STRATÉGIES D'APPRENTISSAGE AUTOMATIQUE POUR LA PRÉDICTION D'EFFETS TARDIFS ASSOCIÉS AU TRAITEMENT DE LA LEUCÉMIE AIGUË LYMPHOBLASTIQUE INFANTILE

par

Nicolas Raymond

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 28 décembre 2022

Le 28 décembre 2022

Le jury a accepté le mémoire de Nicolas Raymond dans sa version finale

Membres du jury

Professeur Martin Vallières
Directeur
Département d'informatique
Faculté des sciences
Université de Sherbrooke

Professeur Félix Camirand Lemyre
Membre interne
Département de mathématiques
Faculté des sciences
Université de Sherbrooke

Professeure Aïda Ouangraoua
Présidente-rapporteuse
Département d'informatique
Faculté des sciences
Université de Sherbrooke

Sommaire

La leucémie aiguë lymphoblastique (LAL) est le cancer le plus fréquemment diagnostiqué chez l'enfant. Approximativement deux tiers des survivants de la LAL infantile présentent une ou plusieurs complications de santé à l'âge adulte. Connues sous le nom d'effets tardifs, ces complications sont plutôt le fruit du traitement que de la maladie elle-même. Les mesures actuellement mises en place pour les visites de suivi post-traitement sont généralement uniformes à l'ensemble des survivants de cancers infantiles et ne sont pas nécessairement adaptées précisément aux survivants de la LAL infantile. Conséquemment, les effets tardifs peuvent être sous-diagnostiqués et, dans la plupart des cas, seulement pris en charge après leurs apparitions. D'autre part, les directives de soins actuelles peuvent également mener à un suivi plus intensif que nécessaire, entraînant parfois des inquiétudes chez les survivants en plus d'augmenter les coûts de soins. Ainsi, il est nécessaire de prédire l'apparition des effets tardifs plus tôt pour contribuer à la santé et au bien-être des survivants. Plusieurs travaux se sont concentrés sur la recherche de biomarqueurs pouvant aider à la prédiction des effets tardifs et, notamment, un article a mis de l'avant l'utilisation d'un modèle d'apprentissage automatique pour prévenir les effets liés à la détérioration de la forme cardio-respiratoire. Toutefois, aucune solution n'a fait usage de réseaux neuronaux jusqu'à présent. Dans ce projet de recherche, nous avons développé des réseaux de neurones graphiques efficients et mis en valeur leur interprétabilité à l'aide de multiples analyses conduites suite à leurs entraînements. En premier lieu, nous avons proposé un nouveau modèle d'estimation de la consommation d'oxygène maximale (c.-à-d.,VO_2 max) qui ne nécessite aucune participation à un test physique (e.g., test de marche de six minutes). Le VO_2 max est reconnu comme la meilleure mesure de la forme cardio-respiratoire, qui à son tour, est un bon indicateur du risque de

développement de certaines morbidités (e.g., obésité, dépression) chez les survivants. En second lieu, nous avons développé un modèle de prédiction de l'obésité utilisant des variables cliniques disponibles dès la fin du traitement de la LAL infantile, ainsi que plusieurs marqueurs génétiques (c.-à-d., polymorphismes à un seul nucléotide). Les réseaux de neurones graphiques mis en place durant ce projet ont permis d'obtenir de meilleures performances que d'autres modèles à structures arborescentes ou neuronales.

Mots-clés: leucémie aiguë lymphoblastique, réseaux neuronaux graphiques, apprentissage supervisé, médecine de précision, données multi-omiques

Remerciements

Je tiens d'abord à remercier mon directeur de recherche Martin Vallières pour ses conseils et son implication dans la réalisation de ce projet. Sa passion et son ouverture d'esprit m'ont permis de faire évoluer la recherche médicale dans un milieu créatif et convivial.

Je tiens également à remercier l'ensemble des collaborateurs du centre hospitalier universitaire Sainte-Justine ayant mis sur pieds l'étude PETALE et participé à l'acquisition des données utilisées au sein de ce projet. En particulier, je remercie Maxime Caru et Daniel Sinnett pour le partage de leur expertise clinique et la confiance qu'ils ont eu à mon égard.

Je remercie notamment Mehdi Mitiche, non seulement pour ses contributions au projet, mais également pour son écoute, sa motivation et sa rigueur de travail lors de son stage au sein de l'équipe.

Je remercie entre autres l'ensemble des étudiants du laboratoire MEDomics UdeS pour leur esprit de camaraderie et d'entraide.

Merci à mes parents Gaétan Raymond et Nicole Vachon pour leur support. Vous m'avez toujours encouragé à poursuivre mes ambitions et j'en serai toujours reconnaissant.

Finalement, un merci spécial à ma compagne Marjorie Dubois avec qui j'ai pu partager mon quotidien durant la réalisation de ma maîtrise. Tu as été là pour me soutenir et m'encourager dans chaque étape de cette expérience. Ce mémoire n'est qu'un chapitre à notre aventure.

- Nicolas

Table des matières

Liste des figures

Liste des tableaux

Abréviations

ADBG Arbres de décision avec boosting de gradient

CHUSJ Centre hospitalier universitaire Sainte-Justine

CSP Cellule souche pluripotente

DGS Descente de gradient stochastique

EPA Estimateur de Parzen à structure arborescente (TPE au chapitre 3)

LAL Leucémie aiguë lymphoblastique (ALL au chapitre 3)

PSN Polymorphisme à un seul nucléotide (SNP au chapitre 3)

RNGC Réseau de neurones graphique convolutif (GCN au chapitre 3)

SGE Séquençage du génome entier

Notations mathématiques

$f(\cdot)$	Fonction
$\mathcal{X}$	Ensemble
$\mathbf{x}$	Vecteur
x, X	Scalaire
$\lvert\mathcal{X}\rvert$	Cardinalité de l'ensemble $\mathcal{X}$
$\lVert\cdot\rVert$	Norme euclidienne
$\mathbb{P}(A\lvert B)$	Probabilité de A sachant B.
$\mathcal{M}_{m,n}(\mathcal{A})$	Ensemble des matrices $m \times n$ à coefficients dans $\mathcal{A}$
$\emptyset$	Ensemble vide
$\forall$	Pour tout
$t.q, \lvert, :$	Tel que

Introduction

Les interactions que nous initions ou entretenons quotidiennement avec les outils technologiques à notre disposition contribuent à la création et la sauvegarde d'un grand volume de données au sein de systèmes informatisés. C'est en particulier le cas pour le milieu de la santé où, après chaque intervention médicale, un vaste éventail d'informations se retrouve entreposé au sein des dossiers électroniques des patients [43]. Ce volume de données représente une opportunité importante pour l'amélioration des soins de santé, notamment pour le domaine de la médecine de précision qui priorise le traitement d'un patient par la mise en place de directives de soins personnalisées plutôt que la poursuite d'un programme commun à l'ensemble de la population atteinte du même problème de santé. En effet, les données des dossiers médicaux peuvent être recueillies pour créer des algorithmes d'apprentissage automatique permettant la sélection de la meilleur modalité de traitement disponible pour un patient en fonction de ses caractéristiques spécifiques.

L'hétérogénéité associée à la nature des différentes structures de données inhérentes aux dossiers médicaux électroniques pose toutefois un défi qui nécessite l'innovation de diverses solutions d'apprentissage automatique (c.-à-d., différentes architectures de modèles). Les réseaux neuronaux convolutifs, par exemple, sont grandement mis de l'avant pour la classification et la segmentation d'images [29, 70]. Les modèles à structure arborescente (p. ex., forêt aléatoire), quant à eux, sont souvent utilisés pour la réalisation de tâches impliquant des données entreposées sous une forme tabulaire (p. ex., données démographiques) [3]. Récemment, une attention particulière est portée vers le développement de modèles abordant l'utilisation de données structurées sous forme de graphes [41]. Ces derniers, portant le nom de réseaux de neurones graphiques, permettent de traiter simultanément les informations associées aux noeuds

et aux arêtes. Plusieurs ensembles de données médicales se présentent naturellement à l'aide d'un graphe. Nous pouvons entre autres penser aux structures moléculaires des différents médicaments administrés à un individu, ou même, avoir une vision plus élargie, et considérer l'ensemble des dossiers médicaux d'une clinique de médecine familiale, connectés l'un à l'autre par des liens de filiations.

L'interprétabilité est également un enjeu d'importance dans la construction de modèles d'apprentissage automatique pour le domaine de la santé. Dans un contexte d'aide à la prise de décision, il peut être requis qu'un utilisateur soit en mesure de décortiquer le cheminement amenant un modèle à émettre une certaine prédiction. En outre, la compréhension d'un modèle renforce la confiance qui lui est attribuée [42]. Récemment, plusieurs travaux portent sur le développement de techniques d'interprétation des réseaux neuronaux [22, 81]. En particulier, certaines d'entre-elles visent à analyser les modèles suite à leur optimisation (p. ex., visualisation des éléments d'une couche cachée, illustration du mécanisme d'attention) [3, 46, 51, 52]. Alors que ces techniques de post-analyse n'apportent qu'une compréhension sommaire du processus de prédiction derrière chaque observation, elles doivent être mises de l'avant pour motiver l'adoption des modèles à architectures neuronales lorsqu'ils permettent d'atteindre de meilleures performances de prédiction que d'autres algorithmes considérablement plus interprétables (p. ex., régression linéaire, arbre de décision).

L'amélioration des soins de suivi post-traitement associés à la leucémie aiguë lymphoblastique (LAL) infantile constitue un sujet de recherche d'intérêt pour la médecine de précision. Bien que près 9 enfants sur 10 survivent à cette maladie [67], 65% de ceux-ci présentent une ou plusieurs complications de santé liées au traitement lors de l'âge adulte [60]. Celles-ci, en plus d'affecter leur qualité de vie, peuvent parfois présenter un degré de sévérité élevé entraînant la mort [60]. Il est donc important d'adopter des mesures permettant la prévention des effets tardifs du traitement de la LAL infantile. Pour ce faire, les récents travaux de recherche proposent de prédire l'apparition de ces différents effets pour permettre d'entreprendre des démarches de suivi adaptées le plus tôt possible. En particulier, plusieurs analyses statistiques réalisées sur la cohorte de PETALE, une étude consacrée à la prévention des effets tardifs liés au traitement de la leucémie aiguë lymphoblastique chez l'enfant, ont permis de mettre en lumière des associations entre des biomarqueurs et certaines morbidités

liées au traitement de la LAL infantile [2, 7, 12, 18, 30, 35, 36, 38, 65, 66]. Alors que ces méthodes permettent de cerner des profils génériques et de leur associer des recommandations de soins telles que nous pouvons le voir dans le guide du *Children's Oncology Group* [25], celles-ci faillent à mettre sur pieds des directives de soins individuelles. Ainsi, les solutions incorporant des techniques d'apprentissage automatique comportent un grand potentiel dans la quête de personnalisation des soins de suivi post-traitement. Néanmoins, à l'exception du modèle de régression linéaire présenté par Labonté *et al.* [37] pour prédire certains effets tardifs par l'évaluation de la forme cardio-respiratoire, aucun travail n'a fait objet de l'utilisation de telles méthodes jusqu'à présent.

Dans ce mémoire, nous aborderons l'utilisation des réseaux de neurones graphiques pour la prédiction d'effets tardifs liés au traitement de la LAL infantile et mettrons de l'avant leur interprétabilité à l'aide de différentes techniques de post-analyse. D'abord, nous décrirons le processus biologique derrière le développement de cette maladie, présenterons les principaux effets tardifs recensés et ferons le survol d'un précédent projet de recherche visant la prévention de ceux-ci. Ensuite, nous introduirons les concepts fondamentaux de l'apprentissage supervisé et décrirons d'un point de vue théorique les différents modèles considérés durant ce projet. Après, nous présenterons un article soumis au journal *Communications Medicine*, dans lequel s'inscrivent nos principaux résultats de recherche. Finalement, nous ferons un récapitulatif des motivations et contributions de ce mémoire et discuterons des perspectives futures liées à notre projet de recherche.

Chapitre 1

Leucémie aiguë lymphoblastique

La leucémie est un terme utilisé pour identifier les différents cancers pouvant se former au sein des cellules sanguines d'un individu. En particulier, la leucémie aiguë lymphoblastique (LAL) représente le cancer le plus fréquemment diagnostiqué chez les enfants [36]. Considérant que la LAL infantile se trouve au coeur de notre projet de recherche, nous débuterons ce chapitre en établissant le portrait cette maladie. Nous poursuivrons ensuite en faisant un survol de PETALE, une étude dédiée principalement à l'identification de biomarqueurs pour la prédiction d'effets tardifs associés au traitement de la LAL infantile [47].

1.1 Portrait de la maladie

Dans cette section, nous explorerons les aspects principaux de la LAL infantile. Nous décrirons le processus de développement de la maladie, mettrons en lumière les différentes étapes du traitement puis discuterons des effets tardifs liés à celui-ci.

1.1.1 Description

Se trouvant au centre des os, la moelle osseuse produit les cellules souches pluripotentes (CSPs) destinées à se transformer en différentes cellules sanguines, soit les globules rouges, les globules blancs et les plaquettes [56]. Chacune de ces cellules

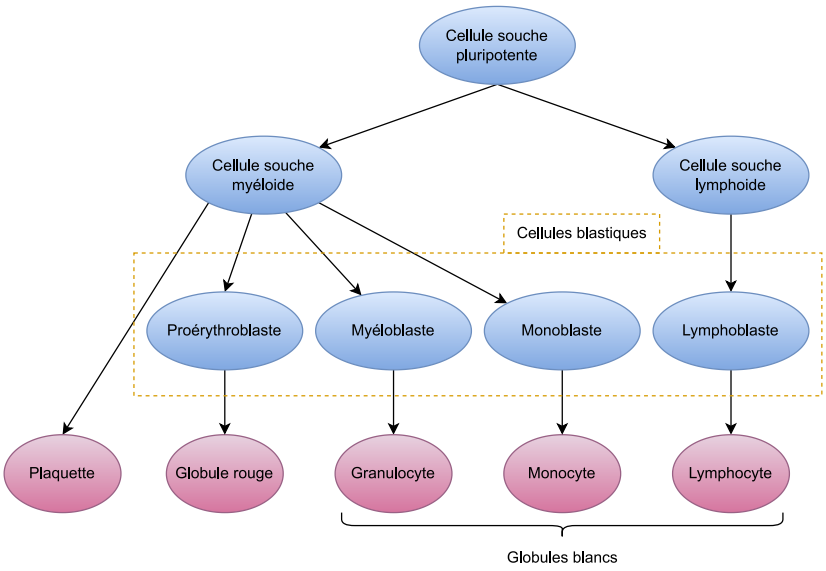

Figure 1.1 – Processus de développement des cellules sanguines matures (hématopoïèse). Chaque cellule sanguine est obtenue suite à la transformation d'une cellule souche de la moelle osseuse. Les ellipses roses et les ellipses bleues représentent respectivement les cellules matures et immatures.

sanguines est dite mature puisqu'elle possède un rôle particulier. Les globules rouges transportent l'oxygène vers les tissus du corps, les globules blancs combattent les infections et les maladies, tandis que les plaquettes s'occupent d'arrêter les saignements entraînés par la blessure d'un vaisseau sanguin [39, 56]. Le processus durant lequel une CSP se transforme en cellule sanguine mature se nomme hématopoïèse [56] (Figure 1.1). Dépendamment de la cellule résultante de ce processus, la CSP de départ devra nécessairement prendre la forme d'une cellule souche myéloide ou lymphoide lors d'une étape intermédiaire. Dans le cas des globules rouges et des globules blancs, une transformation de cellule souche en cellule blastique sera additionnellement requise [56].

Les leucémies aiguës se caractérisent par une surproduction de cellules blastiques (c.-à-d., blastes), en particulier, les lymphoblastes lorsqu'il s'agit de la LAL [39, 68]. Il arrive que ces cellules n'atteignent pas la maturité et se prolifèrent à cause d'une mutation génétique survenue durant leur conception [39, 68]. L'espace et les ressources de la moelle osseuse étant limités, la présence invasive de ces blastes anormaux (c.-à-d., cancéreux) freine le développement des autres cellules sanguines [39, 68]. En raison

du manque de globules blancs, de globule rouges et de plaquettes, un enfant atteint d'une LAL présentera potentiellement des symptômes tels que l'immunodéficience, l'anémie et des troubles de coagulation [39, 68].

1.1.2 Traitement

Le traitement de la LAL infantile vise à éliminer les lymphoblastes cancéreux pour mettre fin à leur multiplication et permettre la génération de nouvelles cellules sanguines en santé. La chimiothérapie constitue la modalité de traitement principale [39, 68]. Celle-ci comporte les phases principales suivantes : l'induction, la consolidation, la maintenance provisoire, l'intensification différée et la maintenance (Figure 1.2). Chacune de ces phases nécessite l'administration régulière de différents médicaments. Le choix de ceux-ci dépend conjointement du groupe de risque associé au patient et de la phase de traitement [68].

Induction Cette phase intensive consiste à éliminer les lymphoblastes cancéreux présents dans la moelle osseuse et le sang jusqu'à l'atteinte du statut de rémission. Selon la *Leukemia & Lymphoma Society* [39], la rémission se caractérise par le respect des cinq points suivants :

— blastes indétectables par microscope dans la moelle osseuse ;
— pourcentage de blastes contenus dans la moelle osseuse $\leq 5\%$;
— absence de blastes dans les vaisseaux sanguins ;
— aucun symptôme de la LAL.

Celle-ci est généralement atteinte après quatre semaines de traitement [39, 68]. N'étant pas différenciées par les médicaments administrés, les cellules sanguines en santé se voient également éliminées lors du processus. Les patients traités durant cette phase

Figure 1.2 – Phases du traitement par chimiothérapie.

sont parfois contraints de rester à l'hôpital considérant qu'ils sont susceptibles de développer des complications de santé [39]. Bien qu'elle soit moins utilisée aujourd'hui en raison de son association à plusieurs effets tardifs [18, 35, 61], la radiothérapie peut également être appliquée à cette étape de traitement chez les patients à plus hauts risques [68]. La radiothérapie consiste à exposer l'enfant atteint de LAL à de fortes radiations pour éliminer les cellules cancéreuses.

Consolidation Après l'atteinte de la rémission, la phase de consolidation vise à éliminer les blastes cancéreux qui n'ont potentiellement pas été détectés à la fin de l'induction [39]. Si ces cellules ne sont pas prises en charge, elles peuvent éventuellement amener la maladie à récidiver. Cette phase dure environ quatre à huit semaines [39].

Maintenance provisoire D'une durée approximative de huit semaines, la phase de maintenance provisoire a pour but de maintenir le statut de rémission du patient tout en permettant à sa moelle osseuse de se rétablir des phases de traitement précédentes [39]. Cet objectif est poursuivi avec l'utilisation de médicament prévenant la croissance des blastes cancéreux résiduels, mais n'éliminant aucune cellule sanguine.

Intensification différée Semblable aux phases d'induction et de consolidation, cette phase se consacre à l'éradication des lymphoblastes cancéreux pouvant être encore présents dans l'organisme [39]. Celle-ci dure habituellement huit semaines.

Maintenance Se distinguant par son utilisation de dosages plus faibles et sa durée allant de deux à trois ans, la phase de maintenance a pour but de prévenir la récidive de la LAL [68].

1.1.3 Post-traitement

Des lignes directrices sont proposées par le *Children's Oncology Group* pour le suivi des survivants de la LAL infantile [25]. Celles-ci concernent principalement la détection, la prise en charge et la prévention des effets tardifs liés au traitement, et sont mises à jour tous les cinq ans pour assurer la mise en place des meilleures pratiques de soins [25]. Ces recommandations, liées aux procédures à suivre lors des

rencontres, se basent sur les caractéristiques des patients pris en charge et considèrent principalement l'historique de traitement de ceux-ci [25]. Une importance est également attribuée à des variables telles que l'âge au diagnostic et le sexe puisqu'elles constituent des facteurs influençant les risques d'apparition d'effets tardifs [25]. Les efforts de recherche continuellement portés envers la personnalisation des procédures post-traitement contribuent à l'avancement de la médecine de précision et l'amélioration de la qualité de vie des survivants. En particulier, la mise en place de bonnes pratiques de suivi permet d'éviter la sur-évaluation des patients à faibles risques et la sous-évaluation des patients à hauts risques. Alors que les évaluations moins pertinentes peuvent entraîner des coûts et des inconforts inutiles, celles qui n'ont pas lieu peuvent, quant à elles, nuirent à la prévention de problèmes de santé chez les survivants [25]. De manière générale, des visites ont lieu tous les mois durant la première année suivant le traitement, tous les trois à six mois durant les quatre années subséquentes et une fois par année ensuite [1].

1.1.4 Effets tardifs

Aujourd'hui, approximativement 90% des enfants atteints de la LAL survivent au traitement [67]. Néanmoins, à l'âge adulte, environ deux tiers des survivants présentent un ou plusieurs effets tardifs liés à celui-ci [60]. Ces différents effets ont été recensés et classés par groupe (Tableau 1.1). Au cours de cette sous-section, nous donnerons des détails additionnels concernant chacun de ces groupes. Entre autres, nous identifierons certains facteurs de prédisposition liés aux différents effets énumérés au sein de ceux-ci.

Syndrome métabolique

Les différents effets répertoriés dans ce groupe constituent les problèmes de santé qui doivent nécessairement être observés chez un individu pour que celui-ci soit diagnostiqué d'un syndrome métabolique. Différents facteurs de prédisposition ont été mis

1. Canadian Cancer Society, «Follow-up after treatment for childhood leukemia», https ://cancer.ca/en/cancer-information/cancer-types/leukemia-childhood/treatment/follow-up, page consulté le 19 octobre 2022

Tableau 1.1 – Effets tardifs principaux associés au traitement de la LAL infantile. Ceux-ci sont classés en fonction des groupes identifiés dans l'étude PETALE [47].

Groupe	Effet tardif
Syndrome métabolique	Dyslipidémie, hypertension, insulino-résistance, obésité [47]
Cardiotoxicités	Cardiomyopathie, dysfonction ventriculaire gauche, insuffisance cardiaque congestive [30]
Morbidités osseuses	Densité minérale osseuse réduite, ostéonécrose [2, 66]
Effets neurocognitifs	Déficits neurocognitifs, changements neuroanatomiques [7, 65]
Qualité de vie	Anxiété, dépression, difficultés scolaires et professionnelles [1, 65]

de l'avant pour ces composantes du syndrome métabolique. Par exemple, l'exposition à la radiothérapie crânienne augmente les risques d'insulino-résistance [35]. C'est également le cas pour la dyslipidémie lorsqu'une faible dose de corticostéroïdes est conjointement administrée [12]. D'autre part, les faits d'être obèse avant le traitement de la LAL et de disposer du sexe masculin à la naissance coïncident respectivement avec une augmentation des risques d'obésité et d'hypertension [35]. Enfin, à l'exception de l'hypertension, plusieurs polymorphismes génétiques ont été identifiés comme facteurs de prédisposition aux effets tardifs de ce groupe [18].

Cardiotoxicités

Les effets identifiés dans de ce groupe réfèrent à des problèmes cardiaques induits par la nature cardiotoxique des anthracyclines, des substances anticancéreuses administrées durant la chimiothérapie. Parmi les anthracyclines, nous retrouvons la doxorubicine, qui est fréquemment utilisée dans le traitement de la LAL infantile. Sa dose cumulative contribue au développement de cardiotoxicités [30]. En outre, le risque d'apparition de troubles cardiaques causés par cet agent cardiotoxique est plus grand lorsque certains polymorphismes génétiques sont présents au sein des gènes NOS3 et ABCC5 [30].

Morbidités osseuses

Comme son nom l'évoque, le groupe des morbidités osseuses englobe l'ensemble des problèmes de santé observés au niveau squelettique. En plus des agents chimiothérapeutiques, plusieurs facteurs indirectement liés au traitement de LAL infantile

peuvent contribuer à l'ostéonécrose, notamment, le manque de vitamine D et la réduction de masse musculaire encourus par les périodes d'isolement [55]. En ce qui a trait à la densité minérale osseuse réduite, celle-ci reste observable à l'âge adulte principalement chez les patients ayant été exposés à la radiothérapie [53]. En effet, les radiations peuvent non seulement affecter directement la densité des os, mais également avoir une incidence sur la production d'hormones de croissances [53]. En supplément, différentes variations génétiques ont été identifiées comme facteurs aggravant les risques d'ostéonécrose et de densité minérale osseuse réduite [2, 66].

Effets neurocognitifs

Les déficits neurocognitifs représentent une grande part des effets neurocognitifs répertoriés. Parmi ceux-ci, nous observons la réduction de la mémoire de travail, de la vitesse de traitement de l'information et de la capacité de concentration [7, 65]. Ces déficits sont principalement liés aux dosages des médicaments anticancéreux administrés et sont généralement plus marqués chez les patients ayant eu recourt à la radiothérapie crânienne [7]. Certains agents neurotoxiques utilisés durant la chimiothérapie (p. ex., methotrexate) sont nuisibles au développement du cerveau étant donné les changements neuroanatomiques qu'ils entraînent au niveau des substances blanches et grises [7]. Notons que l'âge au diagnostic, ainsi que le sexe, constituent des facteurs importants dans le développement des effets neurocognitifs. Ceux-ci témoignent conjointement de la maturité cérébrale des patients durant leurs traitements et conséquemment, de la sensibilité de leurs cerveaux face aux agents neurotoxiques [7]. Additionnellement, certains polymorphismes génétiques peuvent aussi augmenter les risques de développement d'effets neurocognitifs [7].

Qualité de vie

Les éléments associés au groupe suivant réfèrent particulièrement aux effets sociaux et émotionnels suivant le traitement. Ceux-ci se développent généralement par le biais des autres effets tardifs mentionnés jusqu'à présent. Par exemple, l'anxiété et la dépression sont généralement observées chez les survivants éprouvant des difficultés à réaliser des activités quotidiennes en raison de leur état de santé affaibli [1].

Les difficultés scolaires et professionnelles présentes chez les survivants de la LAL sont, quant à eux, aggravées directement par les déficits neurocognitifs [1, 7]. Certains facteurs de risque liés à la détérioration de la qualité de vie restent toutefois indépendants des effets tardifs des autres groupes. Précisément, certains polymorphismes génétiques peuvent augmenter les risques de développement de l'anxiété et de la dépression [7, 65].

1.2 Étude PETALE

Entre 2013 et 2016, une étude importante fut mise en place et réalisée par une équipe interdisciplinaire du centre hospitalier universitaire de Sainte-Justine (CHUSJ). Celle-ci, portait le nom de PETALE puisqu'elle contribuait à la **P**révention des **E**ffets tardifs liés au **T**raitement de la leucémie **A**iguë **L**ymphoblastique chez l'**E**nfant. Le plan de cette étude était d'abord d'identifier des biomarqueurs permettant de prédire l'apparition d'effets tardifs pour ensuite mettre sur pieds de meilleures directives de suivi post-traitement [47]. Nous débuterons cette sous-section en dressant un portrait général de l'étude, puis discuterons finalement de certains résultats liés à celle-ci.

1.2.1 Description de l'étude

L'étude PETALE était constituée de deux phases (Figure 1.3). Lors de la première, 250 survivants de la LAL infantile ont accepté de procéder à une série de 32 tests se déroulant au sein d'une seule journée, et visant principalement à mesurer différents marqueurs cliniques, génétiques et biochimiques [47, 48]. Parmi ces tests, nous pouvons noter quelques exemples tels que le séquençage du génome entier (SGE), la réalisation d'une échographie cardiaque et la participation à des questionnaires concernant la qualité de vie. Pour être éligible à cette première phase, plusieurs critères devaient être respectés par les patients. En particulier, ceux-ci devaient être d'origine caucasienne, avoir été traités au CHUSJ selon des protocoles spécifiques et avoir reçu un diagnostic entre 1987 et 2010 avant l'âge de 19 ans [47]. Notamment, une période d'au moins cinq ans devait avoir été écoulée depuis leurs diagnostics [47].

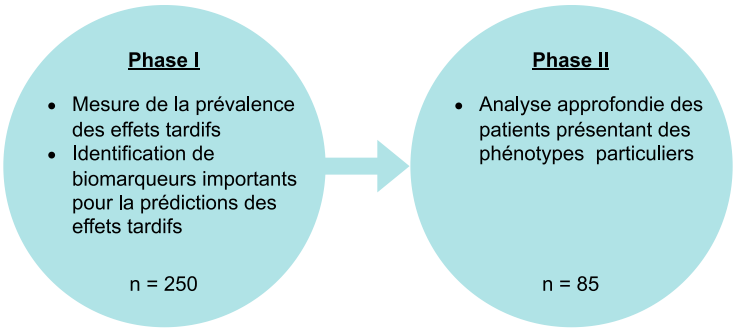

Figure 1.3 – Phases de l'étude PETALE

La seconde phase, quant à elle, comprenait les patients ayant démontrés un phénotype exceptionnel pour au moins l'un des groupes d'effets tardifs (c.à-d., syndrome métabolique, cardiotoxicités, morbidités osseuses, effets cognitifs et qualité de vie), lors de la première phase [47]. Ces particularités pouvaient être soit positives ou négatives du point de vue de la santé des participants. Par exemple, pour la catégorie du syndrome métabolique, seulement les patients disposant de tous les traits, ou bien du plus petit nombre de traits observés dans la cohorte, étaient considérés comme des participants disposant d'un phénotype particulier [48]. Au cours de cette phase, les survivants de la LAL infantile devaient participer à des tests additionnels n'ayant pas eu lieu dans la phase I [48].

1.2.2 Discussion des résultats

Plusieurs travaux de recherche furent publiés suite à l'acquisition des données sur la cohorte PETALE. Entre autres, plusieurs analyses statistiques ont permis l'identification de biomarqueurs importants pour la prédiction de certains effets tardifs du traitement de la LAL infantile. Les principaux ont d'ailleurs été mentionnés dans la sous-section précédente (Section 1.1.4).

Tandis que certains travaux ont simplement proposé de poser une attention sur des marqueurs spécifiques [2, 7, 12, 18, 30, 35, 65, 66], d'autres les ont plutôt utilisés pour avancer des actions concrètes à entreprendre avant ou après la fin du traitement. En particulier, plusieurs articles ont fait la promotion de l'activité physique pour la prévention de diverses morbidités associées au traitement [36, 38]. L'adoption d'un

mode de vie actif contribue au maintient de la forme cardio-respiratoire, qui à son tour, se révèle déterminante dans la prévention d'effets tardifs tels que l'obésité, le cholestérol et la dépression [36]. En complément à ces différents résultats, Labonté *et al.* [37] ont mis sur pieds un modèle d'évaluation de la forme cardio-respiratoire par estimation de la consommation maximale d'oxygène (c.-à-d., VO_2 max). Celui-ci, reposant sur des variables mesurables via un test de marche de six minutes, constitue une alternative plus accessible et moins coûteuse au test d'effort maximal cardio-respiratoire normalement employé pour la mesure du VO_2 max.

Néanmoins, à l'exception du travail de Labonté *et al.* [37], aucune étude n'a exploité le potentiel de l'apprentissage automatique pour développer des modèles prédictifs dans le contexte de prévention des effets tardifs de la LAL infantile. Alors que les analyses statistiques réalisées ont mené à la découverte de tendances importantes dans la population des survivants, l'apprentissage automatique constitue une voie de recherche prometteuse dans le développement d'outils de suivi hautement personnalisés.

Chapitre 2

Apprentissage supervisé

L'apprentissage automatique est un domaine de l'intelligence artificielle qui s'intéresse au développement d'algorithmes ayant la capacité de s'améliorer automatiquement dans la réalisation d'une tâche [54]. En particulier, la branche de l'apprentissage supervisé s'intéresse à la création de fonctions apprenant à associer une cible spécifique à un ensemble de valeurs réelles en se basant sur un jeu de données annotées. Nous débuterons ce chapitre en étudiant les concepts fondamentaux liés à l'apprentissage supervisé, puis étudierons individuellement les caractéristiques de plusieurs solutions utilisées pour développer des modèles de prédiction au sein de contextes où les données disponibles prennent une forme tabulaire.

2.1 Concepts fondamentaux

Dans cette section, nous introduirons des concepts fondamentaux associés à l'apprentissage supervisé. Nous définirons d'abord mathématiquement l'apprentissage supervisé sous forme de problème de minimisation de risque empirique. Nous survolerons ensuite certains éléments dont la connaissance est essentielle pour la mise en place d'un cadre expérimental rigoureux dédié à la conception de modèles prédictifs.

2.1.1 Minimisation de risque empirique

Les données annotées constituent un élément distinctif de l'apprentissage supervisé. Celles-ci présentent deux composantes, soit un ensemble de n observations et leurs cibles respectives. Plus précisément, chacune de ces observations, notée $\mathbf{x}_i$, constitue un élément de l'espace des matrices à coefficients réels comprenant m lignes et une colonne (c.-à-d., $\mathbf{x}_i \in \mathcal{M}_{m,1}(\mathbb{R})\ \forall i \in \{1, \ldots, n\}$) et est associée à une cible $t_i \in \mathbb{R}$. Par exemple, dans un contexte médical, nous pourrions imaginer que nous disposons du poids, de la taille et de l'âge (c.-à-d., les attributs) de plusieurs patients, en plus d'avoir la pression artérielle qui a été mesurée pour chacun de ceux-ci (c.-à-d., la cible). Dans ce cas spécifique nous aurions $m = 3$.

De ces données nous émettons l'hypothèse que l'ensemble des cibles t_i sont issues d'une fonction $y : \mathcal{M}_{m,1}(\mathbb{R}) \to \mathbb{R}$ à laquelle un bruit gaussien de moyenne nulle a été ajouté, c'est-à-dire que $t_i = y(\mathbf{x}_i) + \epsilon_i, \forall_i$ où $\epsilon_i \sim \mathcal{N}(0, \sigma)$. Considérant l'exemple mentionné ci-dessus, nous poserions l'hypothèse que la pression artérielle dépend potentiellement du poids, de l'âge et de la taille ($\mathbf{x}_i$), mais également de facteurs externes inconnus que nous décidons de caractériser par un effet ϵ_i.

Notre but est donc de trouver une approximation de y à l'aide d'une fonction f muni de paramètres modifiables $\boldsymbol{\omega} \in \mathcal{W}$ et d'hyperparamètres fixes $\boldsymbol{\gamma} \in \Gamma$, où $\mathcal{W}$ et Γ représentent des domaines arbitraires. Cette approximation constituera un modèle d'apprentissage supervisé $f_{\boldsymbol{\omega},\boldsymbol{\gamma}} : \mathcal{M}_{m,1}(\mathbb{R}) \to \mathbb{R}$ dont les définitions de $\mathcal{W}$ et Γ dépendront de l'architecture sur lequel celui-ci reposera.

