الجمهورية الجزائرية الديمقراطية الشعبية

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 8 Mai 1945 - Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la Matière

Département d'Informatique


## Mémoire de Fin d'Études – Master

**Filière : Informatique**

**Option : Science et Technologie de l'Information et de la Communication**

---

# Prédiction des maladies d'insuffisance cardiaque en utilisant des modèles de Deep Learning

---

**Présenté par :** Zahra BOAOUANE

**Les membres du jury :**

**Président :** Dr. Said BRAHIMI

**Encadreur :** Prof. Dr. Ouarda ZEDADRA

**Examinateur :** Dr. Saida BAALIA

**Juin 2025**

# Remerciement

Avant tout, je rends grâce à Dieu, qui m'a accordé la force, la patience et la persévérance nécessaires pour mener à bien ce travail. *Alhamdoulillah*, j'ai enfin atteint cette étape tant attendue.

Je tiens à exprimer toute ma gratitude à ma directrice de mémoire, *Professeure ZEDADRA Ouarda*, pour son accompagnement précieux, sa disponibilité constante, ainsi que pour ses conseils avisés et son encadrement rigoureux tout au long de ce travail.

Je remercie également les membres du jury pour l'honneur qu'ils m'ont fait en acceptant d'évaluer ce mémoire, et pour leurs remarques constructives qui ont permis d'enrichir cette recherche.

Enfin, je tiens à remercier chaleureusement toutes les personnes qui m'ont soutenu, de près ou de loin, tout au long de mon parcours universitaire.

# Dédicace

Je dédie ce travail à toutes les personnes qui m'aiment et qui ont toujours été à mes côtés.
À ma chère famille, pour son soutien constant, son amour inconditionnel et sa patience envers moi, particulièrement durant ces cinq dernières années. C'est grâce à vos prières et à votre présence que j'ai pu atteindre cette étape.
À mon père **Rachid** bien-aimé, qui a été mon pilier tout au long de ce parcours, m'accompagnant dans chaque détail, m'encourageant sans relâche… Merci infiniment pour tout.
À ma chère mère **Chaib Nora**, pour ses mots réconfortants et son soutien indéfectible, qui me rappelait toujours que le chemin était bientôt terminé. Que Dieu te garde pour moi.
À mes frères **Hani** et **Haroun**, je vous envoie tout mon amour et ma gratitude.
À toute ma grande famille, pour sa présence précieuse et son soutien dans toutes les circonstances.
À mes amies d'enfance, de parcours scolaire et à celles rencontrées à l'université… Je vous souhaite à toutes réussite et bonheur dans vos vies.
À toutes les personnes qui m'ont soutenue d'un mot gentil, d'un conseil ou d'une prière… Recevez toute ma reconnaissance.

À la mémoire de mon oncle bien-aimé **Mohammed Chaib**, mon second père, qui nous a quittés la veille de cette soutenance. Sa disparition soudaine a été un choc pour nous tous. J'étais impatiente de partager ce moment de réussite avec toute la famille, mais telle est la volonté de Dieu. Alhamdoulillah en toute chose. Qu'Allah lui fasse miséricorde, lui pardonne ses fautes, élève son rang parmi les justes, et lui ouvre les portes de Son vaste paradis.

Enfin, un immense merci à moi-même, pour avoir persévéré et continué malgré tout. Merci *Zahrati*.

**Zahra**

# Résumé

L'insuffisance cardiaque constitue l'un des défis majeurs dans le domaine médical en raison de sa gravité et de sa prévalence croissante. Dans le cadre de ce projet, notre objectif est de développer un modèle d'intelligence artificielle capable de prédire les cas d'insuffisance cardiaque en combinant des images ECG et des données cliniques. Pour cela, nous avons utilisé les Graph Neural Networks (GNN), notamment les modèles GCN et GAT. Les patients et leurs données ont été représentés sous forme de graphe afin d'exploiter les relations entre eux. Les expérimentations ont montré que le modèle GCN a offert de meilleures performances par rapport aux autres modèles utilisés. Ce travail met en évidence l'efficacité de l'intégration des données médicales visuelles et cliniques dans une approche basée sur les graphes, et ouvre la voie à une utilisation plus large de ces techniques dans les systèmes d'aide à la décision médicale.

**Mots clés :** Insuffisance cardiaque, Intelligence artificielle, Graph Neural Networks, ECG, Données cliniques, Prédiction.

# Abstract

Heart failure is considered one of the major challenges in the medical field due to its severity and increasing prevalence. In this project, our objective was to develop an artificial intelligence model capable of predicting heart failure cases by combining ECG images with clinical data. To achieve this, we employed Graph Neural Networks (GNN), particularly the GCN and GAT models. Patients and their data were represented as a graph structure to effectively exploit the relationships between them. The experimental results showed that the GCN model delivered better performance compared to the other models used. This work highlights the effectiveness of integrating visual and clinical medical data within a graph-based approach and paves the way for broader adoption of these techniques in medical decision support systems.

**Keywords :** Heart failure, Artificial intelligence, Graph Neural Networks, ECG, Clinical data, Prediction.

# الملخص

تُعدّ الإصابة بفشل القلب من أبرز التحديات في المجال الطبي، نظراً لخطورتها وانتشارها المتزايد. وفي إطار هذا المشروع، كان هدفنا تطوير نموذج ذكاء اصطناعي قادر على التنبؤ بحالات فشل القلب من خلال دمج صور تخطيط القلب الكهربائي (ECG) مع البيانات السريرية. ولتحقيق ذلك، تم الاعتماد على تقنيات الشبكات العصبية البيانية (GNN)، وبشكل خاص على نموذجي GCN وGAT، حيث جرى تمثيل المرضى وبياناتهم في شكل رسم بياني يسمح باستغلال العلاقات والروابط فيما بينهم. وأظهرت نتائج التجارب أن نموذج GCN قدّم أداءً أفضل مقارنة بالنماذج الأخرى المستخدمة. يُبرز هذا العمل فعالية دمج البيانات الطبية الصورية والسريرية ضمن مقاربة تعتمد على الرسوم البيانية، ويفتح آفاقًا واسعة نحو توسيع استخدام هذه التقنيات في أنظمة دعم اتخاذ القرار الطبي.

**الكلمات المفتاحية:** فشل القلب، الذكاء الاصطناعي، الشبكات العصبية البيانية، تخطيط القلب الكهربائي، البيانات السريرية، التنبؤ.

# Table des matières

# Table des figures

# Liste des tableaux

Liste des tableaux

# Liste des Abréviations

**IA** Intelligence Artificielle

**ECG** Electrocardiogramme

**GNN** Graph Neural Network

**GAT** Graph Attention Network

**GCN** Graph Convolutional Network

**AVC** Accident Vasculaire Cérébral

**SCA** Syndromes Coronariens Aigus

**ML** Machine Learning

**CNN** Convolutional Neural Network

**ResNet** Residual Network

**RNN** Recurrent Neural Network

**LSTM** Long Short-Term Memory

**DNN** Deep Neural Network

**KNN** K-Nearest Neighbors

**DT** Decision Tree

**SVM** Support Vector Machine

**MLP** Multi-Layer Perceptron

**ReLU** Rectified Linear Unit

**Pooling** Opération de sous-échantillonnage dans les CNN

**Multi-head attention** Mécanisme d'attention multi-tête

**Softmax** Fonction d'activation transformant un vecteur en probabilités

**LeakyReLU** Leaky Rectified Linear Unit

**IRM** Imagerie par Résonance Magnétique

**DES** Dossiers de Santé Électroniques

**AUC** Area Under Curve

**GRU** Gated Recurrent Unit

**ANN** Artificial Neural Network

**UCI** University of California Irvine

**CHD** Coronary Heart Disease (maladie corona-rienne)

**Adam** Adaptive Moment Estimation

**EHR** Electronic Health Record

**Word2Vec** Technique de vectorisation des mots

**chi²** Test statistique du chi carré

**BiLSTM** Bidirectional Long Short-Term Memory

**BiGRU** Bidirectional Gated Recurrent Unit

**SMOTE** Synthetic Minority Over-sampling Technique

**HBA** Honey Badger Algorithm

**R-CNN** Region-based Convolutional Neural Network

**DCT** Discrete Cosine Transform

**FFT** Fast Fourier Transform

**PPV** Positive Predictive Value

**BiLSTM-BO** Bidirectional LSTM with Bayesian Optimization

**DCNN** Deep Convolutional Neural Network

**MNN** Multilayer Neural Network

**MIMIC-II** Multiparameter Intelligent Monitoring in Intensive Care II

**GT** Graph Transformer

**IC** Insuffisance Cardiaque

**Z-score** Score de normalisation statistique

**GIN** Graph Isomorphism Networks

**RMSProp** Root Mean Square Propagation

**Adadelta** Adaptive Delta Optimization

**Adagrad** Adaptive Gradient Algorithm

**SGD** Stochastic Gradient Descent

**EF** Fraction d'Éjection

**BNP** Brain Natriuretic Peptide

**NYHA** New York Heart Association

**SBP** Systolic Blood Pressure

**RGB** Red Green Blue (codage couleur)

# Introduction Générale

L'insuffisance cardiaque constitue l'un des problèmes de santé les plus graves à l'échelle mondiale, en raison des dommages structurels et fonctionnels qu'elle engendre au niveau du muscle cardiaque. Elle résulte d'un ensemble de facteurs et de causes qui perturbent les fonctions de contraction ou de relaxation ventriculaire. L'insuffisance cardiaque est considérée comme le stade ultime de l'évolution de diverses maladies cardiovasculaires. Selon les statistiques de l'American College of Cardiology, les maladies cardiovasculaires sont responsables d'un tiers des décès dans le monde, avec environ 550 000 nouveaux cas diagnostiqués chaque année. En Chine seulement, près de 1,3 million de personnes âgées de plus de 35 ans souffrent d'insuffisance cardiaque, ce qui illustre la propagation croissante de cette pathologie. Par conséquent, l'insuffisance cardiaque est devenue un véritable enjeu de santé publique à l'échelle mondiale.

Le diagnostic de cette maladie nécessite de réaliser plusieurs examens, tels que la mesure de la tension artérielle, du taux de sucre sanguin, des signes vitaux, l'électrocardiogramme (ECG), l'analyse de l'historique médical du patient et divers tests cliniques complémentaires. Cependant, ces procédures peuvent être sujettes à des erreurs, nécessitent parfois un temps important et génèrent des coûts considérables, estimés à environ 29 milliards de dollars par an [42].

Dans ce contexte, les dernières années ont connu un recours croissant aux techniques d'intelligence artificielle, notamment celles de l'apprentissage automatique et de l'apprentissage profond, afin d'aider à la prédiction des maladies cardiovasculaires et de l'insuffisance cardiaque. Ces technologies reposent sur l'analyse et le traitement des données médicales des patients, dans le but de construire des modèles prédictifs performants et précis, capables d'assister les médecins dans la prise de décision thérapeutique, d'améliorer la qualité de vie des patients et de réduire les dépenses de santé.

Dans le cadre de ce projet, nous avons pour objectif de développer un modèle d'intelligence artificielle basé sur des techniques d'apprentissage profond pour la prédiction de l'insuffisance cardiaque, en combinant des données cliniques et des images d'électrocardiogramme (ECG). Pour ce faire, nous avons adopté les réseaux de neurones graphiques (Graph Neural Networks) en utilisant les modèles GCN et GAT, afin de représenter les patients et leurs données sous forme de graphe exploitant les relations qui existent entre

eux.

L'organisation de ce travail s'articule autour de quatre chapitres principaux :

**Chapitre 1 :** présente l'insuffisance cardiaque et l'intelligence artificielle. Il définit les types, causes et symptômes de la maladie, ainsi que les méthodes de diagnostic et leurs limites. Ensuite, il introduit les concepts d'IA, d'apprentissage automatique, de deep learning et de réseaux de neurones graphiques, avec leurs applications médicales.

**Chapitre 2 :** propose une revue des travaux existants sur la prédiction de l'insuffisance cardiaque, en comparant les principales approches et en discutant leurs limites. Il se conclut par des suggestions pour améliorer la précision des méthodes actuelles.

**Chapitre 3 :** décrit la conception du système proposé : description des données, prétraitement, extraction des caractéristiques ECG avec ResNet18, création d'un graphe de similarité et présentation des modèles GCN et GAT utilisés, ainsi que l'entraînement.

**Chapitre 4 :** détaille l'implémentation : environnement de développement, traitement des données, fusion des caractéristiques, construction du graphe et comparaison des résultats expérimentaux. Enfin, il présente une interface interactive de prédiction à partir d'une image ECG et de données cliniques.

Et nous clôturons cette étude par une conclusion générale et des perspectives, ouvrant la voie à des travaux futurs qui pourront être réalisés par d'autres étudiants.

# Chapitre 1

# État de l'art : Insuffisance Cardiaque et Intelligence Artificielle

## 1.1 Introduction

Les maladies cardiovasculaires sont la première cause de décès dans le monde, causant environ 17,9 millions de cas par an (selon l'Organisation mondiale de la santé). Parmi elles, l'insuffisance cardiaque, qui est considérée comme la dernière étape du développement des maladies cardiaques[42] , est difficile à diagnostiquer et nécessite de nombreux indicateurs biologiques et facteurs de risque, y compris l'âge, le sexe, l'hypertension artérielle, le diabète, le taux de cholestérol et de nombreux autres indicateurs cliniques[7].
Ces dernières années, les chercheurs se sont concentrés sur des méthodes pour réduire ces risques ou les prédire à l'aide de l'intelligence artificielle, en particulier les méthodes d'apprentissage profond. Dans ce chapitre, nous en apprenons davantage sur les principaux types de maladies cardiovasculaires en nous concentrant sur l'insuffisance cardiaque.
Ce chapitre est divisé en deux grandes parties : Aspects médicaux de l'insuffisance cardiaque et l'intelligence artificielle.
Dans la première partie, nous abordons les maladies cardiaques en présentant leur définition, les différents types existants, notamment l'insuffisance cardiaque et ses classifications. Nous détaillons également les facteurs de risque, les causes de l'insuffisance cardiaque ainsi que les symptômes associés. Ensuite, nous présentons les méthodes classiques de diagnostic utilisées en cardiologie : les examens cliniques (ECG, échocardiogramme, échographie Doppler), les biomarqueurs (comme la troponine), ainsi que les limites des approches traditionnelles.
La seconde partie est consacrée à l'intelligence artificielle et modélisation prédictive. Elle débute par une définition de l'IA, suivie d'une présentation de son intérêt croissant dans le diagnostic médical, puis de l'apprentissage automatique (machine learning) et de ses

types. Nous y introduisons ensuite les réseaux de neurones artificiels, puis l'apprentissage profond (deep learning) et ses différentes catégories.
Enfin, ce chapitre présente également les Graph Neural Networks (GNN), en abordant leur définition, les différents types existants, leurs applications potentielles, ainsi que leurs avantages et leurs limites. Cette approche novatrice sera exploitée dans la suite de ce travail.

# Partie I – Aspects médicaux de l'insuffisance cardiaque

## 1.2 Les maladies cardiaques

Les maladies cardiovasculaires comprennent toutes les maladies du cœur et des vaisseaux sanguins, qui se produisent en raison des dépôts de cholestérol sur les parois des artères. Ces dépôts entravent et bloquent le flux sanguin alimentant le cœur, le cerveau et les jambes, ce qui entraîne des douleurs thoraciques, des crises cardiaques et des AVC (accidents vasculaires cérébraux)[52].

### 1.2.1 Types des maladies cardiaques

Il existe différents types de maladies cardiaques selon la partie du cœur ou des vaisseaux touchée. La Figure1.1 présente une catégorisation des maladies cardiaques existantes, dont l'explication se trouve ci-dessous :

- Maladies coronariennes : atteinte des artères coronaires qui alimentent le muscle cardiaque, ce qui entraîne une ischémie myocardique (manque d'oxygène).
- Cardiomyopathies : dysfonctionnement du muscle cardiaque lui-même.
- Troubles du rythme cardiaque : arythmies qui peuvent parfois être mortelles.
- Insuffisance cardiaque : perte progressive de la capacité du cœur à pomper le sang efficacement[79].
- Maladie cérébro-vasculaire : atteinte des vaisseaux sanguins qui alimentent le cerveau.
- Maladie artérielle périphérique : atteinte des artères qui irriguent les bras et les jambes.
- Cardiopathie rhumatismale : atteinte des valves cardiaques causée par une infection streptococcique[83].

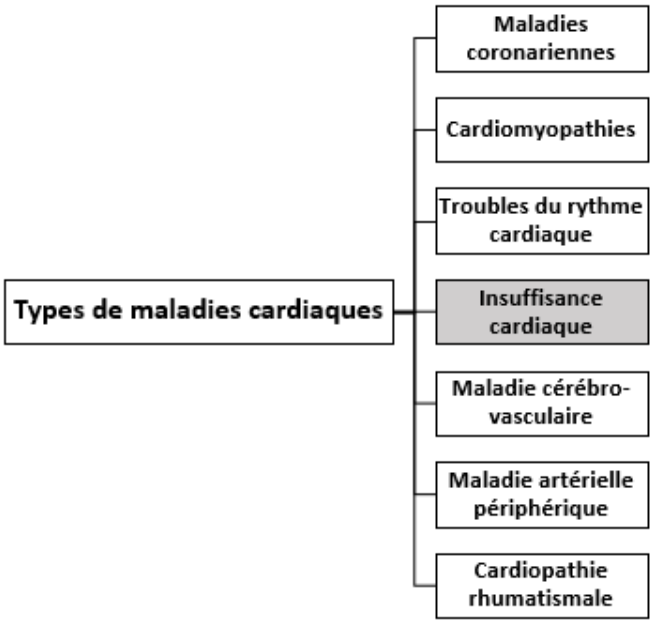



**Figure 1.1 :** Types des maladies cardiaques

### 1.2.2 Insuffisance cardiaque

L'insuffisance cardiaque est un syndrome clinique grave et fréquent qui se produit lorsque le cœur n'est pas capable de pomper le sang de manière efficace, ne pouvant pas fournir une quantité suffisante d'oxygène et de nutriments pour répondre aux besoins du corps. Elle peut également survenir en cas d'accumulation anormale de pression à l'intérieur du cœur[15].

#### 1. Types d'insuffisance cardiaque

L'insuffisance cardiaque peut être classée selon la fraction d'éjection du ventricule gauche (FEVG), qui mesure l'efficacité de pompage du cœur. Les types existantes sont résumés dans la Figure 1.2.

