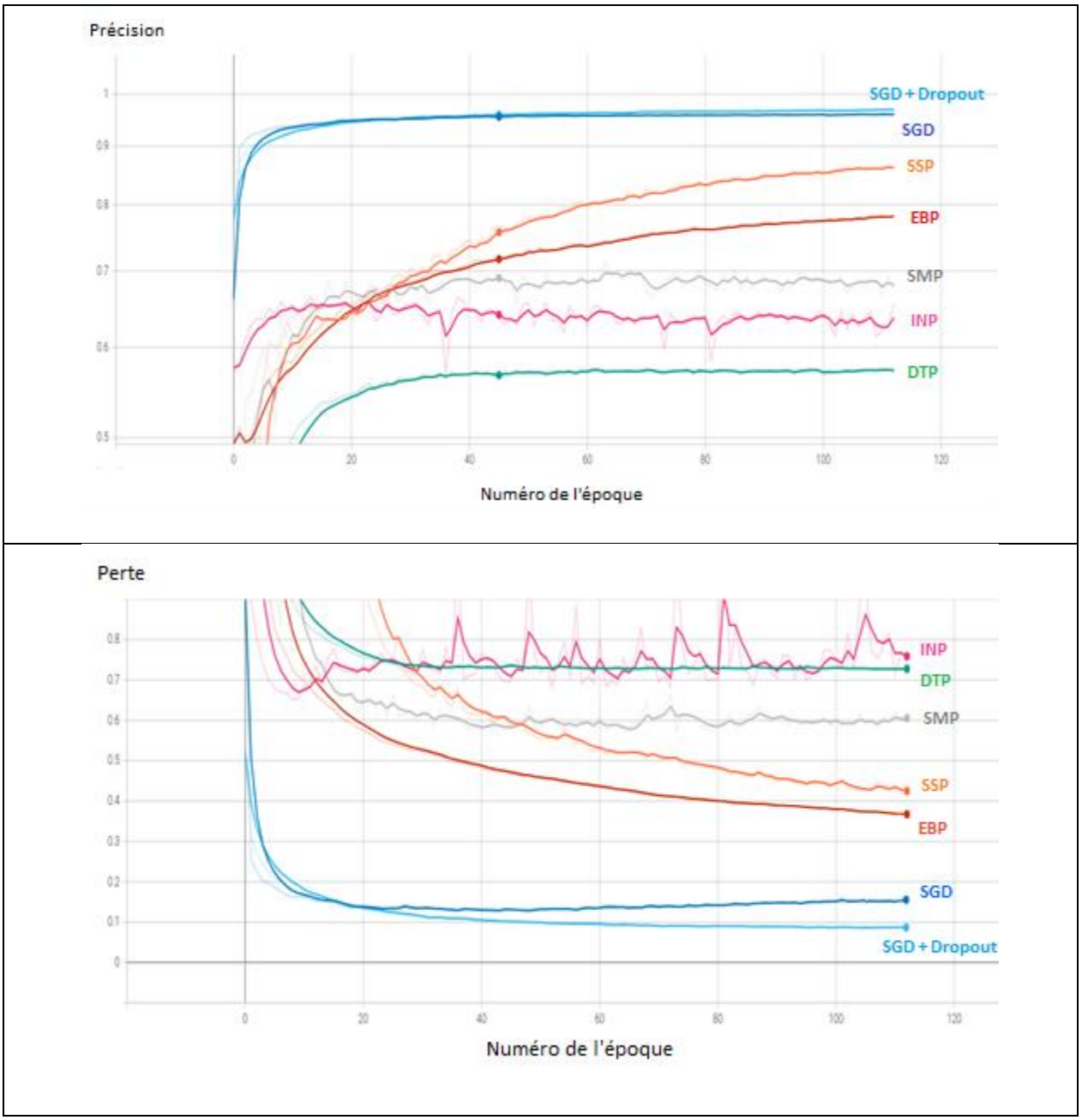


Tableau 9 Performances des 7 modèles sur des données de validation (Forest Cover Type).

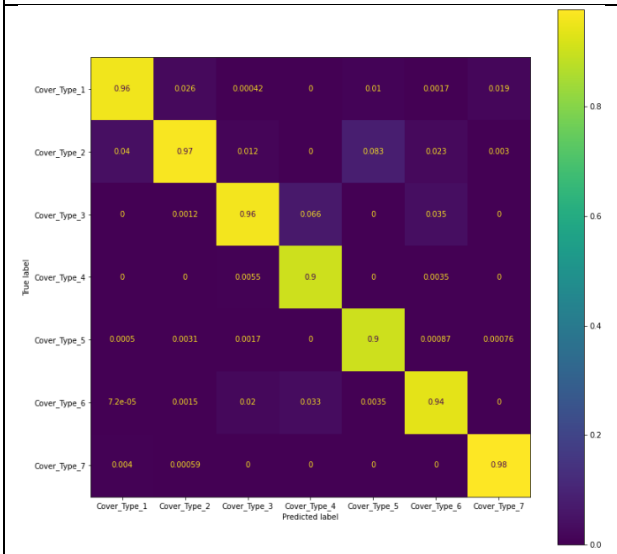
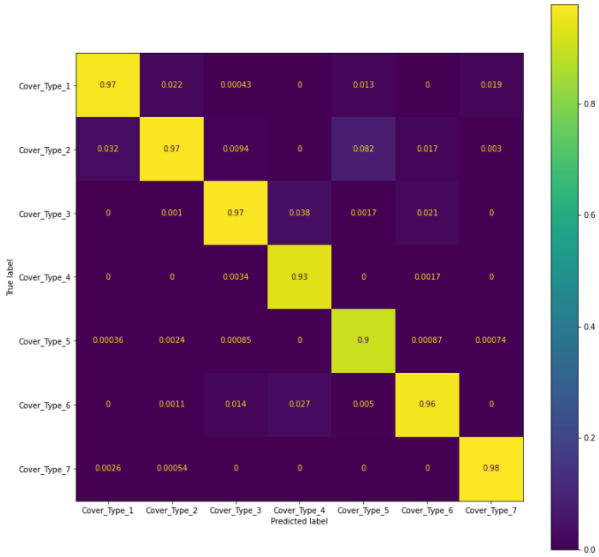
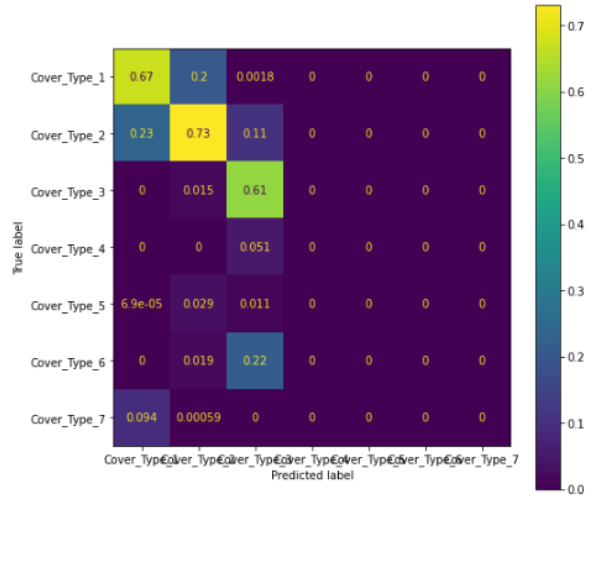
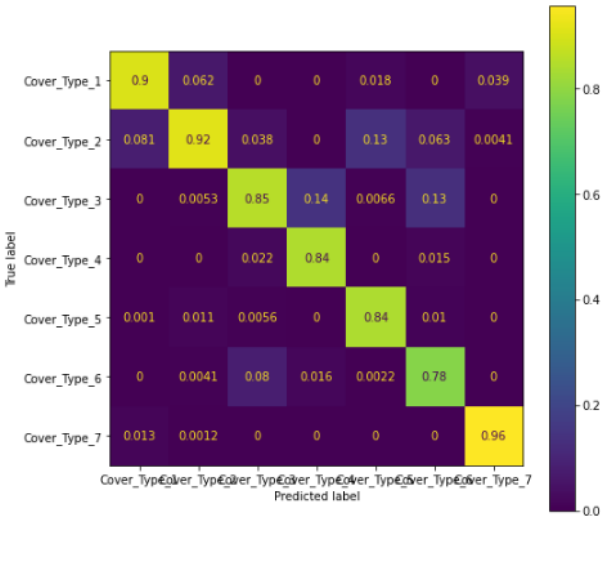
Le tableau suivant explique en détail le résultat de chaque modèle bayésien en utilisant quatre métriques pour spécifier les classes avec le plus d'erreur de prédiction, et comment le modèle se comporte lorsqu'il s'agit de classes avec des données déséquilibrées. SMP a donné les pires résultats, tandis que INP et DTP ont du mal avec des classes avec moins de données. Nous pouvons observer que la dernière classe Cover_type_7 est la classe la plus facile à prédire en ce qui concerne la taille des données.

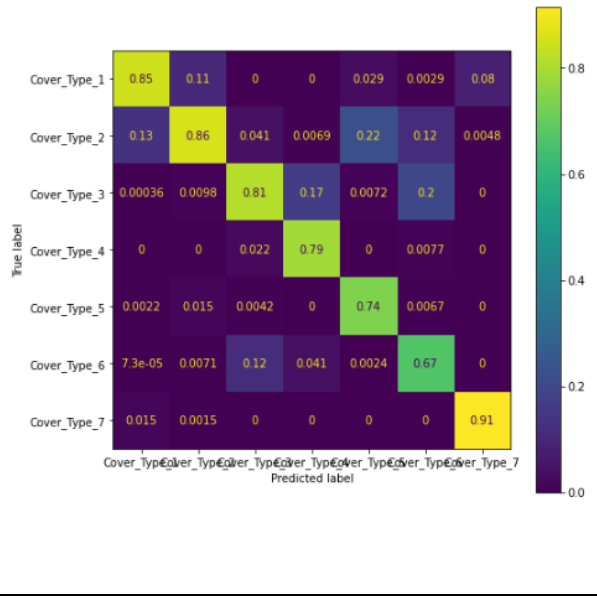
Modèle	Métriques des classes
SMP	precision recall f1-score support 0 0.67 0.70 0.69 13854 1 0.73 0.80 0.76 18554 2 0.61 0.87 0.72 2381 3 0.00 0.00 0.00 174 4 0.00 0.00 0.00 635 5 0.00 0.00 0.00 1126 6 0.00 0.00 0.00 1377
SSP	precision recall f1-score support 0 0.90 0.91 0.91 13854 1 0.91 0.92 0.92 18554 2 0.84 0.89 0.86 2381 3 0.79 0.63 0.70 174 4 0.83 0.61 0.70 635 5 0.78 0.72 0.75 1126 6 0.96 0.84 0.90 1377
EBP	precision recall f1-score support 0 0.85 0.84 0.85 13854 1 0.86 0.88 0.87 18554 2 0.81 0.82 0.82 2381 3 0.77 0.66 0.71 174 4 0.74 0.47 0.58 635 5 0.66 0.60 0.63 1126 6 0.91 0.83 0.87 1377
INP	precision recall f1-score support 0 0.76 0.68 0.72 13854 1 0.74 0.82 0.78 18554 2 0.71 0.33 0.45 2381 3 0.37 0.32 0.35 174 4 0.32 0.01 0.02 635 5 0.30 0.66 0.41 1126 6 0.76 0.72 0.74 1377
DTP	precision recall f1-score support 0 0.84 0.70 0.76 13854 1 0.77 0.90 0.83 18554 2 0.62 0.91 0.74 2381 3 0.00 0.00 0.00 174 4 0.62 0.29 0.40 635 5 0.75 0.01 0.02 1126 6 0.87 0.69 0.77 1377

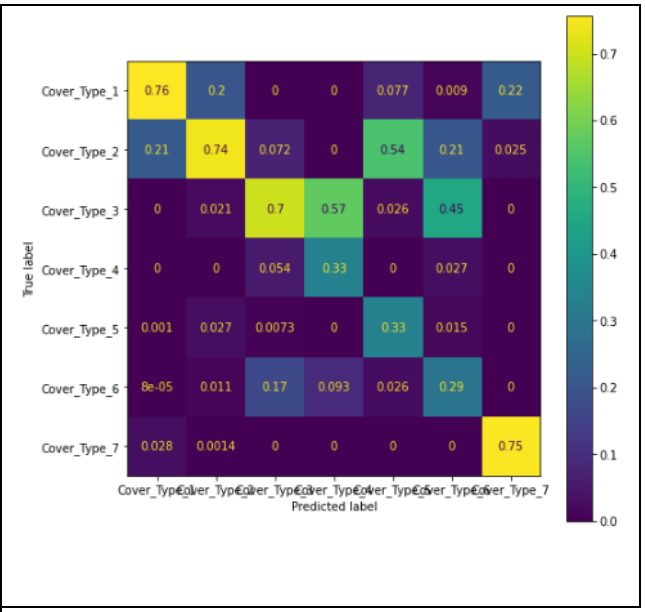
Tableau 10 Métriques des classes.

5.7.2. Matrices de confusion

Des matrices de confusion normalisées sont fournies pour visualiser les erreurs des modèles dans ses prédictions. Comme nous l'avons mentionné, nous pouvons observer que tous les modèles bayésiens luttent avec les classes avec des données déséquilibrées.

SGD	SGD + Dropout
	
SMP	**SSP**
	
EBP	**INP**





DTP

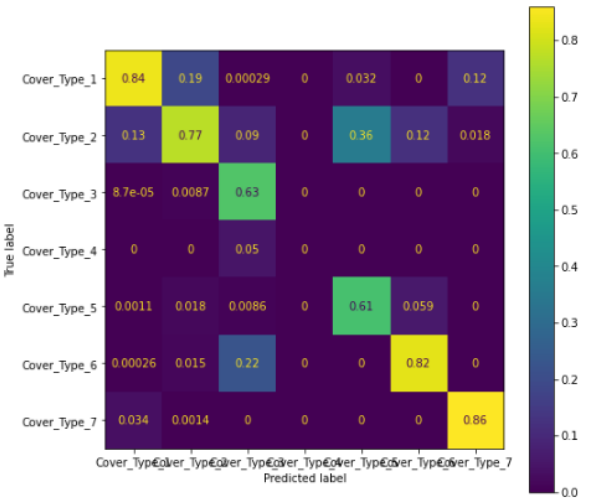
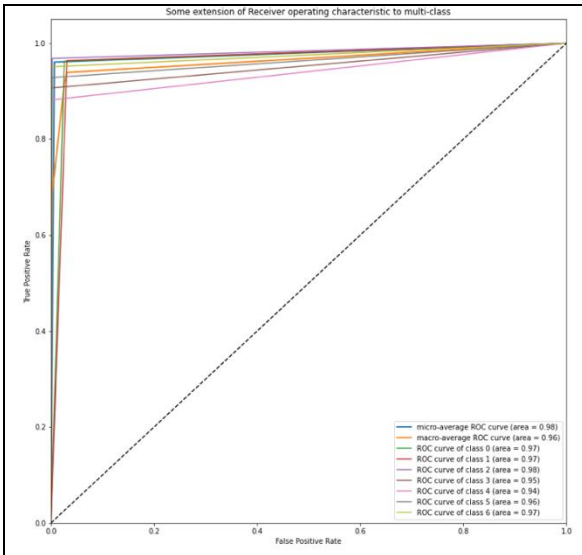


Tableau 11 Matrices de confusion pour les 7 modèles.