Pour ce faire, il nous sera nécessaire de définir a priori une fonction de perte $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ permettant de quantifier l'inexactitude de la prédiction de notre modèle pour n'importe quelle observation annotée donnée, c'est-à-dire $\ell(f_{\boldsymbol{\omega},\boldsymbol{\gamma}}(\mathbf{x}_i), t_i)$, afin d'être en mesure de trouver de nouveaux paramètres permettant d'obtenir une perte réduite. Notons par exemple l'erreur quadratique comme fonction de perte dans un cadre de régression :

$$\ell(f_{\boldsymbol{\omega},\boldsymbol{\gamma}}(\mathbf{x}_i), t_i) = (f_{\boldsymbol{\omega},\boldsymbol{\gamma}}(\mathbf{x}_i) - t_i)^2. \tag{2.1}$$

Ainsi, le problème de l'apprentissage supervisé se traduit par la recherche des paramètres $\boldsymbol{\omega}$ minimisant la perte moyenne sur l'ensemble des données annotées pour

un ensemble d'hyperparamètres définis $\boldsymbol{\gamma}$, c'est-à-dire la résolution du problème de minimisation de risque empirique suivant :

$$\arg\min_{\boldsymbol{\omega}\in\mathcal{W}} \sum_{i=1}^{n} \frac{1}{n} \ell(f_{\boldsymbol{\omega},\boldsymbol{\gamma}}(\mathbf{x}_i), t_i), \tag{2.2}$$

La méthode utilisée pour résoudre le problème présenté à l'équation (2.2) dépendra du modèle $f_{\boldsymbol{\omega},\boldsymbol{\gamma}}$ choisi. Dans certaines circonstances, par exemple lors de l'utilisation d'un modèle de régression linéaire avec la fonction de perte quadratique établie à l'équation (2.1), nous serons en mesure de calculer la solution analytique $\boldsymbol{\omega}^*$ telle que

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}\in\mathcal{W}} \sum_{i=1}^{n} \frac{1}{n} \ell(f_{\boldsymbol{\omega},\boldsymbol{\gamma}}(\mathbf{x}_i), t_i).$$

Dans les cas où il ne sera pas possible de calculer la solution analytique, nous opterons plutôt pour l'emploi d'algorithmes itératifs générant des solutions tentant de se rapprocher de $\boldsymbol{\omega}^*$ au fil des itérations. La solution finale qui sera récupérée de ce type d'algorithme constituera alors une estimation $\hat{\boldsymbol{\omega}}^*$ de $\boldsymbol{\omega}^*$. Une méthode itérative portant le nom de descente de gradient stochastique sera abordée plus en détail dans la section 2.3 portant sur les réseaux neuronaux.

2.1.2 Entraînement et test

Dans le contexte pratique de l'apprentissage supervisé, il n'est pas souhaitable d'optimiser les paramètres d'un modèle en considérant l'ensemble des données annotées qui sont à notre disposition. Il est plutôt de mise d'adopter un cadre expérimental permettant de valider que les paramètres trouvés pour notre modèle permettront à celui-ci d'émettre de bonnes prédictions sur de futurs ensembles de données dont les cibles sont inconnues.

Pour ce faire, les données annotées seront minimalement divisées en deux sous-ensembles distincts $\mathcal{X}_{train}$ et $\mathcal{X}_{test}$, tels que $\mathcal{X}_{train} \cap \mathcal{X}_{test} = \emptyset$. L'ensemble $\mathcal{X}_{train}$ contiendra les observations utilisées pour obtenir les paramètres du modèle $f_{\boldsymbol{\omega},\boldsymbol{\gamma}}$ en résolvant un problème d'optimisation équivalent à celui présenté à l'équation (2.2), c'est-à-dire

$$\arg\min_{\boldsymbol{\omega}\in\mathcal{W}} L(\boldsymbol{\omega}, \mathcal{X}_{train}), \tag{2.3}$$

où $L(\cdot)$ est une fonction définie par

$$L(\boldsymbol{\omega}, \mathcal{X}) = \sum_{i:\mathbf{x}_i \in \mathcal{X}} \frac{1}{|\mathcal{X}|} \ell(f_{\boldsymbol{\omega},\gamma}(\mathbf{x}_i), t_i), \tag{2.4}$$

et telle que $|\mathcal{X}|$ réfère aux nombre d'éléments contenus dans l'ensemble $\mathcal{X}$, soit sa cardinalité. Les données de l'ensemble $\mathcal{X}_{test}$ permettront quant à elles d'évaluer la performance du modèle suite à son optimisation. La phase d'optimisation des paramètres $\boldsymbol{\omega}$ présentée à l'équation (2.3) portera le nom d'entraînement, tandis que la phase d'évaluation sur l'ensemble $\mathcal{X}_{test}$ portera le nom de test. Cette stratégie permet entre autres de détecter si un modèle entraîné présente un problème de sur-apprentissage ou sous-apprentissage.

Sur-apprentissage La capacité d'un modèle $f_{\boldsymbol{\omega},\gamma}$ à approximer la relation entre un ensemble d'attributs et une cible, c'est-à-dire à apprendre, dépend de plusieurs facteurs dont le nombre de paramètres associés à son architecture ainsi que le choix de ses hyperparamètres. Par exemple, un réseau de neurones profond est en mesure de reconnaître des relations très complexes entre certains attributs ainsi que leurs impacts sur la cible à prédire, contrairement à une régression linéaire qui ne peut qu'estimer la cible d'un individu par une combinaison linéaire de ses attributs. Toutefois, avoir une plus grande capacité d'apprentissage vient au détriment d'un risque accru de sur-apprentissage (Figure 2.1). Celui-ci se caractérise par une très petite perte sur l'ensemble d'entraînement et une grande perte sur l'ensemble de test tel que

$$L(\hat{\boldsymbol{\omega}}^*, \mathcal{X}_{train}) \ll L(\hat{\boldsymbol{\omega}}^*, \mathcal{X}_{test})$$

et où $\hat{\boldsymbol{\omega}}^*$ est la solution obtenue en résolvant l'équation (2.3). Ces circonstances laissent sous-entendre que notre modèle a appris comment prédire les cibles des observations se trouvant dans $\mathcal{X}_{train}$, mais n'arrive pas à approximer une relation généralisable à de nouvelles données, celles dans $\mathcal{X}_{test}$. Ce problème est davantage susceptible de se produire lorsque nous disposons d'un petit ensemble de données d'entraînement $\mathcal{X}_{train}$, puisque les observations qui y sont contenues ne représentent potentiellement qu'une faible partie des combinaisons d'attributs et de cibles possibles.

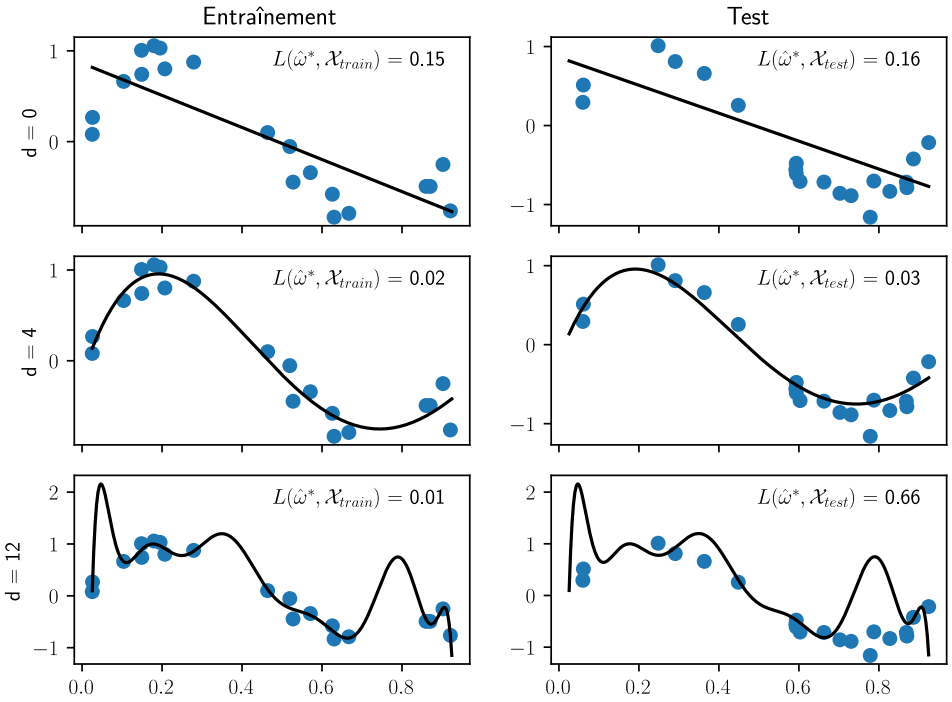

Figure 2.1 – Illustration des phénomènes de sous-apprentissage et de sur-apprentissage. Les figures présentent trois modèles de régression poylnomial de degrés d différents. Ceux-ci ont été entraînés et évalués sur les mêmes ensembles de données d'entraînement et de test respectivement. La ligne du haut illustre le phénomène de sous-apprentissage, la ligne du bas illustre le phénomène de sur-apprentissage et la ligne du centre présente un contexte idéal où le modèle propose à la fois une bonne performance en entraînement et en test. Notons que la fonction $L(\cdot)$ correspond à celle de l'équation (2.4), tandis que $\hat{\boldsymbol{\omega}}^*$ est propre à chaque modèle et correspond aux paramètres obtenus en résolvant l'équation (2.3).

Sous-apprentissage Il est possible qu'un modèle entraîné présente à la fois une grande perte sur l'ensemble d'entraînement et sur l'ensemble de test. Ce scénario particulier montre que notre modèle n'a pas été en mesure d'approximer adéquatement une relation entre les attributs et les cibles provenant de nos données annotées. Autrement dit, celui-ci est en situation de sous-apprentissage (Figure 2.1). Cette situation peut survenir lorsque la capacité d'apprentissage de notre modèle est restreinte par son architecture ou les attributs disponibles dans notre ensemble de données ne permettent pas de prédire les cibles.

2.1.3 Validation croisée

Les caractéristiques propres à un modèle et les attributs disponibles dans un ensemble de données annotées ne sont pas les seuls facteurs pouvant influencer la perte observée sur un ensemble de test quelconque. Les éléments constituant les ensembles $\mathcal{X}_{train}$ et $\mathcal{X}_{test}$ peuvent également avoir une incidence sur celle-ci. Ainsi, la méthode de validation croisée a pour objectif d'estimer l'erreur de prédiction espérée sur un futur ensemble de données externes [28] en évaluant un modèle à l'aide de plusieurs paires d'ensembles d'entraînement et de test. Pour ce faire, l'ensemble de données annotées est d'abord divisé à K reprises ($K \in \mathbb{N}/\{0,1\}$) de façon à obtenir les paires

$$(\mathcal{X}_{train}^{(1)}, \mathcal{X}_{test}^{(1)}), (\mathcal{X}_{train}^{(2)}, \mathcal{X}_{test}^{(2)}), \ldots, (\mathcal{X}_{train}^{(K)}, \mathcal{X}_{test}^{(K)}).$$

Ensuite, l'estimation de la perte de prédiction espérée d'un modèle $f_{\boldsymbol{\omega},\gamma}$ peut être déterminée à partir de la perte moyenne enregistrée sur les différents ensembles de test, c'est-à-dire la valeur

$$\sum_{j=1}^{K} \frac{1}{K} \mathcal{L}(f_{\boldsymbol{\omega},\gamma}, \mathcal{X}_{train}^{(j)}, \mathcal{X}_{test}^{(j)}) \tag{2.5}$$

telle que

$$\mathcal{L}(f_{\boldsymbol{\omega},\gamma}, \mathcal{X}_{train}^{(j)}, \mathcal{X}_{test}^{(j)}) = L(\hat{\boldsymbol{\omega}}^*, \mathcal{X}_{test}^{(j)}),$$

et où $\hat{\boldsymbol{\omega}}^*$ est la solution obtenue en résolvant l'équation (2.3) à l'aide de $\mathcal{X}_{train}^{(j)}$. Plusieurs méthodes peuvent être utilisées pour créer les différentes paires d'ensembles

d'entraînement et de test. Dans la suite de cette section, nous regarderons plus en détail l'échantillonnage stratifié aléatoire.

Échantillonnage stratifié aléatoire L'échantillonnage stratifiée aléatoire vise à créer n'importe quel ensemble $\mathcal{X}_{test}^{(j)}$ de façon à ce que les cibles qui y sont contenues soient représentées de façon proportionnelle à ce qui était observé dans l'ensemble de données annotées complet. Cette stratégie est préférable lorsque les cibles sont des entiers associés à un problème de classification, mais peut également être adaptable dans un contexte de régression en discrétisant les cibles réelles. Ainsi, pour une paire $(\mathcal{X}_{train}^{(j)}, \mathcal{X}_{test}^{(j)})$ quelconque, nous choisissons les observations à incorporer dans l'ensemble de test de manière aléatoire sans remplacement jusqu'à l'obtention du nombre d'éléments souhaités pour chaque classe. Une fois l'ensemble de test complété, les éléments restant forment alors $\mathcal{X}_{train}^{(j)}$.

2.2 Modèles à structure arborescente

Nous catégorisons dans la famille des modèles à structure arborescente, l'ensemble des méthodes d'apprentissage supervisé faisant usage d'arbres de décisions pour arriver à déterminer la cible d'une observation à partir de ses attributs. Dans cette section, nous définirons d'emblée ce qu'est un arbre de décision et détaillerons le processus utilisé pour son entraînement. Par la suite, nous expliquerons le fonctionnement de deux modèles à structure arborescente fréquemment utilisés.

2.2.1 Arbre de décision

Un arbre de décision est un modèle d'apprentissage simple divisant l'espace des attributs en plusieurs régions et associant chacune de celles-ci à une cible particulière. Dans un contexte de régression, les cibles sont des valeurs continues (c.-à-d., $t_i \in \mathbb{R}\ \forall i$), tandis qu'en classification, elles prennent des valeurs entières représentant des classes (c.-à-d., $t_i \in \mathbb{N}\ \forall i$). L'entraînement de ce modèle (c.-à-d., la création des différentes régions) se fait par la division dichotomique successive de l'ensemble des données d'entraînement, en ne considérant qu'un attribut à la fois. Ce processus s'illustre

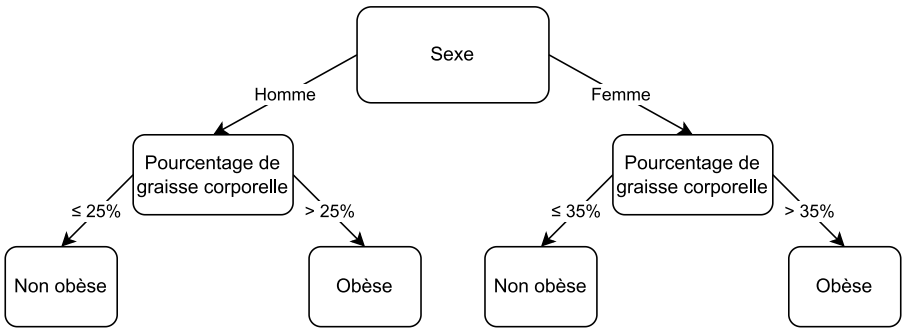

Figure 2.2 – Exemple d'arbre de décision permettant la classification d'un individu comme obèse ou non obèse en fonction de son sexe et de son pourcentage de graisse corporelle.

par un arbre binaire (Figure 2.2) dans lequel chaque noeud représente un attribut et chaque branche représente un critère de séparation subdivisant davantage l'ensemble $\mathcal{X}_{train}$. Les zones délimitées par les derniers critères de séparation se trouvant au bas de l'arbre représentent les différentes régions apprises par le modèle et constituent les feuilles de l'arbre. En cas de classification, la valeur attribuée à chaque feuille correspond à la catégorie majoritaire observée dans le sous-ensemble d'observations d'entraînement qui lui est associé. Autrement, la valeur attribuée à une feuille se calcule par la moyenne des cibles affiliées.
La séquence dans laquelle chaque attribut est considéré, ainsi que le critère de séparation qui lui est appliqué, sont déterminés par un algorithme glouton, tiré de l'anglais *greedy*, qui divise l'ensemble de données itérativement de façon à maximiser une mesure de qualité de séparation $q(\cdot)$ à l'état immédiat. Pour un arbre de décision, nous pouvons définir conceptuellement le domaine des paramètres (c.-à-d., $\mathcal{W}$), comme un ensemble contenant toutes les séquences d'attributs possibles, ainsi que les différentes valeurs que ceux-ci présentent dans $\mathcal{X}_{train}$. Nous considérons les différentes modalités des attributs comme faisant partie de $\mathcal{W}$ puisqu'elles constituent les choix de critères de séparation disponibles. Pour ce qui est des hyperparamètres constituant $\boldsymbol{\gamma}$, nous pouvons considérer des éléments tels que le choix de mesure $q(\cdot)$ et le nombre de feuilles maximal. Des détails concernant l'obtention d'un nombre de feuilles maximal seront discutés à la section 2.2.2.

Pour clarifier le processus d'entraînement d'un arbre de décision, considérons un contexte arbitraire où les attributs disponibles au sein d'un ensemble d'entraînement,

ainsi que les cibles, présentent uniquement des valeurs discrètes. Plus formellement, supposons que $\mathcal{X} \subseteq \mathcal{X}_{train}$ est un sous-ensemble d'observations de l'ensemble d'entraînement que nous souhaitons diviser avec l'ajout d'un noeud, que $\mathcal{T} = \{\tau_1, \ldots, \tau_c\}$ représente l'ensemble des différentes valeurs de cibles (c.-à-d., catégories) possibles, que $\Theta = \{\theta_1, \ldots, \theta_m\}$ constitue l'ensemble des différents attributs disponibles et que $m(\theta_i) = \{\vartheta_{ij}\}_{j=1}^{m_i}$ soit l'ensemble de modalités associées à chaque variable θ_i. La sélection d'un attribut et d'un critère de séparation optimal se traduit par la résolution du problème

$$\underset{\theta_i \in \Theta \ , \ \vartheta_{i,j} \in m(\theta_i)}{\arg\max} q(\mathcal{X}, \theta_i, \vartheta_{ij}).$$

Il existe plusieurs mesures $q(\cdot)$ pour quantifier la qualité de séparation d'un ensemble avec cibles discrètes, notons par exemple le gain d'information (GI) :

$$\begin{aligned} \mathrm{GI}\left(\mathcal{X}, \theta_i, \vartheta_{ij}\right) &= H(\mathcal{X}) - H(\mathcal{X}|\theta_i = \vartheta_{ij}) \\ &= H(\mathcal{X}) - \left(\frac{|\mathcal{X}_{\theta_i = \vartheta_{ij}}|}{|\mathcal{X}|} H(\mathcal{X}_{\theta_i = \vartheta_{ij}}) + \frac{|\mathcal{X}_{\theta_i \neq \vartheta_{ij}}|}{|\mathcal{X}|} H(\mathcal{X}_{\theta_i \neq \vartheta_{ij}}) \right), \end{aligned} \tag{2.6}$$

où $\mathcal{X}_{\theta_i = \vartheta_{ij}}$ est le sous-ensemble de $\mathcal{X}$ formé à partir des observations disposant de la modalité ϑ_{ij} pour l'attribut θ_i, $\mathcal{X}_{\theta_i \neq \vartheta_{ij}} = \mathcal{X} / \mathcal{X}_{\theta_i = \vartheta_{ij}}$ et $H(\cdot)$ est une fonction mesurant l'entropie d'un ensemble par rapport à ses cibles, c'est-à-dire

$$H(\mathcal{X}) = -\sum_{k=1}^{c} p_k \log\left(p_k\right),$$

avec $p_k = \frac{|\mathcal{X}_{t=\tau_k}|}{|\mathcal{X}|}$ et $\mathcal{X}_{t=\tau_k} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{X}, t_i = \tau_k\}$. L'expansion d'une branche de l'arbre prend fin lorsque chaque sous-ensemble d'observations créé suite à l'ajout d'un noeud, et d'un critère de séparation, dispose d'une entropie nulle.

Il est possible d'adapter l'entraînement à des ensembles de données comprenant des cibles continues, en utilisant une mesure $q(\cdot)$ en conséquence. Par exemple, la mesure

$$q(\mathcal{X}, \theta_i, \vartheta_{ij}) = -\left(\frac{|\mathcal{X}_{\theta_i = \vartheta_{ij}}|}{|\mathcal{X}|} \mathrm{Var}(\mathcal{X}_{\theta_i = \vartheta_{ij}}) + \frac{|\mathcal{X}_{\theta_i \neq \vartheta_{ij}}|}{|\mathcal{X}|} \mathrm{Var}(\mathcal{X}_{\theta_i \neq \vartheta_{ij}}) \right),$$

où $\text{Var}(\cdot)$ est une fonction mesurant la variance d'un ensemble par rapport à ses cibles, se définissant par

$$\text{Var}\left(\mathcal{X}\right) = \frac{1}{|\mathcal{X}|} \sum_{i|\mathbf{x}_i \in \mathcal{X}} \left(t_i - \bar{t}\right)^2,$$

avec $\bar{t} = \frac{1}{|\mathcal{X}|} \sum_{i|\mathbf{x}_i \in \mathcal{X}} t_i$. Notons que l'adaptation à des jeux de données comprenant des attributs continus est également réalisable. Comme illustrée à la Figure 2.2, la séparation d'un ensemble à partir d'un attribut continu se caractérise par la sélection d'un seuil. Soit b_i un attribut continu et $m(b_i) = \{\beta_{ij}\}_{j=1}^{m_i}$ l'ensemble ordonné des différentes valeurs prises par b_i dans $\mathcal{X}$, un sous-ensemble d'observations de $\mathcal{X}_{train}$, nous pouvons déterminer la valeur du meilleur seuil à appliquer sur cet attribut en évaluant la qualité de séparation amenée par chacune des valeurs β_{ij}. Par exemple, nous pouvons traduire le gain d'information de l'équation (2.6) par

$$\text{GI}(\mathcal{X}, b_i, \beta_{ij}) = H(\mathcal{X}) - \left(\frac{|\mathcal{X}_{b_i \leq \beta_{ij}}|}{|\mathcal{X}|} H(\mathcal{X}_{b_i \leq \beta_{ij}}) + \frac{|\mathcal{X}_{b_i > \beta_{ij}}|}{|\mathcal{X}|} H(\mathcal{X}_{b_i > \beta_{ij}}) \right),$$

où $\mathcal{X}_{\theta_i \leq \vartheta_{ij}}$ est le sous-ensemble de $\mathcal{X}$ formé à partir des observations pour lesquelles l'attribut b_i est inférieur ou égale à β_{ij} et $\mathcal{X}_{b_i > \beta_{ij}} = \mathcal{X} / \mathcal{X}_{b_i \leq \beta_{ij}}$.

2.2.2 Forêt aléatoire

Les arbres de décision constituent des modèles simples à entraîner et visualiser. Malgré ces aspects positifs, ceux-ci sont toutefois sensibles au sur-apprentissage. Deux causes principales sont à l'origine de ce problème. Premièrement, un arbre de décision peut faire du sur-apprentissage si aucune contrainte n'est préalablement fixé par rapport au nombre de feuilles de celui-ci. En effet, la routine d'entraînement expliquée précédemment permet à un arbre de créer un grand nombre de noeuds et possiblement obtenir une feuille pour chaque observation de l'ensemble d'entraînement. Cette situation n'est pas souhaitable puisque la structure apprise par l'arbre est alors très spécifique aux données d'entraînement et ne reflète pas nécessairement un processus de décision adéquat pour d'autres données. Afin de limiter l'occurrence de ce problème, une opération d'élagage peut être effectuée sur un arbre suite à l'atteinte

d'un nombre pré-déterminé de noeuds [28]. Deuxièmement, un arbre de décision peut également faire du sur-apprentissage par la sélection de critères de séparation caractéristiques aux données utilisées lors de l'entraînement. Afin d'être plus robuste à ce genre de situation, une stratégie consiste à utiliser une forêt aléatoire.

Une forêt aléatoire consiste en un regroupement d'arbres de décision. Ce modèle repose sur les prédictions individuelles de chacun des arbres de la forêt. Dans un contexte de classification, la classe finale prédite consiste en un vote majoritaire des arbres, tandis qu'en cas de régression, le modèle prend plutôt la moyenne des cibles prédites par chacun de ceux-ci. La performance de ce modèle dépend de la corrélation entre les arbres de la forêt [9]. Précisément, plus celle-ci est faible, plus le modèle est susceptible d'avoir de meilleures performances. Afin de réduire la corrélation, chaque arbre de la forêt est entraîné indépendamment sur un regroupement de données échantillonnées avec remise à partir de l'ensemble d'entraînement. Notamment, chaque arbre est entraîné à partir d'un sous-ensemble aléatoire des attributs disponibles.

2.2.3 Arbres de décision avec boosting de gradient

Bien différent des forêts aléatoires, le modèle d'arbres de décision avec boosting de gradient (ADBG) entraîne plutôt séquentiellement plusieurs arbres de décision à faible capacité de façon à ce que chacun de ceux-ci comble les lacunes de l'arbre précédent. La prédiction d'un tel modèle repose sur une somme pondérée des prédictions intermédiaires de chaque arbre de la séquence. Ainsi, en cas de classification, chaque arbre estime les probabilités d'appartenance aux classes possibles plutôt que de donner directement une classe en valeur de sortie. Ces probabilités sont déterminées au sein de chacune des feuilles, lors de l'entraînement, en y calculant le pourcentage d'individus associés à chacune des classes. Notons que cette approche probabiliste peut être également adoptée pour les modèles à structure arborescente sans boosting. Par exemple, pour les forêts aléatoires, les probabilités d'appartenance aux différentes classes peuvent être déterminées, pour un individu donné, en prenant la moyenne des probabilités prédites pour chacune des classes sur l'ensemble des arbres.

Un arbre possède une faible capacité d'apprentissage lorsqu'il est en mesure d'ob-

tenir une structure qui n'offre qu'une division grossière de l'espace des attributs (c.-à-d., dispose de peu de feuilles). C'est-à-dire que celui-ci classifie chaque observation avec seulement un peu plus de succès qu'une décision aléatoire, ou estime les cibles réelles avec une valeur un peu plus juste que la moyenne observée sur l'ensemble d'entraînement.

Pour entraîner un ADBG, une fonction de perte ℓ et un nombre d'arbres doivent préalablement être choisis. Ces valeurs constituent des hyperparamètres du modèle. Le processus d'entraînement d'un ADBG se décrit par la réalisation itérative de cinq étapes jusqu'à l'obtention du nombre d'arbres souhaité. Ces étapes sont les suivantes :

1. Calcul des prédictions du modèle actuel ;
2. Calcul des résidus (c.-à-d. gradients négatifs) ;
3. Entraînement d'un modèle additionnel sur les résidus ;
4. Calcul du poids attribué au modèle additionnel ;
5. Mise à jour du modèle actuel.

Pour mieux expliquer ces étapes, regardons les deux premières itérations du processus dans un contexte où les cibles à prédire prennent des valeurs réelles (c.à-d., contexte de régression) et où la fonction de perte ℓ adoptée correspond à l'erreur quadratique introduite à l'équation (2.1). Dans la première itération, puisqu'aucun modèle n'est encore en place, nous estimons la cible t_i de chaque observation $\mathbf{x}_i$ de l'ensemble d'entraînement par la moyenne μ observée dans l'ensemble lui-même (c.-à-d. $f_{\boldsymbol{\omega},\gamma}(\mathbf{x}_i) = \mu \ \forall i$). À partir de ces estimations, nous sommes en mesure de calculer les résidus du modèle actuel, c'est-à-dire les valeurs $-\partial\ell/\partial f_{\boldsymbol{\omega},\gamma}(\mathbf{x}_i)$, qui dans ce cas correspondent à $t_i - \mu \ \forall i$ [28]. Nous pouvons ensuite entraîner un premier arbre $f^{(1)}$ à prédire ces résidus puis déterminer le poids $\beta^{(1)} \in \mathbb{R}$ à lui attribué en résolvant le problème d'optimisation suivant [21] :

$$\beta^{(1)} = \arg\min_{\beta \in \mathbb{R}} \sum_{i:\mathbf{x}_i \in \mathcal{X}_{train}} \ell\left(\mu + \beta f^{(1)}(\mathbf{x}_i), t_i\right).$$

Nous sommes finalement en mesure de terminer cette itération en mettant à jour le modèle actuel :

$$f_{\omega,\gamma}(\mathbf{x}_i) = \mu + \beta^{(1)} f^{(1)}(\mathbf{x}_i).$$

Dans la seconde itération, nous effectuons les mêmes étapes, mais prenons plutôt la valeur $\mu + \beta^{(1)} f^{(1)}(\mathbf{x}_i)$ pour la prédiction de chaque cible. Le modèle obtenu à la suite de cette itération se représente par

$$f_{\omega,\gamma}(\mathbf{x}_i) = \mu + \beta^{(1)} f^{(1)}(\mathbf{x}_i) + \beta^{(2)} f^{(2)}(\mathbf{x}_i).$$

Ainsi, de manière générale, un ADBG à n arbres se définie par

$$f_{\omega,\gamma}(\mathbf{x}_i) = \mu + \sum_{k=1}^{n} \beta^{(k)} f^{(k)}(\mathbf{x}_i).$$

2.3 Réseaux neuronaux

Les réseaux de neurones constituent des architectures ayant la capacité d'approximer n'importe quelle fonction continue par l'application consécutive de transformations linéaires et non linéaires aux valeurs réelles données en entrée [27]. Dans cette section, nous définirons d'abord mathématiquement le concept de neurone artificiel, un élément clé dans la construction des réseaux neuronaux. Nous survolerons ensuite les propriétés du Perceptron multi-couches, un type de réseaux de neurones dont les fondements sont à la base d'autres types d'architectures courantes. Finalement, nous introduirons l'algorithme de descente de gradient ainsi que le concept de rétropropagation, qui sont des éléments essentiels à l'entraînement des réseaux de neurones.

2.3.1 Neurone artificiel

Un neurone artificiel est défini mathématiquement par la fonction

$$n(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + b)$$

où $\mathbf{w}, \mathbf{x} \in \mathcal{M}_{m,1}(\mathbb{R})$ sont les poids et les entrées du neurone respectivement, $b \in \mathbb{R}$ est un biais et $\sigma(\cdot)$ est une fonction d'activation non linéaire. Une pratique courante

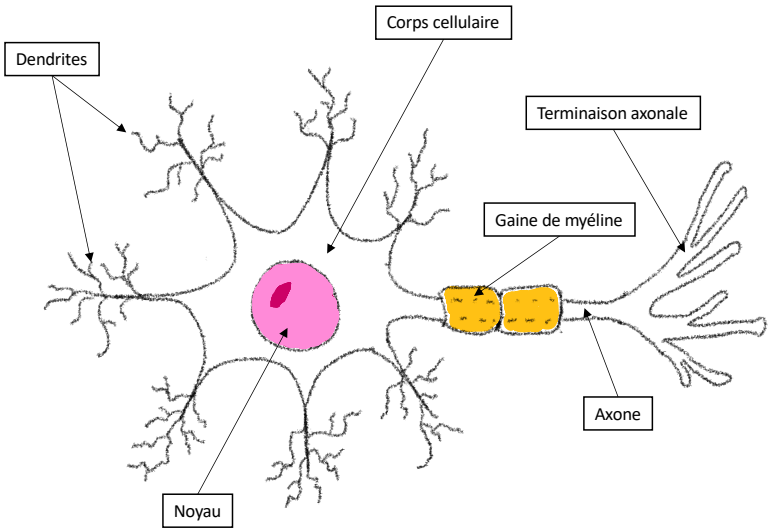

Figure 2.3 – Schéma d'un neurone biologique.

consiste à utiliser la fonction d'activation ReLU$(\cdot)$ (*Rectified Linear Unit*) [26] se définissant par

$$\text{ReLU}(a) = \max(a, 0). \tag{2.7}$$

Le fonctionnement d'un neurone artificiel avec ReLU$(\cdot)$ est conceptuellement comparable à celui d'un neurone biologique (Figure 2.3) acheminant des influx nerveux ($\mathbf{x}$) à son corps cellulaire par l'intermédiaire de ses dendrites ($\mathbf{w}$), et transmettant un nouvel influx par ses terminaisons axonales si un certain seuil d'excitation est surpassé. Précisons que dans le schéma précédent, nous considérons que le neurone dispose déjà d'un certain niveau d'excitation (b) avant la réception des influx nerveux. Notons également qu'il existe plusieurs autres fonctions d'activations, et que chacune d'elle propose des bénéfices et des inconvénients liés à l'entraînement des réseaux de neurones [17].

2.3.2 Perceptron multi-couches

Le Perceptron multi-couches représente la forme la plus simple de réseau de neurones. Ce modèle est constitué d'un nombre défini de couches cachées connectées l'une à la suite de l'autre (Figure 2.4). Chaque couche cachée représente un regroupement de neurones artificiels. Le modèle est construit de sorte que chaque neurone d'une couche cachée prend en entrée l'ensemble des valeurs sortant des neurones de la couche précédente. Ainsi, lors du calcul d'une prédiction associée à une observation, les attributs $\mathbf{x}_i$ de celle-ci sont fournies comme valeurs d'entrées à la première couche cachée pour initier un phénomène de propagation avant via les neurones du modèles.

Soit $\mathbf{x}_i \in \mathcal{M}_{m,1}(\mathbb{R})$ une observation arbitraire provenant de notre ensemble de données, le modèle de Perceptron multi-couches à K couches cachées peut s'exprimer mathématiquement comme la composition de fonctions suivante :

$$f_{\boldsymbol{\omega},\boldsymbol{\gamma}}(\mathbf{x}_i) = f^{(K+1)} \circ f^{(K)} \circ \cdots \circ f^{(1)}(\mathbf{x}_i), \tag{2.8}$$

où

$$f^{(k)}(\mathbf{x}_i) = \sigma^{(k)}\left(\mathbf{W}^{(k)}\mathbf{x}_i + \mathbf{b}^{(k)}\right)$$

est tel que $\mathbf{W}^{(k)} \in \mathcal{M}_{n_k,n_{k-1}}(\mathbb{R})$ et $\mathbf{b}^{(k)} \in \mathcal{M}_{n_{k+1},1}(\mathbb{R})$. Ainsi, $\forall k \in \{1,\ldots,K+1\}$, $f^{(k)}(\cdot)$ constitue une fonction à domaine dans $\mathcal{M}_{n_{k-1},1}(\mathbb{R})$ et image dans $\mathcal{M}_{n_k,1}(\mathbb{R})$. En particulier, pour $k \leq K$, $f^{(k)}(\cdot)$ correspond à une couche cachée de n_k neurones. À l'exception de $\sigma^{(K+1)}(\cdot)$ qui correspond à la fonction identité, $\sigma^{(k)}(\cdot)$ est une fonction non linéaire qui s'applique à chaque élément du vecteur en entrée. Notons également que $n_0 = m$ et que $n_{K+1} = 1$.

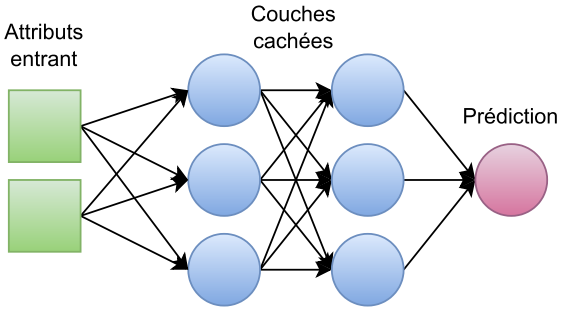

Figure 2.4 – Perceptron à deux couches cachées de trois neurones.

À partir de cette définition, nous pouvons préciser que $\boldsymbol{\omega}$ représente l'ensemble des poids du modèle, c'est-à-dire les différentes valeurs contenues au sein des matrices $\mathbf{W}^{(k)}$ et $\mathbf{b}^{(k)}$. Ainsi, soit P le nombre total de poids, nous pouvons définir l'espace des paramètres $\mathcal{W}$ comme étant $\mathbb{R}^P$. De son côté, l'ensemble d'hyperparamètres $\boldsymbol{\gamma}$ du Perceptron multi-couches peut contenir différentes valeurs réelles et catégoriques telles que le nombre de couches cachées, le nombre de neurones dans chacune des couches cachées et les choix des fonctions d'activation à utiliser.