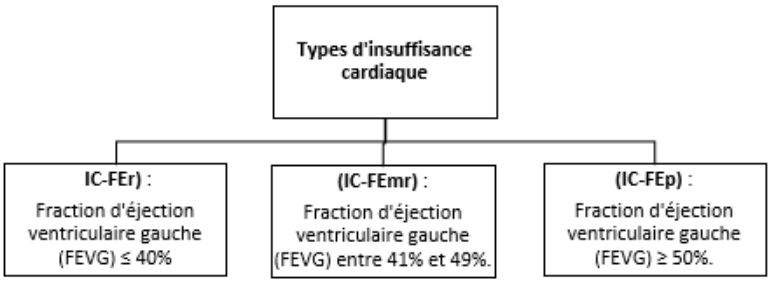



**Figure 1.2 :** Types d'insuffisance cardiaque [74]

## 2. Facteurs de risque de l'insuffisance cardiaque

Plusieurs conditions et habitudes de vie augmentent le risque de développer une insuffisance cardiaque, notamment [65] :

- Hypertension artérielle
- Diabète
- Obésité
- Tabagisme.
- Excès d'alcool
- Dyslipidémie (taux de cholestérol élevé)
- Antécédents familiaux de maladies cardiovasculaires

## 3. Causes de l'insuffisance cardiaque (Étiologie)

L'insuffisance cardiaque peut être causée par plusieurs maladies ou problèmes qui touchent le cœur. Les principales causes sont [65] :

- Maladies coronariennes (ex. : infarctus du myocarde)
- Valvulopathies (maladies des valves cardiaques)
- Arythmies cardiaques
- Maladies du péricarde
- Troubles hormonaux (ex. : dysfonction thyroïdienne)

- Anémie sévère
- Infections et inflammations cardiaques

#### 4. Symptômes de l'insuffisance cardiaque

Les symptômes varient en fonction de la gravité et du type d'insuffisance, voici les symptômes qui indiquent une probabilité de L'insuffisance cardiaque [65] :

- Essoufflement (dyspnée) : au repos ou à l'effort
- Fatigue
- Œdème des chevilles (rétention d'eau)
- Râles crépitants pulmonaires (signes de congestion pulmonaire)
- Tachycardie (rythme cardiaque rapide)
- Hépatomégalie (augmentation du volume du foie due à la congestion sanguine)
- Pression veineuse élevée

Ces symptômes peuvent être difficiles à interpréter, notamment chez les personnes âgées, obèses et les femmes. Il est donc recommandé de confirmer le diagnostic par des tests plus objectifs (échocardiographie, ECG, biomarqueurs, etc.).

## 1.3 Méthodes classiques de diagnostic

Le diagnostic des pathologies cardiaques repose sur un ensemble de méthodes complémentaires, allant de l'examen clinique aux tests médicaux plus approfondis. Voici les principales approches utilisées en pratique courante :

### 1.3.1 Examens cliniques

Les examens cliniques sont essentiels pour évaluer la fonction cardiaque, poser un diagnostic précis et orienter la prise en charge. Voici les principaux examens utilisés en cardiologie :

## 1. Électrocardiogramme (ECG)

L'électrocardiogramme (ECG) est un enregistrement de l'activité électrique du muscle cardiaque effectué à la surface de la peau. Il permet de mesurer la fréquence et le rythme cardiaques, de détecter des anomalies de la conduction électrique, des troubles du rythme, une ischémie myocardique (manque d'oxygène), ou encore des signes d'infarctus du myocarde [24].

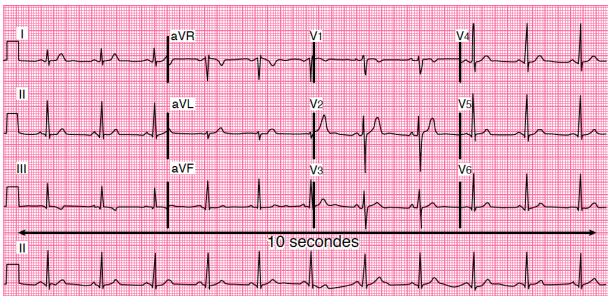



**Figure 1.3 :** Électrocardiogramme (ECG)[53]

## 2. Échocardiogramme

L'échocardiogramme est un examen par ultrasons du cœur, permettant d'évaluer de manière fiable le diamètre des cavités cardiaques, l'épaisseur des parois myocardiques, et de visualiser d'éventuels épanchements péricardiques (liquides autour du cœur). Il y a une autre méthode de mesure : l'échocardiographie Doppler. Elle permet d'analyser la direction et la vitesse du flux sanguin à l'intérieur du cœur, ce qui aide à détecter les anomalies des valves cardiaques, comme les rétrécissements (sténoses) ou les fuites (régurgitations), même de faible intensité[58].

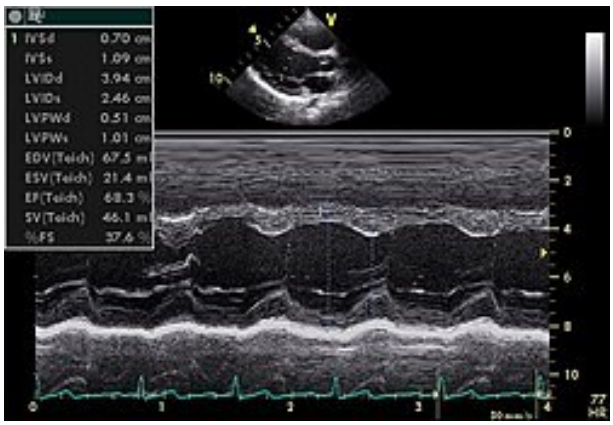


**Figure 1.4 :** Échocardiogramme [41]

**3. Les biomarqueurs**

Les biomarqueurs, en particulier la troponine ultra-sensible (hs-cTn), jouent un rôle essentiel dans la prise en charge des syndromes coronariens aigus (SCA). Cette forme ultra-sensible permet une détection plus précoce et plus précise de la nécrose myocardique, tout en réduisant les délais de surveillance par rapport à la troponine conventionnelle. La troponine cardiaque est un marqueur de lésion myocardique, reconnu pour sa très haute sensibilité et spécificité, ce qui en fait un outil clé pour le diagnostic rapide de l'infarctus du myocarde[45].

### 1.3.2 Limites et difficultés des approches traditionnelles

Malgré leur efficacité prouvée, les méthodes de diagnostic traditionnelles présentent certaines limites techniques et humaines qui peuvent affecter la fiabilité et la rapidité des résultats.
L'interprétation des électrocardiogrammes (ECG) repose sur une analyse approfondie des différentes ondes et intervalles du tracé, nécessitant une expertise médicale confirmée. Cette dépendance à l'expérience du praticien peut conduire à des différences dans la précision diagnostique et influencer la qualité des décisions thérapeutiques [3].
De plus, les opérations de collecte et d'analyse des données sont souvent réalisées manuellement, ce qui peut provoquer des retards dans la prise de décisions médicales cruciales et dans la mise en place des plans de traitement[11].

# Partie II – Intelligence artificielle et modélisation prédictive

# 1 Introduction à l'intelligence artificielle en santé

Les soins de santé dépendaient fortement de l'expérience et des connaissances humaines avant l'apparition de l'intelligence artificielle en médecine, dans lequel les médecins s'appuyaient sur des études antérieures, des expériences personnelles et des évaluations cliniques. Les outils et technologies disponibles étaient limités, ce qui entraînait un manque de précision dans les diagnostics. Avec l'apparition de l'intelligence artificielle, notamment dans le domaine de la cardiologie et de la médecine vasculaire, des avancées majeures ont eu lieu. Cela permet de traiter de grandes quantités de données médicales rapidement et de manière organisée, et ainsi d'améliorer la précision des diagnostics[70].

## 1.1 Définition

L'intelligence artificielle est une science qui fait référence à la simulation de certains aspects de l'intelligence humaine par des machines, en particulier des systèmes informatiques. Cela inclut la résolution de problèmes, l'apprentissage, la prise de décisions, la compréhension du langage naturel et la perception visuelle[44]. L'objectif de l'intelligence artificielle est de permettre aux machines d'effectuer des tâches nécessitant l'intelligence humaine grâce à l'apprentissage automatique et à l'apprentissage profond[17].

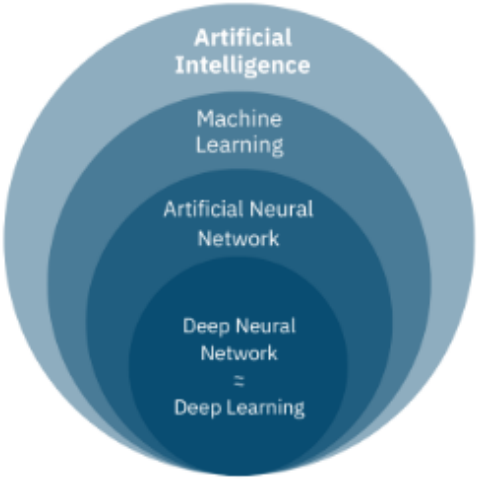



**Figure 1.5 :** Relations entre IA, Machine Learning et Deep Learning[1]

## 1.2 L'intérêt de l'IA dans le diagnostic médical

Face aux limites des approches traditionnelles et grâce aux récentes avancées technologiques, l'intelligence artificielle s'impose aujourd'hui comme une alternative prometteuse et un outil clé dans le domaine du diagnostic médical. L'intelligence artificielle (IA) améliore la rapidité et la fiabilité des diagnostics médicaux. Elle réduit le risque d'erreur humaine grâce à des modèles précis, capables de détecter des signaux faibles dans les données cliniques. Elle permet également de prédire l'apparition de maladies grâce à des modèles prédictifs, facilitant ainsi une prise en charge précoce des patients et allégeant la pression sur les systèmes de santé.

L'IA rend les diagnostics accessibles à l'échelle mondiale, notamment dans les pays en développement ou les zones rurales. Des applications mobiles et dispositifs connectés, soutenus par des algorithmes intelligents, apportent une assistance technique directe aux médecins généralistes et pédiatres sur le terrain [29].

Elle exploite des techniques analytiques avancées pour traiter de grandes quantités de données médicales, avec rapidité et précision. Les algorithmes d'IA reconnaissent des motifs complexes dans les données, améliorant ainsi la précision des diagnostics, accélérant les processus décisionnels et contribuant à de meilleurs résultats thérapeutiques[11].

# 2 Apprentissage automatique et réseaux de neurones

## 2.1 Apprentissage automatique (Machine learning )

Selon Tom M.Mitchell, une référence dans le domaine : "Un programme apprend d'une expérience E par rapport à une tâche T et une mesure de performance P, si sa performance à la tâche T, mesurée par P, s'améliore avec l'expérience E." L'apprentissage automatique (ML) est une branche de l'intelligence artificielle (IA) qui permet aux machines d'apprendre de manière autonome à partir des données et des expériences passées. Il implique l'extraction de motifs à partir des données pour faire des prédictions avec le moins d'intervention humaine possible. Les algorithmes d'apprentissage automatique tirent parti des processus itératifs pour extraire des informations utiles à partir de grands ensembles de données, et apprennent directement à partir des données plutôt que de se fier à des modèles prédéfinis [14].

### 1. Types d'apprentissage automatique

1. **L'apprentissage supervisé :** est un type d'apprentissage automatique où des données étiquetées sont utilisées pour entraîner un modèle à identifier les structures et les relations fondamentales entre les caractéristiques d'entrée et les sorties. L'objectif est de créer un modèle capable de généraliser et de prédire les sorties correctes pour de nouvelles données invisibles [31].

2. **L'apprentissage non supervisé :** est un type d'apprentissage automatique où un ensemble de données non étiquetées est utilisé pour l'entraînement, c'est-à-dire sans sorties cibles connues. L'objectif est d'explorer les données et, à travers elles, de découvrir des motifs et des relations [51].

3. **L'apprentissage semi-supervisé :** est un type d'apprentissage automatique qui combine l'apprentissage supervisé et non supervisé en utilisant des données étiquetées et non étiquetées pour l'entraînement, ce qui aide à améliorer les performances dans les tâches de classification et de régression [30].

4. **L'apprentissage par renforcement :** C'est un type d'apprentissage automatique où l'agent apprend à prendre des décisions en interagissant avec son environnement par essai et erreur, découvrant ainsi quelles actions apportent les plus grandes récompenses. Ce type d'apprentissage repose sur trois composants essentiels : L'agent (celui qui prend les décisions).L'environnement (celui avec lequel l'agent interagit) et Les actions ( ce que l'agent peut accomplir).L'objectif est d'apprendre une stratégie qui maximise la récompense cumulative attendue sur une période de temps

donnée [62].

### 2. Algorithmes classiques de machine learning

L'apprentissage automatique (Machine Learning) repose sur plusieurs algorithmes fondamentaux, tels que le SVM (Support Vector Machine), le KNN (K-Nearest Neighbors) et les arbres de décision.

1. **Les K plus proches voisins (KNN)**

   L'algorithme des k plus proches voisins (KNN) est un algorithme d'apprentissage supervisé utilisé à la fois pour la classification et la régression, même s'il est plus courant en classification. Le principe est simple : pour prédire l'étiquette d'un nouvel échantillon, l'algorithme regarde les k points les plus proches dans les données d'entraînement (selon une distance choisie, souvent la distance euclidienne), puis vote pour la classe majoritaire [67].

2. **Arbre de décision (DT)**

   Un arbre de décision est un algorithme d'apprentissage supervisé, structuré sous forme d'une arborescence comprenant un nœud racine, des branches, et des nœuds feuilles. Les nœuds internes représentent les caractéristiques (ou features) de l'ensemble de données. Les branches correspondent aux résultats des tests effectués sur ces caractéristiques. Chaque nœud feuille indique une prédiction ou une décision finale [55].

3. **Machines à Vecteurs Supports (SVM)** SVM est un algorithme d'apprentissage supervisé utilisé pour la classification et la régression. Il sépare les classes en trouvant un hyperplan qui maximise la marge, c'est-à-dire la distance la plus large possible entre l'hyperplan et les points les plus proches de chaque classe [25].

## 2.2 Réseaux de neurones Artificiels

Les réseaux neuronaux sont une des techniques d'apprentissage automatique inspirées par le cerveau humain. Ils se composent de plusieurs couches de neurones qui reçoivent des entrées et produisent des sorties en effectuant des calculs. L'apprentissage se produit en ajustant les poids des connexions entre les neurones pour réduire la fonction de coût qui mesure la différence entre les valeurs réelles et les prévisions du modèle. Ils sont utilisés pour résoudre des problèmes complexes de classification, de prédiction et de reconnaissance de motifs dans les données[77].

## 2.3 Apprentissage profond (Deep Learning)

L'apprentissage profond représente la dernière avancée en matière d'apprentissage automatique. Il repose sur des algorithmes inspirés du fonctionnement du cerveau humain, permettant aux ordinateurs d'apprendre en simulant certains aspects de l'intelligence humaine. Ces algorithmes sont également appelés réseaux neuronaux profonds ou apprentissage profond[40].

### 1. Types de Deep Learning

Dans l'apprentissage profond, plusieurs algorithmes sont largement utilisés dans diverses tâches et domaines. Parmi les plus notables, on trouve :

1. **CNN ( Convolutional Neural Network) :**

   Un réseau neuronal convolutif (CNN - Convolutional Neural Network) est un type de réseau de neurones artificiel conçu spécifiquement pour analyser les données visuelles. Il est largement utilisé en vision par ordinateur et traitement d'images. Il se compose de couches hiérarchiques comprenant des couches de convolution, des fonctions d'activation non linéaires et des couches de sous-échantillonnage, ce qui lui permet d'extraire automatiquement des caractéristiques sans nécessiter de conception manuelle. Grâce à son architecture profonde et à l'apprentissage par rétropropagation du gradient (backpropagation), le CNN peut reconnaître des motifs complexes et s'adapter à diverses tâches telles que la classification d'images, la segmentation, la détection d'objets et, dans certains cas, le traitement de la vidéo et du langage naturel[33].

   Parmi les architectures CNN les plus performantes et reconnues figure ResNet.

   - **ResNet (Residual Network) :** est une architecture de réseau de neurones profonds introduite par He *et al.* en 2015 [28], qui repose sur le principe de l'apprentissage résiduel. Au lieu d'apprendre directement une fonction de transformation $H(x)$, chaque bloc de couches apprend une fonction résiduelle $F(x) = H(x) - x$, et la sortie finale devient $F(x) + x$. Cette approche utilise des connexions de raccourci (*skip connections*) pour ajouter directement l'entrée à la sortie d'un bloc, ce qui permet d'entraîner efficacement des réseaux très profonds tout en évitant les problèmes de dégradation des performances et de gradients qui disparaissent.

2. **Le réseau de neurones récurrent (RNN) :**

   Le réseau de neurones récurrent (RNN) est une généralisation du réseau de neurones à propagation avant (feedforward) qui possède une mémoire interne. Le RNN

fonctionne de manière récurrente, car il applique la même opération à chaque entrée de données, tandis que la sortie actuelle dépend du calcul de l'entrée précédente, qui est transmise et réinjectée dans le réseau pour renforcer l'apprentissage. Ainsi, lors de la prise de décision, le RNN s'appuie à la fois sur les entrées actuelles et sur les informations apprises à partir des entrées précédentes[73].

3. **Longue mémoire à court terme (LSTM) :** Apparu en 1997, présenté par Hochreiter et Schmidhuber.le Long Short-Term Memory (LSTM) est une extension du RNN, conçue pour éviter le problème de la dépendance à long terme, contrairement au RNN qui peut avoir du mal à se souvenir des données pendant de longues périodes. Les couches cachées des RNN ont une structure simple, tandis que les LSTM sont plus complexes et se composent de plusieurs couches cachées. Sa composante principale est l'état de la cellule, qui permet d'ajouter ou de supprimer des informations de l'état de la cellule. Les portes sont utilisées pour le protéger, en utilisant la fonction sigmoïde (1 = permet la modification, 0 = interdit la modification). Il contient trois portes différentes[38] :

   - *Couche de porte d'oubli :* détermine les informations à supprimer de la cellule de mémoire de l'état précédent.
   - *Couche de porte d'entrée/mise à jour :* définit les nouvelles informations à ajouter à l'état de la cellule.
   - *Couche de sortie :* définit ce qui sera produit en tant que sortie

4. **Deep Neural Network (DNN ) :** Elle se compose de plusieurs couches cachées entre la couche d'entrée et la couche de sortie, chaque couche effectuant différents types de filtrage et de classification spécifiques. Les réseaux neuronaux profonds sont capables d'apprendre des caractéristiques de haut niveau avec une complexité supérieure par rapport aux réseaux neuronaux plus superficiels, offrant ainsi une précision avancée dans de nombreuses tâches d'intelligence artificielle, y compris la vision par ordinateur, la reconnaissance vocale, les robots, etc [76].