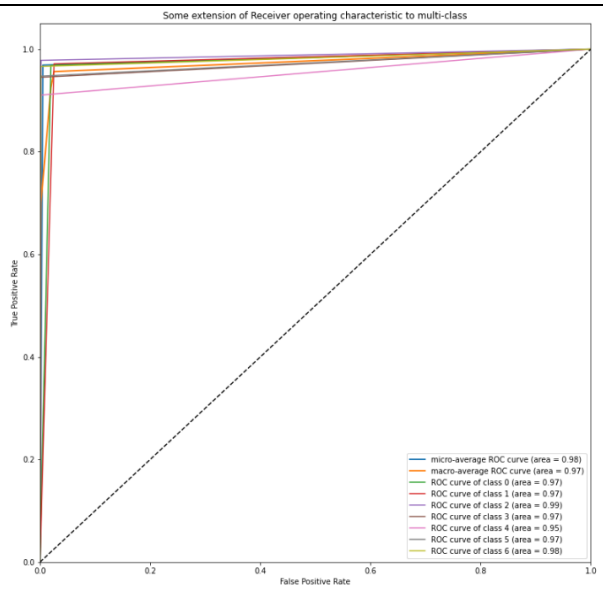
5.7.3. Courbes ROC

La courbe des caractéristiques de fonctionnement du récepteur (ROC) trace la relation entre le taux de vrais positifs (TPR) et le taux de faux positifs (FPR) lorsque le seuil de décision change. Nous pouvons observer que la courbe ROC est moins informatif en raison d'un ensemble de données avec un déséquilibre de classe élevé, car la majorité (Cover_Type_1 et Cover_Type_2) peuvent noyer les contributions des classes minoritaires.

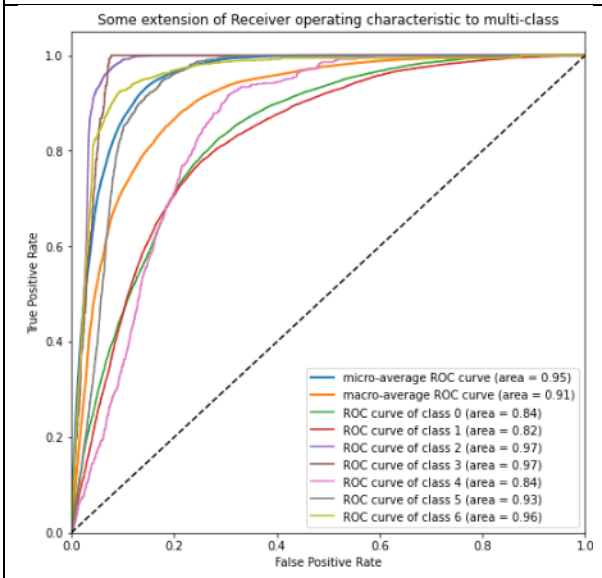
SGD	SGD + Dropout



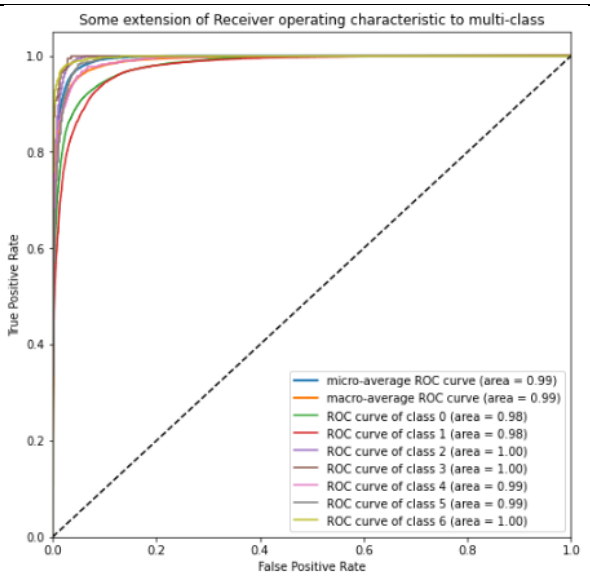
SMP



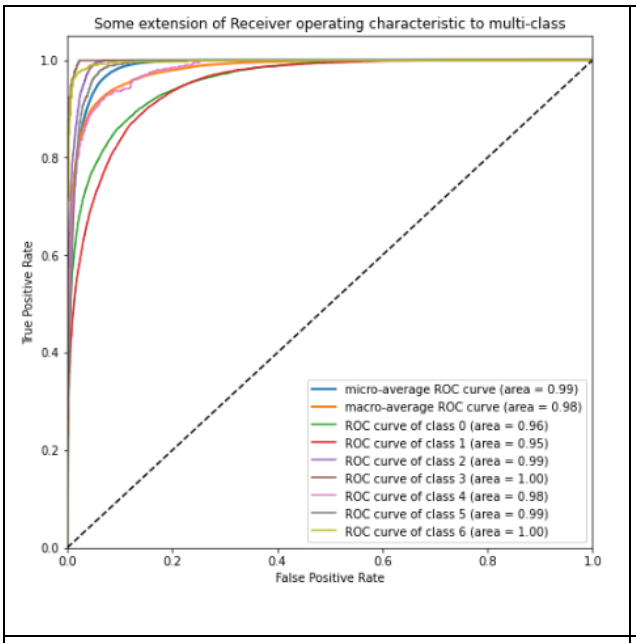
SSP

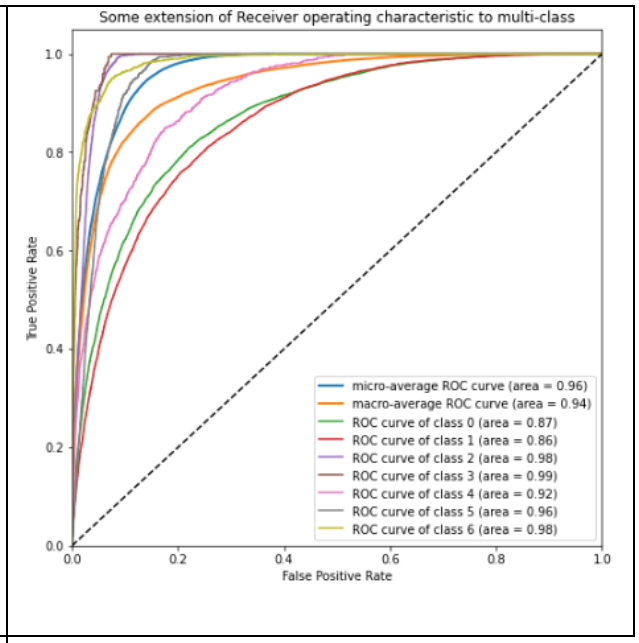


EBP



INP





DTP

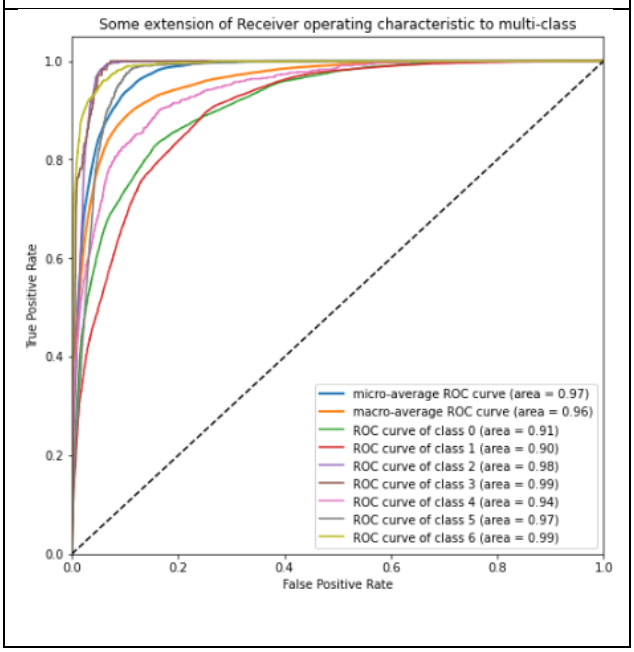
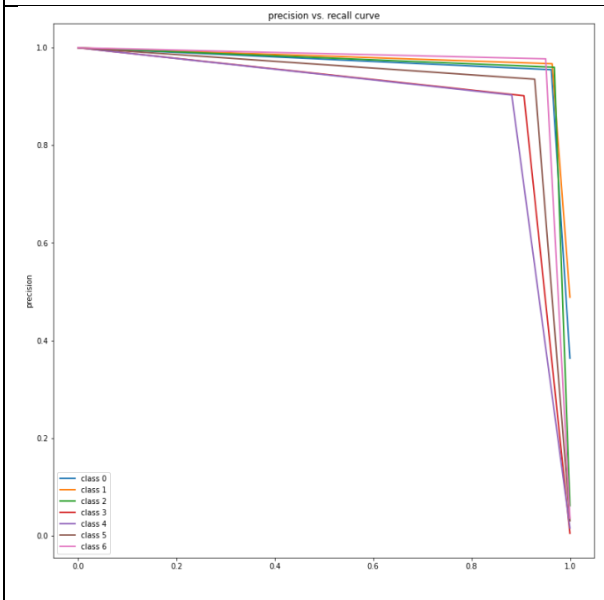
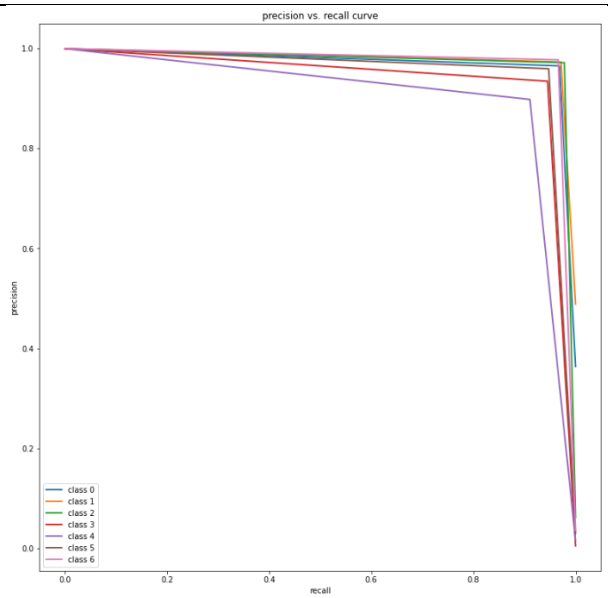
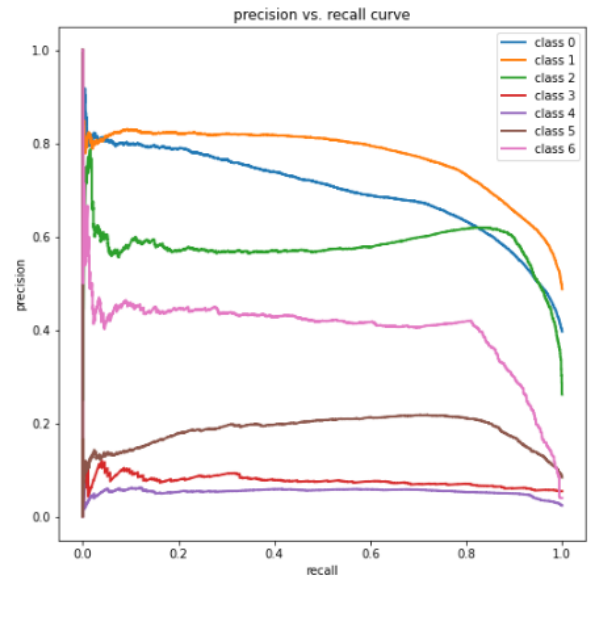
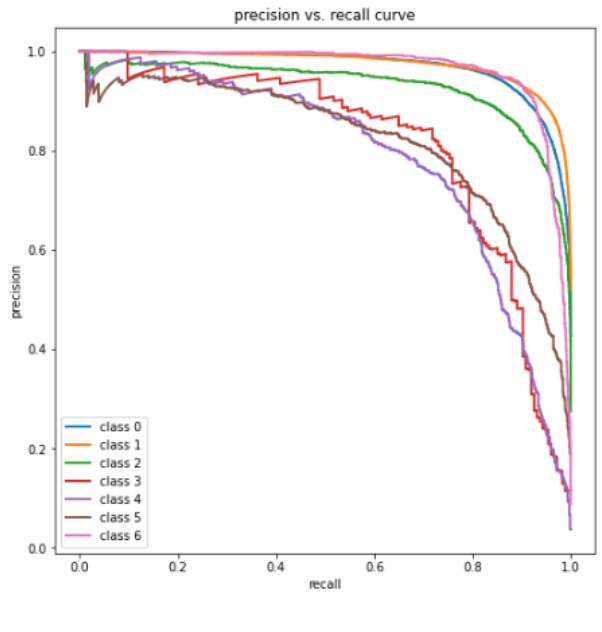


Tableau 12 Courbes ROC pour les 5 modèles.

5.7.4. Courbes rappel-précision

Rappel-précision est une mesure utile du succès de la prédiction dans notre cas avec des classes déséquilibrées. La précision est une mesure de la capacité d'un modèle de classification à identifier uniquement les points de données pertinents, tandis que le rappel est une mesure de la capacité d'un modèle à trouver tous les cas pertinents dans un ensemble de données. On peut

observer que les deux modèles SSP et EBP ont tous deux une mauvaise classification dans le cas de ces trois classes : Cover_Type_4, Cover_Type_5 et Cover_Type_6.

SGD	SGD + Dropout
	
SMP	**SSP**
	
EBP	**INP**

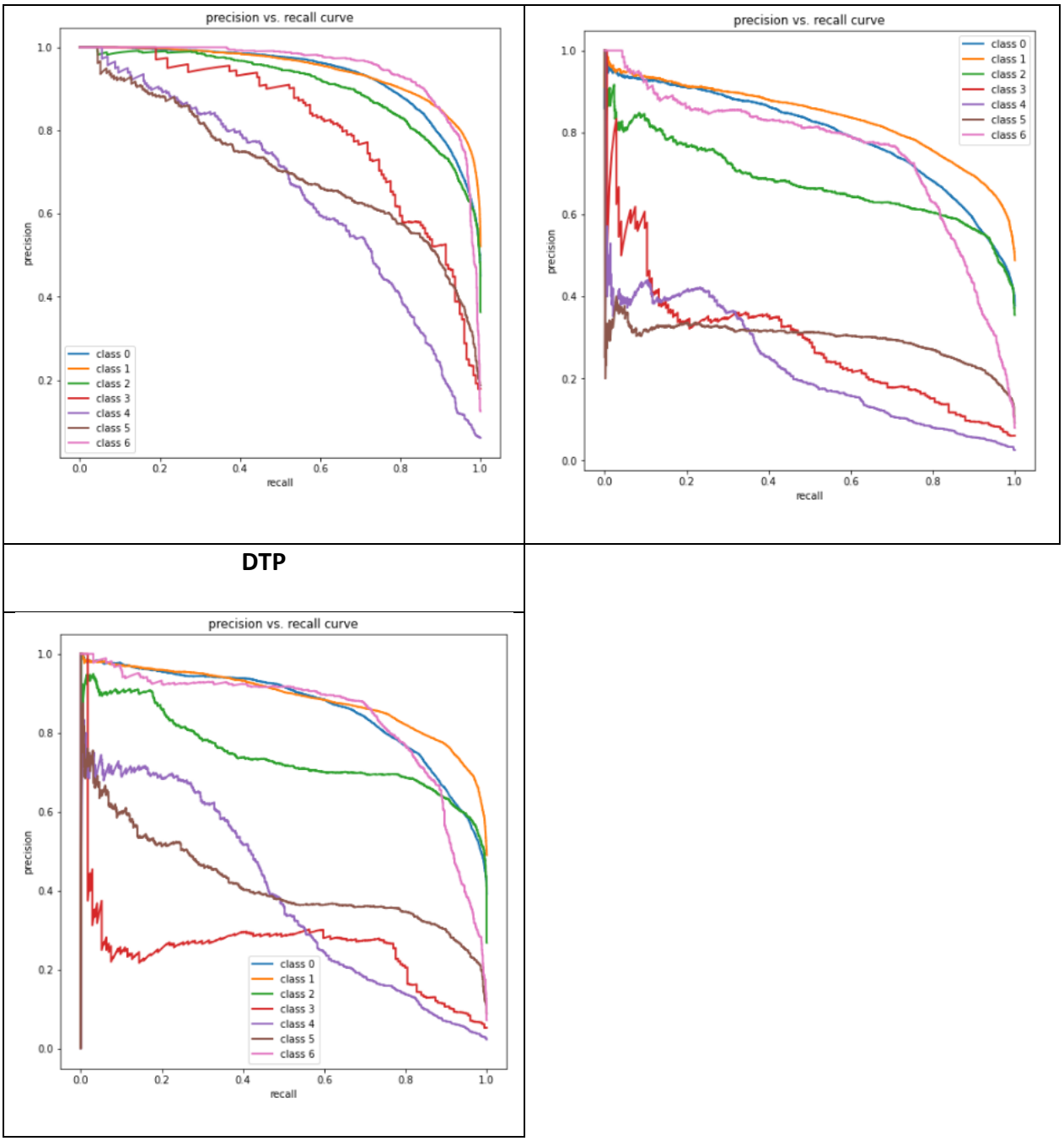


Tableau 13 Courbes rappel-précision pour les 7 modèles.