2.3.3 Descente de gradient

La recherche des paramètres satisfaisant l'équation (2.3) consiste en un problème d'optimisation de taille pour les réseaux de neurones, considérant la nature non convexe de la fonction de perte d'entraînement [28]. Néanmoins, lorsqu'un réseau de neurones n'intègre que des fonctions d'activation dérivables, celui-ci dispose alors d'un caractère différentiable nous permettant d'avoir une idée du relief de la surface de perte en un point donné $\boldsymbol{\omega}$ par le calcul du gradient. Ainsi, considérant que le gradient indique la direction dans laquelle il faut se déplacer dans l'espace réel des paramètres pour augmenter la perte de la manière la plus significative, il suffit de réaliser un déplacement dans la direction opposée pour obtenir la plus grande réduction de celle-ci.

Descente de gradient stochastique et rétropropagation

Le concept expliqué ci-dessus est à la base de l'algorithme de descente de gradient stochastique (DGS). Toutefois, au lieu de considérer l'ensemble des observations de $\mathcal{X}_{train}$ conjointement, cet algorithme propose la mise à jour des paramètres $\boldsymbol{\omega}$ de manière itérative suite au calcul de la perte individuelle de chacune des observations contenues dans $\mathcal{X}_{train}$. La mise à jour des poids suit donc le schéma suivant :

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \eta \nabla_{\boldsymbol{\omega}} \ell(f_{\boldsymbol{\omega},\boldsymbol{\gamma}}(\mathbf{x}_i), t_i), \tag{2.9}$$

où $\eta \in \mathbb{R}^+$ porte le nom de rythme d'apprentissage et représente la taille du saut qui est fait dans la direction opposée du gradient $\nabla_{\boldsymbol{\omega}} \ell(f_{\boldsymbol{\omega},\boldsymbol{\gamma}}(\mathbf{x}_i), t_i)$. Deux étapes sont sous-jacentes à la réalisation de chaque itération de cet algorithme. La première étape

consiste à réaliser une propagation avant, soit le calcul de la prédiction $f_{\boldsymbol{\omega},\gamma}(\mathbf{x}_i)$ et la sauvegarde temporaire des valeurs obtenues à la sortie de chaque neurone du modèle. La seconde consiste à calculer $\nabla_{\boldsymbol{\omega}}\ell(f_{\boldsymbol{\omega},\gamma}(\mathbf{x}_i), t_i)$, c'est-à-dire déterminer la valeur de la dérivée partielle $\frac{\partial}{\partial w}\ell(f_{\boldsymbol{\omega},\gamma}(\mathbf{x}_i), t_i)$ de chaque poids $w \in \mathbb{R}$ contenu dans $\boldsymbol{\omega}$ par rapport à la fonction de perte au point $\mathbf{x}_i$. La réalisation de cette seconde étape requiert l'utilisation de la règle de dérivation en chaîne (Annexe B). En particulier, le calcul de la dérivée partielle d'un paramètre d'une couche $f^{(k)}(\cdot)$ quelconque nécessite les dérivées partielles des paramètres des couches subséquentes. Le processus décrivant les calculs des dérivées partielles de chaque couche du réseau de neurones en ordre inverse se nomme rétropropagation. Celui-ci fait usage des sorties enregistrées pour chaque neurone lors de la propagation avant. Bien que discuté ici pour le Perceptron multicouches, l'algorithme de DGS est valide pour toute architecture de réseau neuronal composée de fonctions différentiables. Nous référons à l'ouvrage *Pattern Recognition and Machine Learning* [8] pour obtenir plus de détails mathématiques concernant les étapes de propagation avant et arrière.

La mise à jour présentée à l'équation (2.9) peut être réalisée plus d'une fois sur l'ensemble des observations contenues dans $\mathcal{X}_{train}$. Nous nommons époques, tiré du terme anglais *epochs*, le nombre de fois où l'ensemble des observations de $\mathcal{X}_{train}$ est visité durant l'exécution de l'algorithme de DGS. Le nombre d'époques réalisées peut être déterminé en fonction d'un critère d'arrêt, soit par exemple l'atteinte d'un nombre d'itérations prédéterminé ou l'absence d'amélioration de la perte observée sur un ensemble de données indépendant de celui d'entraînement (Section 2.3.4).

Descente de gradient par lot

Bien qu'elle soit simple, la méthode de descente de gradient stochastique est rarement employée en pratique. Afin d'accélérer l'entraînement d'un réseau de neurones, il est plus commun d'effectuer une descente de gradient par lot. Cette méthode consiste à diviser d'abord l'ensemble d'entraînement en plusieurs lots $\mathcal{B}^{(i)}$ de tailles similaires, puis effectuer itérativement la mise à jour des poids sur ceux-ci au lieu des observations individuelles. Chaque actualisation de poids se représente donc par

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \eta\nabla_{\boldsymbol{\omega}} L(\boldsymbol{\omega}, \mathcal{B}^{(i)}), \tag{2.10}$$

où $L(\cdot)$ est la fonction introduite à l'équation (2.4). Cette méthode tire avantage du fait que la propagation avant, ainsi que la rétropropagation, peuvent être réalisées simultanément pour l'ensemble des observations d'un même lot par l'utilisation de produits matriciels. Dans le cas particulier où nous faisons usage d'un seul lot, c'est-à-dire $\mathcal{B}^{(1)} = \mathcal{X}_{train}$, nous référons à l'algorithme en tant que descente de gradient. À l'opposé, nous reconnaissons que le cas où $|\mathcal{B}^{(i)}| = 1 \ \forall i$ correspond à l'algorithme de DGS.

Fonctions d'activation dérivables par morceaux

Des ajustements peuvent être apportés pour permettre l'utilisation de l'algorithme de descente de gradient avec un réseau neuronal disposant de fonctions d'activation n'étant pas dérivables à certains points de leurs domaines, c'est-à-dire dérivables par morceaux. Prenons par exemple la fonction ReLU$(\cdot)$, présentée à l'équation (2.7), pour laquelle la dérivée n'est pas définie au point 0. En pratique, des valeurs par défaut sont déterminées pour les dérivées à ces points spécifiques. Généralement, ces valeurs correspondent aux dérivées observées aux limites inférieures ou supérieures des points en question [23].

Optimisation non convexe

La nature non convexe du problème d'optimisation que constitue l'entraînement des réseaux de neurones ne garantit pas l'atteinte d'une solution optimale $\boldsymbol{\omega}^*$. Autrement dit, les différents minimum locaux de la surface $L(\boldsymbol{\omega}, \mathcal{X}_{train})$ ne sont pas nécessairement des minimum globaux (voir l'Annexe B pour les définitions de minimum local et global). Deux problèmes principaux peuvent survenir. D'abord, il est possible d'atteindre un minimum local, qui n'est pas global, et ne pas pouvoir en sortir (Figure 2.5). Ensuite, il est possible de diverger et de n'atteindre aucun minimum local en raison d'un rythme d'apprentissage η qui est trop grand (Figure 2.6). Une solution consiste donc à réaliser l'entraînement à plusieurs reprises avec différents paramètres $\boldsymbol{\omega}$ initiaux pour obtenir plusieurs solutions et favoriser l'atteinte d'un minimum local.

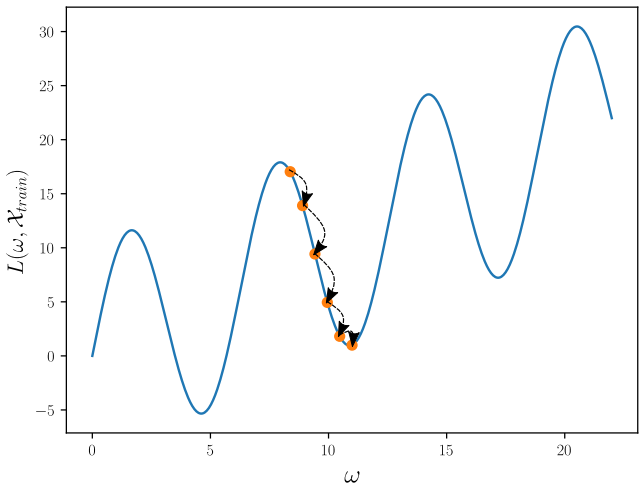

Figure 2.5 – Atteinte d'un minimum local non global par descente de gradient.

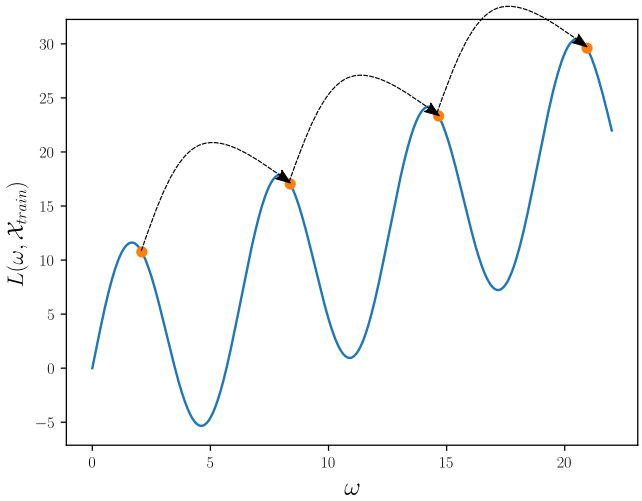

Figure 2.6 – Divergence de l'algorithme de descente de gradient.

Rythme d'apprentissage variable

En vue de favoriser l'atteinte d'un minimum local, une pratique consiste à utiliser un rythme d'apprentissage variable durant l'exécution de l'algorithme de descente de gradient par lot. Les méthodes proposées consistent principalement à réduire le rythme d'apprentissage η à la suite de chaque époque ou multiple d'époques. Par exemple, soit $\tau \in (0, 1)$ un rythme de décroissance donné, E le nombre d'époques réalisées jusqu'à présent, Δ un nombre d'époques pré-établi séparant chaque mise à jour du rythme d'apprentissage et $\mathbb{I}$ la fonction indicatrice, une méthode séquentielle simple consiste à appliquer la procédure d'actualisation suivante après chaque époque :

$$\eta \leftarrow \left(1 - \mathbb{I}_{\{E\%\Delta=0\}}\right) \eta + \mathbb{I}_{\{E\%\Delta=0\}} \tau \eta.$$

En ajout à cet exemple, nous référons à la documentation de la librairie *Pytorch*[1] pour d'autres méthodes d'actualisation du rythme d'apprentissage. Notons également l'existence d'approches adaptatives visant à faciliter la convergence de l'algorithme de descente de gradient par lot en adoptant une procédure d'actualisation des poids tenant compte des gradients observés lors des itérations précédentes, par exemple la descente de gradient avec momentum [75] et Adam [31]. Celles-ci n'ont toutefois pas été élaborées dans cette section puisqu'elles ne mettent pas à jour le rythme d'apprentissage.

2.3.4 Régularisation et arrêt prématuré

Le grand nombre de paramètres modifiables présents dans les réseaux de neurones muni ces architectures d'une grande capacité d'apprentissage. Comme il a été mentionné dans la section 2.1.2, cette puissance vient toutefois avec un risque augmenté de sur-apprentissage. Dans cette sous-section, nous introduirons la régularisation et l'arrêt prématuré, deux techniques pouvant être appliquées de manière complémentaire pour limiter le risque de sur-apprentissage.

1. Pytorch, «TORCH.OPTIM» https ://pytorch.org/docs/stable/optim.html, page consultée le 21 décembre 2022

Régularisation La technique de régularisation vise à limiter la capacité d'apprentissage d'un modèle en pénalisant celui-ci lorsqu'il adopte des paramètres de grandes tailles au sein de son architecture, les poids des neurones par exemple. Pour ce faire, il suffit d'ajouter un terme positif à l'équation (2.4), ne dépendant que des poids du modèle, de façon à obtenir la perte d'entraînement

$$L(\boldsymbol{\omega}, \mathcal{X}_{train}) + \lambda \sum_{w_j \in \boldsymbol{\omega}} |w_j|^q,$$

où $\lambda \in \mathbb{R}^+$ et $q \in \mathbb{N}/\{0\}$ sont des hyperparamètres déterminant la force de restriction appliquée. Ce terme de pénalité est considéré dans le calcul des dérivées partielles lors de la réalisation de la descente de gradient par lot. Ainsi, pour toute valeur q impaire, l'implémentation de la descente de gradient doit être adaptée afin de rester fonctionnelle si un certain poids w_j tend à 0, où la dérivée de la fonction valeur absolue n'est pas définie. À l'opposé, pour toute valeur de q paire, la fonction valeur absolue peut être retirée et ainsi aucune adaptation n'est requise.

Arrêt prématuré La technique d'arrêt prématuré consiste à vérifier les performances du modèle sur un ensemble de données indépendant de $\mathcal{X}_{train}$ et $\mathcal{X}_{test}$ après chaque époque ou multiple d'époques de descente de gradient par lot. Cet ensemble de données, communément identifié comme ensemble de validation $\mathcal{X}_{valid}$, peut être échantillonné à partir des observations restantes suite à la création de l'ensemble $\mathcal{X}_{test}$, de façon à obtenir le tuple $(\mathcal{X}_{train}, \mathcal{X}_{valid}, \mathcal{X}_{test})$ au lieu de la paire $(\mathcal{X}_{train}, \mathcal{X}_{test})$ comme il a été discuté jusqu'à présent. En supposant que la perte calculée sur $\mathcal{X}_{valid}$ suite à une époque offre une bonne approximation de la performance des paramètres courants sur $\mathcal{X}_{test}$, il suffit d'arrêter l'exécution de l'algorithme de descente de gradient par lot lorsque qu'aucune diminution de la perte de validation $L(\boldsymbol{\omega}, \mathcal{X}_{valid})$ n'a été observée durant les p dernières époques. À la suite de cet arrêt, nous conservons seulement les paramètres qui étaient associés à la perte de validation minimale observée durant l'entraînement. La valeur de $p \in \mathbb{N}$ est un hyperparamètre connu sous le nom de patience.

2.4 Réseaux neuronaux graphiques convolutifs

Dans certains contextes, une structure graphique accompagne l'ensemble de données annotées qui est à notre disposition. Pensons entre autres à un jeu de données où les observations $\mathbf{x}_i$ représentent les utilisateurs d'un réseau social. Dans ces circonstances, nous constatons qu'une quantité d'information liée aux utilisateurs se trouve à la fois dans leurs attributs, mais également dans les liens qu'ils partagent avec les autres utilisateurs. Par exemple, deux individus peuvent être connectés l'un à l'autre s'ils exhibent un lien d'amitié sur le réseau. Il est notamment possible que d'autres relations plus complexes impliquent des connections à sens unique entre les utilisateurs. Imaginons le cas où un membre A du réseau serait abonné à un membre B, mais l'inverse serait faux. Toutefois, bien que l'information structurelle liée au graphe d'un jeu de données peut jouer un rôle dans la prédiction des cibles, elle reste imperceptible dans l'entraînement d'un réseau de neurones conventionnel tel que le Perceptron multi-couches. Ainsi, les réseaux de neurones graphiques convolutifs (RNGC) ont été développés pour le type de contexte spécifique où l'ensemble de données annotées prend la forme d'un réseau d'information (voir Réseau d'information) où chaque objet possède un vecteur d'attributs. Nous débuterons cette section en définissant certains éléments théoriques dont la notion est fondamentale pour la compréhension des RNGC. Nous expliquerons ensuite le fonctionnement derrière ces modèles, ferons un bref survol de leur origine et explorerons plus en détail certaines architectures de RNGC.

2.4.1 Éléments théoriques

Dans cette sous-section, nous présentons différents objets mathématiques au coeur de la théorie des RNGC, soit le graphe orienté, le voisinage en k-sauts et le réseau d'information.

Graphe orienté

Un graphe orienté $\mathcal{G}$ est défini mathématiquement par la relation

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}),$$

où $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ est un ensemble de n noeuds et $\mathcal{E} = \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}\}$ est un ensemble d'arêtes. Une illustration de $\mathcal{G}$ peut être obtenue à partir de sa représentation sous forme d'ensembles, et inversement (Figure 2.7). Chaque arête y est représentée par une flèche. L'orientation de chacune de celles-ci est dictée par l'ordre des noeuds la constituant (c.à-d., $(v_i, v_j) \in \mathcal{E} \Rightarrow v_i \rightarrow v_j$). Des matrices peuvent être également générées à partir de la structure d'un graphe. Notons en particulier, la matrice d'adjacence et les matrices de degrés qui sont nécessaires à l'implémentation efficace des modèles discutés à la section 2.4.4.

Matrice d'adjacence La matrice d'adjacence $\mathbf{A} \in \mathcal{M}_{n,n}(\mathbb{N})$ d'un graphe $\mathcal{G}$ est définie de sorte que chaque élément $\mathbf{A}_{i,j}$ représente le nombre de fois où l'arête (v_i, v_j) est présente dans l'ensemble $\mathcal{E}$.

Matrices des degrés Soit un graphe $\mathcal{G}$, nous notons $d^+(v_i)$ et $d^-(v_i)$ le nombre d'arêtes pointant et sortant d'un noeud v_i respectivement. Ces valeurs, portant le nom de degrés entrant et sortant, peuvent être calculées à partir de la matrice d'adjacence comme suit :

$$d^+(v_i) = \sum_i \mathbf{A}_{i,j}, \;\; d^-(v_i) = \sum_j \mathbf{A}_{i,j}.$$

À partir de ces valeurs, nous pouvons déterminer les deux matrices diagonales $\mathbf{D}^+, \mathbf{D}^- \in \mathcal{M}_{n,n}(\mathbb{N})$ contenant respectivement les degrés entrant et sortant des noeuds de l'ensemble du graphe. Celles-ci se définissent de la façon suivante :

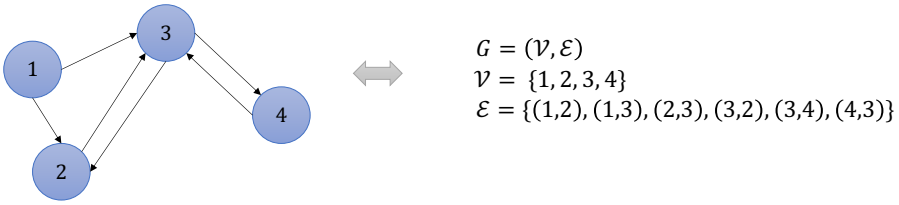

Figure 2.7 – Illustration d'un graphe orienté à partir de sa définition mathématique.

$$\mathbf{D}^{+}_{i,j} = \begin{cases} d^{+}(v_i) & \text{si } i = j \\ 0 & \text{sinon} \end{cases}, \ \mathbf{D}^{-}_{i,j} = \begin{cases} d^{-}(v_i) & \text{si } i = j \\ 0 & \text{sinon} \end{cases}.$$

Voisinage en k-sauts

Soit un graphe orienté $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, nous définissons le voisinage en k-sauts d'un noeud arbitraire $v_i \in \mathcal{V}$, c'est-à-dire $\mathcal{N}_k(v_i)$, par l'ensemble de noeuds capables d'atteindre v_i dans $\mathcal{G}$ en parcourant exactement k arêtes de manière à respecter leurs orientations. Considérant que l'élément aux coordonnées i, j de la k-ième puissance de la matrice d'adjacence (c.-à-d., $\mathbf{A}^{k}_{i,j}$), représente le nombre de chemins connectant v_i à v_j en k-sauts [59], nous pouvons établir que

$$\mathcal{N}_k(v_i) = \{v_j \mid \mathbf{A}^{k}_{j,i} > 0\}.$$

Réseau d'information

Dans le contexte d'apprentissage supervisé, les graphes orientés prennent de l'importance lorsqu'ils font abstraction d'une structure de données réelle, c'est-à-dire, lorsqu'une valeur sémantique est ajoutée à chacune de leurs composantes. Dans ces circonstances, nous employons le terme réseau d'information pour décrire ces structures plus riches.

Soit $\mathcal{O}$ un ensemble d'objets et $\mathcal{R}$ un ensemble constitué des relations possibles entre ces différents objets. Nous définissons un réseau d'information comme un graphe orienté $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, muni des fonctions surjectives $\tau : \mathcal{V} \to \mathcal{O}$, $\phi : \mathcal{E} \to \mathcal{R}$, associant respectivement les noeuds et les arêtes à des types d'objets et de liens spécifiques. Lorsque le nombre d'objets ou de liens est supérieur à un (c.-à-d., $|\mathcal{O}| > 1$ ou $|\mathcal{R}| > 1$), le réseau d'information est dit hétérogène (Figure 2.8). Autrement, nous disons que celui-ci est homogène (Figure 2.9). Pour la suite de ce document, nous ne considérerons que les réseaux d'information homogènes.

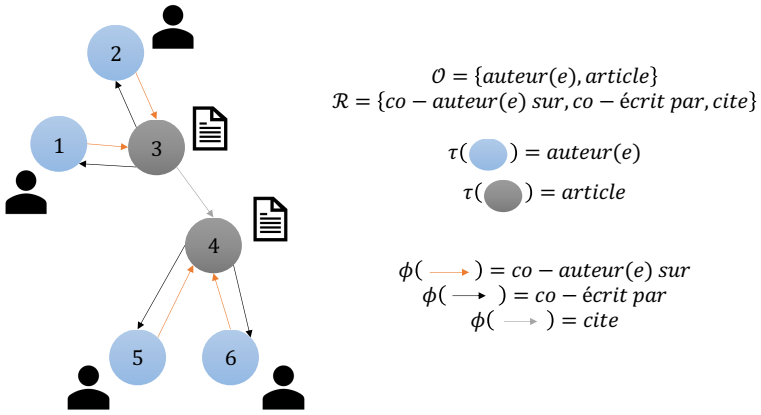

Figure 2.8 – Réseau d'information hétérogène. Celui-ci représente un réseau bibliographique composé d'auteurs et d'articles.

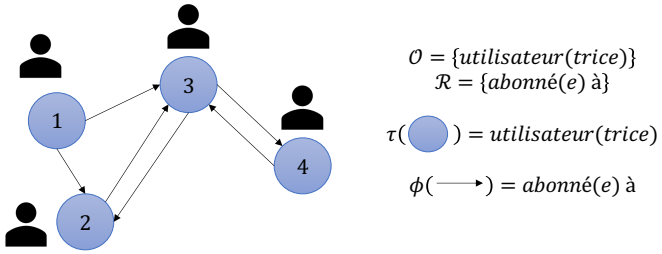

Figure 2.9 – Réseau d'information homogène. Celui-ci représente un réseau social composé d'utilisateurs pouvant être abonnés l'un à l'autre.

2.4.2 Fonctionnement

La présence de jeux de données annotées prenant la forme de réseaux d'information motive le développement continuel de nouvelles architectures de réseaux de neurones pouvant bénéficier à la fois des attributs $\mathbf{x}_i \in \mathcal{M}_{m,1}(\mathbb{R})$ de chaque noeud v_i (c.-à-d., objet), mais également des arêtes (c.-à-d., liens) qui les relient, dans l'accomplissement de tâche d'apprentissage supervisé. En particulier, les réseaux de neurones graphiques convolutifs (RNGC) parviennent à ce défi en généralisant l'opération de convolution à des structures de graphes. Pour calculer la prédiction d'un noeud v_i au sein d'un réseau d'information constitué d'un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ à n noeuds, chaque modèle d'apprentissage $f_{\omega,\gamma}$ issu de la famille des RNGC doit considérer l'index du noeud, la matrice d'adjacence $\mathbf{A}$ du graphe et la matrice $\mathbf{X}$ contenant l'ensemble des attributs

du jeux de données, c'est-à-dire

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}.$$

En particulier, chacun des modèles à K couches se représente par la composition de fonctions suivante :

$$f_{\boldsymbol{\omega},\gamma}(i, \mathbf{X}, \mathbf{A}) = f \circ \psi^{(K)} \circ \psi^{(K-1)} \ \cdots \ \psi^{(1)}(\mathbf{h}_i^{(0)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(0)}), \tag{2.11}$$

où $f : \mathcal{M}_{a_K,1}(\mathbb{R}) \to \mathbb{R}$ est une fonction différentiable quelconque (p. ex., un Perceptron multi-couches) et

$$\psi^{(k+1)}(\mathbf{h}_i^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)}) = \sigma^{(k+1)}\left(\mathbf{W}^{(k+1)}\mathrm{AG}\left(\mathbf{h}_i^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)}\right) + \mathbf{b}^{(k+1)}\right), \tag{2.12}$$

correspond à une opération de convolution, avec

$$\begin{gathered} \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)} = \{\mathbf{h}_j^{(k)} \mid v_j \in \mathcal{N}_1(v_i)\} \\ \mathbf{h}_i^{(k+1)} = \psi^{(k+1)}\left(\mathbf{h}_i^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)}\right), \ \forall k \geq 0 \\ \mathbf{h}_i^{(0)} = \mathbf{x}_i. \end{gathered}$$

Dans l'équation (2.12), $\mathbf{W}^{(k+1)} \in \mathcal{M}_{a_{k+1},a_k}(\mathbb{R})$ et $\mathbf{b}^{(k+1)} \in \mathcal{M}_{a_{k+1},1}(\mathbb{R})$ sont des matrices de paramètres modifiables où $a_0 = m$, $\sigma^{(k)}(\cdot)$ représente une fonction d'activation non linéaire et $\mathrm{AG}(\cdot)$ est une fonction différentiable agrégeant les vecteurs qui lui sont donnés en entrée. Également, notons que la matrice $\mathbf{A}$ indiquée à l'équation (2.11) est implicitement utilisée dans l'équation (2.12) pour déterminer l'ensemble $\mathcal{N}_1(v_i)$.

Plus concrètement, la composition de fonctions $\psi^{(K)} \circ \psi^{(K-1)} \ \cdots \ \psi^{(1)}(\cdot)$ présentée à l'équation (2.11) constitue un processus itératif qui vise à, pour n'importe quel noeud v_i, générer une représentation alternative $\mathbf{z}_i$. Cette représentation encapsule à la fois l'information provenant du noeud lui-même et celle des noeuds inclus dans les voisinages à k-sauts $\mathcal{N}_1(v_i), \mathcal{N}_2(v_i), \ \ldots \ , \mathcal{N}_K(v_i)$, de sorte que l'apport d'information transmis par les noeuds dans $\mathcal{N}_k(v_i)$ soit supérieur à celui de $\mathcal{N}_{k+1}(v_i)$ $\forall k < K$. Au

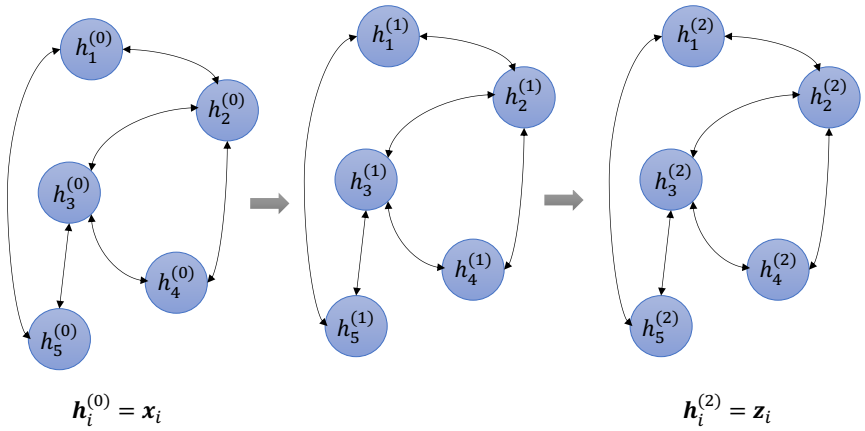

Figure 2.10 – Processus itératif mettant à jour les représentations cachées au sein d'un RNGC avec $K = 2$. Des flèches à double sens sont utilisées pour représenter plus simplement la présence de deux arêtes orientées.

cours de chaque itération k de ce processus, nous notons la nouvelle représentation d'un noeud v_i par $\mathbf{h}_i^{(k)}$ et lui référons sous le terme de représentation cachée. Comme illustrée à l'équation (2.12), la mise à jour de la représentation cachée d'un noeud v_i nécessite les représentations cachées des noeuds avoisinant (c.-à-d., $\mathcal{N}_1(v_i)$). Un exemple du processus itératif est présenté dans la Figure 2.10 pour un graphe orienté à cinq noeuds où $K = 2$.

À l'aide de ce même exemple, nous pouvons visualiser les étapes effectuées pour obtenir $\mathbf{z}_1$ (Figure 2.11). Le mécanisme d'agrégation $\mathrm{AG}(\cdot)$ intégré au sein de chaque boîte noire est l'élément principal permettant de distinguer les différents types de RNGC. Certains mécanismes seront décris plus en détail dans la section 2.4.4.

2.4.3 Origine

Le premier concept de réseau de neurones graphiques fut introduit dans l'article *The Graph Neural Network Model* [72]. Les auteurs y présentent d'abord l'idée d'une fonction paramétrique différentiable $\psi(\cdot)$, partagée par chacun des n noeuds d'un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ au sein d'un réseau d'information, afin de mettre à jour leur

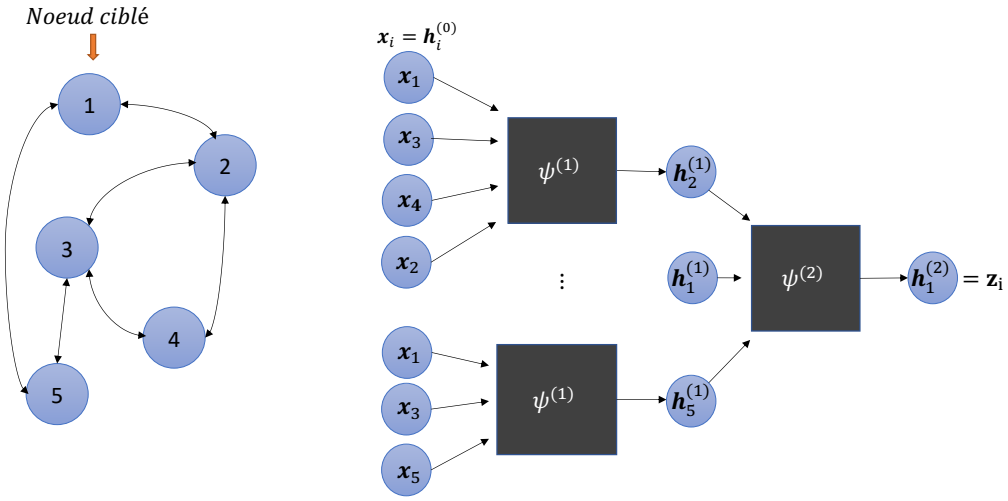

Figure 2.11 – Propagation de l'information au sein d'un RNGC avec $K = 2$ pour la création de la représentation alternative du noeud 1. Les flèches représentent la direction du flux d'information tandis que les boîtes noires représentent le mécanisme d'agrégation et de transformation des messages provenant des noeuds (c.-à-d. les couches de convolution graphiques).

représentation cachée individuelle. En plus d'être conceptualisée de la façon suivante :

$$\psi(\mathbf{h}_i^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)}) = \mathbf{h}_i^{(k+1)}, \tag{2.13}$$

cette fonction portant le nom de fonction de transition locale doit être une contraction. C'est-à-dire que $\forall v_i, v_j \in \mathcal{V}$ il doit exister $\mu \in [0, 1)$ tel que

$$\|\psi(\mathbf{h}_i^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)}) - \psi(\mathbf{h}_j^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_j)}^{(k)})\| \leq \mu \|\mathbf{h}_i^{(k)} - \mathbf{h}_j^{(k)}\|.$$

Par la suite, les auteurs développent l'idée d'une fonction de transition globale $\Psi(\cdot)$ permettant de mettre à jour les représentations cachées de l'ensemble des noeuds du réseau d'information en appliquant simultanément la fonction de transition locale à chacun d'eux. Considérant que $\psi(\cdot)$ soit une contraction, la fonction de transition globale

$$\Psi(\mathbf{H}^{(k)}) = \mathbf{H}^{(k+1)} = \begin{bmatrix} \psi(\mathbf{h}_1^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_1)}^{(k)})^T \\ \vdots \\ \psi(\mathbf{h}_n^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_n)}^{(k)})^T \end{bmatrix} \tag{2.14}$$

représente un système dynamique qui est assuré, par le théorème du point fixe de Banach [32], de converger vers un point $\mathbf{H}^*$ pour toute valeur $\mathbf{H}^{(0)}$, en particulier $\mathbf{H}^{(0)} = \mathbf{X}$. Ainsi, la solution adoptée pour calculer la prédiction d'un noeud v_i consiste d'abord à calculer une estimation $\hat{\mathbf{H}}^*$ du point fixe $\mathbf{H}^*$ par l'application consécutive de $\Psi(\cdot)$ jusqu'à un certain critère d'arrêt, puis y récupérer la représentation $\mathbf{z}_i$ de la façon suivante :

$$\mathbf{z}_i = \hat{\mathbf{H}}^{*^T} \mathbf{o}_i,$$

où $\mathbf{o}_i \in \mathcal{M}_{n,1}(\{0,1\})$ est un vecteur dont l'unique élément non-nul se situe à la position i. Pour ne nommer qu'un exemple, nous pouvons mettre fin au processus itératif lorsque, pour un certain seuil prédéterminé $\epsilon \in \mathbb{R}^+$, nous observons $\|\mathbf{H}^{(k+1)} - \mathbf{H}^{(k)}\|_{max} \leq \epsilon$ (voir l'Annexe B pour la définition de la norme maximale).

Ce premier modèle, catégorisé comme étant un réseau neuronal graphique récurrent [79], fut éventuellement remplacé par les RNGC. L'adoption de ces nouveaux modèles fut motivée par la simplicité et la flexibilité de leurs architectures. Alors que la simplicité des RNGC provient du retrait du critère nécessitant l'atteinte d'un point fixe et par conséquent, la réduction du nombre de couches de convolution, leur flexibilité se manifeste par la possibilité d'utiliser la composition de plusieurs fonctions de transition locale $\psi^{(k)}(\cdot)$ différentes qui ne sont pas nécessairement des contractions. Ces modifications n'impliquent aucun compromis quant à la performance de prédiction des RNGC. L'utilisation d'un grand nombre de couches de convolution implique que chaque noeud d'un graphe intègre de l'information provenant d'une multitude de voisinages en k-sauts, voir même l'entièreté du graphe. Or, dans de telles circonstances, les représentations alternatives de chaque noeuds sont parfois indistinguables, devenant ainsi nuisibles à la tâche de prédiction en question. Ce phénomène, portant le nom de sur-lissage [40], décourage l'atteinte d'un point de convergence $\mathbf{H}^*$ et motive plutôt la mise en place d'architectures favorisant l'intégration des informations

du voisinage local dans la création des représentations alternatives $\mathbf{z}_i$, en particulier les RNGC.

2.4.4 Mécanismes d'aggrégations

Dans cette sous-section, nous explorerons trois mécanismes d'agrégation prenant la forme de somme pondérée. Nous visiterons chacun de ceux-ci de sorte à suivre un ordre de complexité croissant. Nous montrerons également comment intégrer ceux-ci au sein de fonctions de transition globales efficientes.