## 2.4 Limites des approches classiques pour les données relationnelles

Bien que le Machine Learning et le Deep Learning aient connu un grand succès dans plusieurs domaines, ils présentent plusieurs limites pour le traitement des données relationnelles. Voici les principales :

**1. Limites du Machine Learning classique[26] :**

- Le Machine Learning traditionnel (comme la régression logistique, SVM, Random Forest…) fonctionne uniquement avec des données tabulaires où chaque exemple est indépendant des autres.
- Il n'exploite pas directement les relations (liens ou arêtes) entre les entités, comme par exemple des connexions entre patients, produits ou utilisateurs.
- Pour utiliser des graphes, on est obligé de faire du feature engineering manuel (par exemple : le degré d'un nœud, sa centralité, le nombre de voisins…). Ce processus est souvent limité et coûteux en temps.
- Les relations implicites présentes dans la structure du graphe sont perdues.

**2. Limites du Deep Learning classique [84] :**

- Les architectures classiques de Deep Learning (CNN pour les images, RNN pour les séquences, MLP pour les données tabulaires) ne sont pas conçues pour exploiter la structure des graphes.
- Ces modèles supposent des données structurées sous forme de grilles régulières (comme des images ou des séries temporelles), ce qui ne permet pas de capturer les dépendances complexes et irrégulières entre les nœuds dans un graphe.
- Ils ne prennent pas en compte la notion de voisinage dans un graphe, ce qui limite leur capacité à capturer des motifs structuraux importants.

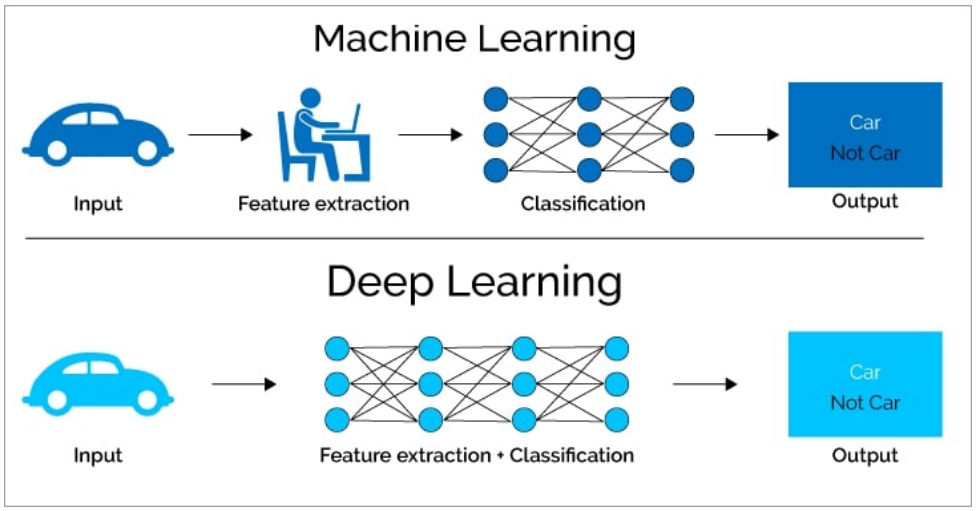



**Figure 1.6 :** Apprentissage automatique et Apprentissage profond [6]

# 3 Les réseaux de neurones graphiques (GNN)

## 3.1 Introduction aux GNNs

Les réseaux de neurones graphiques (GNN) ont été proposés pour la première fois par Gori et Scarselli en 2005, avec une formalisation complète publiée en 2009.Cependant, leur popularité n'a réellement explosé que récemment[34].
Les GNN sont des modèles d'apprentissage profond conçus pour traiter et analyser des données sous forme de graphes, où les nœuds représentent des entités et les arêtes définissent les relations entre elles[13].
Ils reposent sur un mécanisme de passage de messages (Message Passing), permettant à chaque nœud de mettre à jour son état en intégrant les informations de ses voisins. Grâce à cette approche, les GNN capturent à la fois les relations locales et globales au sein du graphe[89].
Aujourd'hui, les GNN sont largement utilisés dans divers domaines, notamment l'analyse des réseaux sociaux, les systèmes de recommandation, la bio-informatique et la prévision du trafic. Dans le secteur de la santé et de la biologie, ils jouent un rôle clé dans la découverte de nouveaux médicaments, la prédiction des interactions entre protéines et l'analyse des similitudes entre patients[13].

## 3.2 Graphe

Un graphe G= (V, E) est une structure mathématique utilisée pour exprimer les relations ou les interactions entre les individus ou les entités. Il est composé de [89] :

- Un ensemble de nœuds (ou sommets) V, Ce sont les entités. Chaque nœud représente un centre ou un point d'information.
- Un ensemble d'arêtes (ou arcs) E, Ce sont les connexions entre ces entités. Elles représentent les canaux ou relations par lesquels l'information circule d'un nœud à un autre.

## 3.3 Types de graphes

Selon les caractéristiques des nœuds, des arêtes et la dynamique du réseau, plusieurs types de graphes peuvent être définis, parmi lesquels on retrouve [89] :

### 1. Graphe orienté / Graphe non orienté

- Graphe orienté : c'est un graphe où les arêtes ont une direction spécifique d'un nœud à un autre, fournissant plus d'informations que les graphes non orientés.

- Graphe non orienté : c'est un graphe où les arêtes n'ont pas de direction ; chaque lien entre deux nœuds est symétrique.

**2. graphe homogène / graphe hétérogène**

- Le graphe homogène : c'est un graphe dont les arêtes et les nœuds sont du même type et appartiennent à la même catégorie.
- Le graphe hétérogène : c'est un graphe qui contient plusieurs types de nœuds et d'arêtes.

**3. graphe statique/ graphe dynamique**

- Le graphe statique : c'est un graphe dont les propriétés et la structure (nœuds et arêtes) restent constantes au fil du temps.
- Le graphe dynamique : c'est un graphe dont les caractéristiques et la structure évoluent avec le temps.

## 3.4 Méthodes de représentation d'un graphe

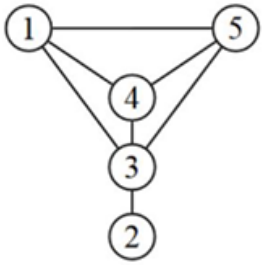
Il existe plusieurs manières de représenter un graphe, parmi lesquelles : liste d'adjacence, la matrice d'incidence, la définition par propriété caractéristique, et la matrice d'adjacence. Mais en général, le graphe est représenté à l'aide d'une matrice d'adjacence.

**1. Matrice d'adjacence**

La matrice d'adjacence est une matrice carrée de dimension n*n où n est le nombre de nœuds. Elle nous permet de savoir si deux nœuds sont adjacents (c'est-à-dire directement connectés) ou non. Formellement, l'élément $A_{ij}$ est défini comme suit [89] :

$$A_{ij} = \begin{cases} 1, & \text{si } i \text{ et } j \text{ sont voisins} \\ 0, & \text{sinon} \end{cases}$$

Dans un graphe non orienté, la matrice d'adjacence est symétrique, c'est-à-dire que $A_{ij} = A_{ji}$.




$$M = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

**Figure 1.7 :** Exemple d'un graphe avec sa matrice d'adjacence

## 3.5 Tâches d'apprentissage sur les graphes :

L'apprentissage sur les graphes vise à exploiter la structure des nœuds et des arêtes pour résoudre différentes tâches de prédiction ou d'analyse. Ces tâches peuvent être classées selon trois niveaux d'observation :

### 1. Au niveau des nœuds (Node-Level) :

Les tâches au niveau des nœuds consistent à déterminer l'identité ou la fonction de chaque nœud dans le graphe. L'objectif principal est de prédire des caractéristiques spécifiques propres à chaque nœud. Ces tâches sont souvent appliquées dans des contextes où certaines données sont manquantes ou non étiquetées.Elles incluent notamment [34] :

- Classification des nœuds : Attribuer une catégorie ou un label à chaque nœud (ex. prédire si une personne est fumeuse).
- Regroupement des nœuds (Clustering) : Diviser les nœuds en plusieurs groupes distincts, où chaque groupe contient des nœuds similaires.
- Régression nodale : Prédire une valeur continue associée à chaque nœud.

### 2. Au niveau des arêtes (Edge-Level) :

Les tâches au niveau des arêtes consistent à analyser les relations entre les paires de nœuds dans le graphe. L'objectif principal est de prédire l'existence, la force ou la nature d'un lien entre deux entités. Ces tâches sont couramment utilisées dans des cas tels que [89] :

- Classification des arêtes : déterminer le type d'une arête reliant deux nœuds.
- Prédiction de liens (Link Prediction) : prédire l'existence future ou l'absence d'une arête entre deux nœuds donnés (ex. recommandation de contenu sur Netflix en fonction des préférences de l'utilisateur).

**3. Au niveau du graphe (Graph-Level) :**

Les tâches au niveau du graphe visent à prédire une propriété ou une caractéristique globale associée à l'ensemble du graphe. Elles sont particulièrement utiles lorsque le graphe représente une entité unique (ex. une molécule, un document). Ces tâches incluent notamment [89] :

- Classification de graphe : attribuer une catégorie spécifique à un graphe entier (ex. déterminer si une molécule est toxique ou non).
- Régression sur le graphe : prédire une valeur continue associée à l'ensemble du graphe.
- Correspondance de graphes (Graph Matching) : identifier des similarités ou entre plusieurs graphes.

## 3.6 Génération des embeddings dans les réseaux de neurones graphiques

L'objectif principal des réseaux de neurones graphiques (GNN) est de créer une nouvelle représentation pour chaque nœud, appelée embedding des nœuds, en exploitant toutes les informations sur le graphe (caractéristiques des nœuds et leurs connexions). Ces embeddings sont des vecteurs de faible dimension qui décrivent la position des nœuds dans le graphe ainsi que la structure de leurs voisins locaux.
L'embedding final est utilisé pour la classification des nœuds, et pour y parvenir, le GNN repose sur un mécanisme de passage de messages qui constitue son principe fondamental [78].

## 3.7 Mécanisme de Message Passing

Le mécanisme de passage de messages est un processus fondamental dans les réseaux de neurones graphiques (GNNs), qui permet aux nœuds d'échanger, d'agréger et de combiner les informations de leurs voisins pour enrichir progressivement leurs représentations. Il repose sur deux étapes essentielles : l'agrégation des messages reçus des voisins et la mise à jour de l'état du nœud. Lors de chaque itération de passage des messages, l'inclusion (ou *embedding*) $h_u^{(n)}$ associée à chaque nœud est mise à jour en fonction des informations accumulées des nœuds voisins $\mathcal{N}(u)$ [69].
Formellement :

$$h_u^{(n+1)} = \text{UPDATE}^{(n)}\left(h_u^{(n)}, \text{AGGREGATE}^{(n)}\left(\left\{h_v^{(n)} \mid v \in \mathcal{N}(u)\right\}\right)\right)$$

Ou, de manière simplifiée :

$$h_u^{(n+1)} = \text{UPDATE}^{(n)}\left(h_u^{(n)}, m_{\mathcal{N}(u)}^{(n)}\right)$$

Avec :

- $h_u^{(n)}$ : représentation (embedding) du nœud $u$ à la $n$-ième itération ;
- $\mathcal{N}(u)$ : ensemble des voisins du nœud $u$ ;
- $\text{AGGREGATE}^{(n)}$ : fonction d'agrégation à l'itération $n$ (par exemple, moyenne, somme, maximum) ;
- $\text{UPDATE}^{(n)}$ : fonction de mise à jour de l'état du nœud à l'itération $n$ ;
- $m_{\mathcal{N}(u)}^{(n)}$ : message agrégé provenant des voisins de $u$ à l'itération $n$.

La Figure suivante 1.8 représente une vue d'ensemble du modèle de passage de messages à deux couches, illustrant comment les nœuds agrègent les messages de leur voisinage. Le modèle collecte les messages des voisins immédiats de A (B, C et D), et ces messages sont eux-mêmes basés sur les informations recueillies auprès de leurs propres voisins.

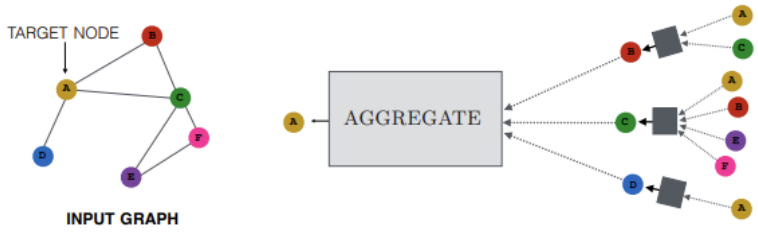



**Figure 1.8 :** Illustration du passage de messages et de l'agrégation dans un GCN[57]

En résumé, chaque nœud collecte des informations des nœuds adjacents à chaque itération. Les étapes suivantes expliquent comment cela se fait [57] :

1. Après la première itération, chaque embedding de nœud contient des informations de ses voisins directs à un saut (1-hop).
2. Dans la deuxième itération, chaque embedding de nœud contient des informations de ses voisins à deux sauts (2-hop).
3. Après plusieurs itérations, chaque nœud devient plus détaillé car son embedding contient des informations sur ses voisins dans une portée de n sauts (n-hop).
4. Enfin, après avoir rassemblé toutes les informations contractuelles disponibles, "le graphe peut être représenté de manière globale.

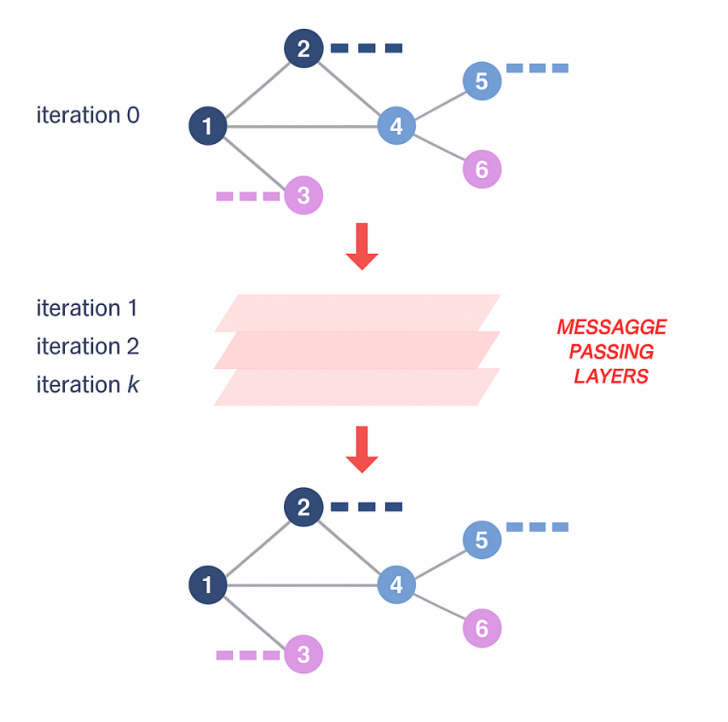



**Figure 1.9 :** Mécanisme de Message Passing [57]

## 3.8 Quelles sont les étapes d'un GNN ?

Les réseaux de neurones graphiques (GNNs) suivent un processus structuré pour apprendre des représentations à partir des graphes. Ce processus s'organise en plusieurs étapes clés, décrites ci-dessous [86] :

1. Représentation du graphe : Représenter les données sous forme de graphe :
   - Nœuds : représentent des entités (ex : utilisateurs, articles, etc.).
   - Arêtes : représentent des relations entre les entités (ex : amitié, citation, etc.).
   - Attributs : les nœuds et arêtes peuvent avoir des attributs (features), comme le nom, l'âge, etc.
2. Initialisation des embeddings des nœuds : Chaque nœud est associé à un vecteur de caractéristiques initial (embedding), basé sur ses attributs ou une initialisation aléatoire si aucune information n'est disponible.
3. Message Passing (Propagation de messages) : Chaque nœud envoie un message (son embedding) à ses voisins.
4. Agrégation des messages : Chaque nœud agrège les messages reçus de ses voisins

selon une fonction d'agrégation définie :

$$a_v^{(k)} = \text{AGGREGATE}^{(k)} \left( \left\{ h_u^{(k-1)} \mid u \in \mathcal{N}(v) \right\} \right)$$

Avec :

- $h_v^{(k)}$ = embedding du nœud $v$ à la couche $k$,
- $N(v)$ = ensemble des voisins de $v$.

5. Mise à jour du nœud : Chaque nœud met à jour son embedding en utilisant l'information agrégée.

6. Répétition : Répéter les étapes de passage de messages, d'agrégation et de mise à jour sur plusieurs couches ou itérations.

7. Utilisation des embeddings finaux : Les embeddings obtenus sont utilisés pour réaliser une tâche spécifique, comme la classification de nœuds, la prédiction de liens ou la classification de graphes.