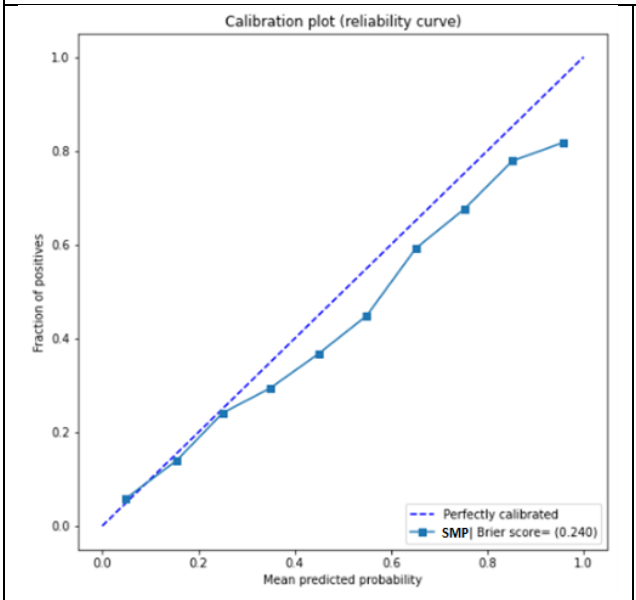
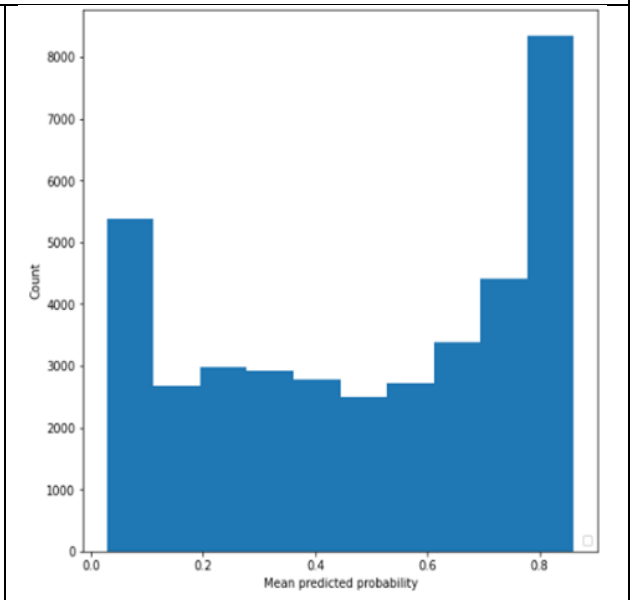
5.7.5. Courbes de calibration

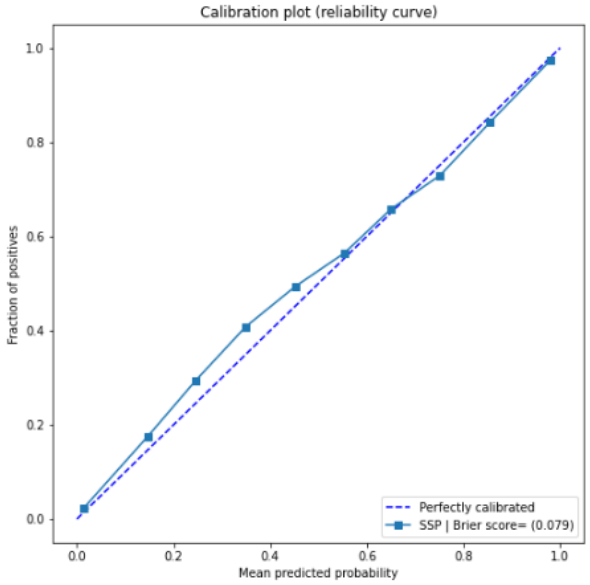
Pour visualiser et mesurer la qualité de l'incertitude et la fiabilité des prédictions de notre modèle, nous utilisons la courbe de calibration (fiabilité) et le score de Brier. Le score Brier permet de vérifier la calibration du modèle, ce qui est important pour éviter une mauvaise prise de décision ou une fausse interprétation.

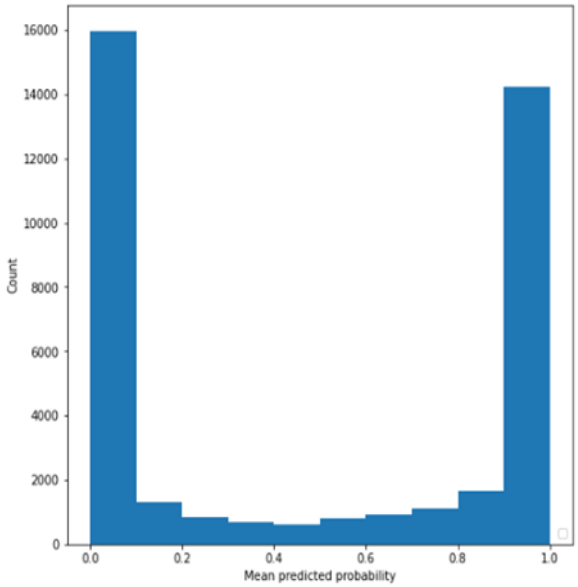
$$BS = \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{R}(f_{ti} - o_{ti})^2$$

Où R est le nombre de classes possibles et est égal à 7 dans notre cas, et N le nombre total d'instances de toutes les classes. Avec f_{it} la probabilité prédite pour la classe i , et où o_{it} vaut 1 s'il s'agit de la $i - ème$ classe à l'instant t ; 0, sinon (Wikiwand n.d.). Le modèle avec le score Brier le plus bas est le meilleur modèle. Le modèle SSP a le score de Brier le plus bas avec 7,9%.

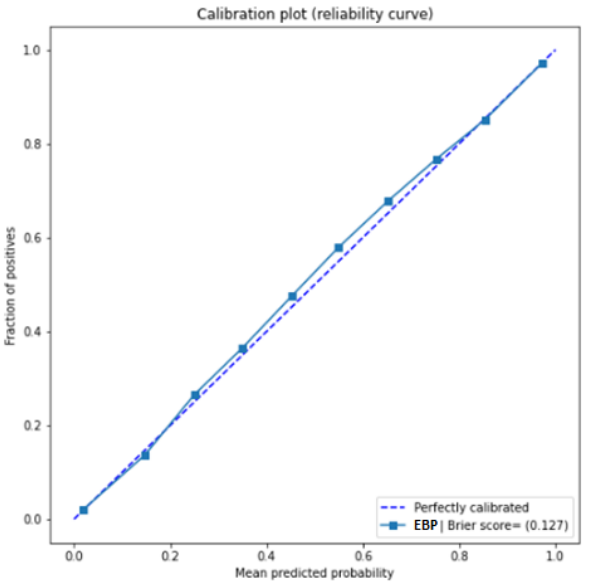
A partir du tableau ci-dessous, le modèle SSP est calibré. Nous pouvons également observer que la probabilité moyenne est bien calibrée dans les probabilités prédictives les plus faibles et les plus élevées de la classe positive, et lorsque la probabilité est d'environ 0,4, le modèle est incertain quant au résultat de la classe. Nous pouvons également observer l'incertitude dans la distribution prédictive dans le tableau ci-dessous des cinq modèles différents, et nous avons également que le modèle EBP est bien calibré que le modèle SSP mais si nous analysons le score de Brier 0,079 > 0,127, nous pouvons trouver le meilleur modèle qui est le modèle SSP.

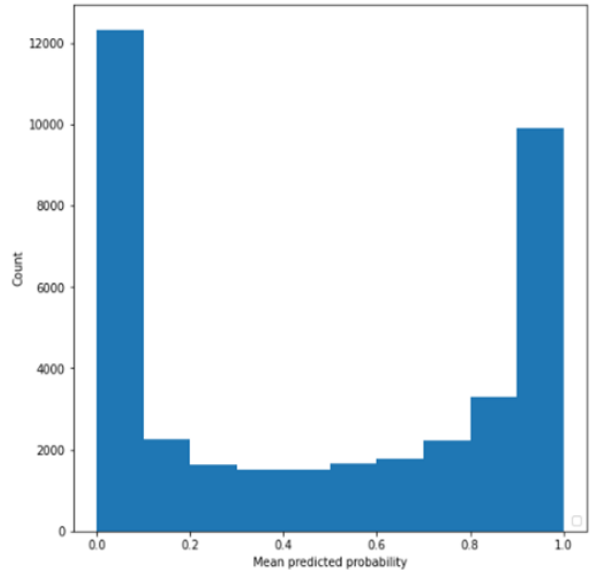
SMP	
	
SSP	


Calibration plot (reliability curve)
Fraction of positives
Mean predicted probability
Perfectly calibrated
SSP | Brier score= (0.079)


Count
Mean predicted probability

EBP


Calibration plot (reliability curve)
Fraction of positives
Mean predicted probability
Perfectly calibrated
EBP | Brier score= (0.127)


Count
Mean predicted probability

INP

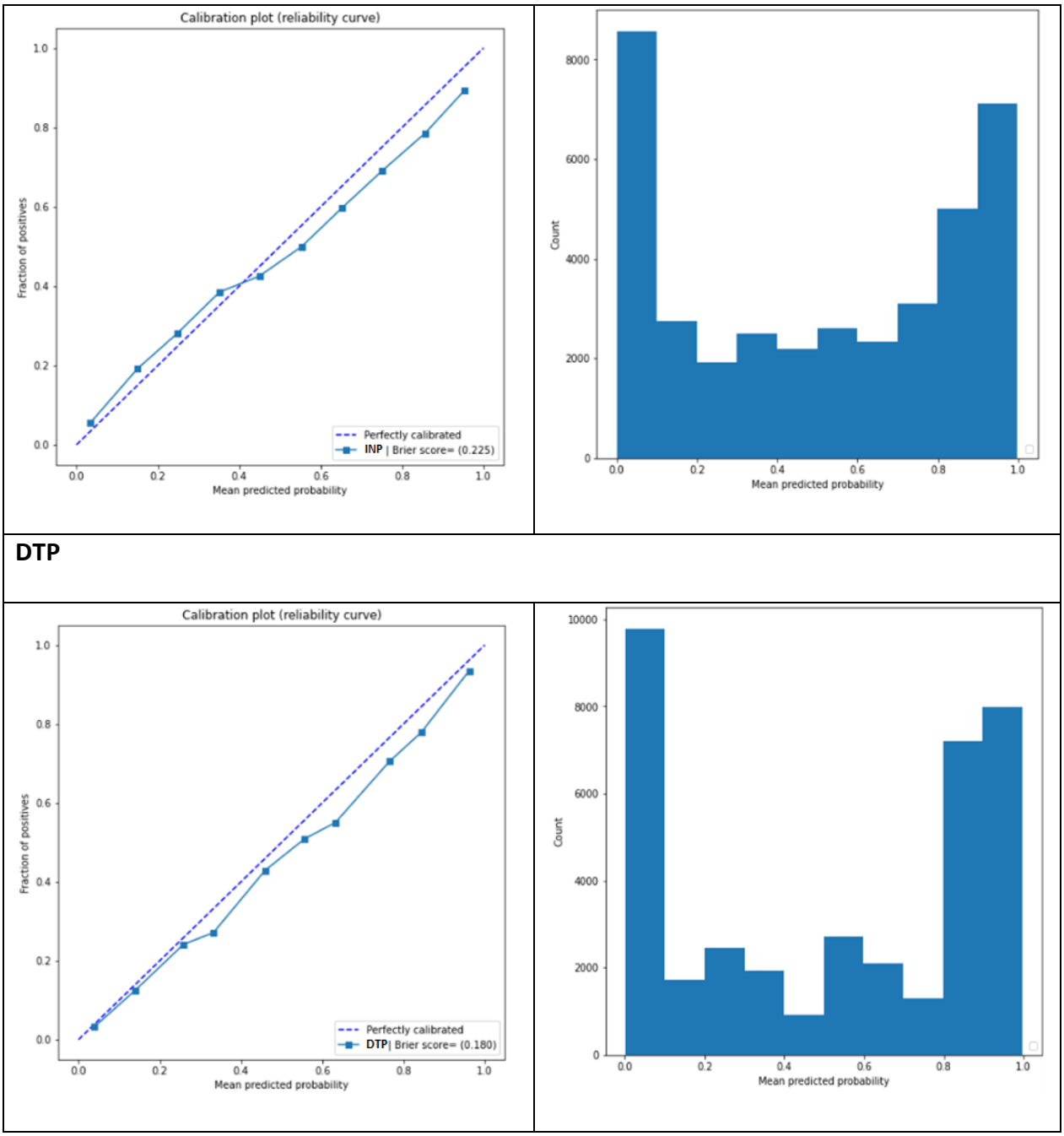


Tableau 14 À gauche: Courbes de calibration des 7 modèles avec score Brier, à droit: l'histogramme de distribution prédictive.

5.8. Estimation de l'incertitude

Pour capturer l'incertitude, nous utiliserons la méthode proposée utilisée dans (Kwon et al., 2020), ils ont basé leur travail sur le travail de (Kendall & Gal, 2017) où ce dernier a ajouté des nœuds supplémentaires à la dernière couche du modèle pour apprendre la variance pour décomposer la source d'incertitude en aléatoire et épistémique. Alors que l'approche du premier

consiste à décomposer la source d'incertitude sans ajouter de composants supplémentaires pour apprendre la variance.