Agrégation par moyenne

Cette méthode d'agrégation consiste simplement à prendre la moyenne de la représentation cachée du noeud ciblé, ainsi que celles des noeuds avoisinant :

$$\mathrm{AG}\left(\mathbf{h}_i^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)}\right) = \sum_{v_j \in \mathcal{N}_1(v_i) \ \cup \ v_i} \frac{\mathbf{h}_j^{(k)}}{|\mathcal{N}_1(v_i) \ \cup \ v_i|}.$$

La mise à jour simultanée des représentations cachées des noeuds d'un graphe $\mathcal{G}$ muni de la matrice d'adjacence $\mathbf{A}$ peut être réalisée par la fonction de transition globale suivante :

$$\Psi^{(k+1)}(\mathbf{H}^{(k)}, \mathbf{A}) = \sigma^{(k+1)}\left(\tilde{\mathbf{D}}^{+^{-1}} \tilde{\mathbf{A}}^T \mathbf{H}^{(k)} \mathbf{W}^{(k+1)^T} + \mathbf{B}^{(k+1)}\right),$$

où $\tilde{\mathbf{A}} = \mathbf{A} + \mathbb{I}$, $\tilde{\mathbf{D}}^+$ est la matrice des degrés entrant associée à $\tilde{\mathbf{A}}$ telle que

$$\tilde{\mathbf{D}}^{+^{-1}}{}_{i,j} = \begin{cases} 1/d^+(v_i) & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$$

et $\mathbf{B}^{(k+1)}$ contient des répliques du vecteur de biais :

$$\mathbf{B}^{(k+1)} = \begin{bmatrix} \mathbf{b}^{(k+1)^T} \\ \vdots \\ \mathbf{b}^{(k+1)^T} \end{bmatrix}.$$

L'addition de la matrice identité $\mathbb{I}$ à la matrice d'adjacence permet l'ajout fictif d'une arête reliant chaque noeud à lui-même dans le graphe original. Notons également que pour minimiser l'espace de mémoire requise, l'ajout des biais peut être fait par diffusion, tiré du terme anglais *broadcasting*[2]. Dans le cas présent, la diffusion est appliquée par l'addition du vecteur $\mathbf{b}^{(k+1)^T}$ à chaque ligne de la matrice $\tilde{\mathbf{D}}^{+^{-1}}\tilde{\mathbf{A}}^T\mathbf{H}^{(k)}\mathbf{W}^{(k+1)^T}$ en ne conservant qu'une seule copie du vecteur $\mathbf{b}^{(k+1)}$ en mémoire.

Agrégation par degrés

Le mécanisme d'agrégation proposé par Thomas N. Kipf et Max Welling [34], pour le modèle GCN (*Graph convolutional network*), c'est-à-dire

$$\mathrm{AG}\left(\mathbf{h}_i^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)}\right) = \sum_{v_j \in \mathcal{N}_1(v_i) \ \cup \ v_i} \frac{\mathbf{h}_j^{(k)}}{|\mathcal{N}_1(v_i)||\mathcal{N}_1(v_j)|},$$

tire ses origines de la théorie spectrale des graphes. Il est défini de sorte à réduire les poids associés aux représentations cachées des noeuds avoisinant disposant eux-même de beaucoup de voisins. Il se conforme donc à l'hypothèse que les noeuds avoisinant un noeud v_i et disposant d'un grand degré entrant transportent moins d'informations pertinentes à la construction de la représentation alternative $\mathbf{z}_i$ de v_i . La fonction de transition globale associée à cette méthode d'agrégation se définie par :

$$\Psi^{(k+1)}(\mathbf{H}^{(k)}, \mathbf{A}) = \sigma^{(k+1)}\left(\tilde{\mathbf{D}}^{+^{-\frac{1}{2}}}\tilde{\mathbf{A}}^T\tilde{\mathbf{D}}^{+^{-\frac{1}{2}}}\mathbf{H}^{(k)}\mathbf{W}^{(k+1)^T} + \mathbf{B}^{(k+1)}\right),$$

où

$$\tilde{\mathbf{D}}^{+^{-\frac{1}{2}}}{}_{i,j} = \begin{cases} 1/\sqrt{d^+(v_i)} & \text{si } i = j \\ 0 & \text{sinon} \end{cases}.$$

2. NumPy, «Broadcasting» https ://numpy.org/doc/stable/user/basics.broadcasting.html, page consultée le 4 novembre 2022

Agrégation par attention

Contrairement aux méthodes précédentes qui calculent les poids de la somme pondérée en fonction de la structure du graphe seulement, la méthode d'agrégation développée par Petar Veličković [76] permet de considérer également les attributs des représentations cachées. Cette flexiblité accrue est supportée par l'intégration d'une fonction paramétrique

$$\begin{aligned} a^{(k+1)} \colon \mathcal{M}_{a_k,1}(\mathbb{R})^2 &\to \mathbb{R}^+ \\ (\mathbf{h}_i^{(k)}, \mathbf{h}_j^{(k)}) &\mapsto \alpha_{i,j}^{(k+1)} \end{aligned}$$

permettant de calculer l'importance d'une représentation cachée $\mathbf{h}_j^{(k)}$ d'un noeud $v_j \in \mathcal{N}_1(v_i)$ du point de vue de n'importe quel noeud v_i, considérant $\mathbf{h}_i^{(k)}$. La valeur $\alpha_{i,j}^{(k+1)}$ représente l'attention portée au noeud v_j par le noeud v_i dans la k-ième couche d'un RNGC et se calcule par

$$a^{(k+1)}(\mathbf{h}_i^{(k)}, \mathbf{h}_j^{(k)}) = \begin{cases} \exp\left(\gamma\left(\mathbf{a}^{(k+1)^T}\left[\mathbf{W}^{(k+1)}\mathbf{h}_i^{(k)} \,||\, \mathbf{W}^{(k+1)}\mathbf{h}_j^{(k)}\right]\right)\right) \text{ si } v_j \in \mathcal{N}_1(v_i) \\ 0 \text{ sinon} \end{cases},$$

avec $\mathbf{W}^{(k+1)} \in \mathcal{M}_{a_{k+1},a_k}(\mathbb{R})$, $\mathbf{a}^{(k+1)} \in \mathcal{M}_{2(a_{k+1}),1}(\mathbb{R})$, γ la fonction d'activation non linéaire LeakyReLU$(\cdot)$ définie à l'Annexe B et $\exp(\cdot)$ la fonction exponentiel e^x. Notons également que la matrice de poids $\mathbf{W}^{(k+1)}$ est partagée avec la fonction de transition locale telle que présentée à l'équation (2.12). Le concept d'attention est intégré au sein du mécanisme d'agrégation de la façon suivante :

$$\text{AG}\left(\mathbf{h}_i^{(k)}, \mathbf{h}_{\mathcal{N}_1(v_i)}^{(k)}\right) = \sum_{v_j \in \mathcal{N}_1(v_i) \,\cup\, v_i} \tilde{\alpha}_{i,j}{}^{(k+1)} \mathbf{h}_j^{(k)},$$

où $\tilde{\alpha}_{i,j}{}^{(k+1)}$ représente l'attention normalisée, c'est-à-dire

$$\tilde{\alpha}_{i,j}{}^{(k+1)} = \alpha_{i,j}^{(k+1)} \,/ \sum_{v_l \in \mathcal{N}_1(v_i) \,\cup\, v_i} \alpha_{i,l}^{(k+1)}.$$

Ainsi, soit la $\mathbf{T}^{(k+1)}$ une matrice définie tel que $\mathbf{T}_{i,j}^{(k+1)} = \tilde{\alpha}_{i,j}{}^{(k+1)}$, nous pouvons mettre en place la fonction de transition globale

$$\Psi^{(k+1)}(\mathbf{H}^{(k)}, \mathbf{A}) = \sigma^{(k+1)}\left(\mathbf{T}^{(k+1)}\mathbf{H}^{(k)}\mathbf{W}^{(k+1)^T} + \mathbf{B}^{(k+1)}\right).$$

2.5 Optimisation d'hyperparamètres

La performance d'un modèle d'apprentissage supervisé dépend des paramètres $\boldsymbol{\omega}$ appris lors de son entraînement, mais également des hyperparamètres $\boldsymbol{\gamma}$ choisis avant cette étape. Notons par exemple le choix du nombre d'arbres de décision dans une forêt aléatoire. Ainsi, plusieurs méthodes ont été mises sur pieds pour aider à la sélection d'un ensemble de valeurs d'hyperparamètres. Dans cette sous-section, nous définirons d'abord formellement le problème d'optimisation d'hyperparpamètres. Nous regarderons ensuite certaines méthodes d'optimisation de base couramment utilisées, puis terminerons en regardant des techniques plus complexes tirées du domaine de l'optimisation bayésienne.

2.5.1 Formulation du problème

Soit un modèle d'apprentissage $f_{\boldsymbol{\omega},\boldsymbol{\gamma}}$ muni de H hyperparamètres fixes $\boldsymbol{\gamma} \in \Gamma$, tel que $\Gamma = \Gamma_1 \times \Gamma_2 \times \cdots \times \Gamma_H$ et où Γ_h est le domaine du h-ième hyperparamètre. Soit également un ensemble d'entraînement $\mathcal{X}_{train}$ obtenu à partir d'un jeu de données annotées et $\{(\mathfrak{X}_{train}^{(j)}, \mathfrak{X}_{test}^{(j)})\}_{j=1}^{T}$ des paires d'ensembles d'entraînement et de test internes définies de sorte que

$$\mathfrak{X}_{train}^{(j)} \cup \mathfrak{X}_{test}^{(j)} = \mathcal{X}_{train}, \text{ et } \mathfrak{X}_{train}^{(j)} \cap \mathfrak{X}_{test}^{(j)} = \emptyset \ \forall j.$$

La recherche d'un ensemble de valeurs d'hyperparmètres $\boldsymbol{\gamma}$ se formule par le problème d'optimisation

$$\arg\min_{\boldsymbol{\gamma}\in\Gamma} \sum_{j=1}^{T} \frac{1}{T} \mathcal{L}(f_{\boldsymbol{\omega},\boldsymbol{\gamma}}, \mathfrak{X}_{train}^{(j)}, \mathfrak{X}_{test}^{(j)}) \tag{2.15}$$

où $\mathcal{L}(\cdot)$ est la perte de validation croisée introduite à l'équation (2.5). Ainsi, comme

illustré à la Figure 2.12, l'optimisation d'hyperparamètres constitue une procédure de validation croisée interne pouvant s'imbriquer au sein du processus de validation croisée déjà mis en place dans la section 2.1.3.

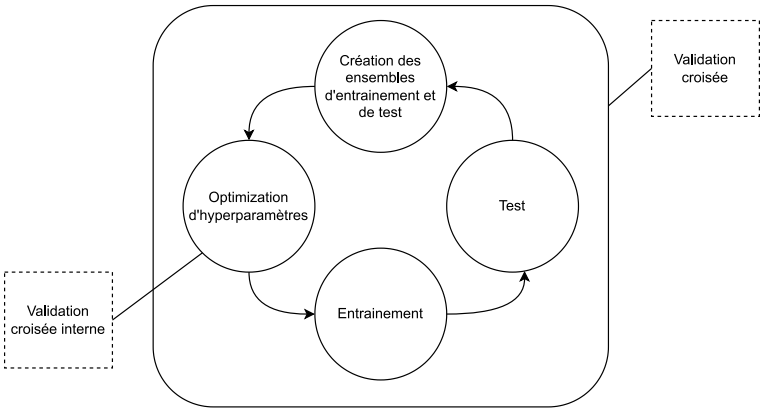

Figure 2.12 – Processus de validation croisée avec optimisation d'hyperparamètres.

2.5.2 Méthodes de bases

Dans cette sous-section, nous regarderons le fonctionnement de deux méthodes de recherche d'hyperparamètres séquentielles dont les processus d'échantillonnage d'ensembles à tester ne tiennent pas compte des résultats observés sur les ensembles de valeurs d'hyperparamètres échantillonnées précédemment.

Recherche en quadrillage

La recherche en quadrillage, tirée du terme anglais *grid search*, consiste à définir d'abord un ensemble de valeurs à tester pour chaque hyperparamètre, de sorte que Γ_i soit un domaine discret et fini $\forall i$, pour ensuite tester toutes les combinaisons de valeurs d'hyperparamètres possibles. Le temps de recherche requis par cette méthode dépend du nombre d'hyperparamètres et de la taille du domaine défini pour chacun d'eux puisque le nombre de combinaisons à évaluer est égale à $\prod_i |\Gamma_i|$.

Recherche aléatoire

Pour la recherche aléatoire, le domaine défini pour chaque hyperparamètre peut être continu ou discret, mais nécessite d'être préalablement associé à une distribution de probabilité. La méthode consiste à évaluer n ensembles d'hyperparamètres formés en échantillonnant leurs composantes à partir des distributions de probabilité préétablies.

2.5.3 Optimisation bayésienne

Dans le cadre de l'apprentissage automatique, l'optimisation bayésienne correspond à une méthode de recherche d'hyperparamètres prenant en considération les résultats passés pour faire une sélection plus judicieuse des futurs ensembles de valeurs d'hyperparamètres à tester. L'évaluation d'un seul ensemble de valeurs d'hyperparamètres, c'est-à-dire, le calcul de la fonction objectif

$$S(\gamma) = \sum_{j=1}^{T} \frac{1}{T} \mathcal{L}(f_{\boldsymbol{\omega},\boldsymbol{\gamma}}, \mathfrak{X}_{train}^{(j)}, \mathfrak{X}_{test}^{(j)}) \tag{2.16}$$

introduite à l'équation (2.15), peut demander du temps et des ressources computationnelles considérables. Ainsi, l'approche bayésienne vise à augmenter l'effort de calcul mené durant la sélection de chaque ensemble pour converger plus rapidement vers une solution satisfaisante. Dans cette sous-section nous regarderons l'algorithme portant le nom d'estimateur de Parzen à structure arborescente.

Estimateur de Parzen à structure arborescente

L'estimateur de Parzen à structure arborescente (EPA) [6], tiré du terme anglais *Tree-structured Parzen estimator*, guide sa recherche d'hyperparamètres par la modélisation d'un modèle $\mathbb{P}(\boldsymbol{\gamma}|s)$ estimant la probabilité qu'un modèle ait fait usage des hyperparamètres $\boldsymbol{\gamma}$ sachant qu'il ait obtenu un score s avec la fonction objectif $S(\boldsymbol{\gamma})$. Pour ce faire, celui-ci modélise $\mathbb{P}(\boldsymbol{\gamma}|s)$ par

$$\mathbb{P}(\boldsymbol{\gamma}|s) = \begin{cases} u(\boldsymbol{\gamma}) & \text{si } s < s' \\ o(\boldsymbol{\gamma}) & \text{si } s \geq s' \end{cases},$$

où $u(\cdot)$ et $o(\cdot)$ sont des estimateurs de $\mathbb{P}(\boldsymbol{\gamma} \,|\, s < s')$ et $\mathbb{P}(\boldsymbol{\gamma} \,|\, s \geq s')$ respectivement (p. ex., des estimateurs de densités à noyaux), tandis que $s' \in \mathbb{R}^+$ est une valeur qui est fixée par l'utilisateur ou estimée en fonction des différents scores obtenus jusqu'à présent dans la recherche, la médiane de ceux-ci par exemple. L'algorithme échantillonne d'abord aléatoirement un nombre prédéterminé d'ensembles de valeurs d'hyperparamètres pour initialiser les estimateurs $u(\cdot)$ et $o(\cdot)$. Par la suite, tenant pour acquis que $S(\boldsymbol{\gamma})$ soit une fonction à minimiser, chaque nouvel ensemble est choisi par la résolution du problème

$$\arg\max_{\boldsymbol{\gamma} \in \Gamma} \frac{u(\boldsymbol{\gamma})}{o(\boldsymbol{\gamma})}. \tag{2.17}$$

Celui-ci se traduit par la recherche de l'ensemble $\boldsymbol{\gamma}$ possédant simultanément une grande probabilité d'être associé à une petite valeur de s et une faible probabilité d'être associé à une grande valeur de s. Notons qu'après chaque itération, ou multiple d'itérations, $u(\cdot)$ et $o(\cdot)$ doivent être mis à jour pour permettre à l'algorithme d'améliorer la sélection des points subséquents.

Visualisons le fonctionnement de cet algorithme avec un exemple simple. Considérons un modèle $f_{\boldsymbol{\omega},\boldsymbol{\gamma}}$ tel que $\boldsymbol{\gamma}$ n'est constitué que d'une valeur réelle pouvant se retrouver dans l'intervalle $[0, 3\pi]$. Considérons également que $S(\boldsymbol{\gamma}) = \sin(\boldsymbol{\gamma}) + 1$ et que nous disposons d'un regroupement de 55 points $\{(\boldsymbol{\gamma}^{(k)}, S(\boldsymbol{\gamma}^{(k)}))\}_{k=1}^{55}$ ayant déjà été évalués suivant un échantillonnage aléatoire (Figure 2.13). À partir des points observés, nous sommes en mesure de créer deux groupes, soit le groupe de points tels que $S(\boldsymbol{\gamma}^{(k)}) < s'$ et celui satisfaisant $S(\boldsymbol{\gamma}^{(k)}) \geq s'$. De ces groupes, nous pouvons ensuite construire $u(\cdot)$ et $o(\cdot)$ (Figure 2.14). Finalement, nous pouvons résoudre le problème de l'équation (2.17) pour déterminer la prochaine valeur d'hyperparamètre à échantilloner (Figure 2.15).

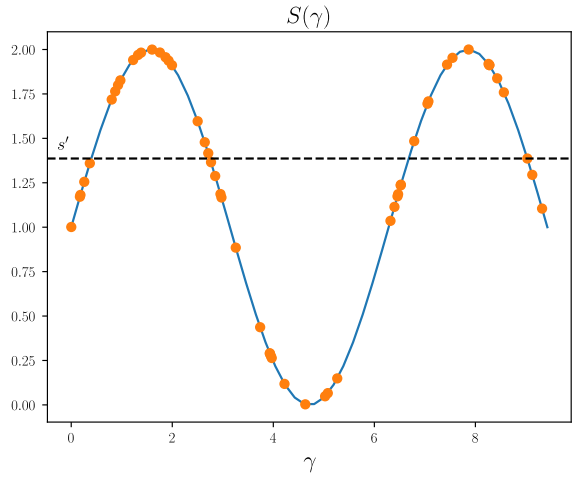

Figure 2.13 – Exemple d'une fonction objectif $S(\boldsymbol{\gamma})$. En orange, nous retrouvons les points associés aux 55 valeurs d'hyperparamètres évaluées aléatoirement. En pointillé, nous pouvons voir s', la médiane des scores observés jusqu'à présent.

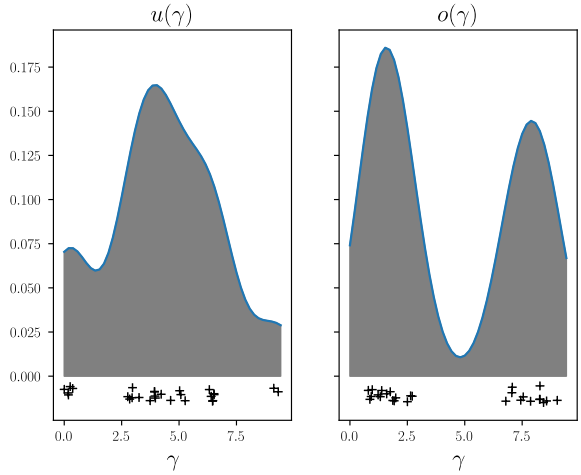

Figure 2.14 – Illustration des estimateurs de densités à noyaux gaussiens obtenus à partir des évaluations effectuées par l'algorithme EPA.

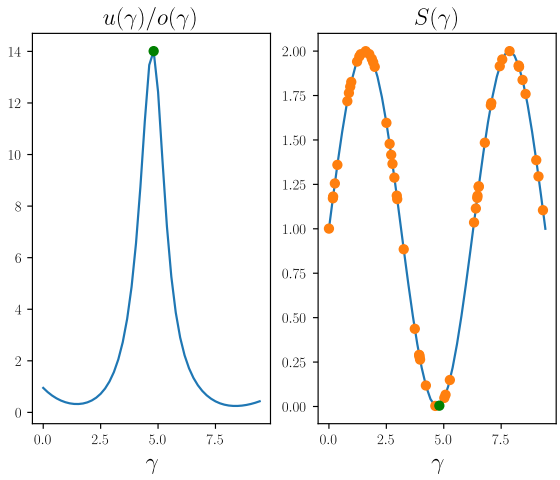

Figure 2.15 – Échantillonnage de la prochaine valeur d'hyperparpamètre avec l'algorithme EPA. Dans cet exemple, la valeur $\boldsymbol{\gamma} \approx 4.81$ maximise le ratio $u(\boldsymbol{\gamma})/o(\boldsymbol{\gamma})$. Dans la figure de droite, nous pouvons voir, en vert, le score associé à cette valeur sur la courbe $S(\boldsymbol{\gamma})$.

Chapitre 3

Stratégies d'apprentissage automatique pour la prédiction d'effets tardifs liés aux traitements de la leucémie aiguë lymphoblastique infantile

Auteur.e.s

Nicolas Raymond	Université de Sherbrooke
Maxime Caru	Penn State College of Medicine
Hakima Laribi	Université de Sherbrooke
Mehdi Mitiche	Université de Sherbrooke
Valérie Marcil	Université de Montréal
Maja Krajinovic	Université de Montréal
Daniel Curnier	Université de Montréal
Daniel Sinnett	Université de Montréal
Martin Vallières	Université de Sherbrooke

Résumé

La leucémie aiguë lymphoblastique (LAL) est le cancer le plus fréquemment diagnostiqué chez l'enfant. Approximativement deux tiers des survivants de la LAL infantile présentent une ou plusieurs complications de santé à l'âge adulte. Connues sous le nom d'effets tardifs, ces complications sont plutôt le fruit du traitement que de la maladie elle-même. Les mesures actuellement mises en place pour les visites de suivi post-traitement sont généralement uniformes à l'ensemble des survivants de cancers infantiles et ne sont pas nécessairement adaptées précisément aux survivants de la LAL infantile. Conséquemment, les effets tardifs peuvent être sous-diagnostiqués et, dans la plupart des cas, seulement pris en charge après leurs apparitions. Ainsi, il est nécessaire de prédire l'apparition des effets tardifs plus tôt pour contribuer à la santé et au bien-être des survivants. Plusieurs travaux se sont concentrés sur la recherche de biomarqueurs pouvant aider à la prédiction des effets tardifs et, notamment, un article a mis de l'avant l'utilisation d'un modèle d'apprentissage automatique pour prévenir les effets liés à la détérioration de la forme cardio-respiratoire. Toutefois, aucune solution n'a fait usage de réseaux neuronaux jusqu'à présent. Dans ce projet de recherche, nous avons développé des réseaux de neurones graphiques efficients et mis en valeur leur interprétabilité à l'aide de multiples analyses conduites suite à leurs entraînements. En premier lieu, nous avons proposé un nouveau modèle d'estimation de la consommation d'oxygène maximale (c.-à-d.,VO_2 max) qui ne nécessite aucune participation à un test physique (e.g., test de marche de six minutes). En second lieu, nous avons développé un modèle de prédiction de l'obésité utilisant des variables cliniques disponibles dès la fin du traitement de la LAL infantile, ainsi que plusieurs marqueurs génétiques (c.-à-d., polymorphismes à un seul nucléotide). Les solutions mises en place ont permis d'obtenir de meilleures performances que d'autres modèles à structures arborescentes ou neuronales sur de petits ensembles de patients (≤ 223) pour les deux tâches.

Présentation de l'article

L'article associé au présent projet de recherche a été soumis le 30 novembre 2022 pour publication au journal *Communications Medicine*. Celui-ci, intitulé *Machine learning strategies to predict late adverse effects in childhood acute lymphoblastic leukemia survivors*, aborde le problème du développement de séquelles liées au traitement de la leucémie aigüe lymphoblastique (LAL) infantile. Il présente d'abord un modèle estimant la consommation d'oxygène maximale (i.e., VO_2 max) d'un survivant de la LAL infantile. Le VO_2 constitue la meilleure mesure de la forme cardio-respiratoire qui à son tour, est un indicateur de la prédisposition à certaines morbidités (p. ex., obésité, dépression). Par la suite, l'article met de l'avant un modèle de prédiction du pourcentage de graisse corporelle. Celui-ci, faisant usage de données multi-omiques, est proposé pour la prédiction de l'obésité chez les survivants de la LAL infantile directement à la fin de leur traitement. L'article a pour objectif de présenter des outils de suivi hautement personnalisés s'appuyant sur des modèles prédictifs intégrant les caractéristiques des patients.

Contributions des auteur.e.s

Nicolas Raymond et Martin Vallières on élaboré l'étude. Nicolas Raymond a rédigé l'article et effectué l'ensemble des analyses de données. Nicolas Raymond et Mehdi Mitiche ont implémenté le logiciel d'apprentissage automatique permettant de réaliser les analyses de données. Hakima Laribi a généré les jeux de données aléatoires permettant de valider la reproductibilité des expériences. Martin Vallières a supervisé l'étude. Maxime Caru, Valérie Marcil, Maja Krajinovic, Daniel Curnier et Daniel Sinnett ont contribué à la conception des expériences. Tous les auteurs ont révisé le manuscrit.

3.1 Introduction

Childhood acute lymphoblastic leukemia (ALL) is the most frequently diagnosed type of cancer in children [36]. The 5-year relative survival rate is currently above 90% [24]. Nevertheless, approximately two thirds of childhood ALL survivors will present one or more health complications [60] known as late adverse effects (LAEs). The LAEs are rather resulting from the treatment (e.g., exposure to chemotherapy, cranial radiation therapy) than the cancer itself [60]. The existing follow-up measures, used in clinical settings and offered to patients during their visits to the hospital, are rather standardized for all childhood cancer survivors and not necessarily personalized for childhood ALL survivors [25]. As a results, LAEs may be underdiagnosed, and in most cases, only taken care of once they have already appeared in adulthood. Thus, it is necessary to predict these treatments related conditions earlier in order to prevent them and enhance the survivors' health.

Between 2013 and 2016, 246 childhood ALL survivors have participated to a series of clinical, physiological, biological and genetic evaluations as part of the PETALE study [47]. The main goal was to pinpoint predictive clinical, genetic and biochemical biomarkers that are relevant to establish personalized intervention plans to reduce LAEs prevalence, while providing knowledge for the improvement of follow-up methods [47].

Using the valuable data acquired from the PETALE cohort, efforts have been made towards the development of better personalized follow-up methods [10, 12, 18, 37, 49, 57]. As an example, an equation based on a linear regression was specifically developed to estimate the maximal oxygen consumption (i.e., VO_2 peak) in childhood ALL survivors following a 6-minute walk test (6MWT) [37]. The VO_2 peak is an excellent predictor of cardiac health in patients with cancer and is recognized as the gold standard in exercise physiology to measure patients' cardiorespiratory fitness [74], which plays an important role towards the prevention of LAEs in childhood ALL survivors. However, the direct measurement of the VO_2 peak, which is usually done by performing a maximal cardiopulmonary exercise test (CPET), is not an optimal solution in clinical settings due to financial and time constraints. Therefore, there is an interest in using a walking test (e.g., 6MWT) when access to comprehensive testing

is limited (e.g., CPET) [50]. Moreover, it has been shown that using a disease-specific VO_2 peak equation from the 6MWT provides a robust tool to estimate the patient's cardiorespiratory fitness with lower costs [50].

More recently, it has also been suggested that childhood ALL survivors' cardiorespiratory fitness is associated with specific trainability genes [10], highlighting the potential impact of some genetic variants in the prediction of VO_2 peak. Another study investigated the association between genetic variants (i.e., single nucleotide polymorphisms) and cardiometabolic LAEs (e.g., obesity, dyslipidemia, hypertension) in childhood ALL survivors [18]. The single nucleotide polymorphisms (SNPs) were grouped according to their associated cardiometabolic conditions and further analyzed with eight other biological and treatment-related variables using a logistic regression. The authors found that multiple common and rare variants were independently associated with cardiometabolic conditions, such as dyslipidemia, insulin resistance and obesity [18]. They also suggested that these associations should be considered as indicators for the early assessment of these LAEs. This is an important aspect to take into consideration since, in the PETALE study, 41.8% of childhood ALL survivors had dyslipidemia, 33% were obese and 18.5% had an insulin resistance [18].

In medical contexts, simple models such as linear regression and logistic regression are often favored over more complex machine learning approaches (e.g., deep learning models) due to their ability of being easily interpreted [22, 42]. Moreover, due to their modest number of parameters to optimize (i.e., reduced capacity), simple models are less inclined to overfit on small training datasets and therefore, have the ability to generalize beyond training samples. Hence, these models are well adapted to a clinical context with a small cohort of patients. However, more sophisticated model architectures (i.e., neural networks) have lately achieved better results in the prediction of clinical events using data from electronic health records [11, 46]. Interpretability of neural networks have also been the subject of many studies over the last years [22, 81]. Post-hoc methods have been investigated to get insights about a neural network's behavior following its training. For example, recent work motivated the usage of attention mechanisms within their models to help depict the decision-making process behind individual samples [3, 46]. Model-agnostic techniques exist as well to compare and visualize features within a layer of a neural network [22, 51, 52]. On the

other hand, interpretability of neural networks can also be strengthened a priori via the design of their architectures by including components with specific functionalities [22]. In particular, some studies explicitly integrated graph-based architectures (i.e., graph neural networks) to leverage the importance of the similarity between patients to solve a prediction task [43, 44].

In this work, our main goal was to design neural networks for the prevention of LAES in childhood ALL survivors population. An overview of the prediction tasks and the experimental setup considered in this study is presented in Figure 3.1. We hypothesized that parameters-efficient neural networks could achieve better prediction performance than linear and tree-based models on small cohorts of patients but should require rigorous post-hoc analyses to provide interpretability of their behaviors. Especially, we believed that graph-based architectures would lead to the best results since they can benefit from the links between patients of the cohorts instead of treating each of them separately. We also suggested that the inclusion of genomic variables would be beneficial in the creation of early LAEs prediction model. Towards our goal, we first proposed a new disease-specific VO_2 peak prediction model that does not require patients to participate to a physical function test (e.g., 6MWT); even if it has some advantages over the cardiopulmonary exercise test, the 6MWT still requires time and human resources. We further created an obesity prediction model using clinical variables that are available from the end of childhood ALL treatment as well as genomic variables. Overall, our results suggest that neural networks can outperform simple models to predict LAEs in small cohorts of ALL survivors. The conducted post-hoc analyses including the visualization of features within specific layers and the visualization of attention maps also demonstrated their usefulness to provide a general understanding of the behaviors of the models.

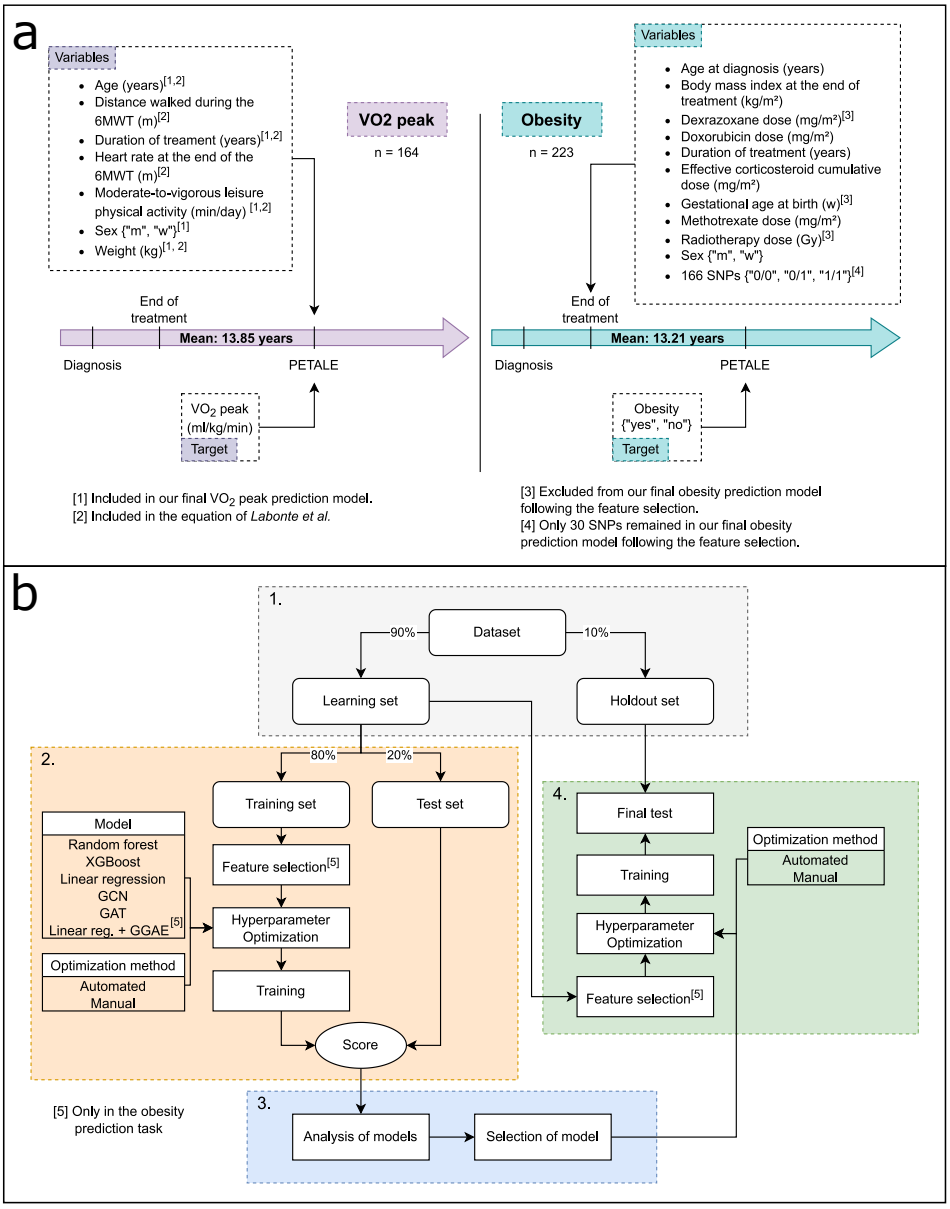

Figure 3.1 – Experimental setup. (**a**) Prediction tasks viewed using childhood ALL treatment timeline. On the left, VO_2 peak is predicted using variables measured on the same day. On the right, obesity is predicted using variables available at the end of childhood ALL treatment. (**b**) Experiment workflow. (1) Separation of the dataset into a *learning set* and an *holdout set*. (2) Evaluation of the models using random stratified subsampling with 10 splits. (3) Comparison of the models. (4) Final evaluation of the selected model on the *holdout set*. Additional details are available in Section 3.4.2

.

3.2 Results

3.2.1 Modeling the maximal oxygen consumption

Construction of the prediction model

We developed a new regression model to estimate the VO_2 peak (mL/kg/min) in childhood ALL survivors. As opposed to the equation from Labonté *et al.* [37], presented in Supplementary Background, we omitted the usage of variables related to the 6MWT but included the sex variable considering that it has an impact on the VO_2 peak [73]. Overall, we considered the age (years), the duration of treatment (DT) (years), the moderate-to-vigorous leisure physical activity (MVLPA) (min/day), the sex and the weight (kg) as our observed variables (Figure 3.1**a**).

Recent works proposing the usage of Graph Neural Networks (GNNs) to improve prediction performance on datasets with no pre-established graph structure [15, 20, 69] motivated us to create an oriented graph from our dataset and build a model by combining the Jumping Knowledge Networks framework [80] to a Graph Attention Network (GAT) [76]. Instead of considering survivors individually, our model captures information from their neighborhood (i.e., the set of survivors connected to them by an oriented edge) when it calculates their predictions. Precisely, a targeted survivor encapsulates the information from his surrounding by calculating a weighted average of his neighbors' standardized features and by applying a transformation to the resulting vector. The weight attributed to each neighbor during the calculation of the weighted average is determined by the attention mechanism of the GAT. Both the attention mechanism and the transformation are parameterized functions for which the parameters are learnt during the training of the model. The vector resulting from the transformation is then concatenated to the initial standardized features to create an enriched representation of the survivor (i.e., an embedding). It follows that the VO_2 peak of a survivor is estimated with a linear combination of the components within his embedding.