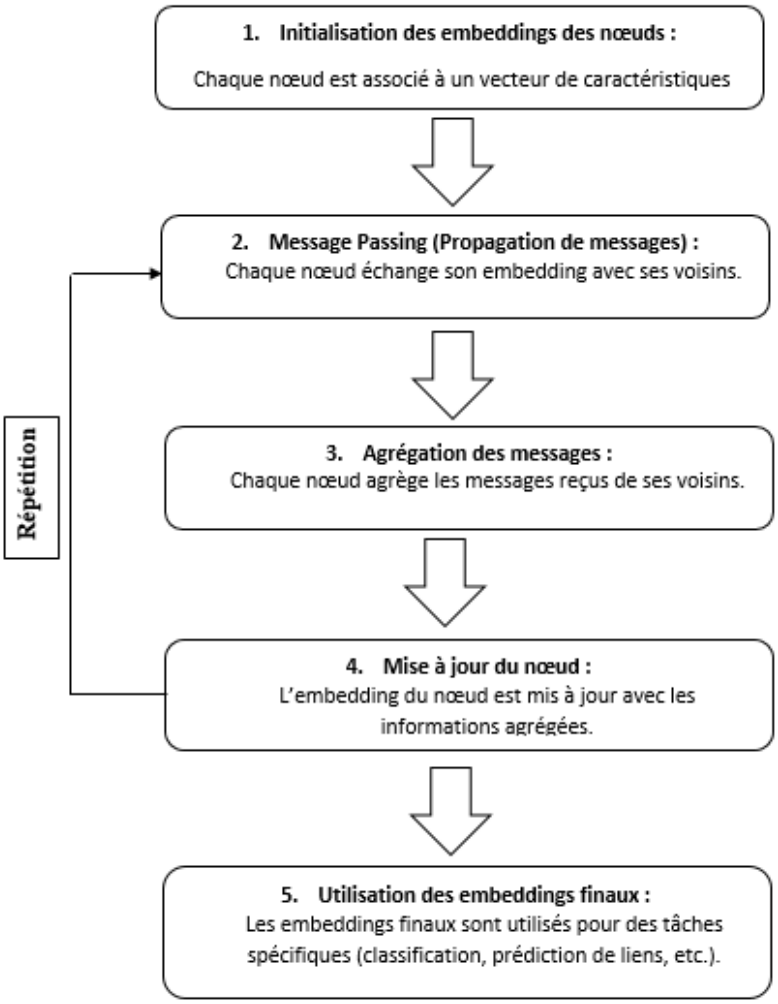



**Figure 1.10 :** les étapes d'un GNN [86]

## 3.9 Pourquoi utiliser les GNN ?

Les réseaux de neurones graphiques (GNN) se distinguent des modèles traditionnels par leur capacité à traiter des données non euclidiennes, qui ne suivent pas une structure organisée en grille. Alors que les réseaux de neurones classiques, comme les CNN, s'adaptent aux images ou aux textes organisés de manière structurée dans l'espace euclidien, les GNN sont conçus pour fonctionner sur des structures graphiques où les connexions entre les entités peuvent être désorganisées, variées et dynamiques. Cela permet de représenter et d'analyser des domaines complexes tels que les réseaux sociaux, les réseaux de transport ou les interactions moléculaires. Les GNN permettent de capturer les dépendances locales et globales au sein du graphe grâce au mécanisme de propagation de messages, ce que les CNN ne peuvent pas faire car ils sont limités à des voisinages fixes et organisés [12].
Pour adapter les opérations de convolution aux données en structure de graphe, plusieurs approches ont été développées. Classées en deux catégories : les méthodes spatiales et les

méthodes spectrales [9].

a. **Convolution Spectrale** : Ce type de convolution repose sur une base mathématique solide. Elle s'appuie sur la théorie du traitement du signal sur les graphes et est basée sur la décomposition spectrale du Laplacien du graphe, ce qui permet de définir la transformée de Fourier adaptée aux graphes. Parmi les méthodes basées sur cette approche, on retrouve notamment le *Graph Convolutional Network* (GCN), qui constitue une forme simplifiée de convolution spectrale, rendant son application plus pratique dans les modèles d'apprentissage profond.

b. **Convolution Spatiale** : La convolution spatiale est une extension des convolutions traditionnelles des réseaux CNN au domaine des graphes. Alors que la convolution dans les CNN est effectuée en rassemblant les unités de pixels adjacents au pixel central à l'aide d'un filtre (ou noyau) contenant des poids apprenables, les réseaux convolutifs spatiaux suivent le même principe en regroupant les informations des nœuds voisins d'un nœud central sans avoir besoin de transformation spectrale.

La Figure suivante 1.11 représente une image tirée de A Comprehensive Survey on Graph Neural Networks. À gauche, on observe la convolution sur un graphe régulier tel qu'une image, tandis qu'à droite, il s'agit d'une convolution appliquée sur une structure de graphe arbitraire.

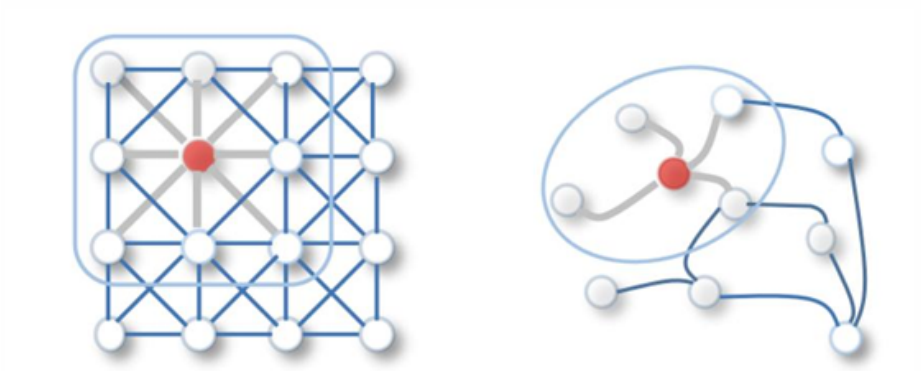


**Figure 1.11 :** Convolution sur un graphe régulier et un graphe arbitraire [84]

## 3.10 Type de Graph Neural Network

Il existe plusieurs types de Graph Neural Networks (GNN), chacun utilisant une méthode différente pour traiter et transmettre l'information entre les nœuds du graphe. Voici les types les plus courants :

1. **GraphSAGE**
   GraphSAGE est apparu en 2017, il effectue un apprentissage efficace sur les don-

nées représentées sous forme de graphes et infère à partir des nœuds invisibles en regroupant les voisins locaux échantillonnés aléatoirement. Il est particulièrement utile pour les grands graphes [48].

2. **Graph Convolutional Networks (GCN)**
   Les Graph Convolutional Networks (GCN), introduits par Kipf et Welling (2017), sont une architecture de réseau de neurones conçue pour travailler sur des données structurées sous forme de graphes. Contrairement aux réseaux de neurones convolutifs traditionnels (CNN), qui fonctionnent bien sur des données euclidiennes (telles que les images, les signaux où les relations entre les données sont régulières et fixes), les GCN permettent de capturer les relations complexes et irrégulières entre les nœuds d'un graphe.

   Ils ont attiré beaucoup d'attention ces dernières années en raison de la croissance des données non euclidiennes dans des applications réelles, telles que les réseaux sociaux, les systèmes de recommandation ou les données biologiques (comme les molécules ou les communications neuronales).

   Les GCN sont considérés comme l'un des outils les plus puissants pour l'apprentissage sur graphes, car ils exploitent à la fois les informations de chaque nœud et la structure du graphe pour apprendre des représentations utiles. Leur architecture permet de généraliser le concept de convolution aux graphes, en regroupant les informations des nœuds voisins pour mettre à jour les représentations de chaque nœud [12].

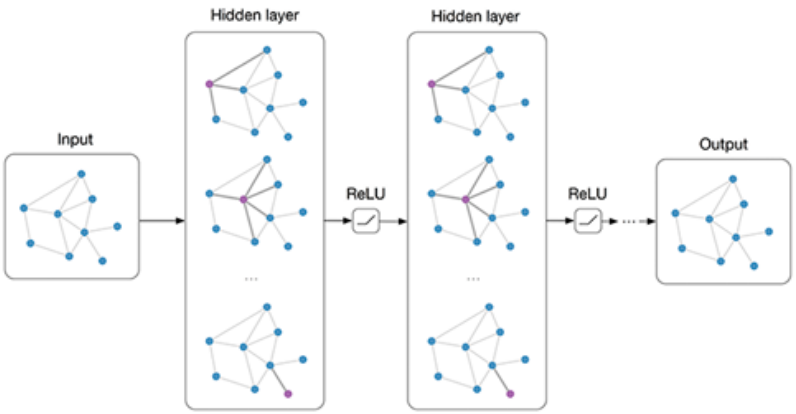



**Figure 1.12 :** Les réseaux convolutifs graphiques [36]

La figure suivante 1.13 illustre l'architecture d'un GCN et son fonctionnement sur un graphe.

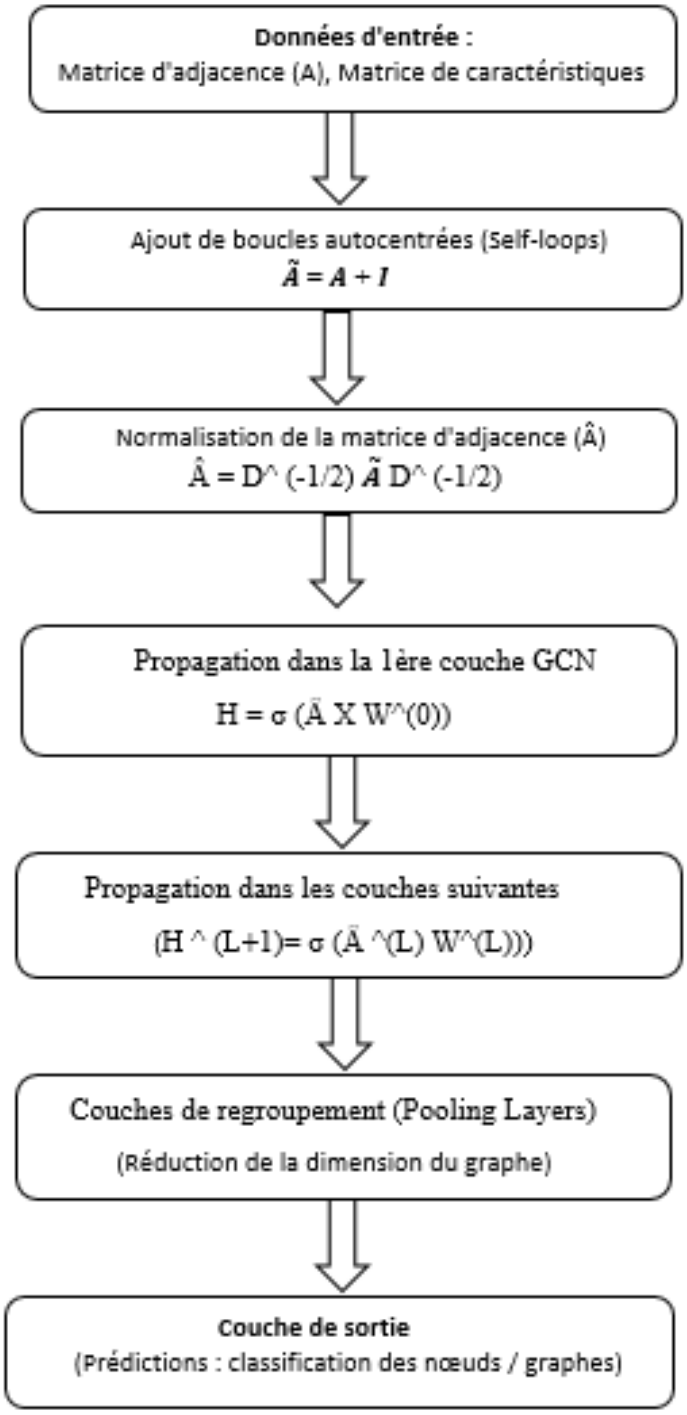



**Figure 1.13 :** Architecture d'un Graph Convolutional Network (GCN)[88]

**Où :**

- A : Matrice d'adjacence du graphe
- X : Matrice des caractéristiques des nœuds
- I : Matrice identité (utilisée pour les boucles autocentrées)

- $\tilde{A}$ : Matrice d'adjacence avec boucles autocentrées (A + I)
- D : Matrice diagonale des degrés des nœuds
- $\hat{A}$ (A chapeau) : Matrice d'adjacence normalisée
- $W^0$, $W^L$ : Matrices de poids des couches GCN
- $\sigma$ : Fonction d'activation non linéaire (ex : ReLU)
- Pooling : Couches de regroupement (réduction de dimension)
- Output : Prédictions (classification des nœuds ou graphes)

3. **Graph attention networks (GAT)**
   Le réseau d'attention graphique (GAT) a été utilisé avec succès en intégrant le mécanisme d'attention pour traiter des graphes de formes variées. Ce mécanisme permet de gérer des entrées de tailles différentes et de se concentrer sur les informations les plus pertinentes. Il est utilisé dans des domaines tels que la modélisation (Bahdanau, Cho, et Bengio 2015 ; Devlin et al. 2019 ; Vaswani et al. 2017), la traduction automatique (Luong, Pham, et Manning 2015), et le traitement visuel (Xu et al. 2015).

   Dans un graphe, l'unité d'attention du GAT se concentre sur le calcul des représentations cachées des nœuds en appliquant une attention répétée aux caractéristiques de leurs voisins. Les coefficients d'attention sont déterminés de manière inductive via une stratégie d'auto-attention. Cette approche a connu un grand succès dans les tâches d'embiding des nœuds et de classification [87].

La figure suivante 1.14 illustre l'architecture d'un GAT et l'intégration du mécanisme d'attention dans le traitement des graphes.

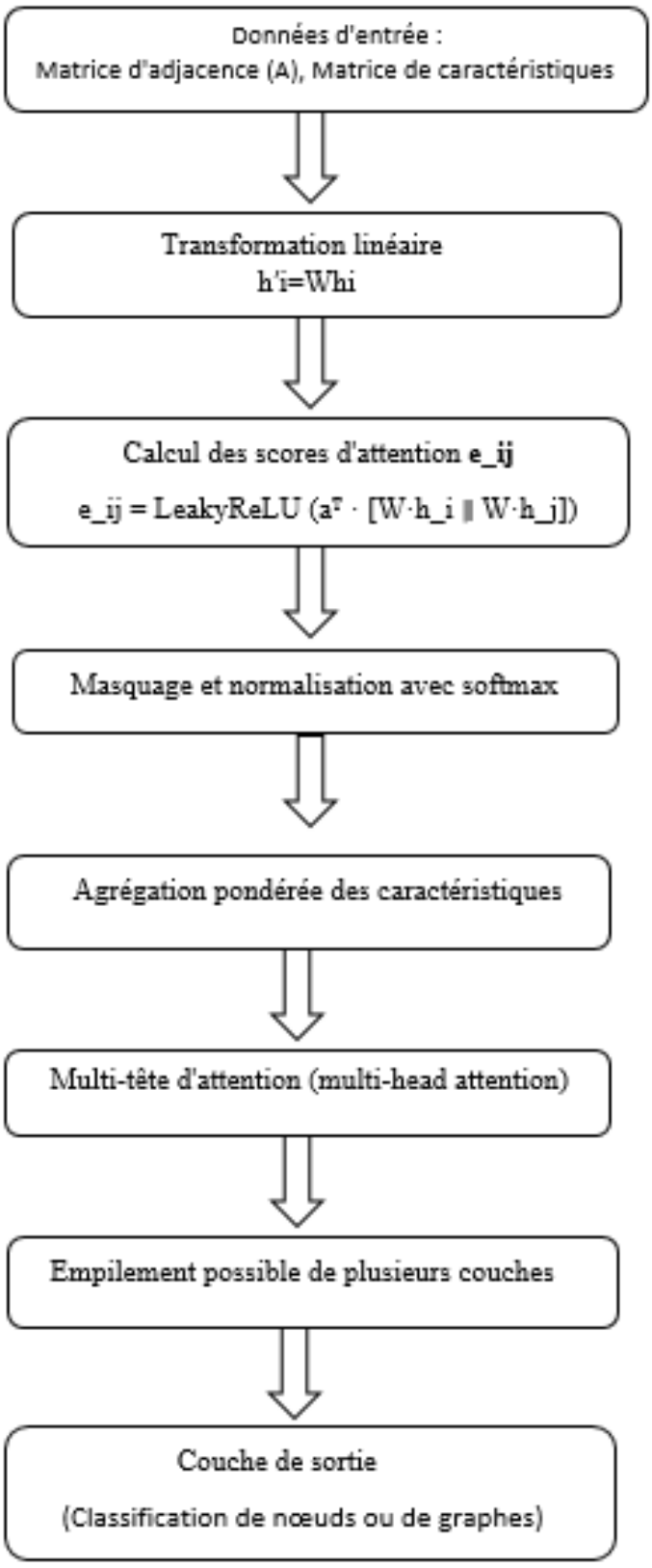



**Figure 1.14 :** Architecture d'un graph attention networks (GAT)[80]

**Où :**

- $A$ : Matrice d'adjacence du graphe
- $X$ : Matrice des caractéristiques des nœuds
- $h_i,\ h_j$ : Représentations des nœuds $i$ et $j$ après transformation linéaire
- $W^a,\ W^b$ : Matrices de poids pour la transformation linéaire des caractéristiques
- $e_{ij}$ : Score d'attention entre le nœud $i$ et le nœud $j$
- $LeakyReLU$ : Fonction d'activation avec pente négative pour les valeurs $< 0$
- *Softmax* : Fonction de normalisation des scores d'attention
- *Multi-head attention* : Mécanisme utilisant plusieurs têtes d'attention parallèles
- *Empilement de couches* : Superposition de plusieurs couches GAT pour capter des relations complexes
- *Output* : Prédictions (classification des nœuds ou du graphe)

## 3.11 Application des GNN

Les réseaux de neurones graphiques (GNN) offrent une large gamme d'applications et peuvent être utilisés dans divers domaines pour résoudre des problèmes complexes impliquant des données structurées sous forme de graphes. Voici quelques exemples [5] :

- **Découverte de médicaments** :- Découverte de médicaments : utilisation des GNN pour prédire l'efficacité des médicaments potentiels en modélisant sous forme de graphes les interactions entre atomes et molécules.
- **Analyse des réseaux sociaux** : utilisation des GNN pour analyser et comprendre la structure des réseaux sociaux. Par exemple, il est possible de prédire la diffusion de l'information et la formation de communautés.
- **Traitement automatique du langage naturel** : un GNN peut analyser le texte en modélisant les relations entre les mots ou les phrases sous forme de graphes. afin de mieux comprendre le contexte sémantique dans les textes.
- **Systèmes de recommandation** : utilisation des GNN pour développer des systèmes de recommandation personnalisés qui proposent des éléments aux utilisateurs en fonction de leur comportement et de leurs préférences.

Les GNN ont montré de bons résultats dans divers domaines et continuent de se développer grâce à l'évolution de leurs architectures et algorithmes.

## 3.12 Limitation des GNN

Bien que les résultats des GNN soient positifs, elles font face à certaines limitations, parmi lesquelles [85] :

- **Complexité computationnelle** : les GNN sont coûteuses en termes de temps et de puissance de calcul, en particulier pour les grands graphes.
- **Gestion des données manquantes** : les GNN rencontrent des difficultés pour traiter les données incomplètes ou manquantes.
- **Risque de sur-apprentissage** : surtout lorsque le nombre de paramètres est élevé et que le nombre de nœuds du graphe est faible.

## 3.13 Avantages des GNNs pour la prédiction en santé

Les GNNs offrent des avantages puissants dans le domaine de la santé, car ils permettent d'exploiter les relations complexes entre patients, examens et maladies pour améliorer la qualité des prédictions médicales. Voici quelques avantages clés :

1. Exploitation des relations complexes entre les patients : Contrairement aux modèles classiques, les GNNs permettent de modéliser explicitement les relations entre patients, médecins, examens, ou maladies. Exemple : connecter les patients ayant des diagnostics ou des caractéristiques cliniques similaires pour améliorer la prédiction du risque [71].
2. Intégration de données hétérogènes (multimodales) : Les GNNs peuvent facilement fusionner différentes sources de données [90] :
   - Données cliniques (âge, tension, historique)
   - Images médicales (ECG, IRM, scanner)
   - Données relationnelles (réseaux de patients, gènes, symptômes…)

   Exemple : combiner un ECG et les antécédents du patient pour prédire une insuffisance cardiaque.
3. Meilleure propagation de l'information (message passing) : Dans un graphe médical, les informations d'un patient peuvent influencer celles de ses voisins. Les GNNs permettent une propagation contrôlée de l'information via les arêtes, ce qui améliore la prédiction surtout en présence de données incomplètes. Exemple : un patient sans diagnostic mais entouré de patients similaires atteints : le modèle peut anticiper son risque[35].