Dans les tâches de classification, notre intérêt est la distribution prédictive $p_{\hat{\theta}}(y^*|x^*)$, où x^* et y^* sont respectivement un échantillon de données et sa classe prédite. Pour un réseau de neurones bayésien, cette distribution est donnée par :

$$p_{\hat{\theta}}(y^*|x^*) = \int p(y^*|x^*, w) p_{\hat{\theta}}(w) dw$$

Sous le VI, BBB approxime la distribution prédictive pour dériver la distribution prédictive variationnelle :

$$q_{\hat{\theta}}(y^*|x^*) = \int p(y^*|x^*, w) q_{\hat{\theta}}(w) dw$$

En raison du manque de conjugaison entre la distribution prédictive catégorielle $p(y^*|x^*, w)$ et les distributions gaussiennes $q_{\hat{\theta}}(w)$, nous pouvons construire un estimateur sans biais de l'espérance en échantillonnant à partir de $q_{\hat{\theta}}(w)$ (Shridhar et al., 2018):

$$\hat{q}_{\hat{\theta}}(y^*|x^*) = \frac{1}{T} \sum_{t=1}^{T} p(y^*|x^*, \widehat{w}_t)$$

Où T est le nombre prédéfini d'échantillons. Cet estimateur permet d'extraire l'incertitude en utilisant la variance prédictive :

$$\begin{aligned} Var_{q_{\hat{\theta}}(y^*|x^*)}(y^*) &= E_{q_{\hat{\theta}}(y^*|x^*)}\{y^{*\otimes 2}\} - E_{q_{\hat{\theta}}(y^*|x^*)}(y^*)^{\otimes 2} \\ &= \underbrace{\int \left[diag\left\{ E_{p(y^*|x^*, w)}(y^*) \right\} - E_{p(y^*|x^*, w)}(y^*)^{\otimes 2} \right] q_{\hat{\theta}}(w) dw}_{Aléatoire} \\ &+ \underbrace{\int \{E_{p(y^*|x^*, w)}(y^*) - E_{q_{\hat{\theta}}(y^*|x^*)}(y^*)\}^{\otimes 2} q_{\hat{\theta}}(w) dw}_{épistémique} \end{aligned}$$

Où l'incertitude prédictive est $E_{q_{\hat{\theta}}(y^*|x^*)}\{y^{*\otimes 2}\} - E_{q_{\hat{\theta}}(y^*|x^*)}(y^*)^{\otimes 2}$, la somme de l'incertitude aléatoire et de l'incertitude épistémique. De plus, $v^{\otimes 2} = vv^T$ et $diag(v)$ est une matrice diagonale avec des éléments du vecteur v. L'incertitude aléatoire capture le caractère aléatoire inhérent d'un résultat y^*. L'incertitude épistémique provient de la variabilité de w. Cette quantité diminue à mesure que la taille de l'échantillon T augmente (Kwon et al., 2020).

La formule proposée suggérait une estimation de deux types d'incertitude comme suit :

$$Incertitude_{prédictive} = \underbrace{\frac{1}{T}\sum_{t=1}^{T} diag(\hat{p}_t) - \hat{p}_t^{\otimes 2}}_{Aléatoire} + \underbrace{\frac{1}{T}\sum_{t=1}^{T} (\hat{p}_t - \bar{p})^{\otimes 2}}_{épistémique}$$

Où $\bar{p} = \sum_{t=1}^{T} \frac{\hat{p}_t}{T}$ et $\hat{p}_t = p(\widehat{w}_t) = Softmax\{f^{\widehat{w}_t}(x^*)\}$, et $f^w(x) = (f_1^w(x), \dots, f_K^w(x))$ la dernière sortie linéaire pré-activée à K dimensions du réseau de neurones avec un vecteur de paramètre w (Kwon et al., 2020).

5.5.1. Incertitude aléatoire et épistémique

Nous avons appliqué la méthode précédente proposée par (Kwon et al., 2020) pour estimer l'incertitude. Tout d'abord, nous avons sélectionné 1190 échantillons à partir des données de test, chaque classe a 170 échantillons, et nous avons initialisé le nombre de passes avant T à 10, puis nous avons calculé l'incertitude épistémique et aléatoire comme nous l'avons expliqué précédemment. Le tableau ci-dessous montre l'incertitude épistémique et aléatoire des 5 modèles bayésiens :

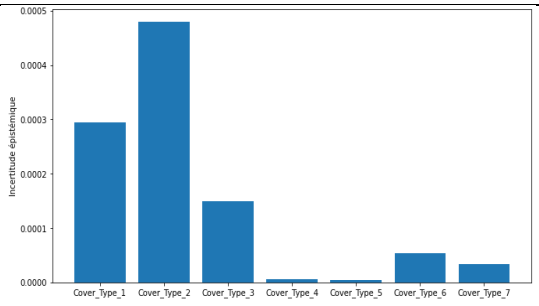
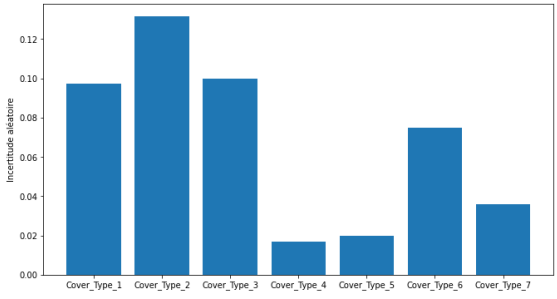
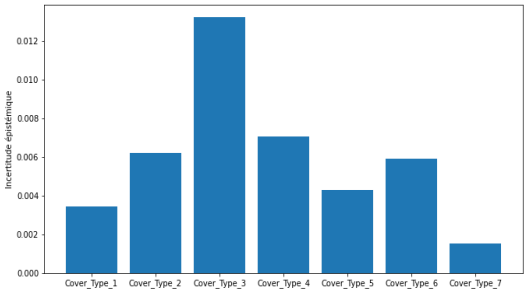
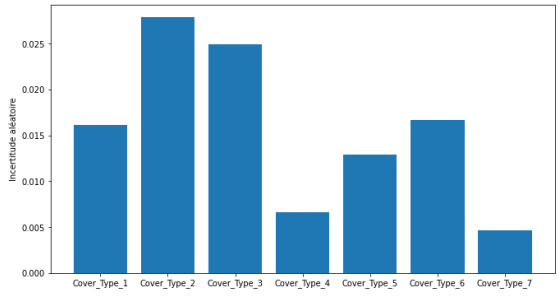
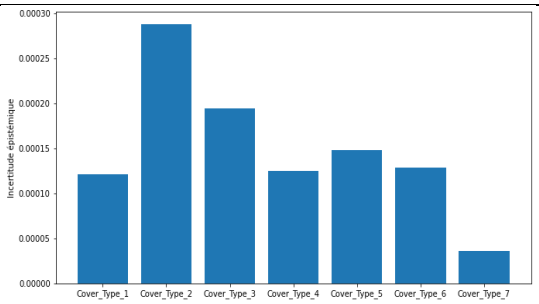
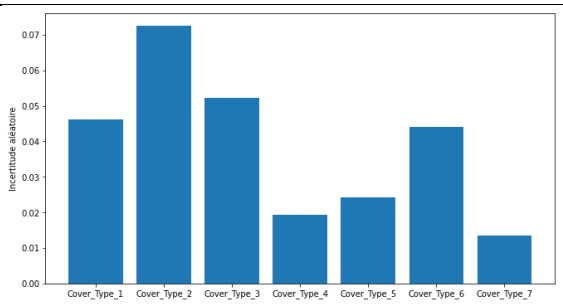
	Incertitude épistémique	Incertitude aléatoire	Précision de la validation	Perte de validation
SMP	**0.00014**	0.06798	0.728	0.572
SSP	0.00594	**0.01568**	**0.861**	0.42
EBP	**0.00014**	0.03882	0.782	**0.366**
INP	0.00589	0.04405	0.655	0.746
DTP	0.00417	0.05288	0.675	0.608

Tableau 15 Incertitude épistémique et aléatoire du modèle 5 bayésien.

En raison des différences dans les architectures et les hyperparamètres des modèles et de la sélection de l'a priori, nous ne devons pas comparer les incertitudes épistémiques des modèles les unes avec les autres. Contrairement à cela, parce que nous avons utilisé le même ensemble de données pour chaque modèle, nous pouvons comparer les 5 modèles sur leur incertitude aléatoire. Nous pouvons remarquer que les deux meilleurs modèles en termes de précision sont

SSP et EBP avec l'incertitude aléatoire la plus faible, et que la précision et l'incertitude épistémique sont négativement corrélées.

Dans le tableau suivant, nous avons estimé les deux types d'incertitudes pour chaque modèle bayésien. En analysant les résultats du tableau parallèlement aux résultats du tableau précédent, nous concluons que la valeur d'incertitude prédictive est principalement déterminée par l'incertitude aléatoire dans le modèle qui exploite les données par rapport à l'a priori. Le meilleur modèle SSP, son incertitude épistémique et aléatoire sont plus proches par rapport aux autres modèles, ce qui valide le bon choix de l'a priori. Comme nous l'avons mentionné précédemment, le modèle EBP exploite les données sur l'a priori pour déterminer la distribution a posteriori, pour cela nous pouvons observer que l'incertitude aléatoire est beaucoup plus grande que l'incertitude épistémique.

	Incertitude épistémique	Incertitude aléatoire
SMP		
SSP		
EBP		

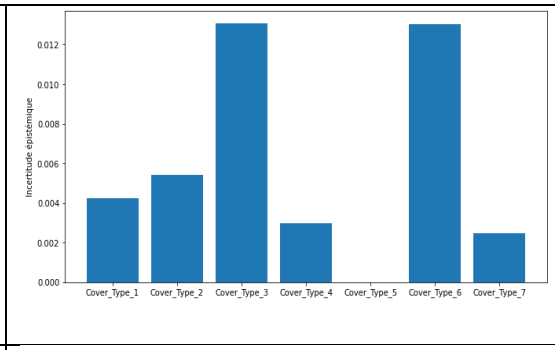
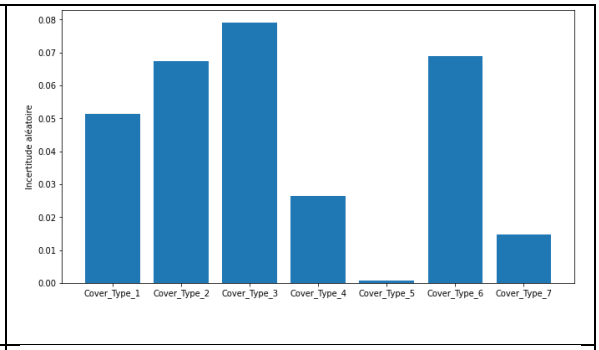
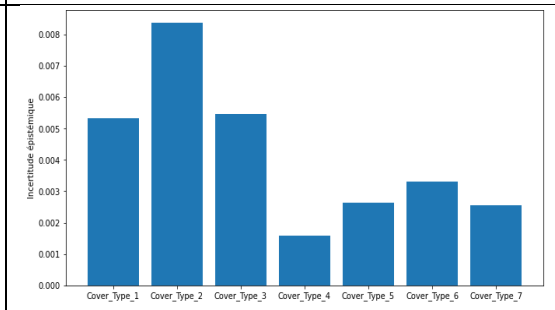
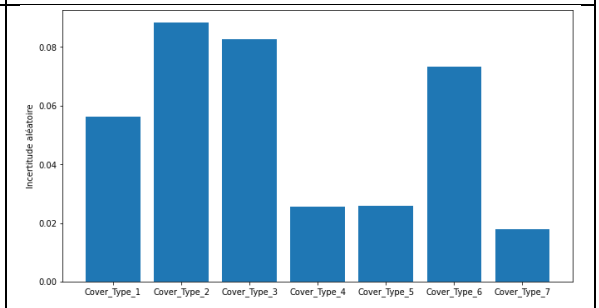
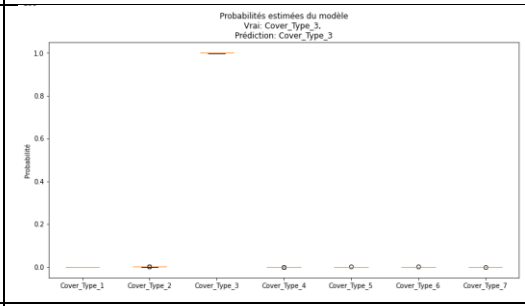
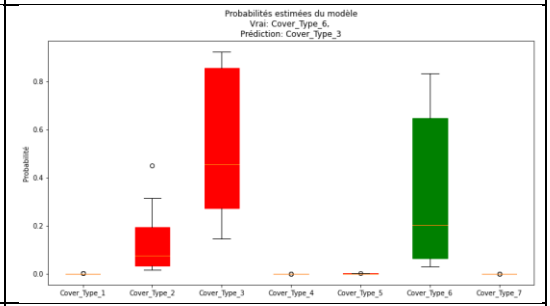
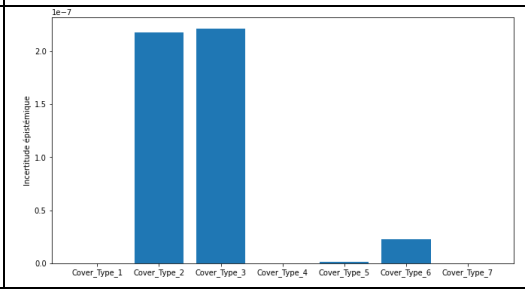
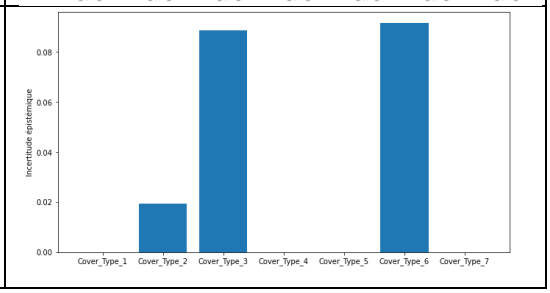
INP		
DTP		

Tableau 16 Incertitude épistémique et aléatoire du modèle à 5 bayésiens par classe.

Dans le tableau suivant, nous avons deux prédictions du modèle SSP. L'exemple avec la prédiction correcte a une incertitude épistémique et aléatoire beaucoup plus faible que l'exemple de la fausse prédiction. Le modèle sait quand il ne sait pas.

	Vraie prédiction	Fausse prédiction
Incertitude prédictive		
Incertitude épistémique		

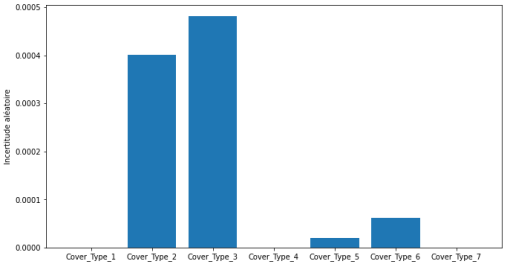
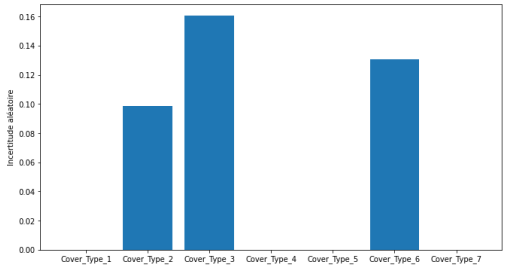
Incertitude aléatoire		

Tableau 17 Représentation de l'incertitude de deux exemples de prédictions du modèle SSP.

Chapitre 6 – Conclusion

Conclusion

Ce travail adopte le réseau neuronal bayésien avec estimation de l'incertitude comme modèle de base pour notre système d'aide à la décision clinique. Nous visons à développer des méthodes qui nous aident à mieux comprendre les données la prédiction et l'incertitude de notre modèle. Pouvoir sélectionner un traitement pour un patient en toute confiance est une propriété indispensable dans le système d'aide à la décision clinique intégré au domaine clinique. L'incertitude produit des doutes qui peuvent conduire à une mauvaise décision et mettre en danger la santé du patient.

Nous proposons un nouveau système d'aide à la décision clinique pour aider à réduire la mortalité par les maladies cardiovasculaires en améliorant la santé des patients en fonction de leurs activités physiques. Nous avons adopté l'architecture de microservices pour réaliser notre application Web de système d'aide à la décision clinique. En l'absence de données sur le patient, nous avons fourni une expérience en utilisant un type d'ensemble de données tabulaire pour simuler les étapes que nous prendrions avec les données originales du patient.