Construction of the graph

We created the oriented graph structure connecting the childhood ALL survivors of our dataset using their attributes. To guide the attention mechanism towards a subset of connections with intuitively more potential to help with the regression task, we restricted the number of oriented edges pointing at each survivor (i.e., node) by selecting only their 10 nearest neighbors of the same sex. The similarity between survivors was determined using the Euclidean distance based on the numerical features. A self-connection was also added to each node so they could be part of their own neighborhood. The survivors from the *holdout set* (Figure 3.1**b**) were not allowed to be connected to each other in order for our experiment to be representative of a real clinical context where each new incoming survivor can only be connected to others that have already been observed (i.e., that are already in the graph).

Performance of the prediction model

We compared our new VO_2 peak prediction model to the equation from Labonté *et al.* [37] by measuring the root-mean-square error (RMSE), the mean absolute error (MAE), the Pearson correlation coefficient (PCC) and the concordance index (C-index) associated to the predictions of both models in the *holdout set*. Except for the concordance index, our model shows an improvement against the equation from Labonté *et al.* [37] (*final test* section of Table 3.1). In Figure 3.2, we can see that the equation from Labonté *et al.* [37] is overestimating the VO_2 peak of childhood ALL survivors while our model is closer to the real observed values (i.e., targets). Moreover, our model does not rely on any measurement acquired from a 6MWT.

The results that led to the selection of our final model (i.e., GAT) are presented in the top section (i.e., *evaluation of models*) of Table 3.1. The *evaluation of models* consists of the second phase of our experimental setup (Figure 3.1**b** box 2), in which a random forest, XGBoost [13], a linear regression, a multi-layer Perceptron (MLP) and two GNNs (GCN [34] and GAT [76]) combined with the Jumping Knowledge Networks framework [80] are compared against each other. Additional results associated to this experiment phase are available in Supplementary Table A.13. Further details about the evaluation procedure and the description of the models are available

in *Experimental setup* and *Models* subsections of Methods section respectively.

		Metric				
Experiment phase	Model	RMSE	MAE	PCC	C-index	HPs optimization
Evaluation of models (*learning set*)	**Labonté *et al.* [37]**	7.95 ± 0.63	6.59 ± 0.56	0.77 ± 0.05	0.79 ± 0.03	-
	Random forest	5.44 ± 0.74	4.28 ± 0.54	0.79 ± 0.05	**0.81 ± 0.02**	Manual
	XGBoost	5.39 ± 0.86	**4.12 ± 0.68**	0.79 ± 0.06	0.81 ± 0.04	Automated
	Linear regression	5.38 ± 0.70	4.22 ± 0.57	0.79 ± 0.07	0.80 ± 0.04	Automated
	MLP	5.76 ± 0.74	4.42 ± 0.54	0.77 ± 0.06	0.80 ± 0.03	Automated
	GCN	5.42 ± 0.86	4.14 ± 0.60	0.79 ± 0.07	0.81 ± 0.03	Manual
	GAT	**5.34 ± 0.80**	4.13 ± 0.59	**0.80 ± 0.07**	0.81 ± 0.03	Manual
Final test (*holdout set*)	**Labonté *et al.* [37]**	8.98	7.84	0.47	**0.72**	-
	GAT	**6.51**	**5.20**	**0.52**	0.70	Manual

Table 3.1 – Performance of the models for the VO_2 peak prediction task. Models are compared using the root-mean-square error (RMSE), the mean absolute error (MAE), the Pearson correlation coefficient (PCC) and the concordance index (C-index). The results reported in the top section (i.e., *evaluation of models*) refer to the *mean* ± *standard deviation* obtained in the second part of our experimental setup (Figure 3.1**b** box 2). The scores recorded following the predictions made by the selected model (i.e., GAT) and the equation from Labonté *et al.* [37] in the *holdout set* are presented in the *final test* section. The *HPs optimization* column indicates if the scores were acquired with hyperparameter values that were manually selected or found by an automated hyperparameter optimization algorithm. **HP**: hyperparameter.

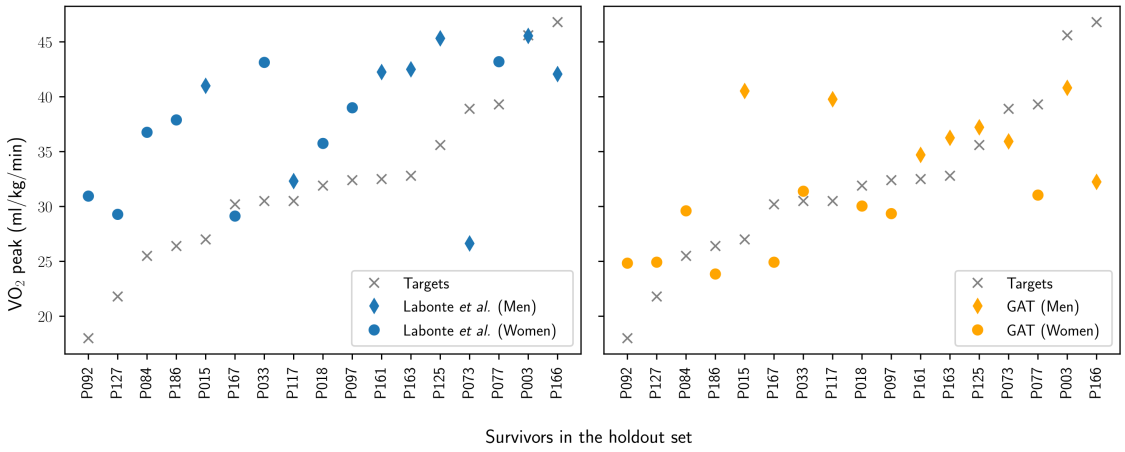

Figure 3.2 – Predictions of the equation from Labonté *et al.* [37] and the new VO_2 peak prediction model in the *holdout set*. On the left, the last established equation overestimates the VO_2 peak of the survivors. On the right, the new model based on a GAT architecture provides predictions that are closer to the targets.

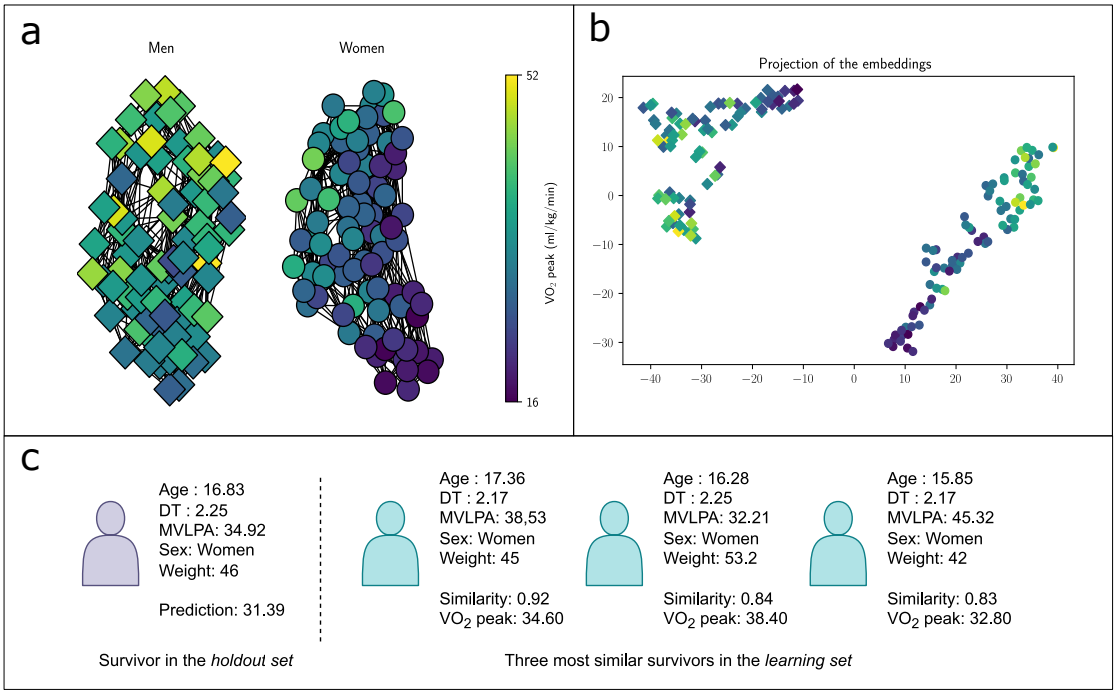

Figure 3.3 – Analysis of the VO_2 peak model. (**a**) Graph constructed for the VO_2 peak regression task. Connected men (diamonds) and women (circles) present similar VO_2 peak levels. The self-connections were omitted for visualisation purpose. (**b**) Projection of the embeddings in a 2D space using t-SNE [51]. Men and women present two distinct groups where close survivors usually share similar VO_2 peaks. (**c**) Comparison between the profile of a survivor in the *holdout set* and the associated three most similar survivors in the *learning set* according to the embeddings learnt by the model.

Analysis of the prediction model

We first visualized the oriented graph structure connecting the childhood ALL survivors of our dataset (Figure 3.3**a**). The resulting graph showed that survivors sharing connections had similar VO_2 peak values while supporting the fact that women in the childhood ALL survivors population have generally lower VO_2 peak values than men, as it is already observed in the non-survivors population [73].

We further projected the embeddings learnt by our model in a 2D space using t-SNE [51] in order to have a better understanding of the model's behavior (Figure 3.3**b**). The projection suggests that our model can learn embeddings that group survivors of the same sex while generally keeping them closer when they share similar VO_2 peak values. Considering that a shorter distance between two survivors' embed-

dings generally means that they have closer VO_2 peak values, we can help a clinician to validate the potential of a new prediction made by our model by comparing the profile of the survivor associated to the predicted value to the profiles of the associated most similar survivors for which we already know the VO_2 peak values. For example, in Figure 3.3**c**, we compared a survivor in the *holdout set* with the associated three most similar survivors in the *learning set*. In this case, the prediction made by the model seems legitimate since the closest patients in the *learning set* have comparable attributes and VO_2 peak values. Note that the similarity measure in Figure 3.3**c** is based on a weighted Euclidean distance between the embeddings, that is, $similarity = 1/(1 + distance)$. The weight attributed to each dimension of the embeddings during the Euclidean distance calculation corresponds to the absolute value of the weight associated to the same dimension in the last linear layer of the model.

3.2.2 Modeling the obesity

Construction of the prediction model

We developed a new model for the early obesity prediction in childhood ALL survivors population. To make our model available to use directly at the end of the childhood ALL treatment, we considered the age at diagnosis (years), the body mass index (BMI) at the end of treatment (kg/m^2), the doxorubicin dose received (mg/m^2), the duration of treatment (DT) (years), the effective corticosteroid cumulative dose received (mg/m^2), the methotrexate dose received (mg/m^2), the sex and 30 SNPs as our observed variables (Figure 3.1**a**). The SNPs are categorical variables that share the three following modalities: homozygous for the reference allele ("0/0"), heterozygous (i.e., one chromosome with the reference allele and the other with the alternate) ("0/1") or homozygous for the alternate allele ("1/1"). All variables mentioned above were kept following a feature selection process (see *Feature selection* in Methods and Figure 3.1**b**-4). The non-genomic variables excluded are shown in Supplementary Tables A.4-A.9.

The obesity status of a single individual can vary according to the measure used (e.g., body mass index (BMI), total body fat percentage (TBF), waist circumference)

and the specific cut-off value associated to it, which can evolve according to guidelines. Thus, we trained our model to directly predict the future TBF of survivors. This way, any cut-off value can be further applied to evaluate if a survivor will be obese or not based on the predicted value. It allows our model to be independent from any cut-off value and consequently ensures that it stays operational with time. In our dataset, the time elapsed between the end of the treatment and the measurement of the TBF was on average 13.21 years (Figure 3.1**a**).

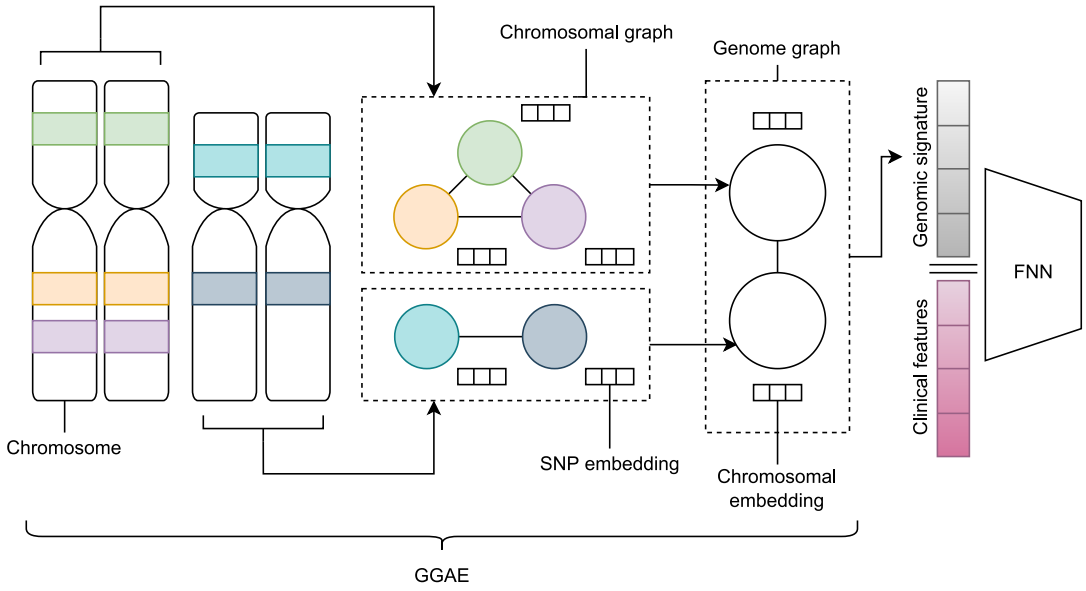

Figure 3.4 – The Gene Graph Attention Encoder (GGAE). In order to create the *genomic signature*, the GGAE first interprets each chromosome pair as a complete graph (i.e., *chromosomal graph*) where the nodes represent the observed SNPs associated to the pair. A real-valued vector is mapped to each node of each *chromosomal graph* according to the SNP's category linked to the node (i.e., "0/0", "1/1" or "0/1"). We refer to each of these vectors as *SNP embedding*. The GGAE further depict the whole genome as another complete graph (i.e., the *genome graph*) where each node represents a pair of chromosomes. A real-valued vector (i.e., a *chromosomal embedding*) is mapped again to each of these nodes by aggregating the *SNP embeddings* of the associated *chromosomal graph* (i.e., applying a readout function). Another readout function is finally applied to the *genome graph* to create the *genomic signature*.

Gene Graph Attention Encoder

We created a novel neural network architecture that efficiently uses the genomic data (i.e., the SNPs) for a regression task while considering the underlying struc-

ture of the genome. This new architecture called the Gene Graph Attention Encoder (GGAE) (Figure 3.4) encodes the data from the SNPs in a low-dimensional vector named the *genomic signature*. The latter is further concatenated to the other standardized clinical features to create a given patient embedding that can be used as an input to any feedforward neural network (FNN) architecture. The parameters of the GGAE and the subsequent connected FNN architecture are learned in an *end-to-end* fashion during the training of the model. It follows that the model generates *genomic signatures* that are specific to the regression task. In our case, we used a simple linear regression (i.e., a linear layer) as the FNN part to reduce the number of parameters to optimize and allow the post-hoc analysis of the coefficients associated to the standardized clinical features. See section 2.2.6 of Supplementary Methods for further details on the architecture of the GGAE.

Performance of the prediction model

We compared the best model obtained with the inclusion of SNPs (linear regression + GGAE) to the best model obtained without the SNPs (linear regression) according to four different regression metrics (RMSE, MAE, PCC and C-index) and two binary classification metrics (sensitivity and specificity) (Table 3.2 *final test* sections). The sensitivity and the specificity of the models were calculated following the categorization of the survivors in the *holdout set* as obese or not obese considering both the real TBFs and the model predictions with the cut-off values presented by Lemay *et al.* [36]: >25% (men), >35% (women) and >95th percentile (children). More precisely, for any child, the cut-off value was the 95th percentile measured from a sample of U.S children of the same sex and age group [62]. The combination of the linear regression with the GGAE achieved the best scores on all metrics except for the specificity since it misclassified a non-obese survivor (P226) by 0.29 percentage point (Figure 3.5). Additional results associated to the *evaluation of models* phase mentioned in the top sections of Table 3.2 are available in Supplementary Tables A.14-A.15.

3.2. Results

Experiment phase	Model	Metric						HPs optimization
		RMSE	MAE	PCC	C-index	Sensitivity	Specificity	
Evaluation of models (w/o SNPs) (*learning set*)	**Random forest**	8.97 ± 0.53	7.36 ± 0.46	0.65 ± 0.04	0.73 ± 0.02	**0.80 ± 0.06**	0.71 ± 0.12	Manual
	XGBoost	9.00 ± 0.50	**7.27 ± 0.41**	0.65 ± 0.04	0.74 ± 0.02	0.77 ± 0.10	0.71 ± 0.13	Automated
	Linear regression	**8.91 ± 0.52**	7.36 ± 0.45	**0.66 ± 0.05**	**0.75 ± 0.03**	0.77 ± 0.07	0.70 ± 0.11	Automated
	MLP	9.07 ± 0.59	7.27 ± 0.48	0.64 ± 0.07	0.74 ± 0.04	0.68 ± 0.10	**0.76 ± 0.10**	Manual
	GCN	9.03 ± 0.61	7.47 ± 0.53	0.65 ± 0.04	0.74 ± 0.02	0.76 ± 0.12	0.75 ± 0.10	Manual
	GAT	9.03 ± 0.52	7.44 ± 0.52	0.64 ± 0.04	0.73 ± 0.02	0.72 ± 0.09	0.75 ± 0.07	Manual
Evaluation of models (w/ SNPs) (*learning set*)	**Random forest**	9.61 ± 0.71	7.84 ± 0.67	0.61 ± 0.04	0.72 ± 0.02	**0.79 ± 0.09**	0.68 ± 0.10	Automated
	XGBoost	9.27 ± 0.58	7.54 ± 0.47	0.62 ± 0.04	0.72 ± 0.02	0.72 ± 0.12	0.70 ± 0.10	Automated
	Linear regression	9.56 ± 0.76	7.95 ± 0.67	0.58 ± 0.08	0.70 ± 0.03	0.71 ± 0.11	0.69 ± 0.08	Automated
	MLP	9.73 ± 1.08	7.80 ± 0.83	0.60 ± 0.09	0.72 ± 0.04	0.68 ± 0.11	**0.74 ± 0.08**	Automated
	GCN	9.82 ± 0.86	7.92 ± 0.75	0.58 ± 0.05	0.70 ± 0.03	0.65 ± 0.12	0.68 ± 0.14	Automated
	GAT	9.64 ± 0.67	7.80 ± 0.63	0.59 ± 0.05	0.71 ± 0.02	0.72 ± 0.14	0.70 ± 0.11	Automated
	Lin. reg. + GGAE	**9.02 ± 0.57**	**7.47 ± 0.55**	**0.65 ± 0.05**	**0.74 ± 0.02**	0.77 ± 0.08	0.72 ± 0.12	Manual
Final test (w/o SNPs) (*holdout set*)	**Linear regression**	7.96	6.31	**0.67**	0.75	**0.83**	**0.88**	Manual
Final test (w/ SNPs) (*holdout set*)	**Lin. reg. + GGAE**	**7.87**	**6.27**	**0.67**	**0.76**	**0.83**	0.82	Manual

Table 3.2 – Performance of the models for the obesity prediction task. Regression metrics are calculated using the predictions of TBF. Classification metrics are calculated considering the obesity class (obese or not obese) associated to each prediction following the application of the cut-off values. The results reported in the *evaluation of models* sections refer to the *mean* ± *standard deviation* obtained in the second part of our experimental setup (Figure 3.1**b** box 2). The scores achieved by the selected model without SNPs (linear regression) and the selected model with SNPs (linear regression + GGAE), in the *holdout set*, are displayed in the *final test* sections. The *HPs optimization* column indicates if the scores were acquired with hyperparameter values that were manually selected or found by an automated hyperparameter optimization algorithm. **HP**: hyperparameter.

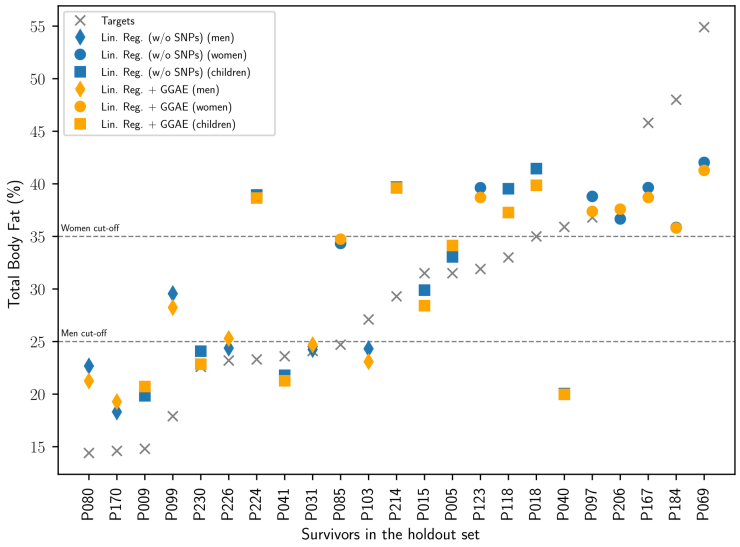

Figure 3.5 – Comparison of the linear regression with GGAE to the linear regression without the SNPs. The only obese man in the *holdout set* was not identified by our model. Nonetheless, all obese women in the *holdout set* were correctly identified.

Analysis of the prediction model

Attention mechanism

The attention mechanism in the GGAE demonstrated its capability to focus on different SNPs to make a prediction. The attention of the GGAE was mainly directed toward the SNPs 4:120241902, 12:48272895, 15:58838010, 16:88713262, 21:4432365 and 22:42486723 (Figure 3.6). The SNPs 4:120241902, 15:58838010, 16:88713262 and 21:4432365 had higher attention scores when they were homozygous for the reference allele. Finally, the attention scores given to SNPs 12:48272895 and 22:42486723 were high for any category.

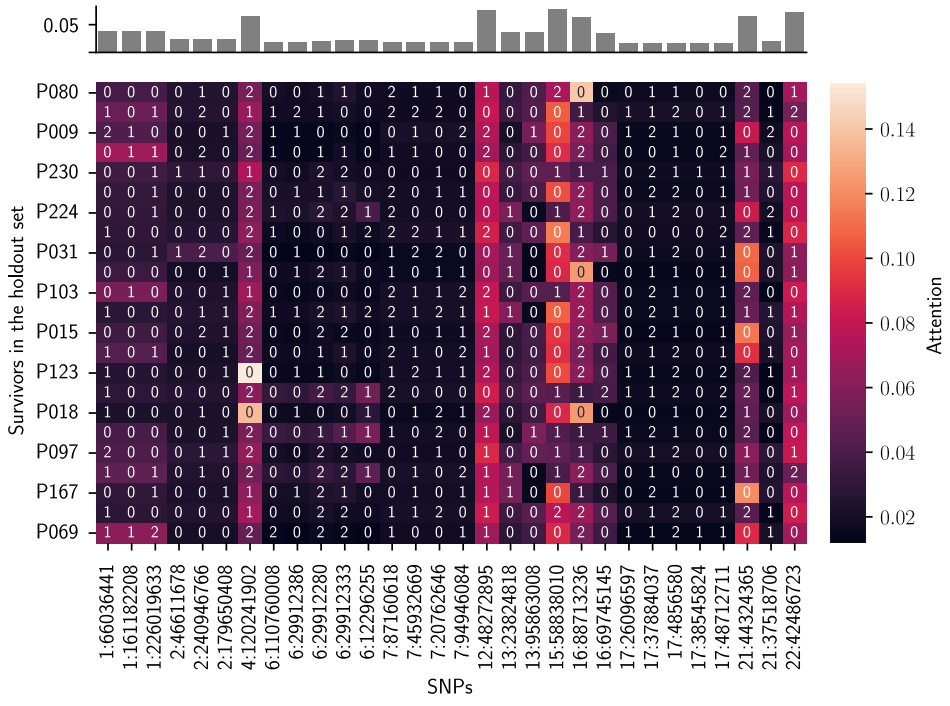

Figure 3.6 – SNPs attention heatmap. We display the attention score (in the range $[0, 1]$) given to each SNP by each survivor in the *holdout set*. The survivors are sorted from the lowest TBF to the highest. The attention scores in each row are summing to 1. Each part of the grid is annotated with a "0", a "1" or a "2" to show if the SNP of a survivor is respectively homozygous for the reference allele ("0/0"), heterozygous ("0/1") or homozygous for the alternate allele ("1/1"). A bar plot with the average attention score of each SNP is presented over the heatmap.

Impact of the clinical features

We analyzed the coefficients associated to the standardized clinical features and the intercepts related to each sex. The age at diagnosis (0.05), the BMI at the end of treatment (2.33), the doxorubicin dose received (0.65) and the effective corticosteroid cumulative dose received (0.44) were found to increase the prediction of the TBF. The obesity at the end of treatment was already identified as a prevalence factor of the obesity at the interview for the survivors in the PETALE cohort [35]. Additionally, corticosteroids have also been reported to increase the obesity risk in other studies [14, 77]. Therefore, it is legitimate that positive coefficients are associated to the BMI at the end of treatment and the cumulative corticosteroids dose received. On the other hand, the duration of treatment (-0.32) and the methotrexate dose (-1.56) were found to decrease the prediction of the TBF. The intercepts calculated for both sex (men: 16.31, women: 30.35) support the fact that women have generally higher TBF than men, which is common in the non-childhood ALL survivors [33].

3.3 Discussion

Over the years, efforts have been pursued towards the development of better personalized follow-up methods for childhood ALL survivors using data from the PETALE study [10, 12, 18, 37, 49, 57]. Other recent works have presented interesting results associated to the usage of neural networks in prediction tasks related to clinical contexts [11, 46]. However, until now, these machine learning approaches remained underexplored for the prediction of LAEs in childhood ALL survivors. In our work, we developed graph-based parameters-efficient neural networks for the LAEs prediction in childhood ALL survivors. In addition to contributing to precision medicine, our solutions constitute a promising avenue for the usage of artificial intelligence in clinical settings with restricted numbers of patients.

We first created a new disease-specific VO_2 peak prediction model based on a Graph Attention Network [76]. The VO_2 peak is the gold standard to measure the cardiorespiratory fitness [74], which in turn is a key element for the prevention of LAEs such as obesity, cholesterol and depression [36]. To use this type of neural network

architecture and handle the VO_2 peak prediction task as a node regression problem, we created a graph structure with our dataset (Fig. 3.3**a**). In addition to achieving better performance than the equation from Labonté *et al.* [37], our model does not rely on a walking test (e.g., 6MWT). The removal of this constraint represents a strong advantage in the context of healthcare considering that the 6MWT requires time and financial resources. Moreover, with our new model, the VO_2 peak prediction becomes more accessible since all variables needed by the model can be self-reported by the survivors. Therefore, our model could be associated to an online survey that survivors would be requested to fill at different time points. The resulting predictions could be further analysed by an exercise physiologist with the support of an interface providing comparisons between the current patient and the most similar survivors for which the VO_2 peak is already known (Fig. 3.3**c**). Even tough it obtained satisfying results, the VO_2 peak prediction model presented development challenges that should be further addressed in following works. The manual construction of an optimal graph structure represents a tedious task. Until now, we only explored solutions based on the calculation of distances between the observations of our dataset. However, even if these solutions are conceptually simple, they involve the selection of several additional hyperparameters such as the choice of a distance metric and the number of neighbors associated to each node. Machine learning methods enabling to simultaneously learn the graph structure relative to the data as well as the parameters of the model should be considered. Among the recent developments relevant to the subject, the *Graph Convolutional Transformer* [16] is a an example of model that simulates the presence of an edge between any pair of observations within a dataset and learn how to calculate a weight for each of them. In other words, this model features a flexible mechanism allowing each node to determine the best candidates to be part of its neighborhood. Whilst such model's complexity is growing according to the number of nodes in a graph, it is a plausible solution to consider in future work given the small cohort size enrolled in our study. In addition to simplifying the construction of the graph, this approach could lead to the improvement of our current state-of-the-art solution.

We also proposed an obesity prediction model using clinical variables that are available at the end of childhood ALL treatment as well as genomic variables. In addition to showing promising result in the prediction of future obesity, our work pre-

sented a novel neural network architecture (i.e., the GGAE) that efficiently encodes the information associated to the SNPs (Fig. 3.4). Its design improve the modeling of genomic data while allowing to manage the obesity prediction task similarly to a graph classification problem. The attention map produced by the GGAE (Fig. 3.6) demonstrates the degree of follow-up personnalization pursued by our work. Not only it allows to generate hypothesis about the importance of certain SNPs regarding the survivors population in general, but it also enables to see the contribution of the allelic constituents within each individual. For example, the map produced for the *holdout set* suggests that certain SNPs (i.e., 4:120241902, 12:48272895, 15:58838010, 16:88713262, 21:4432365 and 22:42486723) could be generally more relevant than the others for the prediction of the future TBF (Top of Fig. 3.6). Meanwhile, it also indicates on which SNPs the model was focusing the most according to each patient. Among others, we can mention the higher attention level given to the SNP 4:120241902 by patient P018 and P123 (Fig. 3.6). We acknowledge that the performance gain provided by the usage of SNPs through the GGAE was small (Fig. 3.5) and therefore, a study comparing the benefits and the cost of executing a whole exome sequencing should be conducted. Nonetheless, we consider the GGAE as an innovative solution for the integration of heterogeneous oncological data in parameters-efficient neural networks and we plan to further explore its potential in future work. It should be noted that the association between the SNP 12:48272895 (i.e., the VDR FokI polymorphism) and different obesity traits has already been investigated in multiple studies but results were found to be inconsistent [4].

We acknowledge that the performance gain provided by the usage of SNPs through the GGAE was small (Figure 3.5). Therefore, the usage of the GGAE stays relevant only if the genomic variables are already acquired during the childhood ALL treatment. The execution of a whole exome sequencing requires considerable resources and might not be worth the slight performance increase that we observed by including the SNPs in our model. Furthermore, the superiority of the GGAE was only observed in the *holdout set* (Table 3.2). Nonetheless, we consider the GGAE as an innovative solution for the integration of heterogeneous oncological data in parameters-efficient neural networks and we plan to further explore its potential in future work.

We further highlighted limitations related to our study. First, only a small num-

ber of samples were available in the PETALE dataset. Therefore, the scores obtained in the *holdout sets* related to both prediction tasks might not be fully representative of the future performance of our models on bigger and unseen datasets. Moreover, we hypothesize that the limited number of samples reduced the effectiveness of the automated hyperparameter optimization. More precisely, even though each set of hyperparameters selected by the algorithm was evaluated on multiple sub-samples, it is possible that their sizes were too small to provide valuable estimations of the hyperparameters' reliability. Second, all survivors of our datasets were from a monocentric cohort where individuals had only European origins. Hence, our findings may not translate to other ethnicity groups of childhood ALL survivors. Third, the concept of future is not clearly defined within the current design of the obesity prediction model. The time from the end of treatment could eventually be integrated as an additional variable to predict this LAE within a more precise time frame while potentially contributing to an increase of its accuracy [2].

Next, we believe that future work could be separated in three different phases: (i) the validation of the current work using an external dataset; (ii) the application of the current work to other LAEs; (iii) the development of new architectures based on multi-task learning [71]. During the validation phase, performance of the new models could be first tested on other cohorts of childhood ALL survivors with European origins. Tests could also be performed on cohorts with different ethnicity to acquire more information about the clinical settings in which our models are reliable. Additionally, investigation could be pursued concerning the possible association between the TBF and the SNPs that received higher attention scores by the GGAE. Except for the VDR FokI polymorphism, we did not find any work that reported relevant results regarding the link between these SNPs and the TBF. At last, investigations could also be conducted to further quantify the theoretical benefits of using GNNs with datasets that do not naturally have underlying graph structures. In the second phase, considering the promising results achieved in our work for the prediction of the VO_2 peak and the future TBF, the development of prediction models for other LAEs such as dyslipidemia and insulin resistance would be an interesting avenue to explore. In terms of the third phase, we hypothesize that, on small cohorts, neural networks designed for the simultaneous prediction of multiple LAEs via multi-task

learning could provide better performance than neural networks built to produce a single output. This hypothesis follows the intuition that a neural network trained to predict LAEs within a same family (e.g., metabolic disorders) should benefit from the common underlying patterns linking the variables to each specific LAE, while being less vulnerable to overfitting since it has to learn parameters that help conjointly for the different tasks.

In conclusion, we demonstrated in our work that graph-based parameters-efficient neural networks can achieve better results than linear and tree-based models for prediction tasks in clinical contexts with small cohorts ($\leq$ 223 childhood ALL survivors). We also showed that an improvement of regression performance can be leveraged from the creation of graph-based architectures by either connecting patients of a dataset together or providing a better modeling of their individual information (e.g., genomic data). Additionally, we displayed that it is feasible to have a better understanding of the behaviors of these more complex machine learning solutions with post-hoc analysis methods such as the visualization of patients' embeddings and the study of attention maps. Overall, we strongly believe that the design of efficient model architectures and the achievement of post-hoc analyses are the key to increase the progress and the trust associated to the usage of machine learning with small cohorts in healthcare.

3.4 Methods

3.4.1 Datasets

All data was taken from the PETALE study. All participants of this study were survivors with European origins who have been diagnosed for childhood ALL between 1987 and 2010 before the age of 19 and were at least 5 years post-diagnosis (see the article from Marcoux *et al.* [47] for a complete list of the eligibility criterion). Descriptive analyses of the datasets and procedures of their construction are presented in Supplementary Tables A.1-A.12 and Supplementary Figures A.1-A.2 respectively. The VO_2 peak dataset consisted of 164 survivors who reached a valid maximal oxygen consumption while performing a cardiopulmonary exercise test [37]. 90% of the survivors with a VO_2 peak under or equal to the median were women while 76% of the

survivors with a VO_2 peak over the median were men. The obesity dataset consisted of 223 survivors for which the TBF was measured by dual energy x-ray absorptiometry [36]. 76% of the survivors with a TBF under or equal to the median were men while 77% of the survivors with a TBF over the median were women.

3.4.2 Experimental setup

We developed a framework (Figure 3.1**b**) to compare the performance of different models (see Models) for the VO_2 peak and the obesity regression tasks. In this framework, 10% of the dataset is first extracted using random stratified sampling (see Random stratified sampling) to create a *holdout set*. The *holdout set* remains hidden until the final best model is selected and ready to be evaluated. The search of the best model is done using the 90% of data left, which is referred to as the *learning set*. The latter is divided 10 times into different training sets and test sets. Each test set is constituted of 20% of the *learning set* and is also extracted using random stratified sampling. For the early obesity problem, a selection of features is conducted on each of these data splits considering the data from the training set (see Feature selection) to exclude variables that are not helpful for the prediction.

Each model is evaluated on these 10 data splits at least twice. The models are first evaluated considering manually selected sets of hyperparameter values. Then, models are evaluated using a set of hyperparameter values obtained from an automated hyperparameter optimization algorithm (see Hyperparameter optimization). For each model evaluation, we save the empirical means and standard deviations of the metrics calculated on the test sets for further analyses. The model that achieved the best performance for the greatest number of metrics during one of its evaluations is kept as our final model. The best manually selected set of hyperparameter values, as well as the hyperparameters' search spaces used for the automated hyperparameter optimization of each model, are reported for each model in Supplementary Methods (Section A.2.3).

The selected model is finally trained and evaluated twice on the *learning set* and the *holdout set* respectively. The first training is done using the best manually selected set of hyperparameter values and the second training is done using automated

hyperparameter optimization. For the early obesity problem, a selection of features is conducted beforehand on the *learning set* to exclude variables that are not helpful for the prediction. The selected features are reported in Results.