4. Robustesse aux données manquantes : Les données médicales souffrent souvent de valeurs manquantes. Grâce à la structure du graphe et au message passing, les GNNs peuvent estimer les informations manquantes en s'appuyant sur les voisins[10].

5. Amélioration de la détection précoce et du diagnostic personnalisé : En combinant relations et données patient, les GNNs peuvent détecter plus tôt des cas à risque et proposer des scores de risque personnalisés en fonction du contexte réseau du patient [47]. Exemple : Prédire une complication après une opération en se basant sur les patients similaires.

# 4 Conclusion

Dans cette section, nous avons d'abord présenté les aspects médicaux des maladies cardiaques, en définissant leurs types, leurs causes, leurs symptômes et les méthodes classiques de diagnostic utilisées en cardiologie.
Ensuite, nous nous sommes intéressés à l'apport de l'intelligence artificielle dans le domaine médical, en décrivant ses concepts clés, du machine learning au deep learning, ainsi que leurs limites face aux données relationnelles.
Nous avons enfin introduit les réseaux de neurones graphiques (GNN), en expliquant leur principe, leur fonctionnement à travers le mécanisme de message passing, et leurs principales architectures (GCN, GraphSAGE, GAT). Nous avons conclu en évoquant les applications des GNN en santé et les avantages qu'ils offrent pour la prédiction médicale.
Dans le chapitre suivant, nous présenterons les travaux existants dans le domaine de la prédiction de l'insuffisance cardiaque, ainsi que les solutions proposées pour améliorer les performances des modèles prédictifs dans ce contexte.

# Chapitre 2

# Synthèse des Travaux existants

## 1 Introduction

L'insuffisance cardiaque est un problème de santé majeur qui touche un grand nombre de personnes et impacte directement leur qualité de vie. Ce sujet intéresse beaucoup de chercheurs, ce qui a conduit à la réalisation de nombreux travaux visant à améliorer sa détection et sa prise en charge. Plusieurs approches basées sur l'intelligence artificielle ont été proposées, et la recherche dans ce domaine reste toujours active.
Dans ce chapitre, nous présentons une revue des études pertinentes sur la prédiction de l'insuffisance cardiaque. Une comparaison entre les travaux existants est ensuite proposée sous forme de tableau synthétique. Par la suite, nous discutons les limites des approches actuelles. Enfin, nous présentons quelques idées pour améliorer les méthodes de prédiction.

## 2 Revue des études pertinentes

Afin de mieux cerner l'état de l'art dans le domaine, nous avons procédé à une revue de la littérature portant sur seize (16) articles scientifiques. Ces travaux couvrent diverses approches, allant des techniques classiques d'apprentissage automatique aux modèles plus récents basés sur les réseaux de neurones profonds et graphiques.

### Using recurrent neural network models for early detection of heart failure onset [16]

Les auteurs dans [16] étudient la détection précoce de l'insuffisance cardiaque en utilisant les Dossiers de Santé Électroniques (DSE), en soulignant l'importance de prévoir cette maladie chronique pour améliorer le suivi des patients. Le principal défi est de prendre en compte l'évolution des données médicales dans le temps. Pour cela, les auteurs proposent

l'utilisation de réseaux de neurones récurrents (RNN), bien adaptés à l'analyse des données organisées en séquences.
Les auteurs utilisent des données provenant des DME de patients en soins primaires, avec un échantillon de 3 884 cas d'IC et 28 903 témoins. Les dossiers incluent diagnostics, prescriptions, procédures et autres événements horodatés. Premièrement, les événements médicaux sont encodés en séquences de vecteurs one-hot, enrichies par le délai jusqu'au diagnostic. Pour capter les relations entre les codes et réduire la dimension, des vecteurs de concepts médicaux sont créés via la méthode Skip-gram. Par la suite, un modèle GRU (Gated Recurrent Unit) est utilisé pour capturer les relations temporelles entre les événements médicaux et produire un score de risque d'apparition de l'IC. Les performances du GRU sont ensuite comparées à celles de méthodes classiques : régression logistique, perceptron multicouche (MLP), machine à vecteurs de support (SVM) et K-plus proches voisins (KNN).
Le modèle GRU avec des vecteurs de concepts médicaux et des informations temporelles a obtenu une AUC de 0,883 avec une fenêtre d'observation de 18 mois, surpassant ainsi les méthodes traditionnelles. Les performances du modèle sont meilleures avec des fenêtres d'observation plus longues, notamment de 18 mois, et l'utilisation de vecteurs de concepts médicaux a permis d'améliorer la prédiction de l'insuffisance cardiaque. Le modèle GRU est scalable et adapté pour une application clinique.
Le modèle propose présente plusieurs limitations : Les données proviennent d'un seul système de santé, ce qui limite la généralisation des résultats à d'autres populations. Les résultats de laboratoire et les notes médicales n'ont pas été inclus dans les données utilisées. Les modèles GRU, bien qu'ils soient puissants, sont complexes et difficiles à interpréter. De plus, la qualité des données (manquantes ou inexactes) peut affecter les résultats.

### An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction [68]

Les auteurs dans [68] proposent un système d'aide à la décision clinique pour prédire les risques d'insuffisance cardiaque, en intégrant des réseaux de neurones artificiels (ANN) avec la méthode Fuzzy_AHP, afin d'améliorer la précision du diagnostic par rapport aux approches classiques fondées uniquement sur l'intelligence artificielle.
Les auteurs ont utilisé des données provenant du célèbre Cleveland Heart Disease Dataset de l'UCI, comprenant 297 patients et 13 attributs médicaux (âge, tension artérielle, cholestérol, électrocardiogramme, etc.).
Comme première étape les auteurs ont commencé par la pondération des Attributs par

Fuzzy_AHP dont un expert cardiologue détermine l'importance relative des 13 attributs via des comparaisons par paires. La méthode Fuzzy_AHP permet de calculer les poids globaux des attributs et de leurs alternatives, établissant ainsi leur contribution respective au diagnostic. Par la suite, ces poids globaux issus de Fuzzy_AHP sont intégrés dans un réseau de neurones à propagation avant (feedforward) pour entraîner le modèle de prédiction. Le réseau est configuré avec 13 entrées (correspondant aux attributs), une couche cachée optimisée, et deux sorties (présence ou absence de l'insuffisance cardiaque). Le système hybride ANN-Fuzzy_AHP atteint une précision de $91,10\%$, supérieure à celle des méthodes utilisant uniquement ANN ou d'autres techniques classiques. Le modèle hybride affiche une sensibilité (rappel) de 100%, détectant ainsi tous les patients à risque d'insuffisance cardiaque.
L'étude présente plusieurs limitations : elle repose sur un échantillon restreint de 297 patients, ce qui peut limiter la généralisation des résultats en contexte clinique réel. Le temps de calcul du modèle n'a pas été évalué, compromettant son applicabilité en temps réel dans un environnement médical. De plus, les paramètres du réseau de neurones ont été optimisés manuellement, sans recours à des techniques systématiques d'optimisation

## Coronary Heart Disease Diagnosis using Deep Neural Networks [46]

Les auteurs dans [46] abordent le diagnostic de la maladie coronarienne (CHD) en exploitant le potentiel des réseaux neuronaux profonds (DNN). Le diagnostic de cette maladie repose traditionnellement sur des méthodes cliniques qui peuvent être coûteuses ou invasives. Dans ce contexte, l'étude vise à évaluer l'efficacité des DNN comme outil non invasif pour prédire la présence de CHD à partir de données cliniques.
Les auteurs ont utilisé des données médicales issues du UCI Machine Learning Repository, comportant 303 cas avec 14 attributs cliniques comme l'âge, le sexe, la tension artérielle, le cholestérol, le taux de sucre dans le sang à jeun, etc. Le jeu de données est relativement équilibré entre les cas positifs et négatifs de CHD. Un nettoyage a été réalisé en traitant les valeurs manquantes et en transformant les attributs catégoriels en représentations numériques. Les variables continues ont été normalisées afin d'améliorer l'apprentissage du réseau. Les auteurs développent un réseau de neurones profond (DNN) avec plusieurs couches entièrement connectées, activées par ReLU et entraînées par rétropropagation avec l'algorithme Adam.
Le DNN surpasse tous les autres modèles classiques, atteignant une précision de $93,3\%$ sur le jeu de test, contre 86% pour la régression logistique et 88% pour KNN. L'étude met en évidence l'avantage des architectures profondes pour capturer des interactions complexes

entre les variables cliniques.
Cette proposition souffre de plusieurs limitations comme la taille réduite de l'échantillon, limitant la représentativité des résultats, le risque de sur-apprentissage en l'absence de validation croisée robuste ; le manque de diversité des données issues d'une seule source (Cleveland Clinic) ; et la comparaison limitée à des méthodes classiques, sans recours à d'autres techniques d'apprentissage profond.

### Predicting the Risk of Heart Failure With EHR Sequential Data Modeling [32]

Les auteurs de l'étude [32] s'intéressent à l'amélioration de la prédiction de l'insuffisance cardiaque à partir des dossiers de santé électroniques (EHR). L'insuffisance cardiaque étant souvent diagnostiquée tardivement, l'exploitation des données médicales sous forme de séries temporelles permettrait une détection plus précoce. Cependant, ces données sont souvent non structurées et difficiles à utiliser directement. Pour répondre à ce défi, l'article propose une approche basée sur les réseaux de neurones LSTM afin d'analyser ces données et d'améliorer la précision des prédictions, facilitant ainsi une prise en charge plus efficace des patients.
Les auteurs exploitent deux ensembles de données réelles : 5000 patients diagnostiqués et 15000 non diagnostiqués en insuffisance cardiaque, avec pour chaque patient des séquences temporelles d'événements médicaux et leurs dates. Les séquences d'événements diagnostiques sont prétraitées via l'encodage one-hot et le word embedding vectoriel (Word2Vec), permettant de mieux représenter les similarités entre événements médicaux. Les auteurs conçoivent un modèle LSTM enrichi de word vectors, capable de traiter la dimension temporelle des données médicales séquentielles. Ce modèle est comparé à des méthodes classiques telles que la régression logistique, Random Forest et AdaBoost.
Les résultats montrent que l'approche LSTM avec word embeddings est la plus performante, confirmant ainsi l'importance de prendre en compte la dimension temporelle des données médicales. L'utilisation des word embeddings (Word2Vec) améliore les performances du modèle par rapport au one-hot encoding, car elle capture mieux les relations entre les événements médicaux. Le modèle LSTM surpasse ainsi les autres approches (régression logistique, forêts aléatoires, AdaBoost) en offrant une meilleure précision et une capacité prédictive accrue.
L'étude présente plusieurs limitations : la qualité variable des données EHR peut affecter la précision des résultats, rendant difficile leur généralisation à d'autres populations. Le modèle LSTM est gourmand en ressources informatiques et manque d'intégration de l'expertise médicale pour affiner les prédictions. De plus, l'interprétation des décisions du

modèle est complexe en raison de son caractère de ”boîte noire”.

### Heart Disease Prediction Using Deep Neural Network [63]

Les auteurs dans [63] s'intéressent à la prédiction des maladies cardiaques. Ils proposent un modèle basé sur un Deep Neural Network (DNN) combiné à un modèle statistique du chi² pour la sélection de caractéristiques. Le modèle est entraîné via une stratégie train-test split à 80%-20% et optimisé par une recherche par grille (grid search) pour ajuster les paramètres et prévenir le sur-apprentissage.
Les auteurs utilisent l'ensemble de données Cleveland Heart Disease, issu de l'archive UCI Machine Learning, comprenant 303 cas et 14 attributs cliniques tels que l'âge, le sexe, etc. Une sélection des caractéristiques a été réalisée en utilisant le test statistique du chi² afin d'évaluer la dépendance entre chaque variable et la classe cible, et garder les plus utiles. Les attributs redondants ou peu informatifs ont été supprimés avant l'entraînement du réseau, améliorant ainsi la qualité des données d'apprentissage.
Les résultats expérimentaux montrent que le modèle DNN proposé surpasse les approches antérieures (comme ANN ou SVM), avec une précision atteignant jusqu'à 99% à certains niveaux d'entraînement. Le système améliore la prédiction en réduisant à la fois le sur-apprentissage et le sous-apprentissage.
L'étude présente plusieurs limitations : une taille d'échantillon restreinte (303 cas) limitant la généralisation du modèle, l'absence de gestion avancée du déséquilibre des classes, et une validation réalisée uniquement sur le dataset Cleveland, ce qui réduit la démonstration de sa robustesse.

### Ensemble Deep Learning Models for Heart Disease Classification : A Case Study from Mexico [8]

Les auteurs dans [8] explore le diagnostic des maladies cardiaques, en mettant en avant la complexité due à divers facteurs cliniques et de risque. Les hôpitaux utilisent les Dossiers de Santé Électroniques (DSE) pour une meilleure surveillance des patients, mais les données déséquilibrées compliquent l'entraînement des algorithmes de classification. Pour pallier ce problème, les auteurs proposent des modèles d'apprentissage profond par ensemble, appliqués à un ensemble de 800 dossiers médicaux mexicains avec 141 indicateurs, afin d'améliorer la précision du diagnostic et de surmonter les défis liés aux données déséquilibrées.
Les auteurs ont collecté les données à partir des dossiers médicaux de 800 patients à l'Hôpital Medica Norte au Mexique, contenant 141 indicateurs, tels que l'âge, le poids, le glucose, et la pression artérielle. L'analyse des données a révélé un déséquilibre significatif

entre les différentes classes de maladies cardiaques, ce qui pose un défi pour les modèles de classification. Pour cela, un prétraitement des données a été réalisé par les auteurs. Premièrement, une sélection des caractéristiques a été mise en œuvre pour réduire le bruit et l'irrélevance des données, améliorant ainsi la qualité des données d'entraînement, puis un sous-échantillonnage aléatoire a été utilisée pour équilibrer les classes, combinée à une technique d'agrégation des échantillons pour mieux gérer les déséquilibres.
Les auteurs proposent une approche intégrant plusieurs modèles de réseaux de neurones : BiLSTM et BiGRU et CNN.
Les performances du cadre d'apprentissage proposé ont été comparées et analysées pour démontrer l'efficacité de l'approche face à des données déséquilibrées. L'approche par ensemble combinant BiLSTM/BiGRU avec un CNN offre les meilleurs résultats pour la classification des maladies cardiaques. Les modèles atteignent des scores F1 et une exactitude entre 91% et 96%, prouvant leur efficacité. Les modèles utilisés seuls affichent des performances moindres, avec des scores F1 et d'exactitude atteignant 87% et 81% pour certaines classes. Le sous-échantillonnage et la sélection de caractéristiques ont amélioré les performances, soulignant l'importance de la gestion des données déséquilibrées.
L'étude présente plusieurs limitations : elle ne repose pas sur des données de référence, ce qui rend la comparaison difficile, et la taille des données pourrait limiter la généralisation des résultats. La sélection des caractéristiques risque d'exclure des variables importantes, tandis que certaines maladies rares restent mal classifiées malgré le sous-échantillonnage. Les résultats étant spécifiques aux données mexicaines, leur applicabilité ailleurs est incertaine.

### Hybrid deep learning model using recurrent neural network and gated recurrent unit for heart disease prediction [37]

Les auteurs dans [37] présentent un modèle d'apprentissage profond hybride innovant pour la prédiction des maladies cardiaques. Ce modèle combine des réseaux de neurones récurrents (RNN) avec plusieurs unités de récurrence à porte (GRU), des réseaux à mémoire à court et long terme (LSTM) et utilise l'optimiseur Adam. L'objectif principal de cette étude est d'améliorer la précision et l'efficacité de la prédiction des maladies cardiaques en surmontant les limitations des modèles existants, notamment en termes de complexité des réseaux neuronaux, de redondance des neurones et de déséquilibre des données.
Les auteurs utilisent des données provenant du Cleveland Heart Disease Dataset. Ces données ont subi un nettoyage des données et une normalisation. Un équilibrage des données avec SMOTe a été réalisé avec génération de données synthétiques pour corriger le déséquilibre du dataset.

Les résultats montrent que Le modèle hybride RNN et GRU atteint une précision de $98,68\%$, surpassant ainsi les modèles existants (RNN seul : $98,23\%$, DNN : $98,5\%$). Grâce aux GRU, le modèle retient les informations essentielles tout en éliminant les données inutiles, ce qui le rend plus efficace et rapide. L'utilisation de la technique SMOTe a permis d'équilibrer les données et, par conséquent, d'améliorer la fiabilité des prédictions. L'étude présente plusieurs limitations : une taille limitée de jeu de données (303 échantillons), ce qui peut restreindre la généralisation du modèle ; une dépendance à SMOTe, pouvant introduire des biais dans les données ; un temps de calcul élevé, nécessitant des ressources informatiques importantes ; une absence de validation clinique, le modèle n'ayant pas été testé sur des patients réels ; une sensibilité aux hyperparamètres, nécessitant des ajustements précis pour de meilleures performances.

## Cardiac Disease Prediction using Supervised Machine Learning Techniques [27]

Les auteurs dans [27] explorent la prédiction des maladies cardiaques à l'aide de techniques d'apprentissage automatique supervisé tels que la régression logistique, les k-plus proches voisins (KNN), les arbres de décision, le modèle à vecteurs de support (SVM) et le classifieur de Naïve Bayes. L'objectif est de réduire le risque de faux négatifs et d'accroître la précision globale des prédictions.
Les auteurs utilisent des données issues de l'UCI contenant des enregistrements de 303 patients avec 14 attributs cliniques. Pour le prétraitement de données il y a eu une vérification et suppression des valeurs manquantes, suivies de la normalisation des données à l'aide d'un Standard Scaler afin d'uniformiser les échelles des variables et d'optimiser les performances des modèles.
La régression logistique a obtenu les meilleurs résultats avec une matrice de confusion de 35 vrais négatifs, 49 vrais positifs, 2 faux négatifs et 5 faux positifs. Elle a atteint une précision de $92,31\%$, un rappel de $96,08\%$, une spécificité de $87,5\%$, une précision de $90,74\%$ et un score F1 de $93,34\%$, démontrant ainsi une efficacité élevée pour la prédiction des maladies cardiaques.
L'étude présente plusieurs limitations : les données sont limitées à un ensemble de 303 échantillons, ce qui peut affecter la généralisation des résultats. Elle dépend fortement de la qualité des données médicales utilisées, ce qui peut influencer l'exactitude des résultats. Les modèles utilisés sont uniquement supervisés, et il n'y a pas d'exploration des techniques d'apprentissage profond ou non supervisé qui pourraient offrir des avantages. De plus, l'absence d'analyse en temps réel signifie que les modèles sont basés sur des données statiques sans suivi médical évolutif.