Nous avons utilisé l'optimisation bayésienne pour sélectionner les meilleurs modèles en utilisant l'approche fréquentiste et bayésienne pour générer différentes architectures et hyperparamètres. Nous avons montré comment différents a priori peuvent affecter la précision et la fiabilité globales du modèle. Après avoir utilisé plusieurs méthodes pour évaluer nos modèles, nous avons sélectionné le meilleur modèle avec l'a priori Spike and Slab (SSP) parmi les quatre autres a priori différents. Nous avons estimé l'incertitude épistémique et aléatoire de notre modèle. Différentes techniques ont été utilisées à cette fin. Les résultats de l'analyse indiquent que les modèles BNN peuvent capturer l'incertitude pour aider à une meilleure compréhension des données et du modèle qui représente les données, et pour une meilleure prise de décision.

Travaux futurs

Les résultats obtenus dans ce travail suggèrent de multiples pistes prometteuses pour de futures recherches. La première étape consiste à définir un protocole pour évaluer le système

d'aide à la décision clinique de manière pratique. Comme la façon dont le clinicien interagira avec le système d'aide à la décision clinique pour renforcer la confiance, et pour déterminer le rôle du système d'aide à la décision clinique dans la clinique, aussi comment intégrer la décision du système d'aide à la décision clinique avec la décision de clinicien, et quel degré de responsabilité nous donnons au système d'aide à la décision clinique. Il y a beaucoup de questions à se poser et de décisions à prendre avant d'intégrer le système d'aide à la décision clinique dans les établissements de santé.

La disponibilité des données est un problème possible lorsqu'il s'agit d'entraîner les modèles. L'apprentissage incrémental peut être une solution en combinant différents modèles formés sur différentes périodes avec différents ensembles de données. Les modèles hiérarchiques bayésiens peuvent être une solution car ils créent le modèle de manière hiérarchique et dans l'approche bayésienne. L'objectif est de créer la première et la deuxième version du modèle après avoir reçu les données en deux périodes, chacune pour entraîner une version du modèle. Suivant l'approche bayésienne, nous utiliserions le premier modèle pour trouver l'a priori pour la deuxième version du modèle. Une ancienne distribution de probabilité a posteriori devient une distribution de probabilité a priori à la nouvelle distribution de probabilité a posteriori.

La diversité de nos sources de données pourrait intégrer un autre problème provenant des données hors distribution (en anglais : out-of-distribution OOD), pour cela nous devrions tester nos décisions de modèle sur les données hors distribution pour évaluer la qualité de notre estimation de l'incertitude.

L'amélioration des modèles bayésiens peut se faire avec différentes techniques et méthodes proposées par différents chercheurs. Expérimentez avec la technique de distribution de probabilité a posteriori froide ou calibrez le modèle pour obtenir les meilleurs résultats.

En utilisant l'optimisation bayésienne, nous pourrions explorer d'autres espaces de recherche avec plus d'hyperparamètres et de grands espaces. Nous pourrions l'utiliser pour trouver les hyperparamètres de l'optimiseur utilisé pour optimiser les paramètres du modèle. Aussi, pour trouver le meilleur a priori pour notre modèle. On pourrait même essayer d'optimiser les hyperparamètres de l'optimisation bayésienne elle-même en commençant par les composantes du processus gaussien, par exemple la fonction noyau et ses hyperparamètres.

Références bibliographiques

Ainsworth, B., Cahalin, L., Buman, M., & Ross, R. (2015). The Current State of Physical Activity Assessment Tools. *Progress in Cardiovascular Diseases*, *57*(4), 387–395. https://doi.org/10.1016/j.pcad.2014.10.005

Baig, M. M., GholamHosseini, H., Moqeem, A. A., Mirza, F., & Lindén, M. (2017). A Systematic Review of Wearable Patient Monitoring Systems – Current Challenges and Opportunities for Clinical Adoption. *Journal of Medical Systems*, *41*(7). https://doi.org/10.1007/s10916-017-0760-1

Baskerville, R., Ricci-Cabello, I., Roberts, N., & Farmer, A. (2017). Impact of accelerometer and pedometer use on physical activity and glycaemic control in people with Type 2 diabetes: a systematic review and meta-analysis. *Diabetic Medicine*, *34*(5), 612–620. https://doi.org/10.1111/dme.13331

Berrar, D. (2018). Bayes' theorem and naive Bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, *403*.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2016). *Variational Inference: A Review for Statisticians*. https://doi.org/10.1080/01621459.2017.1285773

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). *Weight Uncertainty in Neural Networks*.

Brian J. Reich, & Sujit K. Ghosh. (2019). *Bayesian Statistical Methods*.

Brochu, E., Cora, V. M., & de Freitas, N. (2010). *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*.

Charles A Kamhoua, Christopher D Kiekintveld, Fei Fang, & Quanyan Zhu. (2021). *Game Theory and Machine Learning for Cyber Security* (C. A. Kamhoua, C. D. Kiekintveld, F. Fang, & Q. Zhu, Eds.). Wiley. https://doi.org/10.1002/9781119723950

Davenport, T., & Kalakota, R. (2019). The potential for artificial intelligence in healthcare. *Future Healthcare Journal*, *6*(2), 94–98. https://doi.org/10.7861/futurehosp.6-2-94

Dhawale, T., Steuten, L. M., & Deeg, H. J. (2017). Uncertainty of Physicians and Patients in Medical Decision Making. *Biology of Blood and Marrow Transplantation*, *23*(6), 865–869. https://doi.org/10.1016/j.bbmt.2017.03.013

Elsken, T., Metzen, J. H., & Hutter, F. (2018). *Neural Architecture Search: A Survey*.

Frazier, P. I. (2018). *A Tutorial on Bayesian Optimization*.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., & Zhu, X. X. (2021). *A Survey of Uncertainty in Deep Neural Networks*.

Gleason, P. M., & Harris, J. E. (2019). The Bayesian Approach to Decision Making and Analysis in Nutrition Research and Practice. *Journal of the Academy of Nutrition and Dietetics*, *119*(12), 1993–2003. https://doi.org/10.1016/j.jand.2019.07.009

Helou, M. A., DiazGranados, D., Ryan, M. S., & Cyrus, J. W. (2020). Uncertainty in Decision Making in Medicine. *Academic Medicine*, *95*(1), 157–165. https://doi.org/10.1097/ACM.0000000000002902

Hodkinson, A., Kontopantelis, E., Adeniji, C., van Marwijk, H., McMillan, B., Bower, P., & Panagioti, M. (2019). Accelerometer- and Pedometer-Based Physical Activity Interventions Among Adults With Cardiometabolic Conditions. *JAMA Network Open*, *2*(10), e1912895. https://doi.org/10.1001/jamanetworkopen.2019.12895

Houle, J., Doyon, O., Vadeboncoeur, N., Turbide, G., Diaz, A., & Poirier, P. (2012). Effectiveness of a Pedometer-Based Program Using a Socio-cognitive Intervention on Physical Activity and Quality of Life in a Setting of Cardiac Rehabilitation. *Canadian Journal of Cardiology*, *28*(1), 27–32. https://doi.org/10.1016/j.cjca.2011.09.020

Jospin, L., Buntine, W., Boussaid, F., Laga, H., & Bennamoun, M. (2020). *Hands-on Bayesian Neural Networks – a Tutorial for Deep Learning Users*.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, *2017-December*.

Kwon, Y., Won, J. H., Kim, B. J., & Paik, M. C. (2020). Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation.

Computational Statistics and Data Analysis, *142*. https://doi.org/10.1016/j.csda.2019.106816

Kyriakides, G., & Margaritis, K. (2020). *An Introduction to Neural Architecture Search for Convolutional Networks*.

Mahadevaiah, G., RV, P., Bermejo, I., Jaffray, D., Dekker, A., & Wee, L. (2020). Artificial intelligence-based clinical decision support in modern medical physics: Selection, acceptance, commissioning, and quality assurance. *Medical Physics*, *47*(5). https://doi.org/10.1002/mp.13562

Noè, U., & Husmeier, D. (2018). *On a New Improvement-Based Acquisition Function for Bayesian Optimization*.

Pedersen, C. L., & Ritter, T. (2022). Updating the theory of industrial marketing: Industrial marketing as a Bayesian process of belief-updating. *Industrial Marketing Management*, *102*, 403–420. https://doi.org/https://doi.org/10.1016/j.indmarman.2022.02.008

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*.

Shridhar, K., Laumann, F., & Liwicki, M. (2018). *Uncertainty Estimations by Softplus normalization in Bayesian Convolutional Neural Networks with Variational Inference*.

Shridhar, K., Laumann, F., & Liwicki, M. (2019). *A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference*.

Spiegelhalter, D. J., Abrams, K. R., & Myles, J. P. (2003). *Bayesian Approaches to Clinical Trials and Health-Care Evaluation*. John Wiley & Sons, Ltd. https://doi.org/10.1002/0470092602

Strath, S. J., Kaminsky, L. A., Ainsworth, B. E., Ekelund, U., Freedson, P. S., Gary, R. A., Richardson, C. R., Smith, D. T., & Swartz, A. M. (2013). Guide to the Assessment of Physical Activity: Clinical and Research Applications. *Circulation*, *128*(20), 2259–2279. https://doi.org/10.1161/01.cir.0000435708.67487.da

Sutton, R. T., Pincock, D., Baumgart, D. C., Sadowski, D. C., Fedorak, R. N., & Kroeker, K. I. (2020). An overview of clinical decision support systems: benefits, risks, and strategies for success. *Npj Digital Medicine*, *3*(1), 17. https://doi.org/10.1038/s41746-020-0221-y

Tolstikhin, I., Bousquet, O., Schölkopf, B., Thierbach, K., Bazin, P. L., de Back, W., Gavriilidis, F., Kirilina, E., Jäger, C., Morawski, M., Geyer, S., Weiskopf, N., Scherf, N., Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Doersch, C. (2018). An overview of gradient descent optimization algorithms. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11046 LNCS* (Issue NeurIPS).

van Erp, S. (2020). A Tutorial on Bayesian Penalized Regression with Shrinkage Priors for Small Sample Sizes. In *Small Sample Size Solutions*. https://doi.org/10.4324/9780429273872-6

Anil, Nish. 2022. *Communication in a microservice architecture.* 13 04. Accès le 04 17, 2022. https://docs.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/communication-in-microservice-architecture.

Berner, Eta S. 2016. *Clinical Decision Support Systems: Theory and Practice.* Springer Nature.

Blackard, Jock A., Denis J. Dean, et Charles W. Anderson. 1998. «Covertype Data Set.»

Bright, Tiffani J, Anthony Wong, Ravi Dhurjati, Erin Bristow, Lori Bastian, Remy R Coeytaux, Gregory Samsa, et al. 2012. «Effect of clinical decision-support systems: a systematic review.» *Annals of Internal Medicine.*

Capstera. 2022. *Business Capabilities determine Microservices.* 05 01. Accès le 04 17, 2022. https://www.capstera.com/business-capabilities-determine-microservices/.

Chen, Jonathan H, Mary K Goldstein, Steven M Asch, Lester Mackey, et Russ B Altman. 2017. «Predicting inpatient clinical order patterns with probabilistic topic models vs conventional order sets.» *Journal of the American Medical Informatics Association.*

Dehghani, Zhamak. 2018. *How to break a Monolith into Microservices.* 28 04. Accès le 04 17, 2022. https://martinfowler.com/articles/break-monolith-into-microservices.html.

Edinburgh, University of. s.d. *COVER TYPE DATA.* Accès le 2022. http://www.inf.ed.ac.uk/teaching/courses/dme/fredduc/covtype.html.

2016. *Forest covertypes.* Accès le 2022. https://scikit-learn.org/0.18/datasets/covtype.html.

Forrest, Graeme N., Trevor C. Van Schooneveld, Ravina Kullar, Lucas T. Schulz, Phu Duong, et Michael Postelnick. 2014. «Use of Electronic Health Records and Clinical Decision Support Systems for Antimicrobial Stewardship.» *Clinical Infectious Diseases.*

Fowler, Martin. 2002. *Patterns of Enterprise Application Architecture.*

Gamma, Erich, Richard Helm, Ralph Johnson, et John Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional.

Head, Tim, et Holger Nahrstaedt. 2020. *Visualizing optimization results.* Accès le 2022. https://scikit-optimize.github.io/0.8/auto_examples/plots/visualizing-results.html.

Hohpe, Gregor, et Bobby Woolf. 2003. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions.*

Houle, Julie, et Maria-Cecilia Gallani. 2017. «Faisabilité et acceptabilité d'un programme d'activité physique basé sur le podomètre associé à une intervention sur Internet auprès de patients atteints d'une maladie coronarienne (ePOD-Cardio).»

Jospin, Laurent Valentin, Hamid Laga, Farid Boussaid, Wray Buntine, et Mohammed Bennamoun. 2022. «Hands-on Bayesian Neural Networks – A Tutorial.»

Kamperis, Stathis. 2021. *Acquisition functions in Bayesian Optimization.* 11 06. Accès le 06 10, 2022. https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html.

Klann, Jeffrey G, Peter Szolovits, Stephen M Downs, et Gunther Schadow. 2014. «Decision support from local data: creating adaptive order menus from past clinician behavior.» *Journal of Biomedical Informatics.*

Microsoft. s.d. *Retry pattern.* Accès le 2022. https://docs.microsoft.com/en-us/azure/architecture/patterns/retry.

Newman, D.J., et A. Asuncion. 2007. https://archive.ics.uci.edu/ml/datasets/covertype.

Pettigrew, Myriam. 2021. «Scénario rencontre infirmière.»

Rajkomar, Alvin, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, et al. 2018. «Scalable and accurate deep learning with electronic health records.» *npj Digital Medicine.*

Raymond, Eric Steven. 2003. *The Art of UNIX Programming.* Addison-Wesley Professional. Accès le 2022. http://www.catb.org/~esr/writings/taoup/html/index.html.

Richardson, Chris. 2018. *Microservices Patterns: With examples in Java.*

scikit-learn. s.d. *Matern.* https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.Matern.html.

—. s.d. *White.* https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.WhiteKernel.html.

scikit-optimize. s.d. https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html.

Vernon, Vaughn. 2013. *Implementing Domain-Driven Design.* Addison-Wesley Professional.

Vounatsos, Panagiotis. 2019. *Epistemic vs. Aleatory uncertainty.* 03. Accès le 04 2022. http://apppm.man.dtu.dk/index.php/Epistemic_vs._Aleatory_uncertainty.

Weinberger, Kilian. 2018. *Lecture 15: Gaussian Processes.* Accès le 2022. https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html.

Wikiwand. s.d. *Brier score.* Accès le 2022. https://www.wikiwand.com/en/Brier_score#/Original_definition_by_Brier.