Noteworthy, the model comparison step (Figure 3.1**b** box 2) was conducted twice for the early obesity prediction task. We have first selected the best model by running the comparisons without the SNPs and then selected the best model considering the SNPs. Both models were finally evaluated on the *holdout set* (Table 3.2).

3.4.3 Models

For each regression task and each set of variables tested, we evaluated the performance of a random forest, XGBoost [13], a linear regression (trained with gradient descent), a multi-layer Perceptron (MLP) and two GNNs (GCN [34] and GAT [76]) combined with the Jumping Knowledge Networks framework [80]. The linear regression with the GGAE was only evaluated for the obesity task with the set of variables including the SNPs. The random forest and the XGBoost implementations were taken respectively from *scikit-learn* [63] and *xgboost* libraries. The other models were implemented using *PyTorch* [64] and *DGL* [78] librairies. Each model had predetermined sets of values and search spaces associated to their hyperparameters (Section A.2.3). These sets were respectively used to execute the experiments without hyperparameter optimization and with hyperparameter optimization. The architectures of the models and the descriptions of their hyperparameters are available in Section A.2.2 and Section A.2.3 of Supplementary Methods.

3.4.4 Hyperparameter optimization

Hyperparameter optimization (Supplementary Figure A.3) was conducted by evaluating 200 sets of hyperparameter values sampled using the Tree-structured Parzen Estimator algorithm (TPE) [6]. Each set was evaluated on 10 *internal training sets* and *internal test sets* created by sub-dividing the training set (as well as the *learning set*) using stratified random sampling (see Random stratified sampling). The average RMSE observed in the 10 different *internal test sets* was used to estimate the performance related to a set of hyperparameter values. The set of hyperparameter

values associated to the lowest average RMSE was selected. All the hyperparameter optimization process was executed using the *optuna* [5] library. The settings of the TPE algorithm are reported in Supplementary Tables A.27-A.28.

3.4.5 Random stratified sampling

All test sets created (as well as the *holdout sets* and the *internal test sets*) were sampled using random stratified sampling. The stratification was performed each time on a temporary column combining sex and discretized versions of the targets (i.e., the VO_2 peak values or the TBFs depending on the regression task) based on the median in the complete dataset (i.e., the dataset before the extraction of the *holdout set*). The temporary column had four modalities: women ($\leq$median), women (>median), men ($\leq$median) and men (>median). The sex was considered in the stratification since we knew beforehand that it had an impact on the the VO_2 peak [73] and the TBF [33] in the non-childhood ALL survivors population.

Moreover, two additional criteria were established to verify if a test set (as well as a *holdout set* or an *internal test set*) was valid. The first criterion was that, for any test set sampled, the remaining dataset must contain any possible modality associated to the categorical variables. This criterion ensures that any categorical modality has been considered during the training of a model and can further be recognized during the evaluation of the same model on the test set. The second criterion was that the numerical values observed within any numerical column of the test set must not be further than 6 interquartile ranges away from the first and third quartiles of the same column in the remaining dataset. This criterion ensures that the numerical values in the test set lies in a similar region than the one see in the set used for the training.

3.4.6 Feature selection

For each training set (as well as each *learning set*), we trained 10 different random forests using the default hyperparameters of version 0.24.1 of *scikit-learn*. We further extracted the feature importance calculated by each random forest for each feature. All the features with an average feature importance greater or equal to 0.01 were kept for the training. The selection of the clinical features and the genomic features (i.e.,

the SNPs) was done independently.

3.4.7 Data imputation and transformation

For each pair of training and test sets created (as well as *learning/holdout* and *internal training/internal test* sets pairs), we imputed missing data in the numerical columns using the empirical means calculated with the observed data in the training set and imputed the missing data in the categorical columns using the modes of the observed data in the training set. Once imputed, transformation steps were applied to each pair of training and test sets. Numerical columns were reduced and centered using the empirical means and standard deviations calculated with the observed data in the training set. The modalities of each categorical column were changed for nominal encodings.

3.4.8 Graph construction

The directed graphs considered to train and evaluate the GNNs were built by considering the attributes of the survivors. Especially, the oriented edges pointing at each survivor (i.e., node), were coming from the nodes of the k-nearest neighbors of the same sex. During the *evaluation of models* phase (Figure 3.1**b** box 2), values of k of 4, 6, 8 and 10 were considered for the evaluation of each GNN model with manually selected hyperparameters (Supplementary Tables A.16-A.18). The value of k associated to the best performance of a GNN, with manually selected hyperparameters, was further used during the automated hyperparameter optimization of the same model (Supplementary Tables A.24-A.25). The similarity between each survivor was calculated on all the standardized numerical features and categorical features excluding the sex. More precisely, we used the value $1/(1 + \text{Euclidean distance})$ in cases where no categorical attributes were available and considered the cosine similarity otherwise. The categorical attributes were converted to one-hot encodings to calculate the cosine similarities. The similarity values were set as the weights of the edges for the GCN (Section A.2.2).

Survivors in the test sets (as well as the *holdout sets* and the *internal test sets*) were always excluded from each graph during the training of GNNs. Moreover, once added for the evaluation, survivors in the test sets were only allowed to have oriented edges coming from the nodes of the associated training graph. As an example, for each prediction task, during the execution of the second phase of the experimental setup (Figure 3.1**b** box 2), training graphs were only constituted of patients in the training set while patients of the associated test sets were only added in the graphs for the evaluations after the training.

3.4.9 Ethics declarations

All the analyses conducted for the PETALE study were compliant with the Declaration of Helsinki and approved by the Institutional Review Board of Sainte-Justine University Health Center. Written informed consent was obtained from study participants or parents/guardians.

3.5 Code Availability

All software code allowing to run the experiments used to produce all the results presented in this work is freely shared under the GNU General Public License v3.0 on the GitHub website at: https://github.com/Rayn2402/ThePetaleProject.

3.6 Data Availability

The datasets analysed during the current study are not publicly available for confidentiality purposes. However, randomly generated datasets with the same format as used in our experiments are publicly shared in our GitHub repository to test the code implemented for this work.

3.7 Acknowledgements

This work was supported by the Canadian Institutes of Health Research (CIHR), in collaboration with C17 Council, Canadian Cancer Society (CCS), Cancer Research Society (CRS), Garron Family Cancer Centre at the Hospital for Sick Children, Ontario Institute for Cancer Research (OICR) and Pediatric Oncology Group of Ontario (POGO). VM is supported by the Fonds de recherche Québec-Santé. DS holds the research chair FKV in pediatric oncogenomics. MV acknowledges funding from the CIFAR AI Chairs program.

3.8 Author contributions statement

NR and MV conceived the study. NR wrote the manuscript and performed data analysis. NR and MM implemented the machine learning framework used to perform data analysis. HL generated random datasets to validate experiment reproducibility. MV supervised the study. MC, VM, MK, DC and DS contributed to the experimental design. All authors revised the manuscript.

3.9 Competing interests statement

The author(s) declare no competing interests.

Conclusion

La leucémie aiguë lymphoblastique (LAL) constitue la majorité des cancers pédiatriques [39]. Malgré le succès des mesures de traitement actuelle, plusieurs enfants survivant à cette maladie présentent des complications de santé lors de l'âge adulte [60]. Celles-ci, constituant principalement des effets tardifs du traitement et non de la maladie elle-même, ont fait l'objet de plusieurs études. En particulier, plusieurs biomarqueurs, tels que le dosages de différents agents chimiothérapeutiques et la présence de certaines altérations génétiques, ont été mis de l'avant comme facteur d'influence dans le développement d'effets tardifs spécifiques, laissant place à des avenues d'innovations dans la prévention des effets tardifs et le suivi post-traitement des patients. Le projet de recherche présenté au sein de ce mémoire représente un effort supplémentaire dans l'amélioration des soins de suivi et de la qualité de vie des survivants. D'abord, un nouveau modèle d'estimation de la consommation d'oxygène maximale (c.-à-d.,VO_2 max) fut mis sur pieds. En plus d'offrir une plus grande précision que l'ancienne solution proposée par Labonté *et al.* [37], notre modèle ne nécessite pas la participation à un test physique et donc, est plus accessible et efficient. Le VO_2 max est reconnu comme la mesure de référence pour quantifier la forme cardio-respiratoire, qui à son tour, est un bon indicateur du risque de développement de certaines morbidités chez les survivants. Ensuite, un modèle fut développé pour prédire l'apparition future de l'obésité en utilisant des variables cliniques disponibles dès la fin du traitement de la LAL infantile, ainsi que plusieurs marqueurs génétiques. L'utilisation de données accessibles en début de suivi post-traitement constitue un avantage important considérant qu'un patient à risque d'obésité peut rapidement être détecté et pris en charge. Dans l'ensemble, il s'agit du premier travail proposant l'usage de réseaux de neurones pour prédire les effets tardifs du traitement de la LAL infantile.

Conclusion

En plus de contribuer directement au domaine de la médecine de précision en proposant des outils de suivi personnalisés, les modèles de réseaux de neurones graphiques et l'ensemble des modules d'expérimentations implémentés durant ce projet de maîtrise contribuent également à la recherche en apprentissage automatique. Premièrement, le modèle d'estimation du VO_2 max apporte des preuves empiriques supplémentaires à celles des travaux récemment publiés [15, 20, 69] concernant les bénéfices possibles de l'utilisation de réseaux de neurones graphiques avec des ensembles de données structurés manuellement sous forme de graphes. Deuxièmement, le modèle de prédiction de l'obésité présente une toute nouvelle architecture d'encodeur pour la modélisation de l'ensemble des polymorphismes à un seul nucléotide. Troisièmement, le logiciel libre développé durant ce projet constitue un outil flexible pouvant être réutilisé et modifié par la communauté scientifique pour réaliser d'autres expériences en apprentissage automatique. Un pipeline expérimental rigoureux, basé sur l'évaluation de modèles par sous-échantillonnage aléatoire stratifié, y est proposé ainsi que plusieurs modèles d'apprentissage supervisé à l'état de l'art. Notons en outre que, par notre implémentation du logiciel, nous encourageons la transparence dans la recherche scientifique en partageant des scripts permettant de reproduire facilement les expériences menées durant cette étude.

Malgré ses résultats satisfaisant, le modèle de réseau de neurones graphique élaboré pour la prédiction du VO_2 max soulève des limitations importantes qui motivent la réalisation de futurs travaux de recherche. Premièrement, il est fastidieux de déterminer manuellement une structure de graphe connectant les observations d'un ensemble de façon à réduire optimalement les pertes d'entraînement et de test. Jusqu'à présent, nous n'avons qu'exploré des solutions reposant sur le calcul de distances entre les observations. Bien qu'elles soient conceptuellement simple, ces méthodes entraînent la sélection de plusieurs hyperparamètres additionnels dont le choix de la métrique de distance, le nombre de voisins associés à chaque noeud et le choix des attributs sur lesquels calculer la distance. La prochaine étape consisterait donc à évaluer des méthodes d'apprentissage bout en bout où la structure du graphe est déterminée par descente de gradient durant l'entraînement du modèle. Notons particulièrement le modèle *Graph Convolutional Transformer* [16] qui assume la présence d'une arête entre chacune des observations et apprend à calculer un poids pour chacune de celles-ci. Chacun de ces

poids est obtenu par l'application d'une fonction de similarité paramétrique et différentiable sur une projection des attributs associés aux noeuds connectés par l'arête. Autrement dit, ce modèle dispose d'un mécanisme flexible permettant à chaque noeud de déterminer les noeuds avoisinants qui lui sont important pour générer sa représentation alternative optimale. Malgré sa complexité croissante en fonction du nombre de noeuds présents dans un graphe, cet algorithme est envisageable en raison de la petite taille de cohorte inscrite dans notre étude. Deuxièmement, le déploiement d'un réseau de neurones graphique convolutif (RNGC) présente potentiellement un risque quant à la confidentialité des patients utilisés pour l'entraînement du modèle. Plus précisément, pour effectuer une propagation avant, un RNGC nécessite l'ensemble des attributs des noeuds contenus dans le graphe. Ces informations, parfois sensibles, doivent donc être conservées sécuritairement avec le modèle, ce qui n'est pas nécessaire avec les autres algorithmes considérés lors de notre recherche. Le travail de Liu *et al.* [45] constitue une avenue de recherche intéressante pour remédier à ce problème. Celui-ci propose l'apprentissage d'un modèle auto supervisé visant à estimée la probabilité d'observer une certaine configuration d'attributs dans un noeud donné, sachant les attributs observés pour un noeud auquel il est connecté. Une fois entrainé sur la structure de graphe associée à un ensemble d'entraînement, ce modèle, représenté mathématiquement par $\mathbb{P}\left(\mathbf{x}_i \mid \mathbf{x}_j\right)$ pour i, j tel que $v_i \in \mathcal{N}_1(v_j)$, peut être utilisé pour générer des configurations d'attributs potentiellement observables dans le voisinage d'un noeud donné. En particulier, nous pouvons l'utiliser à titre de méthode d'augmentation de données sur les observations initiales pour entraîner des modèles non graphiques. Le graphe requis pour entraîner le modèle probabiliste pourrait être obtenu avec l'entraînement supervisé du modèle *Graph Convolutional Transformer* discuté plus tôt.

Le modèle de prédiction de l'obésité présente également des restrictions importantes qui laisse place à un potentiel d'amélioration. Jusqu'à présent, l'algorithme mis en place a pour objectif de produire une estimation du pourcentage de graisse corporelle à un temps t_1 à partir de variables observées à un temps t_0 tel que $t_1 > t_0$. Dans notre étude, t_0 et t_1 correspondent respectivement à la date de fin du traitement et la date de participation à l'étude PETALE de chaque patient. Néanmoins, alors que le temps depuis la fin du traitement a été ciblé comme variable d'intérêt dans

le développement de certains effets tardifs tels que les morbidités osseuses [2], nous avons omis d'intégrer la différence de temps $t_1 - t_0$ au sein de notre modèle. La prédiction émise par notre modèle réfère donc à un temps futur imprécis. Dans l'ensemble d'apprentissage (c.-à-d., *learning set*) lié à notre étude, les différences entre t_1 et t_0, en années, se trouvaient dans l'intervalle [3.30, 24.49], présentaient une moyenne de 13.21 et un écart type de 5.21. L'ajout de la variable de temps $t_1 - t_0$ constitue une voie de recherche qui présente un potentiel de valorisation important à notre projet. D'abord, un nouveau modèle incorporant cette variable permettrait d'inscrire un temps précis lors de l'évaluation d'un nouveau survivant de la LAL infantile. Ensuite, l'analyse du coefficient associé à cette variable permettrait d'émettre une hypothèse quant à l'impact du temps sur l'obésité.

Les travaux futurs envisageables pour le modèle de prédiction de l'obésité ne concernent pas uniquement la modélisation du problème lui-même, mais également le concept du GGAE (Section 3.2.2) employé pour l'encodage des polymorphismes à un seul nucléotide. Dans l'article présenté au Chapitre 3, l'architecture du GGAE a été proposée comme un outil de réduction de dimensionnalité dont les paramètres sont optimisés en fonction de la tâche à accomplir. Toutefois, la construction d'un modèle d'encodage, entraîné indépendamment du modèle de régression, a également été considéré durant la réalisation de ce projet. Particulièrement, celui-ci était entraîné simultanément à un modèle de décodage dans le but de produire des représentations vectorielles réduites du génome qui, d'une part, possèdent une distance faible lorsque deux patients présentent un génome similaire, et d'autre part, permettent au décodeur de reconstruire correctement les génomes originaux. Les définitions mathématiques de l'encodeur, le décodeur, et la perte d'entraînement auto supervisée sont présentées à l'Annexe B. Les résultats associés aux expériences effectuées avec ce modèle d'encodage n'ont pas été répertoriés dans l'article puisque leurs faiblesses témoignaient d'un besoin de recherche et d'améliorations supplémentaires. L'investigation plus approfondie de cette solution de réduction de dimensionnalité du génome reste pertinente considérant son potentiel d'être utilisée avec des modèles de régression et de classification qui ne sont pas nécessairement des réseaux neuronaux.

En plus de réaliser des projets axés sur l'amélioration des techniques d'apprentissage automatique déjà mises en place, il serait important de mener une étude de

l'utilité clinique des modèles proposés. Entre autres, nous pourrions confirmer ou infirmer si l'utilisation des différentes solutions développées durant cette étude est avantageuse aux recommandations du *Children's Oncology Group* [25]. Spécialement, bien que notre travail contribue à la recherche au niveau de l'utilisation des données génétiques des patients, il serait d'intérêt d'évaluer si les résultats apportés présentement par le modèle de prédiction de l'obésité justifient le séquençage de tous les enfants atteints de LAL.

L'exploration menée dans cette étude pour identifier des méthodes d'interprétabilité de réseaux neuronaux motive également la réalisation d'un projet de recherche indépendant à l'oncologie de précision. Plusieurs travaux mentionnent des techniques d'activation maximale [19, 58] visant à définir les valeurs d'entrées optimales à une architecture neuronale pour permettre l'atteinte d'une valeur de sortie précise. En particulier, ces méthodes sont communément employées au sein de problèmes de classification pour générer des images synthétiques maximisant la probabilité d'appartenance à une certaine classe. Nous croyons qu'il serait pertinent d'étudier la transposition de ces stratégies à des cadres de recherche clinique comportant des données tabulaires. Précisément, à partir d'un modèle entraîné, nous pourrions déterminer les profils de patients maximisant la prédiction de différentes cibles. Ce type de post-analyse n'a pas été élaboré durant notre projet considérant que les méthodes d'activation maximale ne s'appliquent actuellement qu'avec des variables d'entrées continues.

Pour conclure, le projet de recherche réalisé lors de cette maîtrise témoigne du potentiel associé à l'utilisation de stratégies d'apprentissage automatique pour améliorer le suivi post-traitement des survivants de la LAL infantile. Jusqu'à maintenant, les études réalisées à partir des données de la cohorte PETALE portaient principalement sur des analyses statistiques associatives visant à mettre en lumière l'impact général de certains biomarqueurs sur l'ensemble de la population de survivants. Motivée par le récent travail de Labonté *et al.* [37], notre approche permet plutôt la mise en place de directive de soins hautement personnalisées par l'intégration de variables de patients au sein de modèles prédictifs à l'état de l'art. En particulier, l'architecture de GGAE proposée pour intégrer les données génomiques individuelles des patients constitue une preuve considérable du niveau de personnalisation recherché par notre travail. Les résultats soulevés par notre étude concernant la prévention de l'obésité et

des problèmes de santés liés à l'affaiblissement de la forme cardio-respiratoire portent espoir pour le développement de nouveaux modèles d'apprentissage supervisé destinés à prédire d'autres effets tardifs du traitement de la LAL infantile. Nous croyons que les efforts de post-analyse mis en place pour mieux illustrer et décrire les comportements des réseaux de neurones contribuerons à une hausse de la confiance portée envers ces outils d'aide à la décision. Toutefois, nous admettons que des contributions additionnelles seront éventuellement nécessaires pour l'adoption courante des modèles d'apprentissage automatique dans la prise en charge des patients. En particulier, il est important de concentrer davantage d'effort de recherche sur le développement de méthodes pour cerner les patients non admissibles aux solutions implémentées, c'est-à-dire, les patients qui présentent des combinaisons d'attributs pour lesquelles un modèle n'est hypothétiquement pas en mesure d'émettre une prédiction juste. Ainsi, nous croyons fermement en l'utilisation de techniques d'apprentissage automatique pour assister les professionnels de la santé et non les remplacer. Alors que l'intelligence artificielle agit à titre de carte pour guider un médecin dans ses démarches, le savoir et les compétences de celui-ci restent essentiels pour assurer la prise des décisions les plus rationnelles en cas de contextes incertains ou sans précédent.

Bibliographie

[1] A. S. Anestin *et al.*, « Psychological risk in long-term survivors of childhood acute lymphoblastic leukemia and its association with functional health status : A PETALE cohort study, » *Pediatric Blood & Cancer*, vol. 65, no. 11, p. e27356, 2018.

[2] M. Aaron *et al.*, « Identification of a single-nucleotide polymorphism within *CDH2* gene associated with bone morbidity in childhood acute lymphoblastic leukemia survivors, » *Pharmacogenomics*, vol. 20, no. 6, pp. 409–420, 2019.

[3] S. Arık et T. Pfister, « Tabnet : Attentive interpretable tabular learning, » dans *AAAI*, vol. 35, 2021, pp. 6679–6687.

[4] B. Alathari, A. Sabta, C. Kalpana, et K. Vimaleswaran, « Vitamin D pathway-related gene polymorphisms and their association with metabolic diseases : A literature review, » *Journal of Diabetes & Metabolic Disorders*, vol. 19, no. 2, pp. 1701–1729, 2020.

[5] T. Akiba, S. Sano, T. Yanase, T. Ohta, et M. Koyama, « Optuna : A Next-generation Hyperparameter Optimization Framework, » dans *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623—-2631.

[6] J. Bergstra, R. Bardenet, Y. Bengio, et B. Kegl, « Algorithms for Hyper-Parameter Optimization, » *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[7] A. Boulet-Craig, P. Robaey, M. Krajinovic, C. Laverdière, D. Sinnett, S. Sultan, et S. Lippé, « Développement neurocognitif et cérébral des

survivants à long terme de la leucémie lymphoblastique aiguë, » *Revue québécoise de psychologie*, vol. 37, no. 2, pp. 43–63, 2017.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006, ch. 5.

[9] L. Breiman, « Random Forests, » *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] M. Caru *et al.*, « Identification of genetic association between cardiorespiratory fitness and the trainability genes in childhood acute lymphoblastic leukemia survivors, » *BMC Cancer*, vol. 19, no. 1, p. 443, 2019.

[11] E. Choi, M. Bahadori, A. Schuetz, S. WF., et J. Sun, « Doctor AI : Predicting Clinical Events via Recurrent Neural Networks, » dans *Proceedings of the 1st Machine Learning for Healthcare Conference*, 2016, pp. 301–318.

[12] M. Caubet F. *et al.*, « A Bayesian multivariate latent t-regression model for assessing the association between corticosteroid and cranial radiation exposures and cardiometabolic complications in survivors of childhood acute lymphoblastic leukemia : a PETALE study, » *BMC Medical Research Methodology*, vol. 19, no. 1, p. 100, 2019.

[13] T. Chen et C. Guestrin, « XGBoost : A Scalable Tree Boosting System, » dans *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 33, 2016, pp. 785—-794.

[14] E. Chow, C. Pihoker, K. Hunt, K. Wilkinson, et D. Friedman, « Obesity and hypertension among children after treatment for acute lymphoblastic leukemia, » *Cancer*, vol. 110, no. 10, pp. 2313–2320, 2007.

[15] Y. Chen, L. Wu, et M. Zaki, « Iterative Deep Graph Learning for Graph Neural Networks : Better and Robust Node Embeddings, » dans *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 19 314—-19 326.

[16] E. Choi, Z. Xu, Y. Li, M. Dusenberry, G. Flores, E. Xue, et A. Dai, « Learning the Graphical Structure of Electronic Health Records with

Graph Convolutional Transformer, » *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 606–613, 2020.

[17] S. R. Dubey, S. K. Singh, et B. B. Chaudhuri, « Activation functions in deep learning : A comprehensive survey and benchmark, » *Neurocomputing*, vol. 503, pp. 92–108, 2022.

[18] J. England *et al.*, « Genomic determinants of long-term cardiometabolic complications in childhood acute lymphoblastic leukemia survivors, » *BMC Cancer*, vol. 17, no. 1, p. 751, 2017.

[19] D. Erhan, Y. Bengio, A. Courville, et P. Vincent, « Visualizing higher-layer features of a deep network, » *Université de Montréal*, 2009, rapport technique, non publié.

[20] B. Fatemi et S. El A., L. amd Kazemi, « SLAPS : Self-Supervision Improves Structure Learning for Graph Neural Networks, » dans *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 22 667—-22 681.

[21] J. H. Friedman, « Greedy function approximation : a gradient boosting machine, » *Annals of statistics*, pp. 1189–1232, 2001.

[22] F. Fan, J. Xiong, M. Li, et G. Wang, « On Interpretability of Artificial Neural Networks : A Survey, » *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 6, pp. 741–760, 2021.

[23] I. Goodfellow, Y. Bengio, et A. Courville, *Deep Learning*. MIT Press, 2016, ch. 6.

[24] S. Hunger *et al.*, « Improved survival for children and adolescents with acute lymphoblastic leukemia between 1990 and 2005 : a report from the children's oncology group, » *Journal of Clinical Oncology*, vol. 30, no. 14, pp. 1663–1669, 2012.

[25] M. Hudson *et al.*, « Long-term Follow-up Care for Childhood, Adolescent, and Young Adult Cancer Survivors, » *Pediatrics*, vol. 148, no. 3, p. e2021053127, 2021.

[26] K. Hara, D. Saito, et H. Shouno, « Analysis of function of rectified linear unit used in deep learning, » dans *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–8.

[27] K. Hornik, M. Stinchcombe, et H. White, « Multilayer feedforward networks are universal approximators, » *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[28] T. Hastie, R. Tibshirani, et J. Friedman, *The Elements of Statistical Learning*. Springer New York, 2009.

[29] K. He, X. Zhang, S. Ren, et J. Sun, « Deep Residual Learning for Image Recognition, » dans *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[30] M. Krajinovic *et al.*, « Polymorphisms of ABCC5 and NOS3 genes influence doxorubicin cardiotoxicity in survivors of childhood acute lymphoblastic leukemia, » *The Pharmacogenomics Journal*, vol. 16, no. 6, pp. 530–535, 2016.

[31] D. P. Kingma et J. Ba, « Adam : A method for stochastic optimization, » dans *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[32] M. A. Khamsi et W. A. Kirk, *An introduction to metric spaces and fixed point theory*. John Wiley & Sons, 2011.

[33] K. Karastergiou, S. Smith, A. Greenberg, et S. Fried, « Sex differences in human adipose tissues – the biology of pear shape, » *Biology of Sex Differences*, vol. 3, p. 13, 2012.

[34] T. N. Kipf et M. Welling, « Semi-Supervised Classification with Graph Convolutional Networks, » dans *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[35] E. Levy *et al.*, « Cardiometabolic Risk Factors in Childhood, Adolescent and Young Adult Survivors of Acute Lymphoblastic Leukemia – A Petale Cohort, » *Scientific Reports*, vol. 7, no. 1, p. 17684, 2017.

[36] V. Lemay *et al.*, « Prevention of Long-term Adverse Health Outcomes With Cardiorespiratory Fitness and Physical Activity in Childhood

Acute Lymphoblastic Leukemia Survivors, » *Journal of Pediatric Hematology/Oncology*, vol. 41, no. 7, pp. e450–e458, 2019.

[37] L. Labonté *et al.*, « Developing and validating equations to predict VO_2 peak from the 6MWT in Childhood ALL Survivors, » *Disability and Rehabilitation*, pp. 1–8, 2020.

[38] V. Lemay *et al.*, « Physical Activity and Sedentary Behaviors in Childhood Acute Lymphoblastic Leukemia Survivors :, » *Journal of Pediatric Hematology/Oncology*, vol. 42, no. 1, pp. 53–60, 2020.

[39] Leukemia & Lymphoma Society, « Acute Lymphoblastic Leukemia (ALL) in Children and Teens, » 2021, rapport technique, non publié.

[40] Q. Li, Z. Han, et X.-M. Wu, « Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning, » dans *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[41] M. M. Li, K. Huang, et M. Zitnik, « Graph representation learning in biomedicine and healthcare, » *Nature Biomedical Engineering*, pp. 1–17, 2022.

[42] S. Lundberg et S. Lee, « A unified approach to interpreting model predictions, » dans *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4768—-4777.

[43] Z. Liu, X. Li, H. Peng, L. He, et P. Yu, « Heterogeneous similarity graph neural network on electronic health records, » dans *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 1196–1205.

[44] H. Lu et S. Uddin, « A weighted patient network-based framework for predicting chronic diseases using graph neural networks, » *Scientific Reports*, vol. 11, no. 1, p. 22607, 2021.

[45] S. Liu, R. Ying, H. Dong, L. Li, T. Xu, Y. Rong, P. Zhao, J. Huang, et D. Wu, « Local Augmentation for Graph Neural Networks, » dans *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, et S. Sabato, éditeurs, vol. 162, 2022, pp. 14 054–14 072.

[46] F. Ma *et al.*, « Dipole : Diagnosis Prediction in Healthcare via Attention-based Bidirectional Recurrent Neural Networks, » dans *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1903—-1911.

[47] S. Marcoux *et al.*, « The PETALE study : Late adverse effects and biomarkers in childhood acute lymphoblastic leukemia survivors, » *Pediatric Blood & Cancer*, vol. 64, no. 6, p. e26361, 2017.

[48] S. Marcoux *et al.*, « The PETALE study : Late adverse effects and biomarkers in childhood acute lymphoblastic leukemia survivors - Supplementary Table S1, » *Pediatric Blood & Cancer*, vol. 64, no. 6, p. e26361, 2017.

[49] S. Morel *et al.*, « Development and relative validation of a food frequency questionnaire for French-Canadian adolescent and young adult survivors of acute lymphoblastic leukemia, » *Nutrition Journal*, vol. 17, no. 1, p. 45, 2018.

[50] D. Mizrahi *et al.*, « The 6-minute walk test is a good predictor of cardiorespiratory fitness in childhood cancer survivors when access to comprehensive testing is limited, » *International Journal of Cancer*, pp. 847–855, 2020.

[51] L. Maaten et G. Hinton, « Visualizing Data using t-SNE, » *Journal of Machine Learning Research*, vol. 86, no. 9, p. 4, 2008.

[52] L. McInnes, J. Healy, et J. Melville, « Umap : Uniform manifold approximation and projection for dimension reduction, » *arXiv preprint arXiv :1802.03426*, 2018, non publié.

[53] O. Mäkitie, R. Heikkinen, S. Toiviainen-Salo, M. Henriksson, L.-R. Puukko-Viertomies, et K. Jahnukainen, « Long-term skeletal consequences of childhood acute lymphoblastic leukemia in adult males : a cohort study, » *European Journal of Endocrinology*, vol. 168, no. 2, 2013.

[54] T. M. Mitchell et T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997, vol. 1, no. 9.

[55] S. Mostoufi-Moab et J. Halton, « Bone Morbidity in Childhood Leukemia : Epidemiology, Mechanisms, Diagnosis, and Treatment, » *Current Osteoporosis Reports*, vol. 12, no. 3, pp. 300–312, 2014.

[56] F. Martini, M. Timmons, et R. Tallitsch, *Human Anatomy, 7th Edition.* Pearson Inc, 2014, ch. 20, pp. 530–542.

[57] G. Nadeau *et al.*, « Identification of genetic variants associated with skeletal muscle function deficit in childhood acute lymphoblastic leukemia survivors, » *Pharmacogenomics and Personalized Medicine*, vol. 12, pp. 33–45, 2019.

[58] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, et J. Clune, « Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, » dans *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc., 2016.

[59] M. E. J. Newman, *Networks : an introduction.* Oxford University Press, 2010.

[60] P. Nathan, K. Wasilewski-Masker, et L. Janzen, « Long-term Outcomes in Survivors of Childhood Acute Lymphoblastic Leukemia, » *Hematology/Oncology Clinics of North America*, vol. 23, no. 5, pp. 1065–1082, 2009.

[61] K. C. Oeffinger, « Are survivors of acute lymphoblastic leukemia (ALL) at increased risk of cardiovascular disease ? » *Pediatric Blood & Cancer*, vol. 50, pp. 462–467, 2008.

[62] C. Ogden, Y. Li, D. Freedman, L. Borrud, et K. Flegal, « Smoothed percentage body fat percentiles for U.S. children and adolescents, 1999-2004, » *National Health Statistics Reports*, vol. 43, pp. 1–7, 2011.

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

[64] A. Paszke *et al.*, « PyTorch : an imperative style, high-performance deep learning library, » dans *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 8026–8037.

[65] K. Petrykey *et al.*, « Influence of genetic factors on long-term treatment related neurocognitive complications, and on anxiety and depression in survivors of childhood acute lymphoblastic leukemia : The PETALE study, » *PLOS ONE*, vol. 14, no. 6, p. e0217314, 2019.

[66] M. Plesa *et al.*, « Influence of BCL2L11 polymorphism on osteonecrosis during treatment of childhood acute lymphoblastic leukemia, » *The Pharmacogenomics Journal*, vol. 19, no. 1, pp. 33–41, 2019.

[67] C.-H. Pui et W. E. Evans, « A 50-Year Journey to Cure Childhood Acute Lymphoblastic Leukemia, » *Seminars in hematology*, vol. 50, no. 3, pp. 185–196, 2013.

[68] R. S. Porter et J. L. Kaplan, *The Merck manual of diagnosis and therapy - 19th Edition.* Merck Sharp & Dohme Corp., 2011, ch. 117, pp. 1141–1147.

[69] Y. Qian, P. Expert, P. Panzarasa, et M. Barahona, « Geometric graphs from data to aid classification tasks with Graph Convolutional Networks, » *Patterns*, vol. 2, p. 100237, 2021.

[70] O. Ronneberger, P. Fischer, et T. Brox, « U-Net : Convolutional Networks for Biomedical Image Segmentation, » dans *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015.* Springer International Publishing, 2015, vol. 9351, pp. 234–241.

[71] S. Ruder, « An overview of multi-task learning in deep neural networks, » *arXiv preprint arXiv :1706.05098*, 2017, non publié.

[72] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, et G. Monfardini, « The Graph Neural Network Model, » *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[73] K. Santisteban, A. Lovering, H. JR., et C. Minson, « Sex Differences in VO2max and the Impact on Endurance-Exercise Performance, » *International Journal of Environmental Research and Public Health*, vol. 19, no. 9, p. 4946, 2022.

[74] N. A. Smart, « How do cardiorespiratory fitness improvements vary with physical training modality in heart failure patients ? A quantitative

guide, » *Experimental & Clinical Cardiology*, vol. 18, no. 1, pp. e21–e25, 2013.

[75] I. Sutskever, J. Martens, G. Dahl, et G. Hinton, « On the importance of initialization and momentum in deep learning, » dans *International conference on machine learning*. PMLR, 2013, pp. 1139–1147.

[76] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, et Y. Bengio, « Graph Attention Networks, » dans *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[77] J. Van DM *et al.*, « Obesity after Successful Treatment of Acute Lymphoblastic Leukemia in Childhood, » *Pediatric Research*, vol. 38, no. 1, pp. 86–90, 1995.

[78] M. Wang *et al.*, « Deep Graph Library : towards efficient and scalable deep learning on graphs, » dans *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[79] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, et P. S. Yu, « A Comprehensive Survey on Graph Neural Networks, » *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[80] K. Xu *et al.*, « Representation Learning on Graphs with Jumping Knowledge Networks, » dans *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 5453–5462.

[81] Y. Zhang, P. Tino, A. Leonardis, et K. Tang, « A Survey on Neural Network Interpretability, » *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

Annexe A

Matériel supplémentaire

Cette annexe contient principalement les résultats et les détails méthodologiques en support à l'article présenté au Chapitre 3. L'ensemble du texte est présenté en anglais pour être en cohésion avec le format de l'article.

Definitions

6MWD 6-minute walked distance
BMI Body mass index
C-index Concordance index
DT Duration of treatment
ECCD Effective corticosteroid cumulative dose
EOT End of treatment
GAT Graph attention network.
GCN Graph convolutional network
GGAE Gene graph attention encoder
HP Hyperparameter
HRend Heart rate at the end of the 6-minute walk test
MAE Mean absolute error
MLP Multi-layer Perceptron
MVLPA Moderate-to-vigorous leisure physical activity
PCC Pearson correlation coefficient
RMSE Root-mean-square error
SNP Single nucleotide polymorphism
std Standard deviation

A.1 Supplementary results

This section provides additional results regarding the data analyses. We first present tables with descriptive analyses of the datasets used for the experiments. Then, we give further details about the scores obtained by the models evaluated on the *learning set* of each prediction task.