### Hybrid CNN and LSTM Network For Heart Disease Prediction [75]

L'auteur dans [75] a proposé un modèle hybride de prédiction des maladies cardiaques combinant les réseaux de neurones convolutifs (CNN) et les réseaux de neurones à mémoire à long terme (LSTM). Le but est d'améliorer la précision du diagnostic en exploitant les avantages complémentaires de ces deux architectures : la capacité du CNN à extraire automatiquement des caractéristiques locales importantes et la mémoire séquentielle du LSTM adaptée aux relations temporelles entre les données médicales.
Les auteurs ont utilisé un jeu de données issues de la UCI Heart Disease, comprenant 303 enregistrements médicaux. Les données ont été nettoyées pour éliminer les valeurs manquantes, puis normalisées à l'aide de la méthode Z-score afin de garantir l'homogénéité des échelles et la qualité des données pour l'entraînement du modèle. La sélection des variables pertinentes est réalisée à l'aide de la méthode Weight by SVM, qui évalue l'importance de chaque attribut à partir de son score F. Les auteurs conçoivent un modèle hybride combinant 5 couches CNN (convolution et pooling) et un LSTM en sortie, avant passage dans une couche entièrement connectée et un classifieur Softmax pour prédire la présence ou l'absence de maladie cardiaque.
Le modèle hybride CNN-LSTM atteint une précision de 89%, une sensibilité de 81% et une spécificité de 93%. Ces résultats sont comparés à ceux d'autres algorithmes d'apprentissage automatique traditionnels tels que les SVM, les arbres de décision et le naïf Bayes. Le modèle hybride CNN-LSTM se montre plus performant que ces méthodes traditionnelles.
Parmi les limitations de cette étude on cite : une taille et une diversité de jeu de données limitées (303 enregistrements), l'absence de tests sur des données médicales en conditions réelles, peu de comparaison avec d'autres architectures de deep learning avancées, ainsi qu'une absence d'analyse approfondie de l'importance des variables dans les prédictions du modèle.

### An efficient honey badger based Faster region CNN for chronc heart Failure prediction [72]

Les auteurs dans [72] étudient comment prédire les insuffisances cardiaques chroniques à partir de signaux d'électrocardiogramme (ECG). Les signaux ECG contiennent souvent des bruits et des interférences qui compliquent l'analyse. Pour résoudre ce problème, les auteurs ont développé un modèle de réseau de neurones convolutifs rapide (Faster R-CNN) optimisé par un algorithme inspiré du comportement des blaireaux (Honey Badger Algorithm, HBA) nommée HBA-FRCNN. Ce modèle a été appliqué à des bases de données d'ECG de patients atteints d'insuffisance cardiaque chronique et de personnes en bonne

santé, afin d'améliorer la précision du diagnostic. Les auteurs utilisent des signaux ECG provenant de deux bases : l'une sur l'insuffisance cardiaque (BIDMC CHF Database) et l'autre sur des rythmes cardiaques normaux (MIT-BIH NSR Database). Pour le prétraitement des signaux ECG, les auteurs ont utilisé l'algorithme Delayed Normalized Least Mean Square (DNLMS) pour réduire les bruits et les interférences dans les signaux ECG. Il ont réalisé aussi une extraction du complexe QRS (ondes Q, R, S) à partir des signaux ECG, puis application des techniques DCT et FFT pour compresser et transformer les données tout en préservant les informations essentielles.
Le modèle atteint une précision de 98.65%, une sensibilité de 98.2%, une spécificité de 98.5% et un PPV de 97.81%. Le modèle surpasse d'autres approches comme DCNN, SVM, DMSFNet ou BiLSTM-BO en termes de rapidité (0.024 s de convergence) et de consommation mémoire (625 Ko). Il atteint jusqu'à 100% de précision pour certaines classes CHF.
L'étude présente plusieurs limitations : Le modèle n'a été testé que sur des bases de données spécifiques, ce qui limite son évaluation sur d'autres types de données, comme les signaux PPG. La complexité du modèle nécessite des ressources computationnelles importantes. Les jeux de données peuvent être déséquilibrés, affectant les performances sur les classes minoritaires. Le temps de prédiction peut poser un défi pour une utilisation en temps réel.

### Predicting Heart Disease Algorithm Using DNN & MNN in Deep Learning [43]

Les auteurs dans [43] proposent un système de prédiction des maladies cardiaques basé sur des algorithmes de deep learning, s'appuyant sur deux architectures : Deep Neural Network (DNN) et Multilayer Neural Network (MNN). Face aux limites des approches classiques de machine learning dans la détection précoce des maladies cardiovasculaires, cette étude vise à améliorer la précision de prédiction en exploitant les capacités de modélisation des réseaux de neurones profonds.
Les auteurs utilise ont utilisé pour entrainer et tester leurs modèles des données provenant de kaggle, comprenant 1025 dossiers de patients et 14 attributs médicaux : âge, sexe, tension artérielle, etc. Les données sont nettoyées et normalisées afin d'éliminer les valeurs aberrantes et les redondances. Une attention particulière est accordée à la préparation des données pour assurer l'efficacité de l'entraînement des modèles.
L'approche utilisant un réseau multicouche (MNN) offre les meilleurs résultats en termes de précision et d'exactitude. Le modèle MNN atteint des scores d'exactitude et de précision de 80% et 88%, prouvant son efficacité. Le Deep Neural Network (DNN), utilisé seul,

affiche de bonnes performances en rappel (95%) et un score F1 de 79%, mais une précision et une exactitude moindres, respectivement 68% et 75%.
L'étude présente plusieurs limites : l'utilisation d'un seul jeu de données sans validation sur d'autres bases cliniques, une optimisation incomplète des hyperparamètres, une gestion du déséquilibre des classes non explicitement abordée, ainsi qu'une comparaison restreinte aux seuls modèles DNN et MNN sans évaluation d'autres architectures avancées.

### Graph Neural Networks for Heart Failure Prediction on an EHR-Based Patient Similarity Graph [13]

Les auteurs dans [13] explorent la prédiction de l'insuffisance cardiaque (IC) en utilisant des graphes de similarité entre patients dérivés de dossiers de santé électroniques (DSE). Face à la complexité des données cliniques et à leur déséquilibre, l'étude propose une approche novatrice basée sur les réseaux de neurones graphiques (GNN) et un Graph Transformer (GT) pour capturer les relations implicites entre patients.
Les auteurs utilisent des données de MIMIC-III, qui contient des informations sur les diagnostics, procédures et médicaments des patients. La première étape consiste en la construction du graphe de similarité des patients où les auteurs utilisent des embeddings de concepts médicaux pré-entraînés (300 dimensions, méthode Skip-gram) pour représenter chaque visite et chaque patient. Ensuite, un graphe est construit via l'algorithme KNN (K=3) selon la similarité cosinus entre les patients.
Les auteurs proposent trois modèles GNN : GraphSAGE agrège les caractéristiques des nœuds voisins, GAT (Graph Attention Network) pondère leur importance à l'aide de mécanismes d'attention, et le Graph Transformer (GT) applique une attention avancée inspirée des modèles Transformer pour mieux capturer les relations complexes entre patients.
Le modèle Graph Transformer (GT) a montré les meilleures performances avec un F1 score de 0.5361 et une AUROC de 0.7925. L'impact des données cliniques a montré que les codes de médicaments ont eu le plus grand impact sur la prédiction.
L'étude présente plusieurs limitations : L'étude est restreinte aux données MIMIC-III, ce qui limite la généralisation des résultats. Le recours aux codes ICD-9 pour étiqueter l'insuffisance cardiaque peut introduire des imprécisions. De plus, l'utilisation d'une seule division fixe des données et la fiabilité encore discutable des mécanismes d'attention pour l'interprétation constituent d'autres limites importantes du travail.

### Enhancing heart disease prediction using a self-attention-based transformer model [61]

Les auteurs dans [61] notent que les modèles traditionnels du deep learning comme les CNN et RNN présentent des limites dans la modélisation des dépendances complexes entre variables médicales. Pour répondre à ces limites, Les auteurs développent un modèle Transformer à base de self-attention et multi-head attention. Chaque dossier patient est représenté par des vecteurs d'embedding combinant les caractéristiques numériques et catégorielles. Une position encoding est ajouté pour conserver l'ordre des données séquentielles, suivi de couches d'encodage Transformer et d'un réseau feed-forward pour la classification finale.

Les auteurs utilisent des données provenant de la Cleveland de l'UCI, comprenant 303 dossiers de patients avec 14 caractéristiques cliniques comme l'âge, le cholestérol, la tension artérielle, et les résultats d'électrocardiogrammes. Les données ont été nettoyées des valeurs manquantes, les variables catégorielles encodées et les valeurs numériques normalisées. Les valeurs aberrantes ont été détectées via la méthode Z-score.

Le modèle atteint une précision de $96,51\%$ sur le jeu de données principal, surpassant tous les modèles comparatifs en termes de précision et d'interprétabilité. Une validation supplémentaire sur un second jeu de données (Cardiovascular Disease Dataset de Kaggle) confirme la robustesse du modèle avec une précision de $95,2\%$.

L'étude présente plusieurs limitations : L'architecture est difficile à interpréter. L'entraînement nécessite des ressources importantes. Elle est sensible à la qualité et à la diversité des données.

### Heart Disease Detection Using Graph Neural Networks [64]

Les auteurs dans [64] explorent le potentiel des Graph Neural Networks (GNN) pour la prédiction des maladies cardiaques, en réponse aux limites des modèles traditionnels qui peinent à capturer les relations complexes entre attributs médicaux et facteurs de risque. L'étude mobilise des graphes construits à partir de données hétérogènes de patients et évalue quatre architectures GNN : quatre architectures de GNN : Graph Attention Networks (GAT), Graph Convolutional Networks (GCN), GraphSAGE, et Graph Isomorphism Networks (GIN).

Les auteurs utilisent des données médicales diversifiés incluant attributs cliniques, démographiques et historiques. Les données sont nettoyées, normalisées et codées (one-hot et label encoding). Les données patient sont modélisées en graphes où chaque patient est un nœud et les relations médicales ou démographiques sont des arêtes. Les caractéristiques des patients forment les attributs de nœuds et les relations significatives déterminent les

arêtes.
Le modèle GIN offre les meilleures performances grâce à sa capacité à capturer les relations complexes dans les données structurées en graphe. Il atteint une précision de $94,15\%$, un rappel de $98,35\%$, et un F1-score de $94,55\%$, démontrant son efficacité pour le diagnostic précoce et la gestion personnalisée des patients à risque.
L'étude présente plusieurs limitations : une taille de jeu de données limitée réduisant la généralisation des résultats ; des relations entre patients définies manuellement, pouvant introduire des biais ; un coût computationnel élevé pour l'entraînement et le déploiement des modèles GNN ; et l'absence de validation clinique en conditions réelles.

### Early Detection of Heart Disease with Graph Neural Network [82]

Les auteurs dans [82] explorent l'utilisation des Graph Neural Networks (GNNs) pour la détection précoce des maladies cardiaques, en soulignant l'importance de capter les relations complexes entre les attributs cliniques et démographiques. Les auteurs proposent de structurer les données sous forme de graphes pour améliorer la détection des cas à risque et anticiper les complications cardiaques. Les auteurs utilisent deux modèles GNN sont comparés : Graph Convolutional Network (GCN) agrège les informations des nœuds voisins via des convolutions et Graph Attention Network (GAT) pondère les informations des nœuds voisins via un mécanisme d'attention.
Les auteurs utilisent des données issues de la UCI Heart Disease dataset, comprenant 14 attributs principaux (âge, cholestérol, tension, etc.) et un total de 297 patients. Les données ont été nettoyer des valeurs manquantes et doublons. Les variables corrélées (comme oldpeak et slope) ont été identifiées et la distribution des classes montre un léger déséquilibre avec plus de patients sains. Les données tabulaires CSV sont transformés en graphes dirigés où chaque patient devient un nœud et certaines paires d'attributs (par ex. âge et fréquence cardiaque maximale) deviennent des arêtes. Le graph contient 297 nœuds et 297 arêtes.
Les résultats montrent un léger sur-apprentissage dans les deux modèles, avec des écarts d'exactitude acceptables entre l'entraînement et les tests. Le modèle GAT présente des résultats plus stables et un meilleur rappel (75%), tandis que le modèle GCN offre une précision plus élevée ($85,71\%$). Ces différences s'expliquent par leur fonctionnement : GCN utilise des convolutions locales pour propager l'information, tandis que GAT applique des poids d'attention pour donner plus d'importance aux nœuds les plus pertinents.
L'étude présente plusieurs limitations : une taille de jeu de données limitée, ce qui réduit la capacité des modèles GNN à généraliser leurs résultats. De plus, aucune comparaison

n'a été réalisée avec des modèles classiques de machine learning comme SVM ou Random Forest. Enfin, il existe un risque d'overfitting et d'underfitting selon les réglages et la répartition des données.

### Heart Disease Prediction using Graph Neural Network [81]

Les auteurs dans [81] proposent un modèle de prédiction des maladies cardiaques utilisant les Graph Neural Networks (GNN). Les auteurs ont appliqué leur modèle à un ensemble de données publiques issu de quatre bases de données médicales (Cleveland, Hungary, Switzerland et Long Beach V), en utilisant 14 variables caractéristiques.
Le jeu de données utilisé regroupe quatre bases hospitalières de 1988, totalisant 76 attributs dont seulement 14 sont conservés pour la prédiction (âge, sexe, cholestérol, pression artérielle, etc.). Le jeu de données présente un déséquilibre entre les classes (malade / non malade). Les auteurs ont analysé la distribution des caractéristiques et observé qu'il y avait plus d'hommes âgés de 52 à 68 ans. Des visualisations avec la bibliothèque Seaborn ont permis d'explorer ces répartitions.
Le modèle Graph Neural Network (GNN) a été testé avec cinq optimiseurs : RMSProp, Adam, Adadelta, Adagrad et SGD. RMSProp a obtenu la meilleure précision avec 92%, surpassant les autres.
L'étude présente plusieurs limitations : Le modèle utilise un jeu de données ancien (1988) et limité à 14 attributs. Il est testé uniquement sur un jeu de données spécifique, ce qui limite sa généralisation à d'autres populations. De plus, l'étude ne compare pas encore plusieurs modèles GNN et algorithmes classiques. Enfin, un risque de sur-lissage existe si trop de couches sont ajoutées, rendant les représentations des nœuds trop similaires.

## 3 Comparaison qualitative des travaux reliées

Le Tableau 2.1 fournit une comparaison qualitative des travaux reliés présentés précédemment, en se basant sur des critères techniques tels que l'approche adoptée, le jeu de données utilisé, les performances obtenues, la complexité, ainsi que les avantages et les inconvénients de chaque méthode. pdflscape

| # | Auteur / Année | Approche | Dataset | Performance | Complexité | Avantages | Inconvénients |
|---|---|---|---|---|---|---|---|
| 1 | Choi et al.,2016 [16] | GRU sur EHR | Sutter Health (3 884 IC, 28 903 contrôles) | AUC 0,883 | Élevée | Prédiction précoce, gestion temporelle | Données d'un seul système |
| 2 | Samuel et al.,2016 [68] | ANN + Fuzzy AHP | UCI Cleveland (297 patients, 13 attr.) | Acc 91,1%, Sens 100%, Spec 84% | Moyenne | Pondération précise, haute sensibilité | Expertise humaine requise, échantillon limité |
| 3 | Miao et al.,2018 [46] | DNN | UCI Cleveland (282 patients, 28 attr.) | Acc 83,7%, Sens 93,5% | Moyenne | Bonne sensibilité | Risque de surapprentissage |
| 4 | Jin et al., 2018 [32] | LSTM sur EHR | EHR (5 000 IC, 15 000 contrôles) | Non spécifié | Élevée | Exploite la temporalité | Prétraitement complexe, coût élevé |
| 5 | Ramprakash et al., 2020[63] | DNN + $^2$ | UCI Cleveland (303 patients, 14 attr.) | Acc 99% | Élevée | Très haute précision | Échantillon limité |
| 6 | Baccouche et al., 2020 [37] | CNN + BiLSTM/BiGRU | Medica Norte (800 patients, 141 attr.) | Acc 91-96%, AUC 99% | Élevée | Haute précision | Modèle lourd, données spécifiques |

Suite à la page suivante

Tab. 2.1 – suite

| # | Auteur / Année | Approche | Dataset | Performance | Complexité | Avantages | Inconvénients |
|---|---|---|---|---|---|---|---|
| 7 | Krishnan et al., 2021 [37] | RNN + GRU + SMOTe | UCI Cleveland (303 patients, équilibré) | Acc 98,7%, Sens 100% | Très élevée | Précision élevée | Temps d'entraînement long |
| 8 | Gupta et al., 2022 [27] | ML supervisé (LR, SVM, RF, KNN, NB, DT) | UCI Cleveland (303 patients, 14 attr.) | Acc 92,3%, F1 93,3% | Faible | Modèles simples | Moins robuste pour cas complexes |
| 9 | Sudha, 2023 [75] | CNN + LSTM | UCI Cleveland (303 patients, 14 attr.) | Acc 89%, Sens 81%, Spec 93% | Élevée | Combine CNN et LSTM | Complexe à régler |
| 10 | Sherly & Mathivanan, 2023 [72] | Faster R-CNN + HBA | ECG (BIDMC CHF, MIT-BIH NSR) | Acc 98,8%, Sens 98,2% | Élevée | Précis pour ECG | Limité aux ECG |
| 11 | Mary & Ramaprabha, 2024 [43] | DNN & MNN | Kaggle (1 025 patients, 14 attr.) | Acc 80% (MNN) | Moyenne | Simple, rapide | Moins performant sur cas complexes |
| 12 | Boll et al., 2024 [13] | GNN (GraphSAGE, GAT, GT) | MIMIC-III | AUC 0,793, F1 0,54 | Élevée | Bonne modélisation | Données limitées |

Suite à la page suivante

Tab. 2.1 – suite

| # | Auteur / Année | Approche | Dataset | Performance | Complexité | Avantages | Inconvénients |
|---|---|---|---|---|---|---|---|
| 13 | Rahman et al., 2024 [61] | Transformer | UCI Cleveland + Kaggle (303 patients) | Acc 96,5% | Élevée | Haute précision | Complexe, coûteux |
| 14 | Rangisetti et al., 2024 [64] | GNN (GIN, GCN, GAT, GraphSAGE) | Données hétérogènes | Acc 94,2%, F1 94,6% | Élevée | Très performant | Coût élevé |
| 15 | Gunawan et al., 2024 [82] | GCN, GAT | UCI Cleveland (297 patients, 14 attr.) | Acc 78,3% | Moyenne | Modélise relations | Données limitées |
| 16 | Wajgi et al., 2024 [81] | GNN | UCI (Cleveland, Hungary, etc., 14 attr.) | Acc 92% | Moyenne | Bonne précision | Données anciennes |

**Tableau 2.1 :** Comparaison qualitative des travaux liés à la prédiction des maladies cardiaques

# 4 Discussions et limites des approches existantes

Après une revue approfondie de 16 articles scientifiques portant sur la détection des maladies cardiaques, il apparaît que les Graph Neural Networks (GNN), et en particulier le modèle GIN, surpassent nettement les approches traditionnelles. Le modèle GIN atteint une précision de $94,15\%$, un rappel élevé de $98,35\%$ et un F1-score équilibré de $94,55\%$, démontrant ainsi son efficacité à détecter les cas positifs tout en réduisant significativement les erreurs de diagnostic. Ces performances s'expliquent par sa capacité à modéliser les relations complexes entre patients, symptômes, diagnostics et antécédents médicaux sous forme de graphes. Il est vrai que les méthodes classiques, telles que les modèles traditionnels (SVM, RF, LR), les réseaux neuronaux standards (DNN, MLP) ou les architectures séquentielles (LSTM, GRU), offrent des performances satisfaisantes sur des jeux de données tabulaires simples.