Zauderer, Marjorie Glass, Ayca Gucalp, Andrew S. Epstein, Andrew David Seidman, Aryeh Caroline, Svetlana GranovskyJulia Fu, Jeffrey Keesing, et al. 2014. «Piloting IBM Watson Oncology within Memorial Sloan Kettering's regional network.» *Journal of Clinical Oncology.*

Annexes

Outils et spécifications

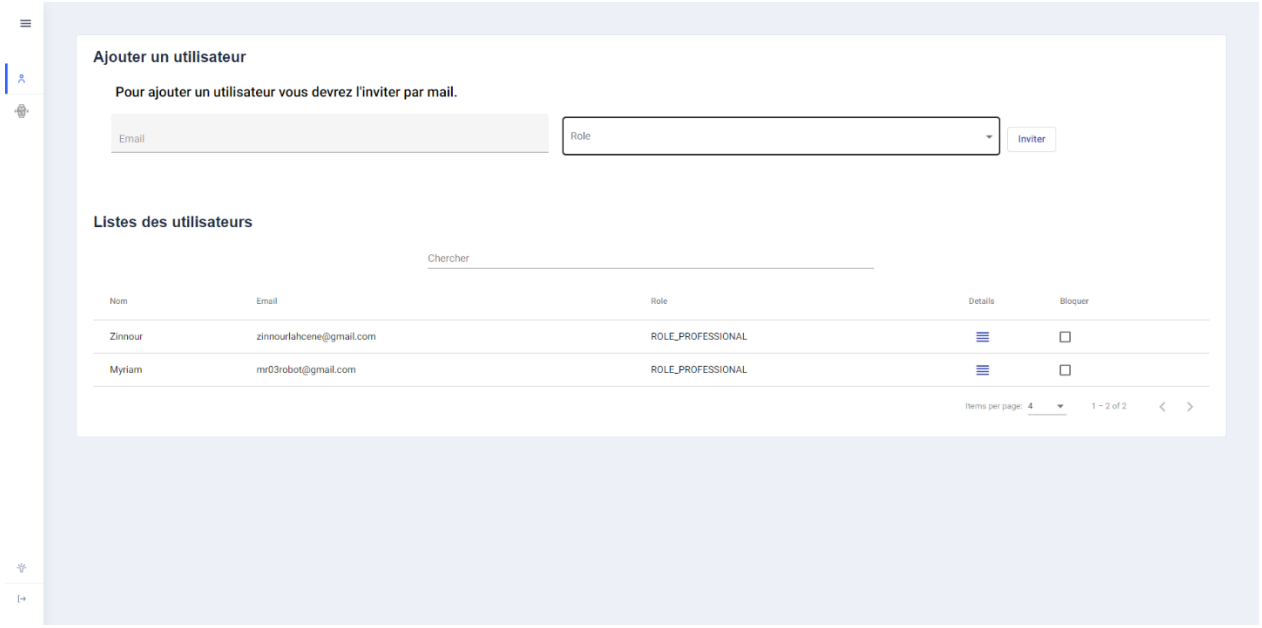
Dans le tableau suivant, nous présentons les outils que nous avons utilisés et leurs spécifications :

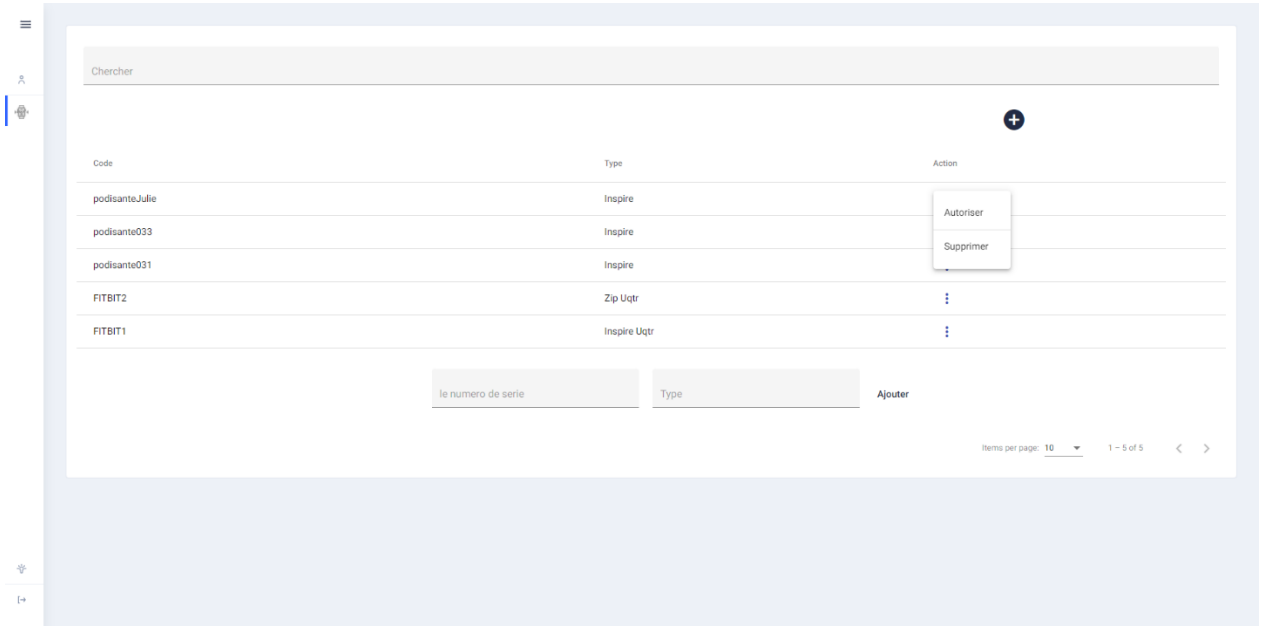
Outil	Description
Tensorflow, Tensorflow probability	Deux bibliothèques pour les probabilités, l'apprentissage automatique et l'intelligence artificielle.
Scikit-learn	Bibliothèque d'apprentissage automatique pour le langage de programmation Python. Mieux connu pour le prétraitement des données.
Scikit-optimize	Une bibliothèque Python pour minimiser les fonctions de boîte noire coûteuses et bruyantes.
Spring	Cadre d'application open source qui fournit un support d'infrastructure pour le développement d'applications Java.
PostgreSQL	Système de gestion de base de données relationnelle open source.
Angular	Une structure logicielle d'application Web gratuit et open source basée sur TypeScript.
Fitbit zip et inspire	Deux moniteurs Fitbit pour détecter les activités et les mesures corporelles.
Heroku	Utilisé pour déployer, gérer et mettre à l'échelle des applications modernes.
Kaggle TPU v3-8	Plate-forme pour entraîner les modèles de réseaux de neurones.
RabbitMQ	Agent de messagerie open source.

Tableau 18 Outils et spécifications.

Interface utilisateur

Administrateur UI





Clinicien UI

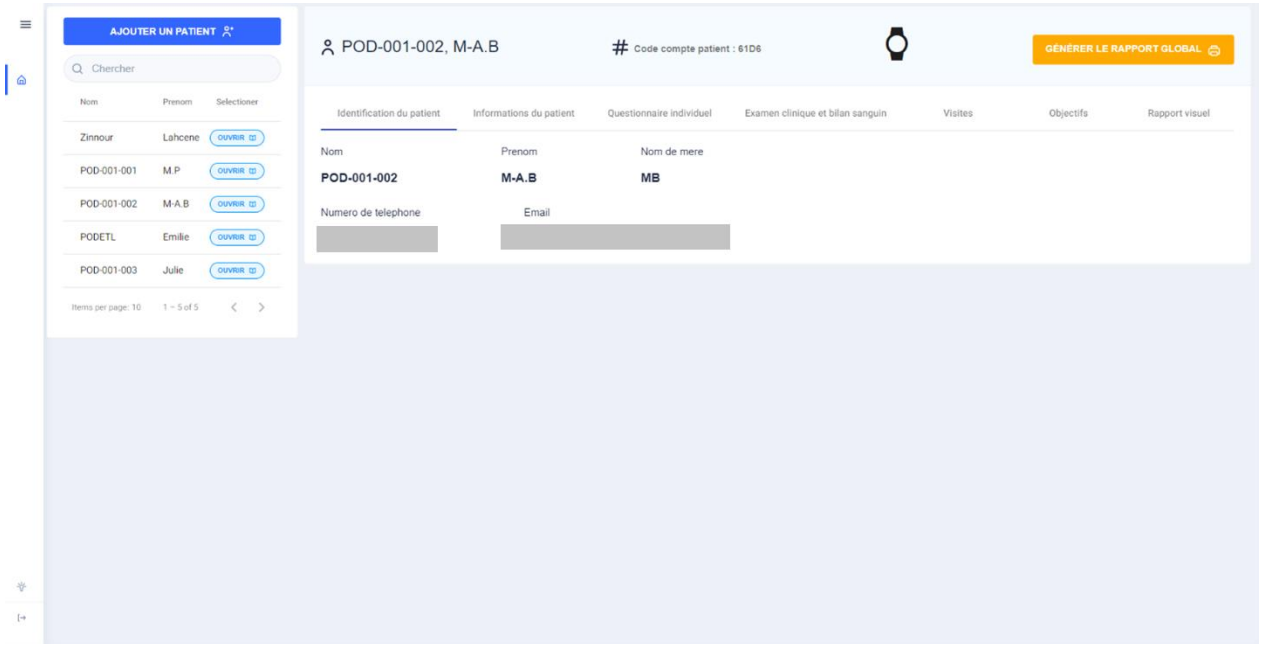


Figure 41 Identification du patient.

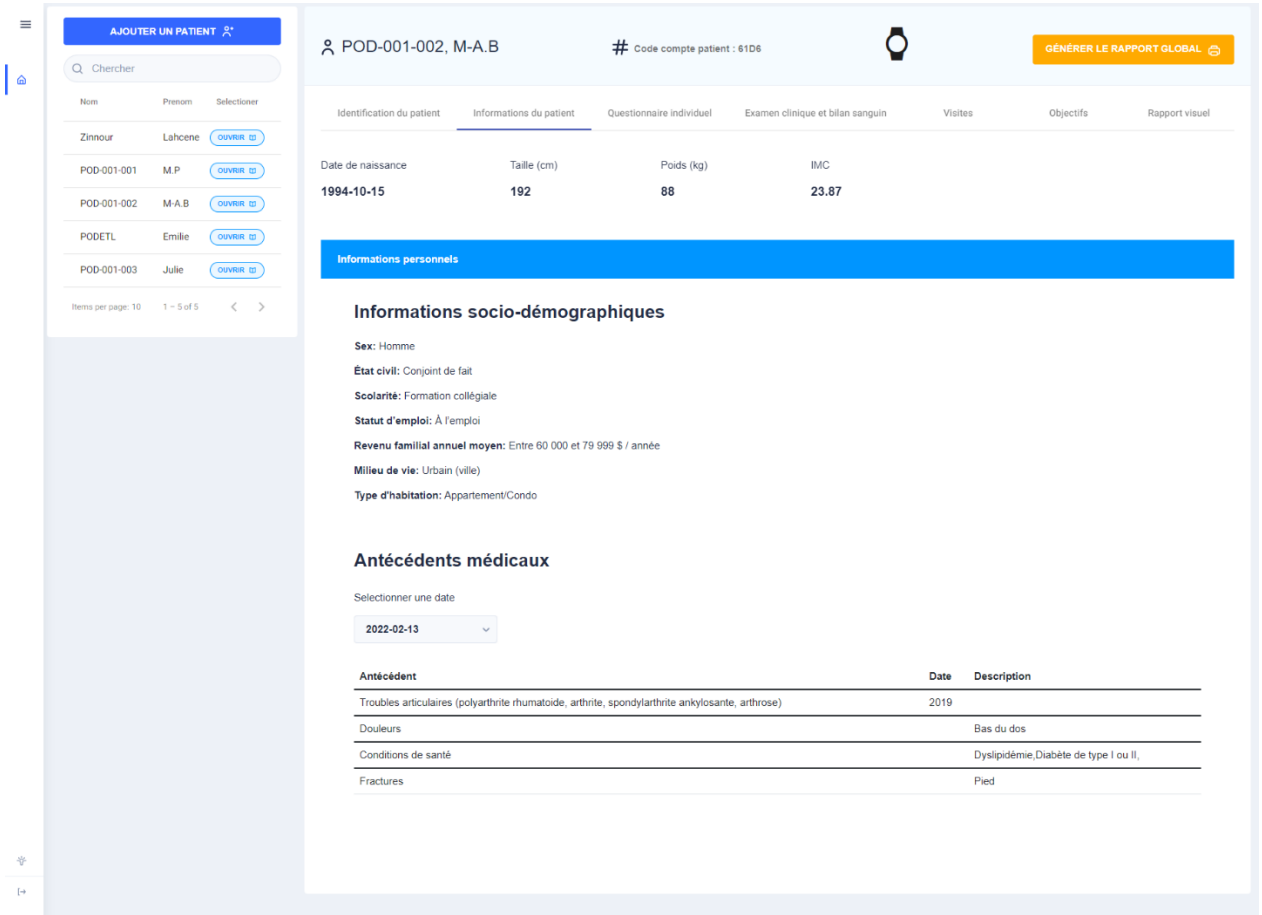


Figure 42 Informations du patient.

AJOUTER UN PATIENT

Chercher

Nom	Prénom	Sélectionner
Zinnour	Lakeane	
POD-001-001	M.P.	
POD-001-002	M-A.B	
PODEL	Emilie	
POD-001-003	Julie	

POD-001-002, M-A B

Code compte patient

Identification du patient | Informations du patient | Questionnaire individuel | Examen clinique et bilan sanguin | Visites | Objectifs | Rapport visuel

INFORMATIONS SOCIO-DÉMOGRAPHIQUES

Complétez les informations ci-dessous. Cochez une seule case par question.

Sexe :
- ◯ Femme
- ◯ Homme
- ◯ Autre
- ◯ Préfère ne pas répondre

État civil :
- ◯ Célibataire
- ◯ Marié
- ◯ Conjoint de fait
- ◯ Veuf/veuve
- ◯ Séparé
- ◯ Divorcé

Scolarité :
- ◯ < 11 ans de scolarité (Diplôme d'études secondaires)
- ◯ Diplôme d'études secondaires (DEP)
- ◯ Formation professionnelle
- ◯ Formation collégiale
- ◯ Formation universitaire 1er cycle
- ◯ Formation universitaire 2ème et 3ème cycle

Statut d'emploi :
- ◯ À l'emploi
- ◯ Retraité
- ◯ Sans emploi
- ◯ Au foyer
- ◯ Autre

Revenu familial annuel moyen :
- ◯ < 20 000 $ / année
- ◯ Entre 20 000 et 39 999 $ / année
- ◯ Entre 40 000 et 59 999 $ / année
- ◯ Entre 60 000 et 79 999 $ / année
- ◯ 80 000 $ / année

Milieu de vie :
- ◯ Urbain (ville)
- ◯ Rural (campagne)

Type d'habitation :
- ◯ Maison unifamiliale
- ◯ Appartement/Condo
- ◯ Maison de retraite
- ◯ CHSLD/Soins longue durée

ANTÉCÉDANTS MÉDICAUX

Indiquez les conditions de santé pour lesquelles vous avez déjà eu un diagnostic dans votre vie ainsi que l'année du diagnostic. Si vous avez eu une même condition plusieurs fois, inscrire les années correspondantes.

Angine
- ◯ Non
- ◯ Oui

Infarctus/crise cardiaque
- ◯ Non
- ◯ Oui

Pontages coronariens
- ◯ Non
- ◯ Oui

Angioplastie coronarienne
- ◯ Non
- ◯ Oui

Maladie valvulaire
- ◯ Non
- ◯ Oui

Arythmie cardiaque
- ◯ Non
- ◯ Oui

Insuffisance cardiaque (NYHA I ou II)
- ◯ Non
- ◯ Oui

Insuffisance cardiaque (NYHA III ou IV)
- ◯ Non
- ◯ Oui

AVC
- ◯ Non
- ◯ Oui

ICT
- ◯ Non
- ◯ Oui

Maladie vasculaire périphérique
- ◯ Non
- ◯ Oui

Cancer
- ◯ Non
- ◯ Oui

Maladie pulmonaire obstructive chronique (MPOC)
- ◯ Non
- ◯ Oui

Asthme
- ◯ Non
- ◯ Oui

Insuffisance rénale/dialyse/transplantation
- ◯ Non
- ◯ Oui

Troubles articulaires (polyarthrite rhumatoïde, Arthrite, Spondylarthrite ankylosante, Arthrose)
- ◯ Non
- ◯ Oui

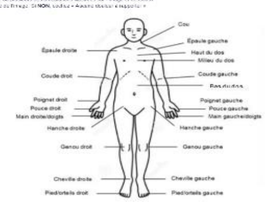
Maladie neuro-dégénérative (ex : sclérose en plaque Parkinson, Alzheimer)
- ◯ Non
- ◯ Oui

Avez-vous l'une des conditions de santé ci-dessous ?
- ☐ Hypertension artérielle
- ☐ Dyslipidémie
- ☐ Dépression
- ☐ Obésité
- ☐ Diabète de type I ou II
- ☐ Trouble anxieux
- ☐ Autre

Troubles musculo-squelettiques

Avez-vous déjà eu des fractures ?
- ◯ Non
- ◯ Oui

À l'heure actuelle, avez-vous des douleurs à l'une ou plusieurs des articulations illustrées sur l'image ci-dessous?
Si **OUI**, cochez la/les case(s) correspondantes à droite de l'image. Si **NON**, cochez « Aucune douleur à rapporter ».