A.1.1 Descriptive analyses of the datasets

In this subsection we present tables with descriptive analyses of the complete dataset, the *learning set* and the *holdout set* related to each prediction task. The statistics associated to each numerical feature are presented into the format *mean* (*std*) [*min*, *max*]. The statistics associated to each categorical feature (e.g., SNP) are presented into the format *count* (*percentage of group*). Note that the statistics are calculated without considering any missing value.

	Group		
Feature	All (n = 164)	Women (n = 84)	Men (n = 80)
Age (years)	22.40 (6.13) [11.33, 38.76]	22.50 (6.22) [11.33, 38.76]	22.29 (6.07) [11.40, 36.92]
DT (years)	2.13 (0.17) [1.17, 2.83]	2.16 (0.18) [1.17, 2.83]	2.10 (0.16) [1.33, 2.41]
HRend (bpm)	149.37 (22.33) [75.00, 191.00]	144.87 (23.21) [75.00, 191.00]	153.92 (20.57) [96.00, 190.00]
MVLPA (min/day)	25.26 (25.03) [0.00, 149.18]	24.09 (25.66) [0.00, 149.18]	26.49 (24.45) [0.00, 110.16]
Weight (kg)	66.75 (15.94) [40.50, 113.00]	64.37 (16.66) [40.50, 113.00]	69.26 (14.83) [43.00, 108.00]
6MWD (m)	612.25 (78.47) [412.00, 873.00]	637.53 (80.55) [412.00, 873.00]	587.60 (68.32) [445.00, 755.00]
VO_2 peak (mL/kg/min)	32.12 (8.32) [15.90, 52.40]	27.32 (6.76) [15.90, 44.60]	37.16 (6.69) [24.20, 52.40]

Table A.1 – Descriptive analysis of the clinical features and the target (VO_2 peak dataset).

	Group		
Feature	All (n = 147)	Women (n = 72)	Men (n = 75)
Age (years)	22.30 (6.16) [11.33, 38.76]	22.48 (6.39) [11.33, 38.76]	22.11 (5.95) [11.40, 36.92]
DT (years)	2.13 (0.17) [1.17, 2.83]	2.15 (0.18) [1.17, 2.83]	2.11 (0.15) [1.33, 2.41]
HRend (bpm)	150.65 (21.95) [75.00, 191.00]	145.97 (23.23) [75.00, 191.00]	155.47 (19.57) [99.00, 190.00]
MVLPA (min/day)	25.13 (25.81) [0.00, 149.18]	23.61 (26.59) [0.00, 149.18]	26.70 (25.06) [0.00, 110.16]
Weight (kg)	67.19 (16.59) [40.50, 113.00]	64.68 (17.39) [40.50, 113.00]	69.81 (15.39) [43.00, 108.00]
6MWD (m)	615.06 (77.21) [420.00, 873.00]	639.97 (76.75) [420.00, 873.00]	590.49 (69.88) [445.00, 755.00]
VO_2 peak (mL/kg/min)	32.12 (8.43) [15.90, 52.40]	27.19 (6.84) [15.90, 44.60]	37.27 (6.69) [24.20, 52.40]

Table A.2 – Descriptive analysis of the clinical features and the target (VO_2 peak *learning set*).

A.1. Supplementary results

	Group		
Feature	All (n = 17)	Women (n = 8)	Men (n = 9)
Age (years)	23.25 (5.98) [14.10, 34.43]	22.72 (4.84) [14.10, 30.67]	23.85 (7.36) [15.46, 34.43]
DT (years)	2.16 (0.20) [1.59, 2.58]	2.23 (0.17) [2.00, 2.58]	2.07 (0.22) [1.59, 2.25]
HRend (bpm)	138.94 (23.34) [96.00, 177.00]	135.25 (22.08) [112.00, 171.00]	142.22 (25.23) [96.00, 177.00]
MVLPA (min/day)	26.46 (17.29) [0.00, 50.49]	28.11 (16.42) [0.22, 50.49]	24.61 (19.18) [0.00, 50.16]
Weight (kg)	62.95 (7.73) [46.00, 73.60]	61.80 (8.61) [46.00, 73.60]	64.25 (6.94) [53.00, 73.00]
6MWD (m)	588.94 (87.28) [412.00, 750.00]	616.12 (112.84) [412.00, 750.00]	564.78 (51.96) [480.00, 631.00]
VO_2 peak (mL/kg/min)	32.10 (7.60) [18.00, 46.80]	28.44 (6.31) [18.00, 39.30]	36.21 (7.08) [27.00, 46.80]

Table A.3 – Descriptive analysis of the clinical features and the target (VO_2 peak *holdout set*).

	Group		
Feature	All (n = 223)	Women (n = 112)	Men (n = 111)
Age at diagnosis (years)	6.00 (4.54) [0.00, 18.00]	5.83 (4.41) [0.00, 17.00]	6.17 (4.67) [1.00, 18.00]
BMI at the EOT (kg/m^2)	18.55 (3.16) [12.71, 31.52]	18.89 (3.44) [12.71, 31.52]	18.21 (2.82) [13.98, 26.43]
Doxorubicin (mg/m^2)	182.09 (120.22) [41.51, 472.85]	181.79 (123.45) [41.51, 472.85]	182.40 (117.43) [43.64, 403.09]
DT (years)	2.14 (0.17) [1.17, 2.83]	2.15 (0.19) [1.17, 2.83]	2.12 (0.15) [1.33, 2.41]
ECCD (mg/m^2)	11088.88 (4992.14) [4028.75, 24918.16]	10684.69 (4808.98) [4424.10, 24918.16]	11496.72 (5159.98) [4028.75, 24095.31]
Methotrexate (mg/m^2)	6287.94 (1578.23) [853.63, 12784.10]	6113.30 (1539.55) [853.63, 9746.50]	6464.15 (1603.98) [1036.32, 12784.10]
Total body fat (%)	31.17 (11.70) [7.00, 59.80]	38.30 (8.94) [19.40, 59.80]	23.98 (9.58) [7.00, 47.30]

Table A.4 – Descriptive analysis of the numerical clinical features and the target (Obesity *dataset*).

		Group		
Feature	Modality	All (n = 223)	Women (n = 112)	Men (n = 111)
Dexrazoxane (mg/m^2)	0	157 (70.40%)	79 (70.54%)	78 (70.27%)
	>0,≤2963.16	32 (14.35%)	13 (11.61%)	19 (17.12%)
	>2963.16	34 (15.25%)	20 (17.86%)	14 (12.61%)
Gestational age at birth (weeks)	<37	13 (6.19%)	10 (9.35%)	3 (2.91%)
	≥37	197 (93.81%)	97 (90.65%)	100 (97.09%)
Radiotherapy dose (Gy)	0	93 (41.70%)	54 (48.21%)	39 (35.14%)
	>0	130 (58.30%)	58 (51.79%)	72 (64.86%)

Table A.5 – Descriptive analysis of the categorical clinical features (Obesity dataset).

	Group		
Feature	All (n = 200)	Women (n = 100)	Men (n = 100)
Age at diagnosis (years)	6.07 (4.56) [0.00, 18.00]	5.94 (4.47) [0.00, 17.00]	6.20 (4.66) [1.00, 18.00]
BMI at the EOT (kg/m^2)	18.53 (3.25) [12.71, 31.52]	18.86 (3.57) [12.71, 31.52]	18.20 (2.87) [13.98, 26.43]
Doxorubicin (mg/m^2)	188.74 (120.12) [41.51, 472.85]	192.22 (123.18) [41.51, 472.85]	185.27 (117.50) [43.64, 403.09]
DT (years)	2.14 (0.18) [1.17, 2.83]	2.15 (0.20) [1.17, 2.83]	2.12 (0.16) [1.33, 2.41]
ECCD (mg/m^2)	11251.40 (5069.78) [4028.75, 24918.16]	10982.62 (4956.43) [4424.10, 24918.16]	11520.18 (5191.62) [4028.75, 24095.31]
Methotrexate (mg/m^2)	6286.33 (1575.20) [853.63, 12784.10]	6132.98 (1468.53) [853.63, 8035.45]	6439.68 (1668.38) [1036.32, 12784.10]
Total body fat (%)	31.35 (11.83) [7.00, 59.80]	38.58 (8.89) [19.40, 59.80]	24.12 (9.84) [7.00, 47.30]

Table A.6 – Descriptive analysis of the numerical clinical features and the target (Obesity *learning set*).

		Group		
Feature	Modality	All (n = 200)	Women (n = 100)	Men (n = 100)
Dexrazoxane (mg/m^2)	0	140 (70.00%)	69 (69.00%)	71 (71.00%)
	>0,≤2963.16	29 (14.50%)	12 (12.00%)	17 (17.00%)
	>2963.16	31 (15.50%)	19 (19.00%)	12 (12.00%)
Gestational age at birth (weeks)	<37	11 (5.85%)	9 (9.47%)	2 (2.15%)
	≥37	177 (94.15%)	86 (90.53%)	91 (97.85%)
Radiotherapy dose (Gy)	0	79 (39.50%)	43 (43.00%)	36 (36.00%)
	>0	121 (60.50%)	57 (57.00%)	64 (64.00%)

Table A.7 – Descriptive analysis of the categorical clinical features (Obesity *learning set*).

	Group		
Feature	All (n = 23)	Women (n = 12)	Men (n = 11)
Age at diagnosis (years)	5.39 (4.43) [2.00, 16.00]	4.92 (3.96) [2.00, 16.00]	5.91 (5.03) [2.00, 16.00]
BMI at the EOT (kg/m^2)	18.72 (2.28) [14.55, 23.01]	19.18 (2.10) [16.47, 23.01]	18.30 (2.45) [14.55, 22.12]
Doxorubicin (mg/m^2)	124.27 (106.89) [44.23, 301.39]	94.87 (89.37) [44.23, 299.34]	156.35 (119.08) [46.43, 301.39]
DT (years)	2.15 (0.09) [2.00, 2.25]	2.17 (0.09) [2.00, 2.25]	2.13 (0.09) [2.00, 2.25]
ECCD (mg/m^2)	9675.70 (4081.64) [6049.08, 21437.53]	8201.96 (2176.78) [6049.08, 13413.93]	11283.41 (5098.02) [6636.30, 21437.53]
Methotrexate (mg/m^2)	6301.94 (1640.15) [2104.66, 9746.50]	5949.30 (2115.48) [2104.66, 9746.50]	6686.64 (826.67) [4986.33, 7687.01]
Total body fat (%)	29.61 (10.62) [14.40, 54.90]	35.95 (9.44) [23.30, 54.90]	22.70 (7.03) [14.40, 35.90]

Table A.8 – Descriptive analysis of the numerical clinical features and the target (Obesity *holdout set*).

		Group		
Feature	Modality	All (n = 23)	Women (n = 12)	Men (n = 11)
Dexrazoxane (mg/m^2)	0	17 (73.91%)	10 (83.33%)	7 (63.64%)
	>0,≤2963.16	3 (13.04%)	1 (8.33%)	2 (18.18%)
	>2963.16	3 (13.04%)	1 (8.33%)	2 (18.18%)
Gestational age at birth (weeks)	<37	2 (9.09%)	1 (8.33%)	1 (10.00%)
	≥37	20 (90.91%)	11 (91.67%)	9 (90.00%)
Radiotherapy dose (Gy)	0	14 (60.87%)	11 (91.67%)	3 (27.27%)
	>0	9 (39.13%)	1 (8.33%)	8 (72.73%)

Table A.9 – Descriptive analysis of the categorical clinical features (Obesity *holdout set*).

A.1. Supplementary results

		Group		
SNP	Modality	All (n = 223)	Women (n = 112)	Men (n = 111)
1:66036441	0/0	123 (55.16%)	62 (55.36%)	61 (54.95%)
	0/1	82 (36.77%)	43 (38.39%)	39 (35.14%)
	1/1	18 (8.07%)	7 (6.25%)	11 (9.91%)
1:161182208	0/0	179 (80.63%)	88 (79.28%)	91 (81.98%)
	0/1	43 (19.37%)	23 (20.72%)	20 (18.02%)
1:226019633	0/0	110 (49.33%)	56 (50.00%)	54 (48.65%)
	0/1	91 (40.81%)	43 (38.39%)	48 (43.24%)
	1/1	22 (9.87%)	13 (11.61%)	9 (8.11%)
2:46611678	0/0	211 (94.62%)	107 (95.54%)	104 (93.69%)
	0/1	12 (5.38%)	5 (4.46%)	7 (6.31%)
2:179650408	0/0	124 (55.61%)	61 (54.46%)	63 (56.76%)
	0/1	85 (38.12%)	44 (39.29%)	41 (36.94%)
	1/1	14 (6.28%)	7 (6.25%)	7 (6.31%)
2:240946766	0/0	86 (41.95%)	45 (44.12%)	41 (39.81%)
	0/1	85 (41.46%)	43 (42.16%)	42 (40.78%)
	1/1	34 (16.59%)	14 (13.73%)	20 (19.42%)
4:120241902	0/0	18 (8.07%)	11 (9.82%)	7 (6.31%)
	0/1	85 (38.12%)	43 (38.39%)	42 (37.84%)
	1/1	120 (53.81%)	58 (51.79%)	62 (55.86%)
6:12296255	0/0	139 (62.33%)	66 (58.93%)	73 (65.77%)
	0/1	74 (33.18%)	41 (36.61%)	33 (29.73%)
	1/1	10 (4.48%)	5 (4.46%)	5 (4.50%)
6:29912280	0/0	41 (18.81%)	14 (12.84%)	27 (24.77%)
	0/1	89 (40.83%)	47 (43.12%)	42 (38.53%)
	1/1	88 (40.37%)	48 (44.04%)	40 (36.70%)
6:29912333	0/0	60 (27.03%)	26 (23.42%)	34 (30.63%)
	0/1	106 (47.75%)	56 (50.45%)	50 (45.05%)
	1/1	56 (25.23%)	29 (26.13%)	27 (24.32%)
6:29912386	0/0	165 (74.32%)	82 (73.87%)	83 (74.77%)
	0/1	49 (22.07%)	23 (20.72%)	26 (23.42%)
	1/1	8 (3.60%)	6 (5.41%)	2 (1.80%)
6:110760008	0/0	131 (58.74%)	73 (65.18%)	58 (52.25%)
	0/1	78 (34.98%)	38 (33.93%)	40 (36.04%)
	1/1	14 (6.28%)	1 (0.89%)	13 (11.71%)
7:20762646	0/0	104 (47.27%)	56 (50.00%)	48 (44.44%)
	0/1	86 (39.09%)	41 (36.61%)	45 (41.67%)
	1/1	30 (13.64%)	15 (13.39%)	15 (13.89%)
7:45932669	0/0	85 (38.12%)	40 (35.71%)	45 (40.54%)
	0/1	99 (44.39%)	50 (44.64%)	49 (44.14%)
	1/1	39 (17.49%)	22 (19.64%)	17 (15.32%)
7:87160618	0/0	38 (17.19%)	18 (16.36%)	20 (18.02%)
	0/1	105 (47.51%)	53 (48.18%)	52 (46.85%)
	1/1	72 (32.58%)	35 (31.82%)	37 (33.33%)
7:94946084	0/0	89 (39.91%)	40 (35.71%)	49 (44.14%)
	0/1	102 (45.74%)	53 (47.32%)	49 (44.14%)
	1/1	32 (14.35%)	19 (16.96%)	13 (11.71%)
12:48272895	0/0	48 (21.72%)	24 (21.62%)	24 (21.82%)
	0/1	96 (43.44%)	47 (42.34%)	49 (44.55%)
	1/1	77 (34.84%)	40 (36.04%)	37 (33.64%)
13:23824818	0/0	172 (79.26%)	87 (80.56%)	85 (77.98%)
	0/1	45 (20.74%)	21 (19.44%)	24 (22.02%)
13:95863008	0/0	203 (91.03%)	100 (89.29%)	103 (92.79%)
	0/1	20 (8.97%)	12 (10.71%)	8 (7.21%)
15:58838010	0/0	99 (44.39%)	48 (42.86%)	51 (45.95%)
	0/1	111 (49.78%)	55 (49.11%)	56 (50.45%)
	1/1	13 (5.83%)	9 (8.04%)	4 (3.60%)
16:69745145	0/0	150 (67.26%)	79 (70.54%)	71 (63.96%)
	0/1	67 (30.04%)	29 (25.89%)	38 (34.23%)
	1/1	6 (2.69%)	4 (3.57%)	2 (1.80%)
16:88713236	0/0	48 (23.08%)	23 (22.33%)	25 (23.81%)
	0/1	72 (34.62%)	32 (31.07%)	40 (38.10%)
	1/1	88 (42.31%)	48 (46.60%)	40 (38.10%)
17:4856580	0/0	33 (14.80%)	16 (14.29%)	17 (15.32%)
	0/1	111 (49.78%)	60 (53.57%)	51 (45.95%)
	1/1	79 (35.43%)	36 (32.14%)	43 (38.74%)
17:26096597	0/0	153 (68.61%)	74 (66.07%)	79 (71.17%)
	0/1	60 (26.91%)	35 (31.25%)	25 (22.52%)
	1/1	10 (4.48%)	3 (2.68%)	7 (6.31%)
17:37884037	0/0	39 (18.06%)	16 (14.81%)	23 (21.30%)
	0/1	99 (45.83%)	58 (53.70%)	41 (37.96%)
	1/1	78 (36.11%)	34 (31.48%)	44 (40.74%)
17:38545824	0/0	207 (92.83%)	106 (94.64%)	101 (90.99%)
	0/1	16 (7.17%)	6 (5.36%)	10 (9.01%)
17:48712711	0/0	23 (12.30%)	12 (12.63%)	11 (11.96%)
	0/1	91 (48.66%)	51 (53.68%)	40 (43.48%)
	1/1	73 (39.04%)	32 (33.68%)	41 (44.57%)
21:37518706	0/0	89 (39.91%)	53 (47.32%)	36 (32.43%)
	0/1	113 (50.67%)	47 (41.96%)	66 (59.46%)
	1/1	21 (9.42%)	12 (10.71%)	9 (8.11%)
21:44324365	0/0	47 (21.96%)	24 (22.43%)	23 (21.50%)
	0/1	96 (44.86%)	47 (43.93%)	49 (45.79%)
	1/1	71 (33.18%)	36 (33.64%)	35 (32.71%)
22:42486723	0/0	105 (47.30%)	57 (51.35%)	48 (43.24%)
	0/1	94 (42.34%)	45 (40.54%)	49 (44.14%)
	1/1	23 (10.36%)	9 (8.11%)	14 (12.61%)

Table A.10 – Descriptive analysis of the SNPs (Obesity dataset).

		Group		
SNP	Modality	All (n = 200)	Women (n = 100)	Men (n = 100)
1:66036441	0/0	112 (56.00%)	59 (59.00%)	53 (53.00%)
	0/1	72 (36.00%)	35 (35.00%)	37 (37.00%)
	1/1	16 (8.00%)	6 (6.00%)	10 (10.00%)
1:161182208	0/0	160 (80.40%)	77 (77.78%)	83 (83.00%)
	0/1	39 (19.60%)	22 (22.22%)	17 (17.00%)
1:226019633	0/0	97 (48.50%)	49 (49.00%)	48 (48.00%)
	0/1	82 (41.00%)	39 (39.00%)	43 (43.00%)
	1/1	21 (10.50%)	12 (12.00%)	9 (9.00%)
2:46611678	0/0	190 (95.00%)	95 (95.00%)	95 (95.00%)
	0/1	10 (5.00%)	5 (5.00%)	5 (5.00%)
2:179650408	0/0	111 (55.50%)	55 (55.00%)	56 (56.00%)
	0/1	75 (37.50%)	38 (38.00%)	37 (37.00%)
	1/1	14 (7.00%)	7 (7.00%)	7 (7.00%)
2:240946766	0/0	73 (39.89%)	37 (40.66%)	36 (39.13%)
	0/1	80 (43.72%)	40 (43.96%)	40 (43.48%)
	1/1	30 (16.39%)	14 (15.38%)	16 (17.39%)
4:120241902	0/0	16 (8.00%)	9 (9.00%)	7 (7.00%)
	0/1	79 (39.50%)	40 (40.00%)	39 (39.00%)
	1/1	105 (52.50%)	51 (51.00%)	54 (54.00%)
6:12296255	0/0	123 (61.50%)	59 (59.00%)	64 (64.00%)
	0/1	69 (34.50%)	37 (37.00%)	32 (32.00%)
	1/1	8 (4.00%)	4 (4.00%)	4 (4.00%)
6:29912280	0/0	36 (18.27%)	13 (13.40%)	23 (23.00%)
	0/1	84 (42.64%)	45 (46.39%)	39 (39.00%)
	1/1	77 (39.09%)	39 (40.21%)	38 (38.00%)
6:29912333	0/0	54 (27.14%)	24 (24.24%)	30 (30.00%)
	0/1	97 (48.74%)	52 (52.53%)	45 (45.00%)
	1/1	48 (24.12%)	23 (23.23%)	25 (25.00%)
6:29912386	0/0	151 (75.88%)	75 (75.76%)	76 (76.00%)
	0/1	41 (20.60%)	18 (18.18%)	23 (23.00%)
	1/1	7 (3.52%)	6 (6.06%)	1 (1.00%)
6:110760008	0/0	115 (57.50%)	64 (64.00%)	51 (51.00%)
	0/1	72 (36.00%)	36 (36.00%)	36 (36.00%)
	1/1	13 (6.50%)	0 (0.00%)	13 (13.00%)
7:20762646	0/0	96 (48.73%)	50 (50.00%)	46 (47.42%)
	0/1	77 (39.09%)	38 (38.00%)	39 (40.21%)
	1/1	24 (12.18%)	12 (12.00%)	12 (12.37%)
7:45932669	0/0	77 (38.50%)	36 (36.00%)	41 (41.00%)
	0/1	88 (44.00%)	43 (43.00%)	45 (45.00%)
	1/1	35 (17.50%)	21 (21.00%)	14 (14.00%)
7:87160618	0/0	32 (16.08%)	14 (14.14%)	18 (18.00%)
	0/1	98 (49.25%)	50 (50.51%)	48 (48.00%)
	1/1	64 (32.16%)	32 (32.32%)	32 (32.00%)
7:94946084	0/0	79 (39.50%)	36 (36.00%)	43 (43.00%)
	0/1	93 (46.50%)	47 (47.00%)	46 (46.00%)
	1/1	28 (14.00%)	17 (17.00%)	11 (11.00%)
12:48272895	0/0	41 (20.71%)	21 (21.21%)	20 (20.20%)
	0/1	87 (43.94%)	40 (40.40%)	47 (47.47%)
	1/1	70 (35.35%)	38 (38.38%)	32 (32.32%)
13:23824818	0/0	156 (80.00%)	80 (83.33%)	76 (76.77%)
	0/1	39 (20.00%)	16 (16.67%)	23 (23.23%)
13:95863008	0/0	182 (91.00%)	88 (88.00%)	94 (94.00%)
	0/1	18 (9.00%)	12 (12.00%)	6 (6.00%)
15:58838010	0/0	85 (42.50%)	41 (41.00%)	44 (44.00%)
	0/1	104 (52.00%)	51 (51.00%)	53 (53.00%)
	1/1	11 (5.50%)	8 (8.00%)	3 (3.00%)
16:69745145	0/0	132 (66.00%)	68 (68.00%)	64 (64.00%)
	0/1	63 (31.50%)	29 (29.00%)	34 (34.00%)
	1/1	5 (2.50%)	3 (3.00%)	2 (2.00%)
16:88713236	0/0	45 (23.81%)	21 (22.34%)	24 (25.26%)
	0/1	65 (34.39%)	29 (30.85%)	36 (37.89%)
	1/1	79 (41.80%)	44 (46.81%)	35 (36.84%)
17:4856580	0/0	31 (15.50%)	15 (15.00%)	16 (16.00%)
	0/1	100 (50.00%)	56 (56.00%)	44 (44.00%)
	1/1	69 (34.50%)	29 (29.00%)	40 (40.00%)
17:26096597	0/0	136 (68.00%)	65 (65.00%)	71 (71.00%)
	0/1	54 (27.00%)	32 (32.00%)	22 (22.00%)
	1/1	10 (5.00%)	3 (3.00%)	7 (7.00%)
17:37884037	0/0	36 (18.46%)	15 (15.46%)	21 (21.43%)
	0/1	89 (45.64%)	50 (51.55%)	39 (39.80%)
	1/1	70 (35.90%)	32 (32.99%)	38 (38.78%)
17:38545824	0/0	186 (93.00%)	95 (95.00%)	91 (91.00%)
	0/1	14 (7.00%)	5 (5.00%)	9 (9.00%)
17:48712711	0/0	20 (11.83%)	11 (13.10%)	9 (10.59%)
	0/1	81 (47.93%)	44 (52.38%)	37 (43.53%)
	1/1	68 (40.24%)	29 (34.52%)	39 (45.88%)
21:37518706	0/0	76 (38.00%)	47 (47.00%)	29 (29.00%)
	0/1	105 (52.50%)	42 (42.00%)	63 (63.00%)
	1/1	19 (9.50%)	11 (11.00%)	8 (8.00%)
21:44324365	0/0	39 (20.21%)	19 (20.00%)	20 (20.41%)
	0/1	91 (47.15%)	43 (45.26%)	48 (48.98%)
	1/1	63 (32.64%)	33 (34.74%)	30 (30.61%)
22:42486723	0/0	92 (46.23%)	51 (51.52%)	41 (41.00%)
	0/1	86 (43.22%)	40 (40.40%)	46 (46.00%)
	1/1	21 (10.55%)	8 (8.08%)	13 (13.00%)

Table A.11 – Descriptive analysis of the SNPs (Obesity *learning set*).

		Group		
SNP	Modality	All (n = 23)	Women (n = 12)	Men (n = 11)
1:66036441	0/0	11 (47.83%)	3 (25.00%)	8 (72.73%)
	0/1	10 (43.48%)	8 (66.67%)	2 (18.18%)
	1/1	2 (8.70%)	1 (8.33%)	1 (9.09%)
1:161182208	0/0	19 (82.61%)	11 (91.67%)	8 (72.73%)
	0/1	4 (17.39%)	1 (8.33%)	3 (27.27%)
1:226019633	0/0	13 (56.52%)	7 (58.33%)	6 (54.55%)
	0/1	9 (39.13%)	4 (33.33%)	5 (45.45%)
	1/1	1 (4.35%)	1 (8.33%)	0 (0.00%)
2:46611678	0/0	21 (91.30%)	12 (100.00%)	9 (81.82%)
	0/1	2 (8.70%)	0 (0.00%)	2 (18.18%)
2:179650408	0/0	13 (56.52%)	6 (50.00%)	7 (63.64%)
	0/1	10 (43.48%)	6 (50.00%)	4 (36.36%)
	1/1	13 (59.09%)	8 (72.73%)	5 (45.45%)
2:240946766	0/0	5 (22.73%)	3 (27.27%)	2 (18.18%)
	0/1	4 (18.18%)	0 (0.00%)	4 (36.36%)
	1/1	30 (16.39%)	14 (15.38%)	16 (17.39%)
4:120241902	0/0	2 (8.70%)	2 (16.67%)	0 (0.00%)
	0/1	6 (26.09%)	3 (25.00%)	3 (27.27%)
	1/1	15 (65.22%)	7 (58.33%)	8 (72.73%)
6:12296255	0/0	16 (69.57%)	7 (58.33%)	9 (81.82%)
	0/1	5 (21.74%)	4 (33.33%)	1 (9.09%)
	1/1	2 (8.70%)	1 (8.33%)	1 (9.09%)
6:29912280	0/0	5 (23.81%)	1 (8.33%)	4 (44.44%)
	0/1	5 (23.81%)	2 (16.67%)	3 (33.33%)
	1/1	11 (52.38%)	9 (75.00%)	2 (22.22%)
6:29912333	0/0	6 (26.09%)	2 (16.67%)	4 (36.36%)
	0/1	9 (39.13%)	4 (33.33%)	5 (45.45%)
	1/1	8 (34.78%)	6 (50.00%)	2 (18.18%)
6:29912386	0/0	14 (60.87%)	7 (58.33%)	7 (63.64%)
	0/1	8 (34.78%)	5 (41.67%)	3 (27.27%)
	1/1	1 (4.35%)	0 (0.00%)	1 (9.09%)
6:110760008	0/0	16 (69.57%)	9 (75.00%)	7 (63.64%)
	0/1	6 (26.09%)	2 (16.67%)	4 (36.36%)
	1/1	1 (4.35%)	1 (8.33%)	0 (0.00%)
7:20762646	0/0	8 (34.78%)	6 (50.00%)	2 (18.18%)
	0/1	9 (39.13%)	3 (25.00%)	6 (54.55%)
	1/1	6 (26.09%)	3 (25.00%)	3 (27.27%)
7:45932669	0/0	8 (34.78%)	4 (33.33%)	4 (36.36%)
	0/1	11 (47.83%)	7 (58.33%)	4 (36.36%)
	1/1	4 (17.39%)	1 (8.33%)	3 (27.27%)
7:87160618	0/0	6 (27.27%)	4 (36.36%)	2 (18.18%)
	0/1	7 (31.82%)	3 (27.27%)	4 (36.36%)
	1/1	8 (36.36%)	3 (27.27%)	5 (45.45%)
7:94946084	0/0	10 (43.48%)	4 (33.33%)	6 (54.55%)
	0/1	9 (39.13%)	6 (50.00%)	3 (27.27%)
	1/1	4 (17.39%)	2 (16.67%)	2 (18.18%)
12:48272895	0/0	7 (30.43%)	3 (25.00%)	4 (36.36%)
	0/1	9 (39.13%)	7 (58.33%)	2 (18.18%)
	1/1	7 (30.43%)	2 (16.67%)	5 (45.45%)
13:23824818	0/0	16 (72.73%)	7 (58.33%)	9 (90.00%)
	0/1	6 (27.27%)	5 (41.67%)	1 (10.00%)
13:95863008	0/0	21 (91.30%)	12 (100.00%)	9 (81.82%)
	0/1	2 (8.70%)	0 (0.00%)	2 (18.18%)
15:58838010	0/0	14 (60.87%)	7 (58.33%)	7 (63.64%)
	0/1	7 (30.43%)	4 (33.33%)	3 (27.27%)
	1/1	2 (8.70%)	1 (8.33%)	1 (9.09%)
16:69745145	0/0	18 (78.26%)	11 (91.67%)	7 (63.64%)
	0/1	4 (17.39%)	0 (0.00%)	4 (36.36%)
	1/1	1 (4.35%)	1 (8.33%)	0 (0.00%)
16:88713236	0/0	3 (15.79%)	2 (22.22%)	1 (10.00%)
	0/1	7 (36.84%)	3 (33.33%)	4 (40.00%)
	1/1	9 (47.37%)	4 (44.44%)	5 (50.00%)
17:4856580	0/0	2 (8.70%)	1 (8.33%)	1 (9.09%)
	0/1	11 (47.83%)	4 (33.33%)	7 (63.64%)
	1/1	10 (43.48%)	7 (58.33%)	3 (27.27%)
17:26096597	0/0	17 (73.91%)	9 (75.00%)	8 (72.73%)
	0/1	6 (26.09%)	3 (25.00%)	3 (27.27%)
	1/1	3 (14.29%)	1 (9.09%)	2 (20.00%)
17:37884037	0/0	10 (47.62%)	8 (72.73%)	2 (20.00%)
	0/1	8 (38.10%)	2 (18.18%)	6 (60.00%)
	1/1	70 (35.90%)	32 (32.99%)	38 (38.78%)
17:38545824	0/0	21 (91.30%)	11 (91.67%)	10 (90.91%)
	0/1	2 (8.70%)	1 (8.33%)	1 (9.09%)
17:48712711	0/0	3 (16.67%)	1 (9.09%)	2 (28.57%)
	0/1	10 (55.56%)	7 (63.64%)	3 (42.86%)
	1/1	5 (27.78%)	3 (27.27%)	2 (28.57%)
21:37518706	0/0	13 (56.52%)	6 (50.00%)	7 (63.64%)
	0/1	8 (34.78%)	5 (41.67%)	3 (27.27%)
	1/1	2 (8.70%)	1 (8.33%)	1 (9.09%)
21:44324365	0/0	8 (38.10%)	5 (41.67%)	3 (33.33%)
	0/1	5 (23.81%)	4 (33.33%)	1 (11.11%)
	1/1	8 (38.10%)	3 (25.00%)	5 (55.56%)
22:42486723	0/0	13 (56.52%)	6 (50.00%)	7 (63.64%)
	0/1	8 (34.78%)	5 (41.67%)	3 (27.27%)
	1/1	2 (8.70%)	1 (8.33%)	1 (9.09%)

Table A.12 – Descriptive analysis of the SNPs (Obesity *holdout set*).

A.1.2 Evaluation of the models

In this subsection, we present tables with the scores obtained by the models following the 10 splits random stratified subsampling evaluation on the *learning set* associated to each prediction task. The scores are reported into the format *mean* ± *std*.

		Metric			
HPs optimization	Model	RMSE	MAE	PCC	C-index
Manual	**Random forest**	5.44 ± 0.74	4.28 ± 0.54	0.79 ± 0.05	0.81 ± 0.02
	XGBoost	5.64 ± 0.90	4.34 ± 0.70	0.76 ± 0.08	0.79 ± 0.04
	Linear regression	5.45 ± 0.71	4.27 ± 0.58	0.78 ± 0.07	0.80 ± 0.04
	MLP	6.00 ± 1.05	4.54 ± 0.75	0.78 ± 0.08	0.80 ± 0.04
	GCN	5.42 ± 0.86	4.14 ± 0.60	0.79 ± 0.07	0.81 ± 0.03
	GAT	**5.34 ± 0.80**	4.13 ± 0.59	**0.80 ± 0.07**	0.81 ± 0.03
Automated	**Random forest**	5.50 ± 0.75	4.32 ± 0.54	0.79 ± 0.06	0.80 ± 0.03
	XGBoost	5.39 ± 0.86	**4.12 ± 0.68**	0.79 ± 0.06	0.81 ± 0.04
	Linear regression	5.38 ± 0.70	4.22 ± 0.57	0.79 ± 0.07	0.80 ± 0.04
	MLP	5.76 ± 0.74	4.42 ± 0.54	0.77 ± 0.06	0.80 ± 0.03
	GCN	5.51 ± 0.76	4.26 ± 0.53	0.79 ± 0.07	0.81 ± 0.04
	GAT	5.47 ± 0.84	4.28 ± 0.67	0.79 ± 0.07	**0.82 ± 0.03**

Table A.13 – Scores obtained by the models following the 10 splits random stratified subsampling evaluation on the *learning set* of the VO_2 peak prediction task.