Toutefois, elles restent limitées dans leur capacité à capturer des interactions cliniques non linéaires et des dépendances complexes entre les différentes variables médicales. À l'inverse, les GNN exploitent la structure relationnelle des données médicales sous forme de graphes, offrant une meilleure capacité à capturer des interactions complexes et à gérer la nature hétérogène des dossiers médicaux. Plusieurs études confirment leur supériorité en termes de précision, de flexibilité et de facilité d'interprétation.

Cependant, l'analyse des études précédentes révèle que la majorité des approches GNN se limitent à des données tabulaires, négligeant ainsi la structure multimodale propre aux dossiers médicaux. En pratique, ces derniers intègrent à la fois des variables cliniques structurées et des signaux physiologiques complexes, tels que les électrocardiogrammes (ECG).

C'est dans cette perspective que notre travail propose une approche multimodale originale, combinant des images ECG et des variables cliniques afin d'enrichir la représentation de chaque patient. Cette combinaison permet de capter à la fois des motifs visuels caractéristiques de pathologies cardiaques et des indicateurs numériques pertinents, améliorant ainsi la capacité du modèle à détecter précocement des situations cliniques complexes et à mieux représenter les cas réels.

Par ailleurs, contrairement aux approches antérieures reposant sur des architectures lourdes et coûteuses comme GIN ou GraphSAGE, notre méthodologie privilégie des modèles plus légers et accessibles, notamment GCN et GAT. Ceux-ci sont appliqués à des graphes de similarité construits à partir de données multimodales, permettant d'assurer un équilibre optimal entre performance, coût computationnel et intégrabilité dans des environnements médicaux réels.

# 5 Conclusion

Dans ce chapitre, nous avons présenté une revue des études pertinentes sur la prédiction de l'insuffisance cardiaque. Nous avons ensuite effectué une comparaison entre les travaux existants sous forme de tableau synthétique. Par la suite, nous avons discuté les limites des approches actuelles. Enfin, nous avons proposé quelques idées pour améliorer les méthodes de prédiction.

Dans le chapitre suivant, nous appliquons notre approche proposée, visant à améliorer les performances de la prédiction de l'insuffisance cardiaque à travers la conception et l'implémentation d'un modèle adapté.

# Chapitre 3

# Conception

## 1 Introduction

Dans le chapitre précédent, nous avons mis en évidence les avancées significatives de l'apprentissage automatique dans le domaine de la détection des maladies cardiovasculaires, et plus particulièrement de l'insuffisance cardiaque. Parmi les différentes approches explorées, les modèles de réseaux de neurones graphiques (GNN) ont démontré une nette supériorité en termes de précision, grâce à leur capacité à capturer les relations complexes entre les entités médicales.

Dans ce chapitre, nous approfondissons cette approche en appliquant des modèles GNN, notamment le Graph Convolutional Network (GCN) et le Graph Attention Network (GAT), sur un jeu de données multimodal combinant des images ECG et des variables cliniques simulées. Cette combinaison vise à reproduire un cadre d'analyse réaliste et riche, où chaque patient est représenté non seulement par ses signaux visuels, mais également par ses paramètres physiologiques pertinents.

Nous commencerons par présenter l'architecture détaillée du système, en débutant par la description du jeu de données, suivie d'un prétraitement complet incluant le nettoyage, l'équilibrage des classes, la génération de variables cliniques réalistes, la division des données et la préparation des caractéristiques. Ensuite, nous aborderons l'extraction des caractéristiques visuelles à l'aide du modèle ResNet18, puis la construction d'un graphe de similarité entre patients. Cette étape sera suivie par la présentation des modèles de graphes neuronaux utilisés (GCN et GAT) avec une description détaillée de leur architecture respective. Enfin, nous terminerons ce chapitre par la section relative à l'entraînement du modèle.

## 2 L'architecture détaillée du système

Pour une prédiction multimodale de l'insuffisance cardiaque, nous avons utilisé deux modèles de GNN, à savoir le Graph Convolutional Network (GCN) et le Graph Attention Network (GAT), sur un jeu de données multimodal combinant des images ECG et des variables cliniques (facteurs de risque). L'architecture de notre proposition est illustrée par la Figure 3.1.Les différents modules composant cette architecture seront détaillés dans les sections suivantes.

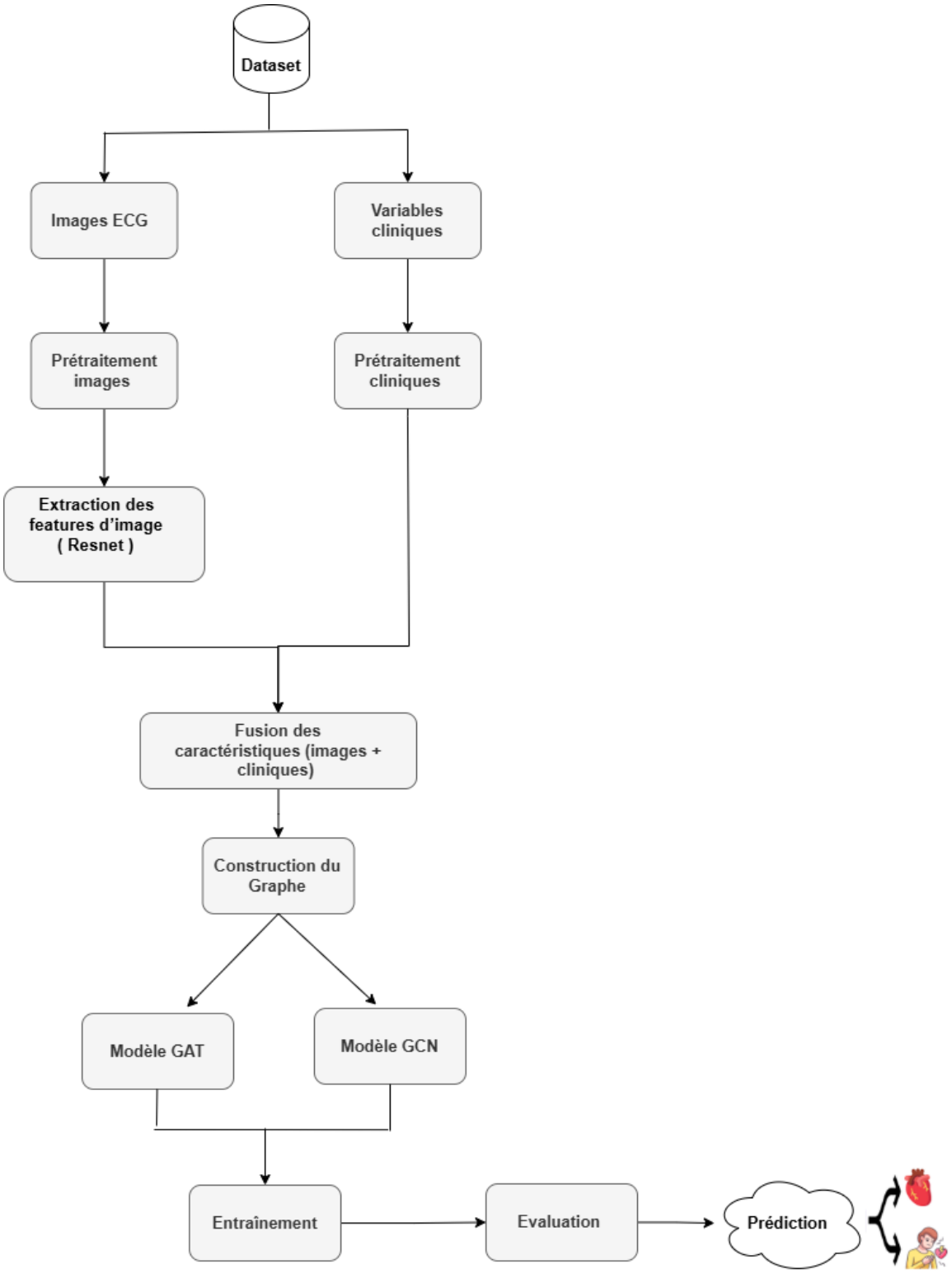



**Figure 3.1 :** Architecture de notre système

## 2.1 Description des données utilisées

Nous avons utilisé un ensemble de données multimodal composé d'images ECG et de cinq caractéristiques cliniques (facteurs de risque). L'objectif de cet enrichissement est de

permettre l'entraînement de modèles multimodaux capables de tirer parti à la fois des signaux visuels et des caractéristiques cliniques, dans le but d'améliorer la précision du diagnostic. Ce type d'approche reflète mieux les pratiques médicales réelles et nos résultats expérimentaux confirment qu'il apporte une amélioration significative des performances par rapport aux approches unimodales.
Nous présentons ci-après une illustration des deux types de données utilisés.

1. **Base d'électrocardiogrammes :** La base d'électrocardiogrammes (ECG) est extraite de la plateforme Kaggle [23], spécifiquement destinée à l'étude des pathologies cardiaques. Initialement composée de plusieurs classes, elle a été regroupée en deux catégories principales : patients malades et normaux (non malades). Structuré en deux fichiers principaux : fichier d'entraînement (Train) et fichier de test (Test) (voir Tableau 3.1).

| **Classe** | **Train** | **Test** |
|---|---|---|
| **Malade** | 2 367 | 678 |
| **Normal** | 791 | 228 |
| **Total** | 3 158 | 906 |

**Tableau 3.1 :** Description de la dataset des ECG.

Les Figures 3.2 et 3.3 montrent un exemple d'images ECG pour un patient sain (normal) et un patient malade.

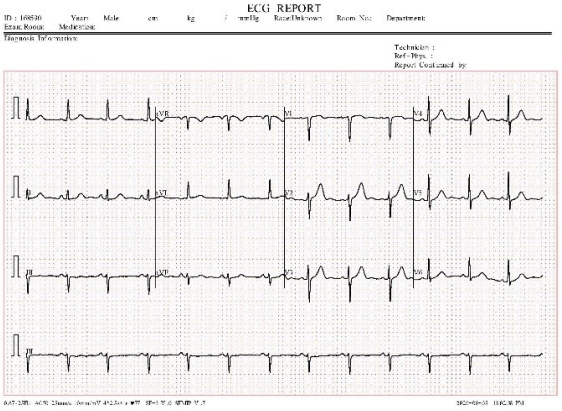



**Figure 3.2 :** ECG d'un patient sain

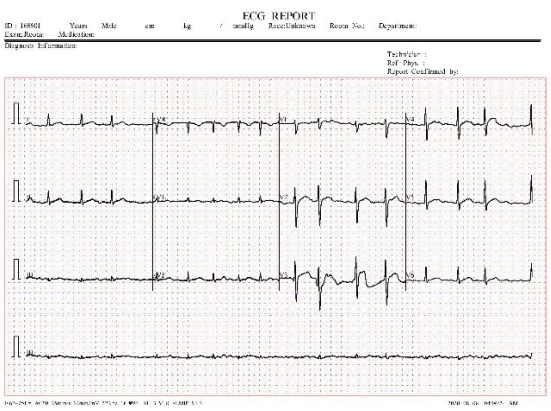



**Figure 3.3 :** ECG d'un patient pathologique

2. **Les cinq caractéristiques cliniques :**

## 1. Introduction

Après une recherche bibliographique, nous avons identifié cinq caractéristiques cliniques reconnues comme facteurs de risque pouvant améliorer la précision prédictive de nos modèles. Il s'agit de la fraction d'éjection (EF), du taux de BNP (Brain Natriuretic Peptide), de la classification NYHA, de la pression artérielle systolique (PAS) et de l'âge. En l'absence d'un jeu de données public combinant à la fois des images ECG et ces informations cliniques dans une même base, nous avons généré aléatoirement ces cinq variables. Leur simulation a été réalisée à l'aide de distributions statistiques conditionnées par l'état de santé (malade ou non), avec des paramètres distincts (moyennes et écarts-types) pour chaque groupe, sur la base de plages de valeurs cliniquement réalistes.

## 2. Description des cinq facteurs cliniques

Nous présentons dans ce qui suit une description détaillée des cinq variables cliniques. Les valeurs ont été générées à l'aide des fonctions suivantes :

*np.random. normal($\mu$, $\sigma$)* pour EF, SBP et l'âge, *np.random.lognormal($\mu$, $\sigma$)* pour BNP, et *np.random.choice(...)* pour NYHA, où $\mu$ est la moyenne (valeur centrale des données) et $\sigma$ l'écart-type (mesurant leur dispersion autour de cette moyenne).

(a) **La fraction d'éjection (EF)** La fraction d'éjection (EF) est le pourcentage de sang que le ventricule gauche envoie à chaque battement du cœur. Une valeur normale est généralement entre 60 % et 70 %. Quand une personne souffre d'insuffisance cardiaque, cette valeur descend souvent en dessous de 40 %, ce qui en fait un indicateur très important pour détecter la maladie[66].

Dans notre travail, nous avons utilisé une distribution normale (gaussienne) pour générer des valeurs d'EF réalistes. Cette méthode permet de créer des données qui ressemblent aux vrais cas cliniques. La distribution normale est définie par deux éléments :

- $\mu$ **(la moyenne) :** c'est la valeur centrale, autour de laquelle se trouvent la majorité des données.
- $\sigma$ **(l'écart-type) :** il montre à quel point les valeurs sont dispersées autour de la moyenne.

Nous avons limité les valeurs de l'EF entre 0,20 (20 %) et 0,75 (75 %). Moins que 0,20 est très rare et souvent dangereux, et plus que 0,75 est irréaliste médicalement. Ensuite, nous avons utilisé deux cas différents selon l'état de la personne :

- Si la personne n'est pas malade : Moyenne = 0,60 (60 %), Écart-type = 0,07 (valeurs proches de la moyenne).
- Si la personne est malade : Moyenne = 0,38 (38 %), Écart-type = 0,09 (valeurs plus dispersées).

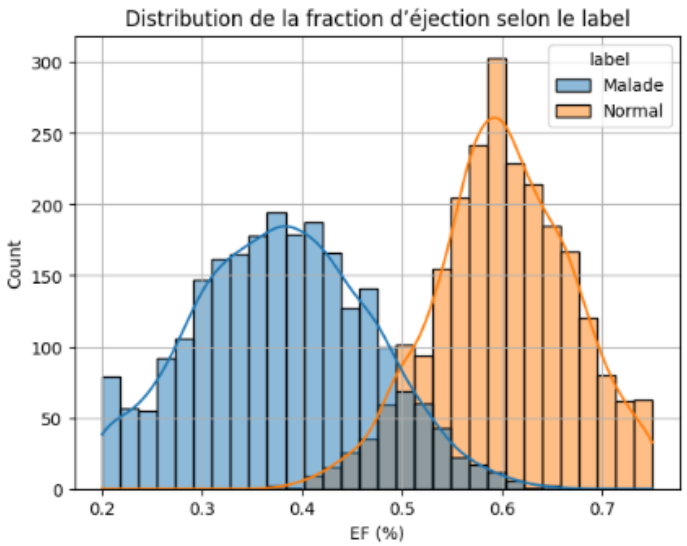



**Figure 3.4 :** Distribution de la fraction d'éjection (EF) selon l'état de santé (Malade / Normal)

(b) **BNP (Brain Natriuretic Peptide)** Le BNP est une hormone libérée par le cœur lorsqu'il est soumis à un stress, par exemple en cas d'insuffisance cardiaque. Plus le taux de BNP est élevé, plus le risque de maladie cardiaque est important. Une valeur normale de BNP est généralement inférieure à 100 pg/mL. Entre 100 et 400 pg/mL, on parle d'une zone intermédiaire qui peut nécessiter des examens complémentaires. Au-delà de 400 pg/mL, cela suggère fortement une insuffisance cardiaque, surtout si les symptômes cliniques sont présents[54].

Dans notre travail, nous avons utilisé une distribution log-normale, adaptée lorsque :

- Les valeurs sont **toujours positives** (aucune valeur négative).
- Il existe une **forte variabilité** (certaines valeurs peuvent être très faibles, d'autres très élevées).

Ensuite, nous avons utilisé deux cas différents selon l'état de la personne :

- Si la personne n'est pas malade : moyenne ($\mu$) du log(BNP) = 4,2.
- Si la personne est malade : moyenne ($\mu$) du log(BNP) = 5,8.
- Écart-type ($\sigma$) fixé à 0,6 pour les deux cas (malade et non malade).