- ☐ Aucune douleur
- ☐ Cou
- ☐ Épaule droite
- ☐ Épaule gauche
- ☐ Milieu du dos
- ☐ Haut du dos
- ☐ Bas du dos
- ☐ Coude gauche
- ☐ Coude droit
- ☐ Poignet gauche
- ☐ Poignet droit
- ☐ Main/doigts gauche
- ☐ Main/doigts droit
- ☐ Pouce gauche
- ☐ Pouce droit
- ☐ Hanche gauche
- ☐ Hanche droite
- ☐ Genou gauche
- ☐ Genou droit
- ☐ Cheville gauche
- ☐ Cheville droite
- ☐ pied/orteils gauche
- ☐ pied/orteils droit

Consommation de produits alcoolisés, tabagiques ou drogues (si actuel/occasionnel, indiquez la fréquence, ex : nombre de cigarette/jour ou nombre de consommations/semaine, etc)

Alcool
- ◯ Jamais
- ◯ Ancien
- ◯ Actuel / Occasionnel

Tabac
- ◯ Jamais
- ◯ Ancien
- ◯ Actuel / Occasionnel

Drogue
- ◯ Jamais
- ◯ Ancien
- ◯ Actuel / Occasionnel

Énumérer tous les médicaments que vous prenez régulièrement. Inscrivez le nom du médicament en lettres moulées, il n'est pas nécessaire de spécifier la dose.

AJOUTER UN MÉDICAMENT +

Ou

☐ **Je ne prends pas de médicaments**

ENREGISTRER LE QUESTIONNAIRE

Figure 43 Questionnaire individuel.

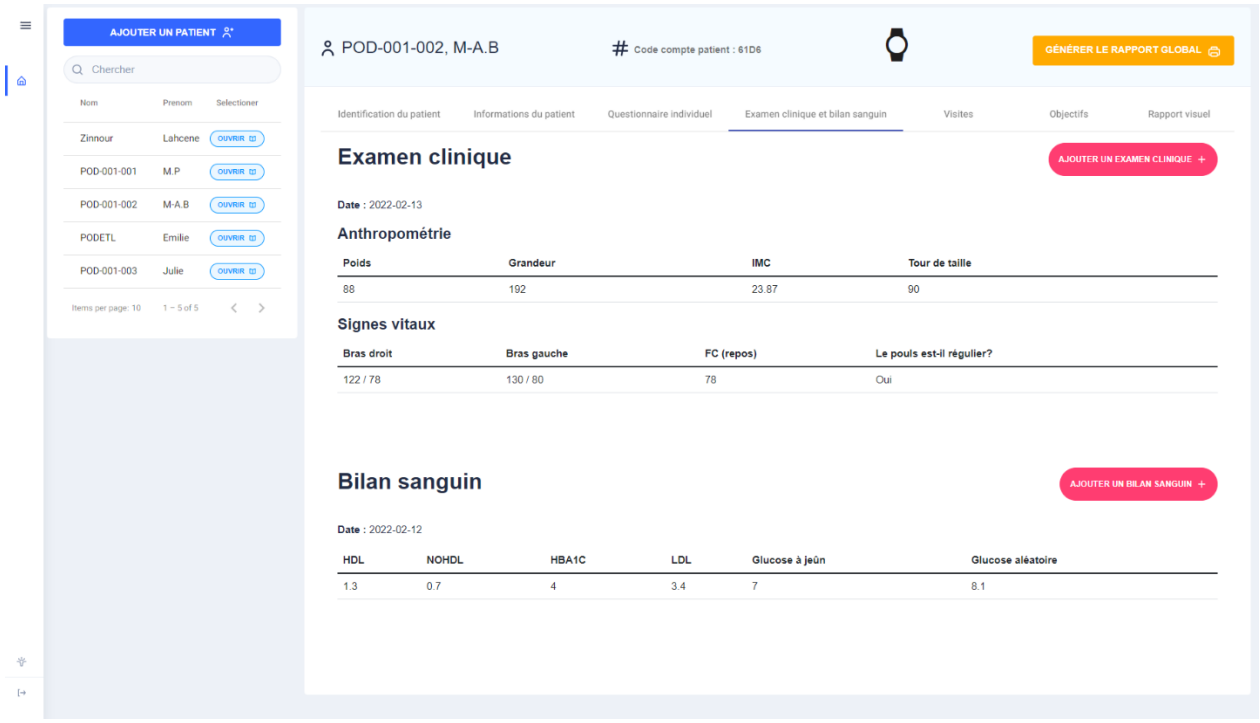


Figure 44 Examen clinique et bilan sanguin.

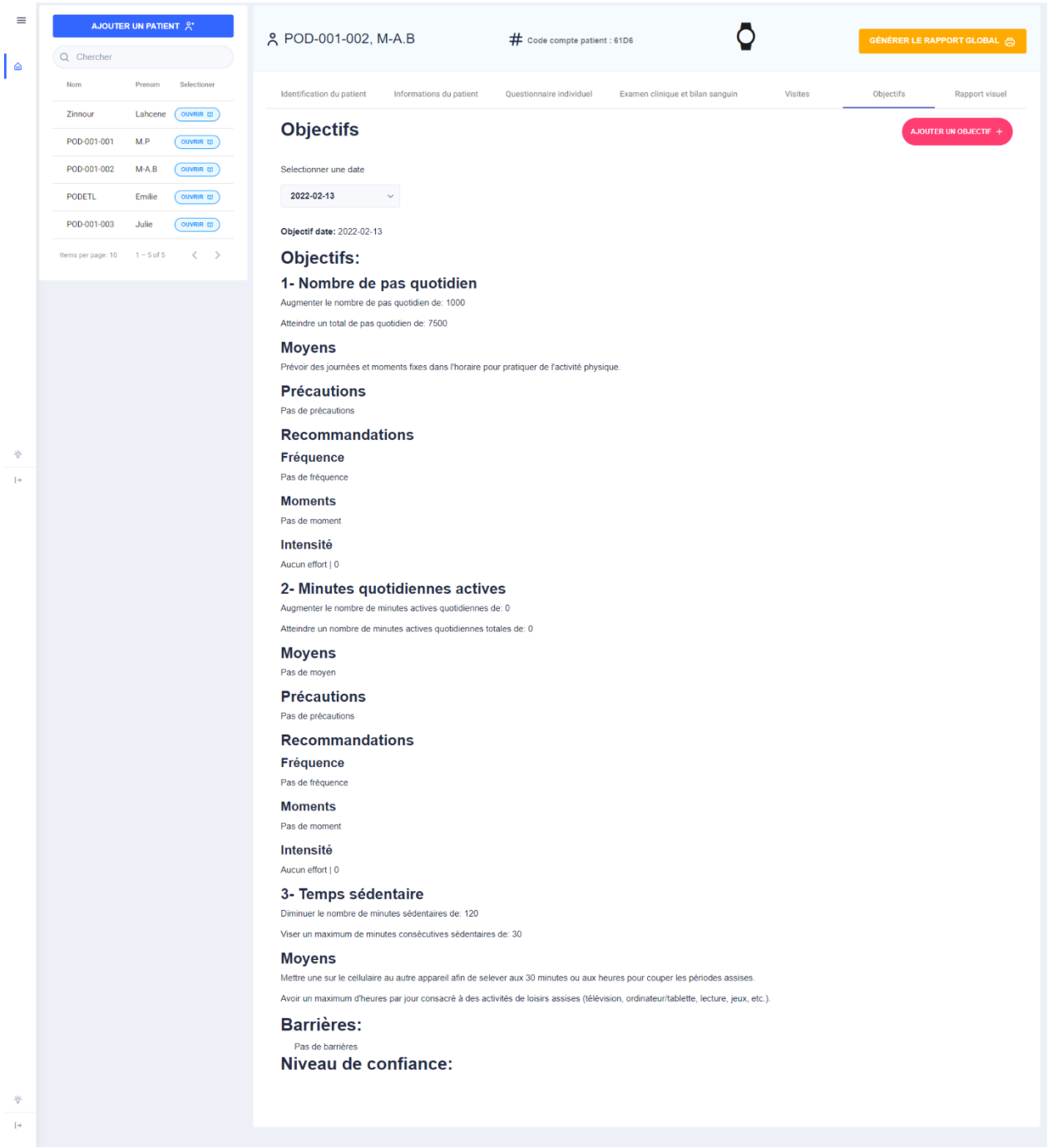


Figure 45 Objectifs.

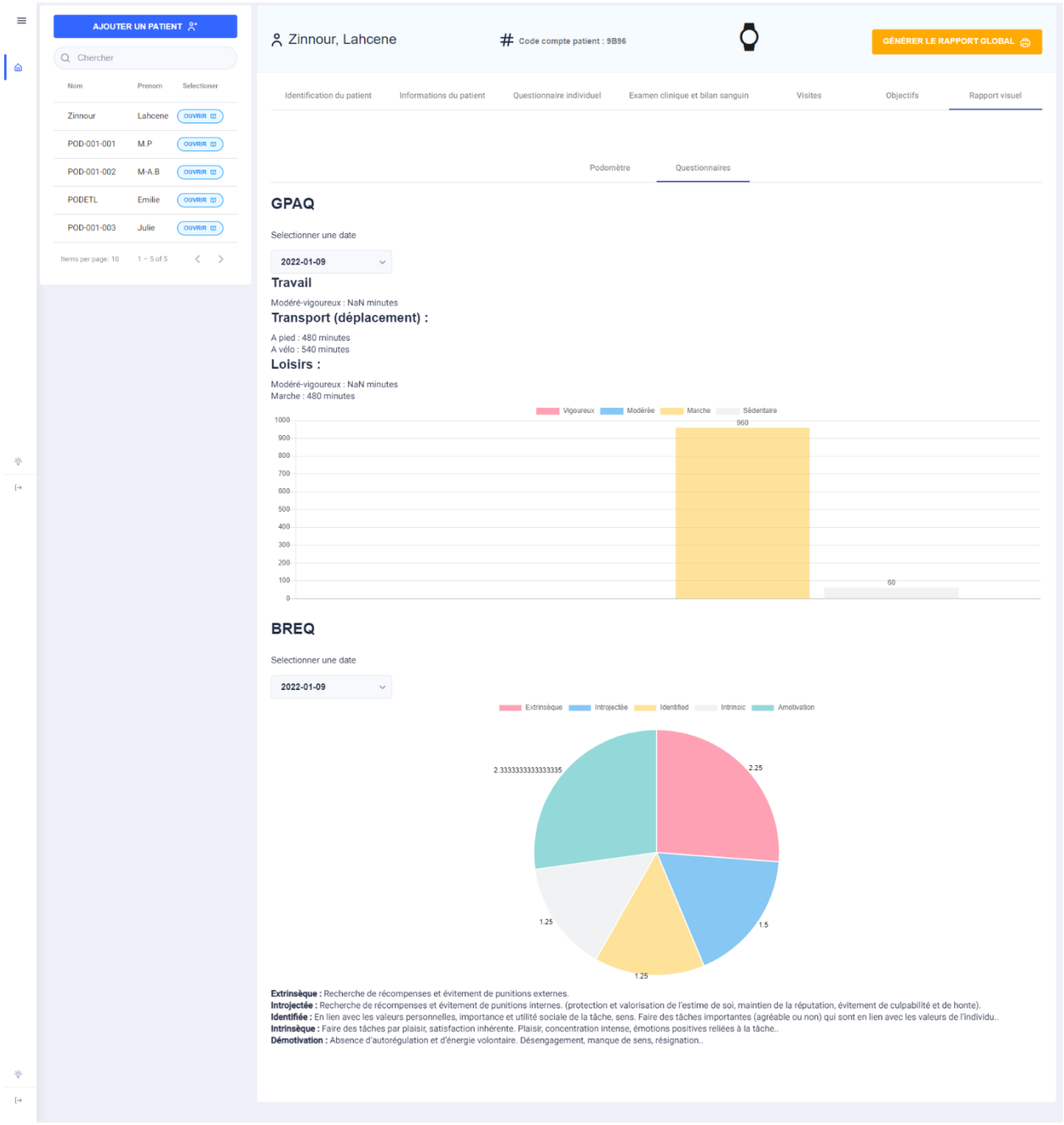


Figure 46 Rapport visuel (podomètre).

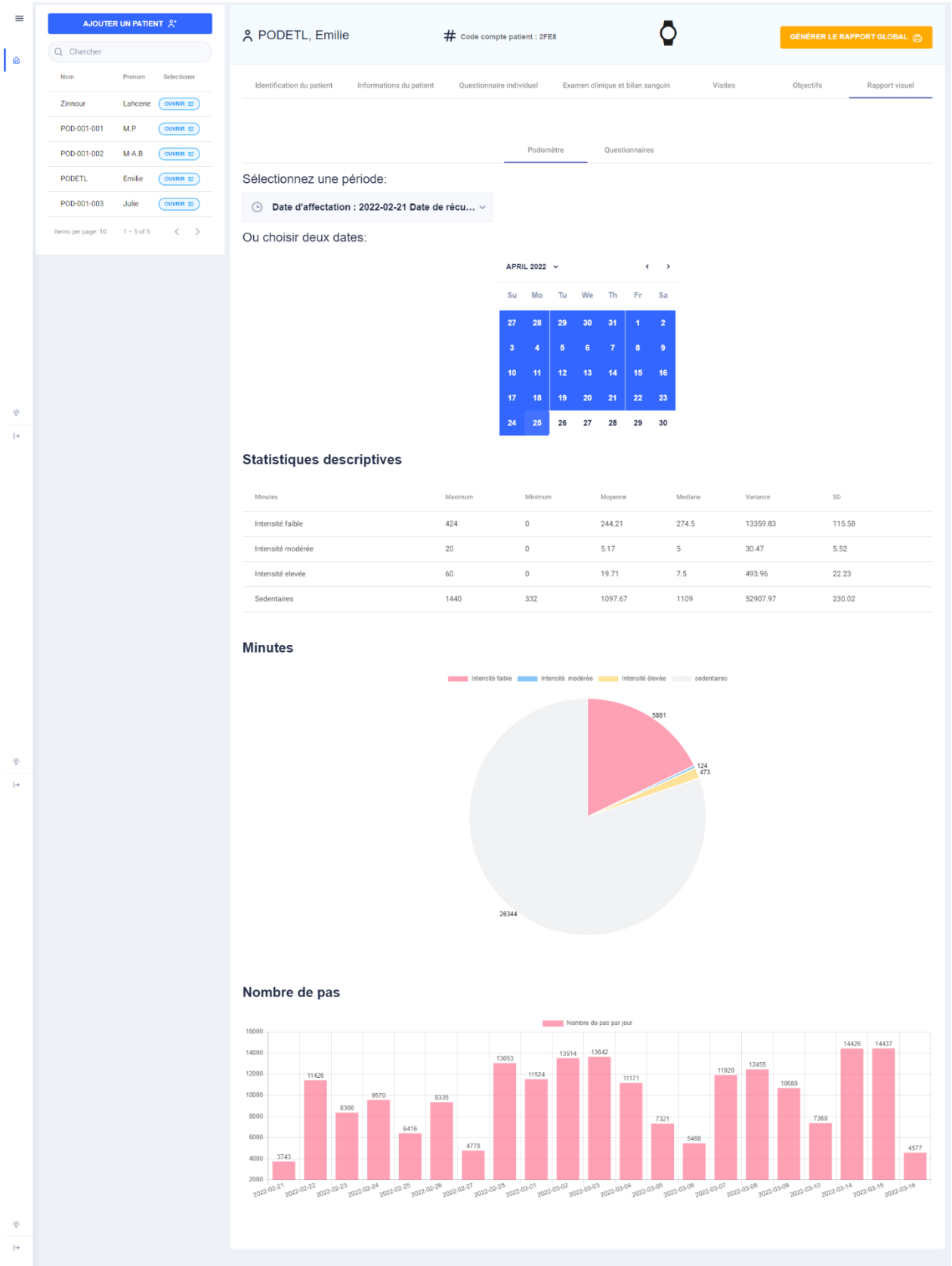


Figure 47 Rapport visuel (questionnaires).