		Metric					
HPs optimization	Model	RMSE	MAE	PCC	C-index	Sensitivity	Specificity
Manual	**Random forest**	8.97 ± 0.53	7.36 ± 0.46	0.65 ± 0.04	0.73 ± 0.02	0.80 ± 0.06	0.71 ± 0.12
	XGBoost	9.12 ± 0.44	7.31 ± 0.35	0.65 ± 0.05	0.73 ± 0.02	0.79 ± 0.05	0.74 ± 0.09
	Linear regression	8.96 ± 0.62	7.39 ± 0.59	**0.66 ± 0.05**	**0.75 ± 0.02**	0.75 ± 0.08	0.72 ± 0.12
	MLP	9.07 ± 0.59	7.27 ± 0.48	0.64 ± 0.07	0.74 ± 0.04	0.68 ± 0.10	**0.76 ± 0.10**
	GCN	9.03 ± 0.61	7.47 ± 0.53	0.65 ± 0.04	0.74 ± 0.02	0.76 ± 0.12	0.75 ± 0.10
	GAT	9.03 ± 0.52	7.44 ± 0.52	0.64 ± 0.04	0.73 ± 0.02	0.72 ± 0.09	0.75 ± 0.07
Automated	**Random forest**	9.02 ± 0.54	7.37 ± 0.45	0.64 ± 0.04	0.73 ± 0.01	**0.81 ± 0.04**	0.69 ± 0.11
	XGBoost	9.00 ± 0.50	**7.27 ± 0.41**	0.65 ± 0.04	0.74 ± 0.02	0.77 ± 0.10	0.71 ± 0.13
	Linear regression	**8.91 ± 0.52**	7.36 ± 0.45	**0.66 ± 0.05**	0.75 ± 0.03	0.77 ± 0.07	0.70 ± 0.11
	MLP	9.14 ± 0.61	7.46 ± 0.65	0.64 ± 0.06	0.73 ± 0.03	0.70 ± 0.11	0.75 ± 0.13
	GCN	9.53 ± 0.55	7.74 ± 0.41	0.60 ± 0.05	0.72 ± 0.02	0.62 ± 0.10	0.74 ± 0.07
	GAT	9.18 ± 0.60	7.59 ± 0.58	0.63 ± 0.05	0.73 ± 0.02	0.75 ± 0.09	0.71 ± 0.12

Table A.14 – Scores obtained by the models following the 10 splits random stratified subsampling evaluation without SNPs on the *learning set* of the obesity prediction task.

		Metric					
HPs optimization	Model	RMSE	MAE	PCC	C-index	Sensitivity	Specificity
Manual	**Random forest**	9.64 ± 0.71	7.84 ± 0.67	0.62 ± 0.04	0.73 ± 0.01	**0.79 ± 0.09**	0.67 ± 0.09
	XGBoost	9.53 ± 0.53	7.80 ± 0.34	0.60 ± 0.04	0.71 ± 0.02	0.74 ± 0.08	0.70 ± 0.10
	Linear regression	10.16 ± 0.73	8.35 ± 0.69	0.52 ± 0.08	0.67 ± 0.03	0.62 ± 0.13	0.70 ± 0.11
	MLP	11.63 ± 1.59	9.05 ± 1.03	0.44 ± 0.12	0.66 ± 0.03	0.59 ± 0.10	0.73 ± 0.11
	GCN	11.46 ± 1.03	9.26 ± 0.64	0.46 ± 0.10	0.65 ± 0.04	0.62 ± 0.14	0.69 ± 0.11
	GAT	10.54 ± 0.64	8.59 ± 0.52	0.49 ± 0.06	0.67 ± 0.02	0.57 ± 0.11	0.73 ± 0.11
	Lin. reg. + GGAE	**9.02 ± 0.57**	**7.47 ± 0.55**	0.65 ± 0.05	**0.74 ± 0.02**	0.77 ± 0.08	0.72 ± 0.12
Automated	**Random forest**	9.61 ± 0.71	7.84 ± 0.67	0.61 ± 0.04	0.72 ± 0.02	**0.79 ± 0.09**	0.68 ± 0.10
	XGBoost	9.27 ± 0.58	7.54 ± 0.47	0.62 ± 0.04	0.72 ± 0.02	0.72 ± 0.12	0.70 ± 0.10
	Linear regression	9.56 ± 0.76	7.95 ± 0.67	0.58 ± 0.08	0.70 ± 0.03	0.71 ± 0.11	0.69 ± 0.08
	MLP	9.73 ± 1.08	7.80 ± 0.83	0.60 ± 0.09	0.72 ± 0.04	0.68 ± 0.11	**0.74 ± 0.08**
	GCN	9.82 ± 0.86	7.92 ± 0.75	0.58 ± 0.05	0.70 ± 0.03	0.65 ± 0.12	0.68 ± 0.14
	GAT	9.64 ± 0.67	7.80 ± 0.63	0.59 ± 0.05	0.71 ± 0.02	0.72 ± 0.14	0.70 ± 0.11
	Lin. reg. + GGAE	9.11 ± 0.63	7.53 ± 0.52	**0.65 ± 0.04**	**0.74 ± 0.02**	0.78 ± 0.06	0.70 ± 0.11

Table A.15 – Scores obtained by the models following the 10 splits random stratified subsampling evaluation with SNPs on the *learning set* of the obesity prediction task.

Evaluations of graphs' neighborhood sizes

Here, we display tables that motivated the choices of neighborhood sizes used in the construction of the graphs associated to the GNNs.

		Metric			
Model	Number of neighbors	RMSE	MAE	PCC	C-Index
GCN	4	5.52 ± 0.81	4.17 ± 0.56	**0.79 ± 0.07**	0.80 ± 0.04
	6	5.46 ± 0.83	4.17 ± 0.59	**0.79 ± 0.07**	**0.81 ± 0.03**
	8	**5.42 ± 0.86**	**4.14 ± 0.60**	**0.79 ± 0.07**	**0.81 ± 0.03**
	10	5.45 ± 0.84	4.18 ± 0.57	**0.79 ± 0.07**	0.81 ± 0.04
GAT	4	5.46 ± 0.86	4.19 ± 0.63	0.79 ± 0.06	**0.81 ± 0.03**
	6	5.45 ± 0.82	4.21 ± 0.61	0.79 ± 0.06	**0.81 ± 0.03**
	8	5.39 ± 0.84	4.21 ± 0.62	0.79 ± 0.07	0.81 ± 0.04
	10	**5.34 ± 0.80**	**4.13 ± 0.59**	**0.80 ± 0.07**	**0.81 ± 0.03**

Table A.16 – Graph neighborhood evaluation (VO_2 peak). Values of neighborhood size were tested with the other manually selected hyperparameters reported in Tables A.24-A.25

A.1. Supplementary results

		Metric					
Model	Number of neighbors	RMSE	MAE	PCC	C-Index	Sensitivity	Specificity
GCN	4	9.86 ± 0.70	7.99 ± 0.63	0.58 ± 0.06	0.70 ± 0.03	0.67 ± 0.07	0.71 ± 0.10
	6	9.06 ± 0.60	7.47 ± 0.54	0.64 ± 0.05	**0.74 ± 0.02**	0.75 ± 0.11	0.74 ± 0.10
	8	**9.03 ± 0.61**	**7.47 ± 0.53**	**0.65 ± 0.04**	**0.74 ± 0.02**	**0.76 ± 0.12**	**0.75 ± 0.10**
	10	9.84 ± 0.52	8.01 ± 0.52	0.58 ± 0.07	0.71 ± 0.03	0.63 ± 0.11	0.74 ± 0.12
GAT	4	9.85 ± 0.84	7.85 ± 0.74	0.57 ± 0.07	0.70 ± 0.03	0.60 ± 0.14	**0.76 ± 0.09**
	6	**9.03 ± 0.52**	**7.44 ± 0.52**	**0.64 ± 0.04**	**0.73 ± 0.02**	0.72 ± 0.09	0.75 ± 0.07
	8	9.18 ± 0.61	7.48 ± 0.59	0.63 ± 0.04	**0.73 ± 0.02**	**0.75 ± 0.10**	0.72 ± 0.13
	10	9.73 ± 0.60	7.86 ± 0.65	0.58 ± 0.06	0.71 ± 0.03	0.69 ± 0.07	0.74 ± 0.11

Table A.17 – Graph neighborhood evaluation (obesity w/o SNPs). Values of neighborhood size were tested with the other manually selected hyperparameters reported in Tables A.24-A.25.

		Metric					
Model	Number of neighbors	RMSE	MAE	PCC	C-Index	Sensitivity	Specificity
GCN	4	12.0 ± 1.48	9.62 ± 1.11	0.41 ± 0.11	0.64 ± 0.03	0.59 ± 0.10	**0.78 ± 0.08**
	6	11.89 ± 0.99	9.46 ± 0.75	0.42 ± 0.09	0.64 ± 0.04	**0.64 ± 0.12**	0.71 ± 0.15
	8	**11.46 ± 1.03**	**9.26 ± 0.64**	**0.46 ± 0.10**	0.65 ± 0.04	0.62 ± 0.14	0.69 ± 0.11
	10	12.15 ± 1.42	9.85 ± 1.38	0.43 ± 0.06	**0.65 ± 0.02**	0.63 ± 0.12	0.68 ± 0.13
GAT	4	11.12 ± 0.90	9.07 ± 0.81	0.40 ± 0.09	0.64 ± 0.03	0.60 ± 0.14	0.71 ± 0.12
	6	10.82 ± 0.85	8.63 ± 0.67	0.44 ± 0.10	0.65 ± 0.04	0.61 ± 0.13	**0.77 ± 0.09**
	8	10.78 ± 0.73	**8.82 ± 0.82**	0.47 ± 0.09	0.66 ± 0.04	**0.62 ± 0.14**	0.69 ± 0.11
	10	**10.54 ± 0.64**	8.59 ± 0.52	**0.49 ± 0.06**	**0.67 ± 0.02**	0.57 ± 0.11	0.73 ± 0.11

Table A.18 – Graph neighborhood evaluation (obesity w/ SNPs). Values of neighborhood size were tested with the other manually selected hyperparameters reported in Tables A.24-A.25.

A.2 Supplementary methods

This section provides additional details regarding the methodology supporting our experiments. We first illustrate how datasets were built for each prediction task. We continue by giving a formal mathematical description of each prediction model considered during the analyses. We further describe the hyperparameters of each of these models and provide their respective search spaces. We end this section by exposing the parameters used for the execution of the Tree-structured Parzen estimator (TPE) algorithm during the hyperparameter optimization.

A.2.1 Construction of the datasets

Figures reported in this subsection illustrate the construction procedures of the two datasets used in the study.

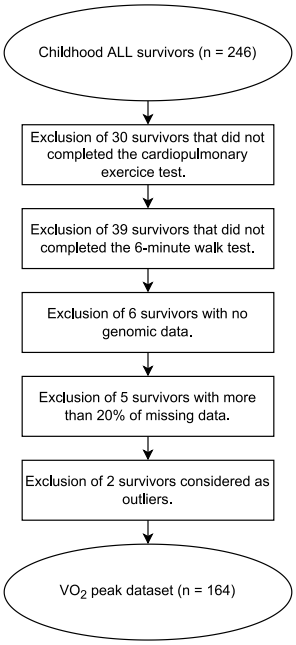

Figure A.1 – Construction of the VO_2 peak dataset. The six survivors without genomic data were excluded to allow the possibility of an eventual experiment including SNPs. The first outlier was excluded due to a short DT of six months. The second outlier was excluded considering its high MVLPA of 238 minutes per day.

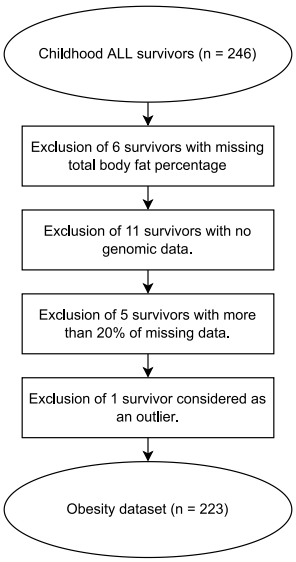

Figure A.2 – Construction of the obesity dataset. The outlier was excluded due to a short DT of six months.

A.2.2 Architectures of the models

In this subsection, we describe the model architectures that were implemented using *Pytorch* [64] and *DGL* [78] librairies. We denote $\mathcal{N}$ and $\mathcal{C}$ the sets of numerical and categorical features observed in a dataset and k_j the number of modalities associated to any categorical feature $j \in \mathcal{C}$. We use the notation $\mathbf{x}_i \in \mathcal{M}_{(|\mathcal{N}|+|\mathcal{C}|),1}(\mathbb{R})$ to represent a single data point in a dataset and $\mathbf{x}_{ij}$ its component associated to the feature $j \in \mathcal{N} \cup \mathcal{C}$. We also identify the prediction associated to a datapoint $\mathbf{x}_i$ as y_i. Hence, considering this notation, we can define

$$\mathbf{x}_i = [\mathbf{m}_i; \mathbf{c}_i],$$

where $\mathbf{m}_i = [\mathbf{x}_{ij}]_{j\in\mathcal{N}} \in \mathcal{M}_{|\mathcal{N}|,1}(\mathbb{R})$ and $\mathbf{c}_i = [\mathbf{x}_{ij}]_{j\in\mathcal{C}} \in \mathcal{M}_{|\mathcal{C}|,1}(\mathbb{R})$ contain the standardized numerical features and the nominal encodings of the categorical features respectively.

Categorical embedding

We integrated the categorical features within each of our architecture using categorical embeddings. More precisely, each categorical feature ($\mathbf{x}_{ij}$ s.t $j \in \mathcal{C}$) is first mapped to a vectorial representation (i.e., embedding) $\mathbf{h}_{ij} \in \mathcal{M}_{k_j,k_j}(\mathbb{R})$. Each embedding is defined as

$$\mathbf{h}_{ij} = \mathbf{W}_{emb}^{(j)} \mathbf{o}_{ij},$$

where $\mathbf{W}_{emb}^{(j)} \in \mathcal{M}_{k_j,k_j}(\mathbb{R})$ is a matrix of parameters initialized independently from a standard normal distribution and $\mathbf{o}_{ij} \in \mathcal{M}_{k_j,1}(\mathbb{R})$ is a one-hot vector with the non-null value at the position represented by the nominal encoding $\mathbf{x}_{ij}$.

All categorical embeddings are further concatenated to represented the enriched representation of $\mathbf{c}_i$ that we denote $\mathbf{c}'_i \in \mathcal{M}_{\left(\sum_{j\in\mathcal{C}} k_j\right),1}(\mathbb{R})$. More precisely,

$$\mathbf{c}'_i = \|_{j\in\mathcal{C}}\, \mathbf{h}_{ij}.$$

Linear regression

The linear regression model was implemented such that

$$y_i = \boldsymbol{\omega}^T \left[\mathbf{m}_i; \mathbf{c}'_i\right] + b$$

with $\boldsymbol{\omega} \in \mathcal{M}_{\left(|\mathcal{N}|+\sum_{j\in\mathcal{C}} k_j\right),1}(\mathbb{R})$ and $b \in \mathbb{R}$.

Multi-layer Perceptron (MLP)

The MLP was designed to have a single hidden layer containing half the number of features of the input layer. Its architecture is defined in order to have

$$y_i = \boldsymbol{\omega}^T \left(\sigma\left(\mathbf{W}\left[\mathbf{m}_i; \mathbf{c}'_i\right] + \mathbf{b}_1\right)\right) + b_2,$$

where $\mathbf{W} \in \mathcal{M}_{round(n/2),n}(\mathbb{R})$, $\boldsymbol{\omega} \in \mathcal{M}_{round(n/2),1}(\mathbb{R})$, $\mathbf{b}_1 \in \mathcal{M}_{round(n/2),1}(\mathbb{R})$, $b_2 \in \mathbb{R}$, $n = |\mathcal{N}| + \sum_{j\in\mathcal{C}} k_j$ and σ is a parametric rectified linear unit (PReLU).

Graph convolutional network (GCN)

Considering that an oriented graph is attached to a dataset, meaning that each data point $\mathbf{x}_i$ is associated to a node v_i. We denote $\mathcal{S}_i$ the set of nodes v_j such that an oriented edge exist from v_j to v_i and d_{ji} the weight associated to this edge.

Following the Jumping Knowledge framework [80], we implemented the GCN [34] such that

$$y_i = \boldsymbol{\omega}^T \left([\mathbf{m}_i; \mathbf{c}'_i; \mathbf{x}'_i] \right) + b_2 \tag{A.1}$$

with

$$\mathbf{x}'_i = \sigma \left(\frac{1}{\sum_{j \in \mathcal{S}_i} d_{ji}} \sum_{j \in \mathcal{S}_i} d_{ji} \left(\mathbf{W} \left[\mathbf{m}_j, \mathbf{c}'_j \right] + \mathbf{b}_1 \right) \right), \tag{A.2}$$

where $\mathbf{W} \in \mathcal{M}_{n,n}(\mathbb{R})$, $\boldsymbol{\omega} \in \mathcal{M}_{2n,1}(\mathbb{R})$, $\mathbf{b}_1 \in \mathcal{M}_{n,1}(\mathbb{R})$, $b_2 \in \mathbb{R}$, $n = |\mathcal{N}| + \sum_{j \in \mathcal{C}} k_j$ and σ is a ReLU.

Graph attention network (GAT)

We designed the GAT [76] by using the equation (A.1) but replacing equation (A.2) by

$$\mathbf{x}'_i = \sigma \left(\sum_{j \in \mathcal{S}_i} \alpha_{ij} \left(\mathbf{W} \left[\mathbf{m}_j, \mathbf{c}'_j \right] + \mathbf{b}_1 \right) \right),$$

where α_{ij} represents the attention score given to node v_j by the node v_i and is calculated with a single attention-head of the mechanism introduced by Velickovic *et al.* [76].

Gene graph attention encoder (GGAE)

Considering that the set of categorical features is composed by the set of SNPs and the set of categorical features that are not SNPs ($\mathcal{C} = \mathcal{C}_{SNPs} \cup \mathcal{C}_{\neg SNPs}$). We further define $\mathcal{C}_{SNPs} = \bigcup_k \mathcal{C}^{(k)}_{SNPs}$ where $\mathcal{C}^{(k)}_{SNPs}$ is the set of SNPs belonging in the chromosome pair k.

Following this notation, the GGAE first calculates the *chromosomal embeddings* $\mathbf{z}_i^{(k)} \in \mathcal{M}_{3,1}(\mathbb{R})$ linked to a data point $\mathbf{x}_i$:

$$\mathbf{z}_i^{(k)} = \sigma\left(\sum_{j \in \mathcal{C}_{SNPs}^{(k)}} \alpha_{ij}^{(k)} \mathbf{h}_{ij}\right),$$

with

$$\alpha_{ij}^{(k)} = \frac{exp\left(\gamma\left(\mathbf{a}^{(k)^T}\mathbf{h}_{ij}\right)\right)}{\sum_{p \in \mathcal{C}_{SNPs}^{(k)}} exp\left(\gamma\left(\mathbf{a}^{(k)^T}\mathbf{h}_{ip}\right)\right)},$$

such that $\mathbf{a}^{(k)} \in \mathcal{M}_{3,1}(\mathbb{R})$ is the attention mechanism specific to the chromosome pair k, $\mathbf{h}_{ij} \in \mathcal{M}_{3,1}(\mathbb{R})$ is the categorical embedding (i.e., *SNP embedding*) associated to SNPs j, σ is a ReLU and γ is a LeakyReLU.

Then, the *genomic signature* $\mathbf{s}_i$ is calculated:

$$\mathbf{s}_i = \mathbf{W}\left(\sigma\left(\sum_k \beta_{ik}\mathbf{z}_i^{(k)}\right)\right),$$

with

$$\beta_{ik} = \frac{exp\left(\gamma\left(\mathbf{b}^T\mathbf{z}_i^{(k)}\right)\right)}{\sum_\ell exp\left(\gamma\left(\mathbf{b}^T\mathbf{z}_i^{(\ell)}\right)\right)},$$

where $\mathbf{W} \in \mathcal{M}_{s,1}(\mathbb{R})$, $\mathbf{b} \in \mathcal{M}_{3,1}(\mathbb{R})$ is the final attention mechansim, s is the *signature size*, σ is a ReLU and γ is a LeakyReLU.

A.2.3 Hyperparameters of the models

In this subsection we present further characteristics concerning the hyperparameters of the models. We begin by providing a description of the latter. We further show the manually selected value associated to each of them as well as their respective search spaces. We conclude this subsection by giving additional details about the algorithm used to proceed to the hyperparameter optimization.

Descriptions of the hyperparameters

Here, we enumerate and give a description of the hyperparameters associated to the models we implemented. All hyperparameters are associated to the models in which they are included in Table A.19. For the description of the hyperparameters associated to the random forest and XGBoost, please refer to the documentation of versions 0.24.1 and 1.4.2 of *scikit-learn* [63] and *xgboost* [13] libraries.

attention dropout	Probability of randomly setting an attention score α_{ij} to 0 for any node v_i in the execution of a forward pass of the GAT during its training.
batch size	Number of elements in each batch during the training of a model.
β	L2 penalty coefficient for the regularization term in the mean squared error loss used during the training of a model.
dropout	Probability of randomly setting a feature (i.e., a neuron) of the hidden layer to 0 in the execution of a forward pass of the model during its training.
feature dropout	Probability of randomly setting a feature (i.e., a neuron) of the input layer to 0 in the execution of a forward pass of the GAT during its training.
genomic dropout	Probability of randomly setting a feature within a *SNP embedding* or a *chromosomal embedding* to 0 during the execution of a forward pass of the linear regression with GGAE during its training.
learning rate	Step size parameter α of the Adam optimizer [31].
number of neighbors	In-degrees of each node in the oriented graph connecting the patients (without considering each node's self-connection).
signature dropout	Probability of randomly setting a feature (i.e., a neuron) of the *genomic signature* to 0 in the execution of a forward pass of the linear regression with GGAE during its training.
signature size	Number of components in the *genomic signature*.

Hyperparameter	Models
attention dropout	GAT
batch size	Linear regression, MLP, Linear regression + GGAE
β	All*
dropout	MLP, GAT, GCN, Linear regression + GGAE
feature dropout	GAT
genomic dropout	Linear regression + GGAE
learning rate	All*
number of neighbors	GAT, GCN
signature dropout	Linear regression + GGAE
signature size	Linear regression + GGAE

Table A.19 – List of the hyperparameters and the models to which they are associated. *****Models implemented with *Pytorch* and *DGL*.

Values and search spaces of the hyperparameters

In the following tables, we report the sets of manually selected hyperparameter values and hyperparameters' search spaces that led to the results obtained for each model. All models implemented with *Pytorch* and *DGL* were trained using the Adam optimizer with a maximal budget of 500 epochs. The training of each of these models was prematurely stop if no improvement of the root-mean-square error was seen on the validation set for 50 consecutive epochs.

Hyperparameter	Manually selected value	Search space
max_features	sqrt	$\{\text{sqrt}, \log 2\}$
max_leaf_nodes	25	$\{5k\}_{k=1}^{10}$
max_samples	0.8	$[0.8, 1]$
n_estimators	2000	$\{1000 + 250k\}_{k=0}^{8}$

Table A.20 – Random forest's hyperparameters. The hyperparameters that are not mentioned were set as the default ones from version 0.24.1 of the *scikit-learn* library.

Hyperparameter	Manually selected value	Search space
max_depth	2	$\{k\}_{k=1}^{5}$
learning_rate	0.1	$[5 \times 10^{-3}, 0.1]$
reg_lambda	5×10^{-4}	$[5 \times 10^{-4}, 1]$
subsample	0.8	$[0.8, 1]$

Table A.21 – XGBoost's hyperparameters. The hyperparameters that are not mentioned were set as the default ones from the scikit-learn wrapper interface of version 1.4.2 of the *xgboost* library.

Hyperparameter	Manually selected value	Search space
batch size	5	$\{5k\}_{k=1}^{5}$
β	5×10^{-4}	$[5 \times 10^{-4}, 1]$
learning rate	0.01	$[5 \times 10^{-3}, 0.1]$

Table A.22 – Linear regression's hyperparameters.

Hyperparameter	Manually selected value	Search space
batch size	5	$\{5k\}_{k=1}^{5}$
β	5×10^{-4}	$[5 \times 10^{-4}, 1]$
dropout	0.25	$[0, 0.25]$
learning rate	1×10^{-3}	$[1 \times 10^{-3}, 0.1]]$

Table A.23 – MLP's hyperparameters.

Hyperparameter	Manually selected value	Search space
β	5×10^{-3}	$[5 \times 10^{-4}, 1]$
dropout	0.25	0.25
learning rate	0.1	$[5 \times 10^{-3}, 0.1]$
number of neighbors	8	8

Table A.24 – GCN's hyperparameters.

Hyperparameter	Manually selected value	Search space
attention dropout	0.5	0.5
β	5×10^{-3}	$[5 \times 10^{-4}, 1]$
dropout	0.25	0.25
feature dropout	0	$[0, 0.25]$
learning rate	0.1	$[5 \times 10^{-3}, 0.1]$
number of neighbors	$6^{[**]}$, $10^{[*,***]}$	$6^{[**]}$, $10^{[*,***]}$

Table A.25 – GAT's hyperparameters. *: Value selected for the VO_2 peak problem. **: Value selected for the early obesity problem (w/o SNPs). ***: Value selected for the early obesity problem (w/ SNPs)

Hyperparameter	Manually selected value	Search space
batch size	5	$\{5k\}_{k=1}^{5}$
β	5×10^{-4}	$[5 \times 10^{-4}, 1]$
genomic dropout	0.25	[0, 0.25]
learning rate	0.1	$[5 \times 10^{-3}, 0.1]]$
signature dropout	0.25	0.25
signature size	4	4

Table A.26 – Linear regression + GGAE's hyperparameters.

A.2.4 Tree-structured Parzen estimator (TPE)

In this subsection, we expose additional details about the parameters (Table A.27) and the statistical distributions (Table A.28) used to run the hyperparameter optimization with the TPE algorithm [6]. Additionally, an illustration of the automated hyperparameter optimization process is displayed (Supplementary Figure A.3)

Parameter	Value
Expected improvement candidates	20
Startup trials	20

Table A.27 – Parameters configuration for the TPE algorithm.

Type	Distribution
Integer ($\mathbb{Z}$)	Integer Uniform
Real ($\mathbb{R}$)	Uniform

Table A.28 – Statistical distributions used to sample hyperparameters according to their types.

A.3 Supplementary background

This section contains supplementary information of a precedent study addressing the creation of models to predic LAEs.

A.3.1 Disease-specific VO_2 peak equation

Here, we present the disease-specific VO_2 peak equation created by Labonté *et al.* [37].

$$\begin{aligned} VO_2\ peak\ (mL/kg/min) &= -0.236 \times Age\ (years) - 0.094 \times Weight(kg) \\ &\quad - 0.120 \times HRend\ (bpm) + 0.067 \times 6WMD\ (m) \\ &\quad + 0.065 \times MVLPA\ (min/day) \\ &\quad - 0.204 \times DT\ (years) + 25.145 \end{aligned}$$

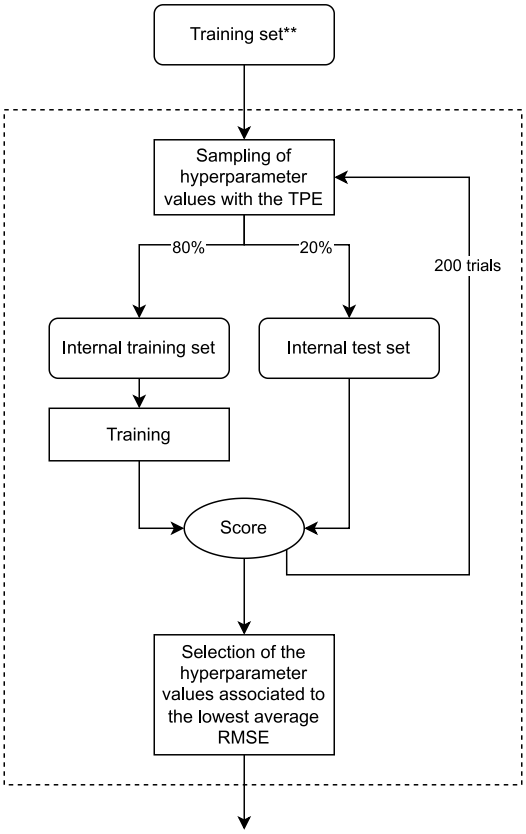

Figure A.3 – Automated hyperparameter optimization with the Tree-Structured Parzen estimator algorithm (TPE). 200 sets of hyperparameter values are sampled sequentially using the TPE and evaluated on the ten same *internal train sets* and *internal test sets*. The set of hyperparameter values associated to the lowest average RMSE is selected. The dashed rectangle refers to the feature selection steps of boxes 2 and 4 in Figure 3.1**b** of the main text. **: training set after the feature selection.

Annexe B

Définitions mathématiques

Cette annexe contient les définitions mathématiques en support à certains concepts mentionnés au cours de ce mémoire.

B.1 Norme maximale

Soit $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{R})$, considérons la notation $\mathbf{A}_{i,j}$ pour identifier le coefficient présent aux coordonnées i, j de cette matrice. La norme maximale de $\mathbf{A}$ correspond à

$$\|\mathbf{A}\|_{max} = \max_{i,j} |\mathbf{A}_{i,j}|.$$

B.2 Règle de dérivation en chaine

Soit $f(\cdot)$ et $g(\cdot)$ deux fonctions et h leur composition telle que $h(x) = f(g(x))$. Suivant la notation de Leibniz, la dérivée de h par rapport à x se calcule

$$\frac{dh}{dx} = \frac{df}{dg} \times \frac{dg}{dx}.$$

B.3 Minimum global et local

Nous nommons $x^* \in \mathcal{A}$ un minimum global de la fonction $f : \mathcal{A} \to \mathbb{R}$ si $f(x^*) \leq f(x)\ \forall x \in \mathcal{A}$. Pour cette même fonction, nous nommons x' un minimum local de f s'il existe $\epsilon > 0$ tel que $f(x') \leq f(x)\ \forall x \in \mathcal{A} \cap [x - \epsilon, x + \epsilon]$.

B.4 Leaky ReLU

Nous nommons *Leaky ReLU* la fonction $f : \mathbb{R} \to \mathbb{R}$ se définissant par

$$f(x) = \begin{cases} x & \text{si } x > 0 \\ 0.01x & \text{sinon} \end{cases}.$$

B.5 Prototypes d'architectures d'encodeur et de décodeur

Les définitions mathématiques des prototypes d'architectures d'encodage et de décodage utilisées avec l'ensemble des polymorphismes à un seul nucléotide sont présentées ci-dessous. En outre la fonction de perte employé pour entraîner conjointement ces architectures est également détaillée.

Chaque polymorphisme à un seul nucléotide (PSN) constitue une variation génétique se trouvant au sein d'une pair de chromosomes. Du point de vue des données, chaque PSN représente une variable catégoriques j prenant la modalité entière 1, 2 ou 3. Pour la suite de cette sous-section, nous noterons $\mathcal{C}_{PSNs}$ l'ensemble des PSN observés dans un ensemble de données, tel que $\mathcal{C}_{PSNs} = \bigcup_k \mathcal{C}^{(k)}_{PSNs}$ où $\mathcal{C}^{(k)}_{PSNs}$ est l'ensemble de PSN associés à la pair de chromosome k. Nous noterons également par $\mathbf{x}_i$ la représentation vectoriel d'un individu au sein d'une base de données et $\mathbf{x}_{ij}$ la valeur qu'il associe à la variable j.

Encodeur

L'encodage de l'ensemble des PSNs d'un individu comporte quelques étapes. D'abord, une représentation vectorielle $\mathbf{h}_{ij} \in \mathcal{M}_{3,1}(\mathbb{R})$ est calculée pour chaque PSN $j \in \mathcal{C}_{PSNs}$.

Précisément, nous obtenons

$$\mathbf{h}_{ij} = \mathbf{W}_{emb}^{(j)} \mathbf{o}_{ij},$$

où $\mathbf{W}_{emb}^{(j)} \in \mathcal{M}_{3,3}(\mathbb{R})$ est une matrice de paramètres à optimisés et $\mathbf{o}_{ij} \in \mathcal{M}_{3,1}(\{0,1\})$ est un encodage *one-hot*, c'est-à-dire un vecteur unitaire dont la seule valeur non nulle se situe à la position représentée par la valeur entière $\mathbf{x}_{ij}$.

Ensuite, une représentation vectorielle de chaque pair de chromosomes est calculée en prenant la moyenne des représentations des PSNs associés à celle-ci et en lui appliquant une fonction d'activation non linéaire. Cette représentation, notée $\mathbf{z}_i^{(k)} \in \mathcal{M}_{3,1}(\mathbb{R})$, est définie par le calcul

$$\mathbf{z}_i^{(k)} = \text{ReLU}\left(\sum_{j \in \mathcal{C}_{PSNs}^{(k)}} \frac{1}{|\mathcal{C}_{PSNs}^{(k)}|} \mathbf{h}_{ij} \right).$$

Pour terminer, un encodage final de l'ensemble des PSNs, noté $\mathbf{s}_i$, est obtenu en effectuant les opérations

$$\mathbf{s}_i = \mathbf{W}\left(\text{ReLU}\left(\left[\|_k \boldsymbol{\omega}^T \mathbf{z}_i^{(k)}\right]\right)\right),$$

où $\boldsymbol{\omega} \in \mathcal{M}_{3,1}(\mathbb{R})$, $\mathbf{W} \in \mathcal{M}_{s,K}(\mathbb{R})$ sont des matrices de paramètres à optimisés, s est la taille d'encodage souhaitée et K correspond au nombre de pairs de chromosomes observés dans la base de données.

Décodeur

La phase de décodage a pour objectif de produire une approximation des différents vecteurs $\mathbf{o}_{ij}$ d'un individu à partir de l'encodage $\mathbf{s}_i$ qui lui est associé. La fonction de décodage est définie

$$\text{Dec}(\mathbf{s}_i) = \text{Sigmoid}\left(\left(\text{ReLU}\left(\boldsymbol{\omega}^{(1)} {\mathbf{s}_i}^T\right)\mathbf{v}\right)\boldsymbol{\omega}^{(2)^T}\right)$$

avec $\boldsymbol{\omega}^{(1)} \in \mathcal{M}_{|\mathcal{C}_{PSNs}|,1}(\mathbb{R})$, $\boldsymbol{\omega}^{(2)} \in \mathcal{M}_{3,1}(\mathbb{R})$ des vecteurs de paramètres à optimisés et $v \in \mathcal{M}_{s,|1|}(\mathbb{R})$ un vecteur dont l'ensemble des éléments correspondent à la valeur $1/s$. Chaque ligne de la matrice résultante de ces opérations correspond à une

approximation d'un vecteur unitaire représentant un PSN, c'est-à-dire

$$\mathrm{Dec}(\mathbf{s}_i) = [\hat{\mathbf{o}}_{ij}]_{j \in \mathcal{C}_{PSNs}} \in \mathcal{M}_{|\mathcal{C}_{PSNs}|,3}(\mathbb{R}).$$

Perte d'entraînement non supervisée

Soit $\mathbf{o}_i = [\|_{j \in \mathcal{C}_{PSNs}} \mathbf{o}_{ij}]$ la concaténation de l'ensemble des encodages *one-hot* représentant les PSNs d'un individu i et N le nombre d'individus total dans un ensemble d'entraînement. Nous définissons la perte associée à cet ensemble par

$$\frac{1}{C}\left(\sum_{i=1}^{N}\sum_{k=1}^{N} \mathrm{jacc}\left(\mathbf{o}_i, \mathbf{o}_k\right) \cdot \left\|\mathbf{s}_i - \mathbf{s}_k\right\|^2\right) + \frac{1}{N}\left(\sum_{i=1}^{N} \left\|\mathbf{o}_i - \hat{\mathbf{o}}_i\right\|^2\right),$$

où $\mathrm{jacc}\left(\cdot,\cdot\right)$ est la distance de Jaccard et $C = 1/\sum_{i=1}^{N}\sum_{k=1}^{N} \mathrm{jacc}\left(\mathbf{o}_i, \mathbf{o}_k\right)$ est une constante de normalisation. La première partie vise à ce que la distance entre les encodages de deux individus soit petite si ceux-ci partage des PSNs semblables. La seconde partie, quant à elle, valorise la qualité de reconstruction des encodages *one-hot*.