Patient UI

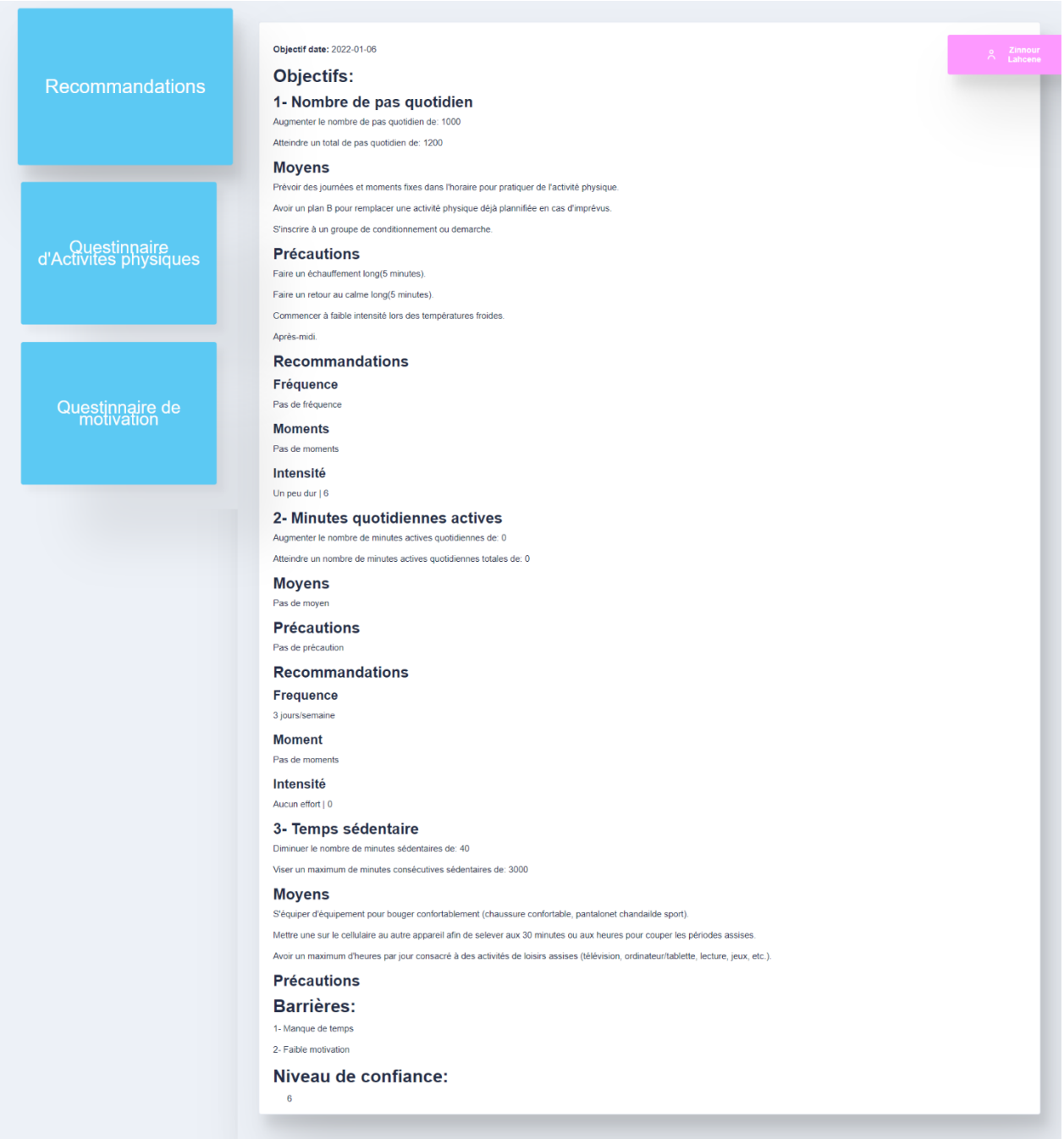


Figure 48 Recommandations et objectifs.

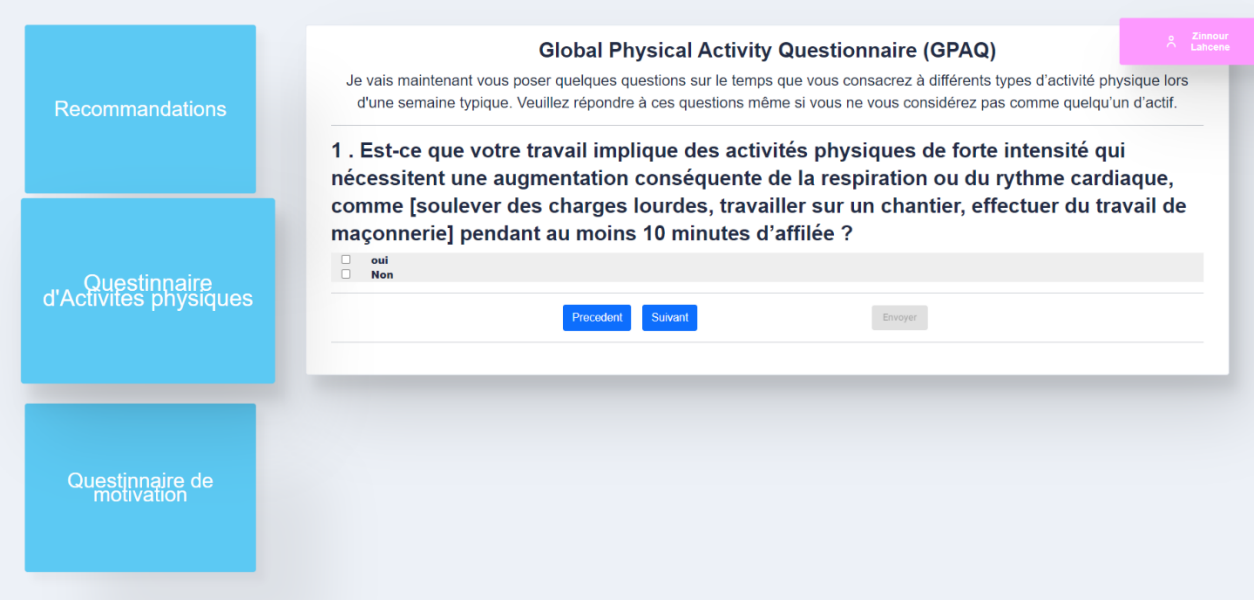


Figure 49 Questionnaire GPAQ.

Recommandations

Questinnaire d'Activités physiques

Questinnaire de motivation

BREQ-2 QUESTIONNAIRE : MOTIVATION A PRATIQUER UNE ACTIVITE PHYSIQUE

Pourquoi prenez-vous part à un programme d'activité physique ? Pourquoi décidons-nous de pratiquer ou non un sport ? Attribuez une cote de 0 à 4 à chacun des points ci-dessous, en fonction de ce qui vo correspond le mieux. Attention : il n'y a pas de "bonne" ou de "mauvaise" réponse ! Notre seul objec connaître votre attitude personnelle face au sport.

Zimmour Lahcene

Je fais de l'activité physique parce que les autres estiment que je dois en faire

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je me sens coupable si je ne fais pas de l'activité physique

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

J'apprécie les avantages que m'apporte l'activité physique

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je fais de l'activité physique parce que j'aime ça

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je ne vois pas pourquoi je devrais faire de l'activité physique

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je fais de l'activité physique parce que mes amis / ma famille / mon partenaire estime(nt) que je dois en faire

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

J'ai honte quand je manque un de mes entraînements

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

J'estime qu'il est important de pratiquer une activité physique régulière

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je ne vois pas pourquoi je devrais prendre la peine de faire de l'activité physique

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

J'apprécie mes séances d'entraînement

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je pratique parce que les autres n'apprécieront pas que je ne le fasse pas

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je ne vois pas l'utilité de l'activité physique

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je me sens minable quand je n'ai pas fait de l'activité physique pendant un certain temps

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

J'estime qu'il est important de faire de l'activité physique régulièrement

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je trouve que l'activité physique est agréable

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je trouve que mes amis / ma famille / mon partenaire font pression sur moi pour que je fasse de l'activité physique

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je me sens nerveux(se) si je ne fais pas de l'activité physique régulièrement

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

L'activité physique m'apporte du plaisir et de la satisfaction

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Je trouve que faire de l'activité physique est une perte de temps

- 0 (Non, pas du tout !)
- 1
- 2 (C'est parfois vrai...)
- 3
- 4 (Oui, tout à fait !)

Envoyer

Figure 50 Questionnaire BREQ-2.

Implémentation d'un réseau de neurones bayésien

A posteriori mean field Q(x)

```
def  posterior_mean_field(kernel_size, bias_size=0, dtype=None):
    n = kernel_size + bias_size
    return tf.keras.Sequential([ tfpl.VariableLayer(
                                            tfpl.IndependentNormal.params_size(n),
                                            dtype=dtype),
                                 tfpl.IndependentNormal(n) ])
```

A priori P(x)

SMP

```
def scale_mixture_prior(kernel_size, bias_size, dtype=None):
    distribution = tfd.MixtureSameFamily(
        mixture_distribution=tfd.Categorical(probs=[0.5, 0.5]),
        components_distribution=tfd.Normal(loc=[0, 0], scale=[0.5, 0.3]))
    prior = Sequential([
        tfpl.DistributionLambda(
            lambda t: distribution , name='scale_mixture_prior'
        )])
    return prior
```

SSP

```
def spike_and_slab_distribution(event_shape, dtype):
    distribution = tfd.Mixture(
        cat=tfd.Categorical(probs=[0.5, 0.5]),
        components=[
            tfd.Independent(tfd.Normal(
                loc=tf.zeros(event_shape, dtype=dtype),
                scale=1.0*tf.ones(event_shape, dtype=dtype)),
                            reinterpreted_batch_ndims=1),
            tfd.Independent(tfd.Normal(
                loc=tf.zeros(event_shape, dtype=dtype),
                scale=10.0*tf.ones(event_shape, dtype=dtype)),
                            reinterpreted_batch_ndims=1)],
    name='spike_and_slab')
    return distribution

def spike_and_slab_prior(kernel_size, bias_size, dtype=None):
    n = kernel_size + bias_size
    prior = Sequential([
        tfpl.DistributionLambda(
            lambda t: spike_and_slab_distribution(n, dtype), name='spike_and_slab_prior'
        )
    ])
    return prior
```

EBP

```
def empirical_bayes_prior(kernel_size, bias_size=0, dtype=None):
    n = kernel_size + bias_size
    c = np.log(np.expm1(1.))
    return tf.keras.Sequential([
        tfp.layers.VariableLayer(2 * n, dtype=dtype),
        tfp.layers.DistributionLambda(
            lambda t: tfd.Independent(
                tfd.Normal(loc=t[..., :n], scale=1e-5 + tf.nn.softplus(c + t[..., n:])),
                reinterpreted_batch_ndims=1), name='empirical_bayes_prior'),
    ])
```

INP

```
def isotropic_trainable_prior(kernel_size, bias_size=0, dtype=None):
    n = kernel_size + bias_size
    return tf.keras.Sequential([
      tfp.layers.VariableLayer(n, dtype=dtype),
      tfp.layers.DistributionLambda(lambda t: tfd.Independent(
          tfd.Normal(loc=tf.zeros(n), scale=1),
          reinterpreted_batch_ndims=1), name='isotropic_trainable_prior')
    ])
```

DTP

```
def diag_trainable_prior(kernel_size, bias_size=0, dtype=None):
    n = kernel_size + bias_size
    return tf.keras.Sequential([
      tfp.layers.VariableLayer(n, dtype=dtype),
      tfp.layers.DistributionLambda(lambda t: tfd.Independent(
          tfd.Normal(loc=t, scale=1),
          reinterpreted_batch_ndims=1), name='diag_trainable_prior'),
    ])
```

Espace de recherche

```
 dim_learning_rate = Real(low=1e-6, high=1e-1, prior='log-uniform',
                          name='learning_rate')
dim_num_dense_layers = Integer(low=1, high=20, name='num_dense_layers')
dim_num_dense_nodes = Integer(low=1, high=700, name='num_dense_nodes')
dim_activation = Categorical(categories=['relu', 'sigmoid'],
                             name='activation')
dimensions = [dim_learning_rate,
              dim_num_dense_layers,
              dim_num_dense_nodes,
              dim_activation]

default_parameters = [2.3e-03, 2, 70, 'sigmoid']
```

Modèle de réseau de neurones bayésien

```
classes = 7
def create_model(learning_rate, num_dense_layers,
                 num_dense_nodes, activation):
    model = Sequential()
```

```
    model.add(InputLayer(input_shape=(training_examples.shape[1],)))
    for i in range(num_dense_layers):
        name = 'layer_{0}'.format(i+1)
        model.add(tfp.layers.DenseVariational(units=num_dense_nodes,
                                              activation=activation,
                                              make_posterior_fn=posterior_mean_field,
                                              make_prior_fn=spike_and_slab_prior
                                              kl_weight=1/train_size,
                                              kl_use_exact=True,
                                              name=name))
    model.add(tfp.layers.DenseVariational(7
                                          make_posterior_fn=posterior_mean_field,
                                          make_prior_fn= spike_and_slab_prior,
                                          kl_weight=1/train_size, kl_use_exact=True))
    model.add(tfp.layers.OneHotCategorical(classes,
                           convert_to_tensor_fn=tfp.distributions.Distribution.mean))

    optimizer = Adam(lr=learning_rate)

    model.compile(optimizer=optimizer,
                  loss=neg_log_likelihood,
                  metrics=['accuracy'])
    model.summary()

    return model
```

La fonction f

```
path_best_model = 'BNN_best_model.h5'
best_accuracy = 0.0
BNN_GLOBAL_MODEL = None

@use_named_args(dimensions=dimensions)
def f(learning_rate, num_dense_layers, num_dense_nodes, activation):

    model = create_model(learning_rate=learning_rate,
                         num_dense_layers=num_dense_layers,
                         num_dense_nodes=num_dense_nodes,
                         activation=activation)

    log_dir = log_dir_name(learning_rate, num_dense_layers,
                           num_dense_nodes, activation)

    erly_stopping_callback = keras.callbacks.EarlyStopping(monitor='val_accuracy',
                              min_delta=0.0005,
                              Baseline=0.30
                              patience=5,
                              verbose=0, mode='max')

    history = model.fit(training_examples, training_targets,
                        epochs=118,
                        batch_size=1024,
                        #batch_size=16*tpu_strategy.num_replicas_in_sync,
                        validation_split=0.25,
                        callbacks=[TqdmCallback(verbose=1)],
                        verbose=0)

    print("Accuracy: {0:.2%}".format(accuracy))
    global best_accuracy
    global BNN_GLOBAL_MODEL

    if accuracy > best_accuracy:
        tf.saved_model.save(model, "/kaggle/working")
```

```
36.         best_accuracy = accuracy
37.         BNN_GLOBAL_MODEL = history
38.
39.     del model
40.     K.clear_session()
41.
42.     return -accuracy
```