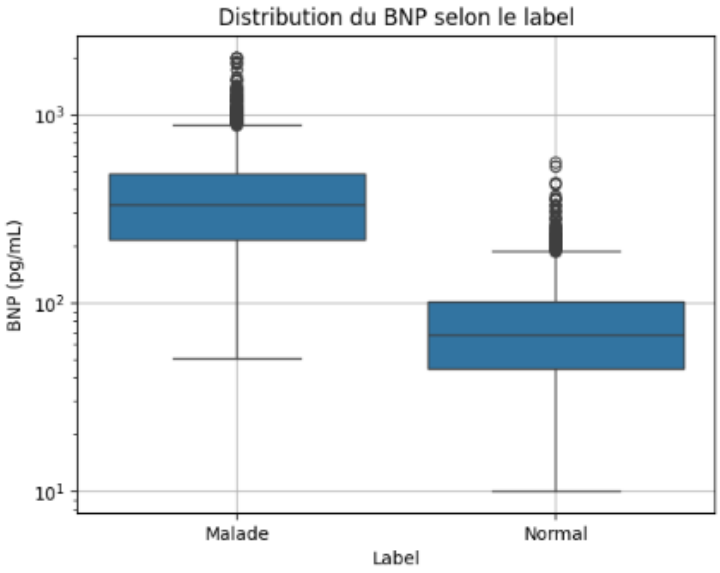



**Figure 3.5 :** Distribution du BNP selon l'état de santé (Malade / Normal)

(c) **NYHA (New York Heart Association)** La classification NYHA est un système clinique utilisé pour évaluer la sévérité des symptômes chez les patients atteints d'insuffisance cardiaque. Elle classe les patients de 1 à 4 en fonction de leur capacité à effectuer les activités quotidiennes[4], où :

- Classe 1 : aucune limitation des activités physiques.
- Classe 2 : légère limitation – le patient est à l'aise au repos.
- Classe 3 : limitation marquée – des symptômes apparaissent même lors d'activités légères.
- Classe 4 : symptômes même au repos, invalidité importante.

Dans notre travail, nous avons attribué une classe NYHA de manière probabiliste, selon que la personne soit malade ou non :

- Si la personne n'est pas malade : Probabilités : [0,50, 0,30, 0,15, 0,05]. La majorité des patients seront en classe 1 ou 2, avec 50 % en classe 1 et 30 % en classe 2.
- Si la personne est malade : Probabilités : [0,10, 0,30, 0,40, 0,20]. La majorité des patients seront en classe 3 ou 4, avec 40 % en classe 3 et 20 % en classe 4.

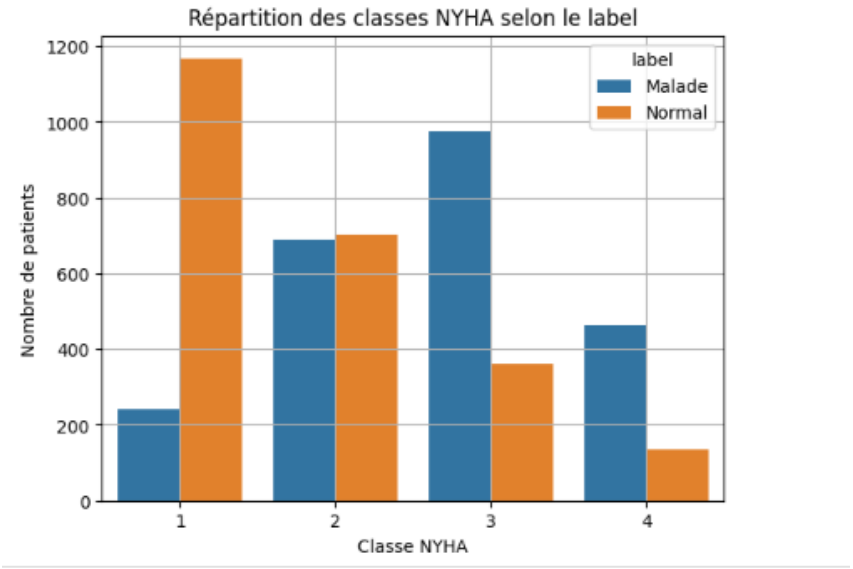



**Figure 3.6 :** Distribution des classes NYHA selon l'état de santé (Malade / Normal)

(d) **SBP (Systolic Blood Pressure)** La pression artérielle systolique (SBP) représente la pression dans les artères lorsque le cœur se contracte pour pomper le sang. C'est l'un des indicateurs essentiels de la santé cardiovasculaire.

- Chez une personne en bonne santé, la SBP est généralement autour de 120 à 130 mmHg.
- Chez une personne atteinte d'insuffisance cardiaque, la pression peut être plus basse en raison d'un débit cardiaque diminué, et peut descendre vers 100 mmHg, voire moins.

Dans notre travail, nous avons utilisé une distribution normale pour simuler la SBP selon l'état du patient :

- Si la personne n'est pas malade : Moyenne ($\mu$) = 125, Écart-type ($\sigma$) = 15.
- Si la personne est malade : Moyenne ($\mu$) = 110, Écart-type ($\sigma$) = 15.

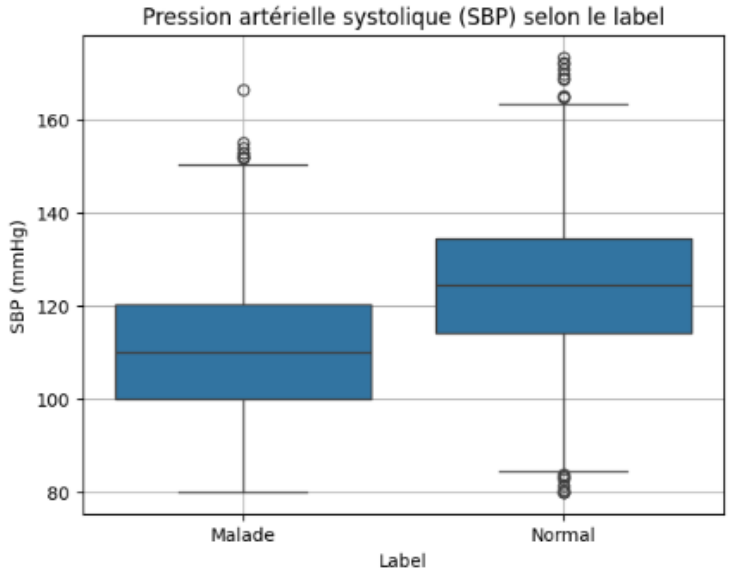



**Figure 3.7 :** Distribution de SBP selon l'état de santé (Malade / Normal)

(e) **Âge (Age)** L'âge est un facteur de risque très important dans les maladies cardiovasculaires. Plus une personne est âgée, plus le risque d'insuffisance cardiaque ou d'autres problèmes cardiaques augmente. Dans notre travail, nous avons utilisé une distribution normale pour simuler l'âge des patients selon leur état de santé :

- Si la personne n'est pas malade : Moyenne ($\mu$) = 55 ans, Écart-type ($\sigma$) = 10 ans.
- Si la personne est malade : Moyenne ($\mu$) = 68 ans, Écart-type ($\sigma$) = 10 ans.

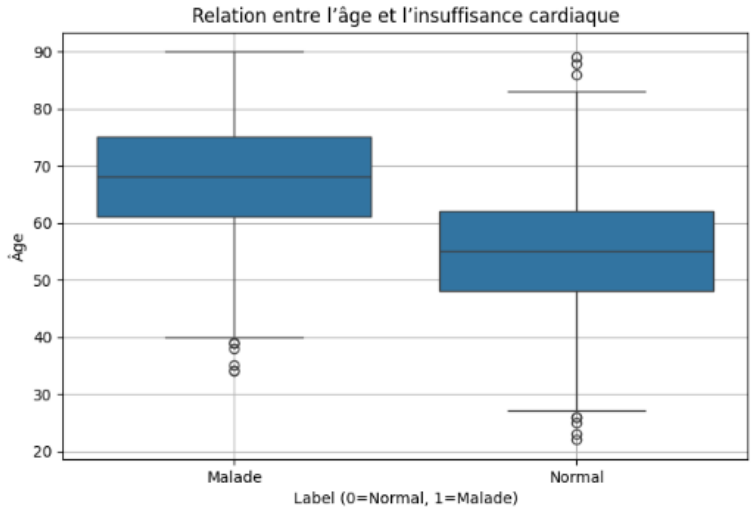



**Figure 3.8 :** Distribution de l'âge selon l'état de santé (Malade / Normal)

Le Tableau 3.2 montre un résumé des caractéristiques cliniques.

| Variable | État de santé | Distribution | Moyenne ($\mu$) | Écart-type ($\sigma$) | Intervalle |
|---|---|---|---|---|---|
| **EF** | Non malade | Normale | 0,60 | 0,07 | [0,20 – 0,75] |
| | Malade | Normale | 0,38 | 0,09 | [0,20 – 0,75] |
| **BNP** | Non malade | Log-normale | $\log(\mu) = 4{,}2$ | 0,6 | [10 – 2000] |
| | Malade | Log-normale | $\log(\mu) = 5{,}8$ | 0,6 | [10 – 2000] |
| **NYHA** | Non malade | Discrète (probas) | – | – | Classe 1 : 50 %<br>Classe 2 : 30 %<br>Classe 3 : 15 %<br>Classe 4 : 5 % |
| | Malade | Discrète (probas) | – | – | Classe 1 : 10 %<br>Classe 2 : 30 %<br>Classe 3 : 40 %<br>Classe 4 : 20 % |
| **SBP** | Non malade | Normale | 125 | 15 | [80 – 180] |
| | Malade | Normale | 110 | 15 | [80 – 180] |
| **Âge** | Non malade | Normale | 55 | 10 | [20 – 90] |
| | Malade | Normale | 68 | 10 | [20 – 90] |

**Tableau 3.2 :** Résumé des caractéristiques cliniques

### 3. Analyse des corrélations entre les cinq facteurs de risque

Afin de mieux comprendre les relations entre les cinq caractéristiques cliniques, nous avons calculé la matrice de corrélation de Pearson. Cette analyse permet d'identifier les dépendances linéaires entre les variables, ce qui peut aider à interpréter les résultats médicaux et à ajuster les modèles prédictifs (voir Figure 3.9).

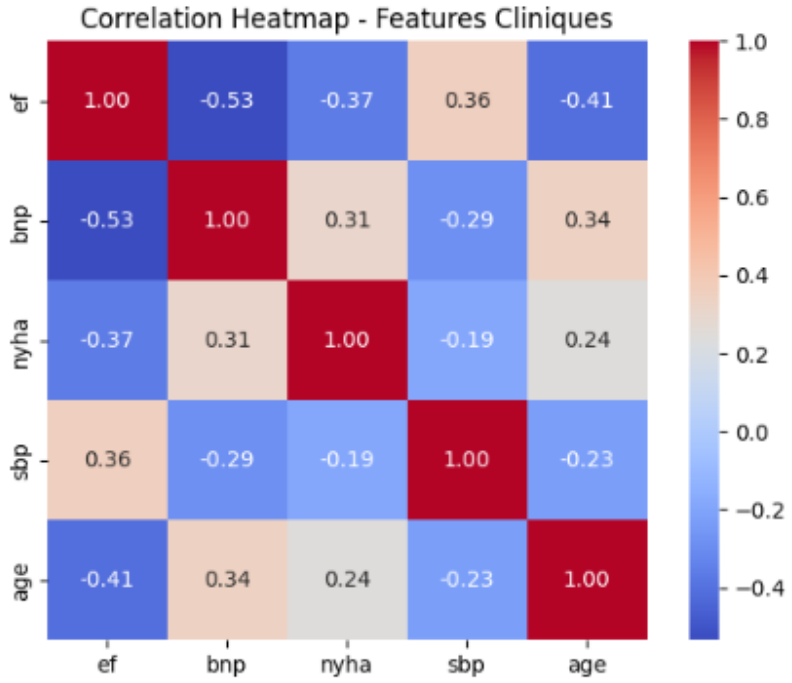



**Figure 3.9 :** Matrice de corrélation entre les variables cliniques

Les principales observations sont les suivantes :

(a) **Corrélations négatives significatives :**

- **EF et BNP :** Corrélation de -0,53. Plus la fraction d'éjection est faible, plus le taux de BNP est élevé, ce qui est cohérent sur le plan clinique, car une mauvaise fonction cardiaque provoque une augmentation du BNP.
- **EF et Âge :** Corrélation de -0,41. On observe une diminution de la fraction d'éjection avec l'âge.
- **EF et NYHA :** Corrélation de -0,37. Une fraction d'éjection réduite est associée à une classe NYHA plus élevée, indiquant des symptômes plus graves.

(b) **Corrélations positives modérées :**

- **BNP et NYHA :** Corrélation de 0,31. Le BNP tend à augmenter avec la gravité des symptômes (classe NYHA).
- **BNP et Âge :** Corrélation de 0,34. Les patients plus âgés présentent en général un taux de BNP plus élevé.
- **SBP et EF :** Corrélation de 0,36. La pression artérielle systolique est modérément liée à la fraction d'éjection.

(c) **Corrélations faibles ($|r| < 0,2$) :**

- Certaines paires comme SBP-NYHA, SBP-Âge, ou NYHA-SBP ont des corrélations faibles, ce qui indique une faible relation entre ces variables.

## 2.2 Prétraitement des données

La Figure 3.10 illustre la chronologie des prétraitements appliqués aux données. Une explication détaillée des différentes étapes est présentée ci-après :

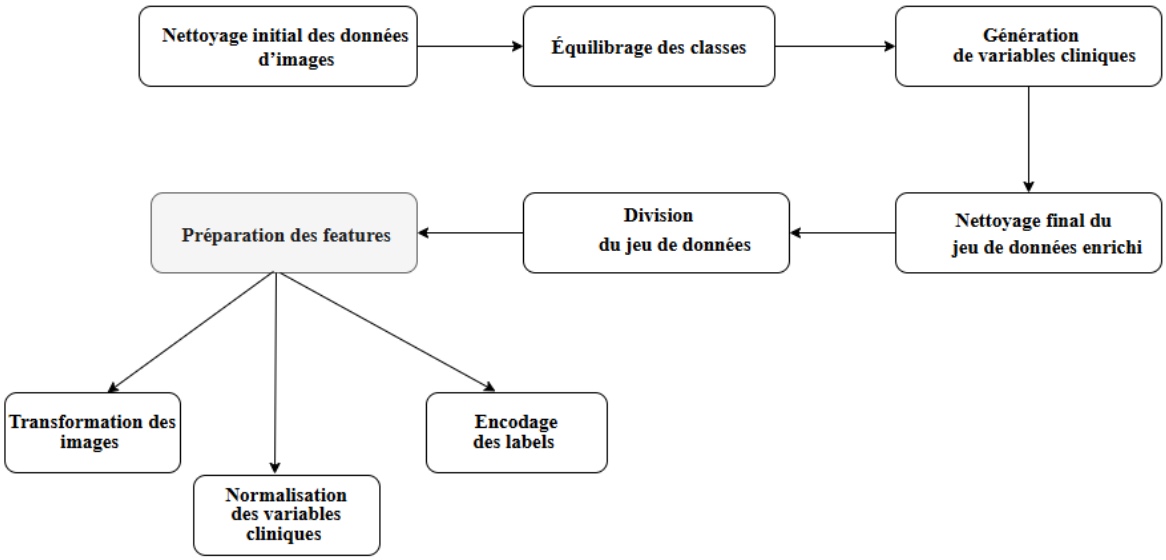



**Figure 3.10 :** Les étapes de prétraitement des données

1. **Nettoyage initial des données d'images** À cette étape, chaque image est vérifiée pour s'assurer qu'elle existe réellement, qu'elle est lisible (non manquante), et qu'elle respecte une taille minimale de 100x100 pixels. Si ces conditions sont remplies, l'image est conservée dans le jeu de données. Sinon, elle est ignorée, et un message s'affiche pour indiquer si le fichier est trop petit ou illisible. L'objectif est d'assurer un dataset propre, homogène et fiable, condition essentielle pour éviter des erreurs lors de l'entraînement du modèle et pour optimiser ses performances finales.

2. **Équilibrage des classes et augmentation des données :** Pour pallier le déséquilibre initial du jeu de données d'images ECG, où la classe « Malade » est surreprésentée par rapport à la classe « Normal », nous avons procédé à une augmentation des données. Nous avons d'abord sélectionné les images de la classe « Normal », puis appliqué des transformations aléatoires telles que rotations, translations, zooms et retournements horizontaux en utilisant le générateur `ImageDataGenerator` de Keras. Un nouveau DataFrame a ensuite été créé avec les chemins des images augmentées, qui ont été ajoutées au DataFrame d'origine afin d'obtenir un jeu de données équilibré, prêt pour l'entraînement. Cette étape est essentielle pour éviter que le

modèle ne développe un biais en faveur de la classe majoritaire, et ainsi améliorer sa capacité à généraliser sur toutes les catégories.

3. **Génération de variables cliniques :** Après le traitement initial des données d'images et l'équilibrage des classes, nous avons enrichi le jeu de données d'images en y ajoutant cinq variables cliniques : EF, BNP, NYHA, SBP et l'âge. Les nouvelles colonnes sont intégrées aux DataFrames initiaux, qui sont ensuite enregistrés au format CSV. Ainsi, chaque ligne du DataFrame contient désormais : le chemin de l'image (`image_path`), le label (`label`), et les 5 nouvelles colonnes : `ef`, `bnp`, `nyha`, `sbp`, `age`.

| image_path | label | ef | bnp | nyha | sbp | age |
|---|---|---|---|---|---|---|
| /content/drive/MyDri | Malade | 0.334 | 463.331 | 3 | 132 | 70 |
| /content/drive/MyDri | Malade | 0.298 | 580.505 | 1 | 104.3 | 75 |
| /content/drive/MyDri | Malade | 0.382 | 640.537 | 2 | 109.8 | 73 |
| /content/drive/MyDri | Malade | 0.4 | 133.121 | 2 | 96.9 | 58 |
| /content/drive/MyDri | Malade | 0.531 | 576.904 | 3 | 117.7 | 60 |
| /content/drive/MyDri | Malade | 0.302 | 776.536 | 3 | 106.1 | 65 |
| /content/drive/MyDri | Malade | 0.486 | 374.236 | 2 | 102.4 | 57 |
| /content/drive/MyDri | Malade | 0.406 | 561.009 | 3 | 115.9 | 89 |

**Figure 3.11 :** La structure du jeu de données

4. **Nettoyage final du jeu de données enrichi :** Le nettoyage final consiste en une dernière vérification des ensembles de données avant leur utilisation dans l'entraînement des modèles. L'objectif est de garantir que le DataFrame final soit propre, cohérent et exploitable directement par les algorithmes d'apprentissage, sans erreurs inattendues pendant l'entraînement ou la validation. C'est une étape cruciale pour la fiabilité des résultats. Concrètement, cette étape vise à :

   (a) **Supprimer les lignes incomplètes ou corrompues :** On élimine les entrées contenant des valeurs manquantes (NaN) ou non valides dans les colonnes cliniques ou les chemins d'images.

   (b) **Vérifier les doublons :** On supprime les doublons éventuels pour éviter que certaines images ou patients soient surreprésentés dans les données, ce qui fausserait l'apprentissage.

   (c) **Assurer la cohérence des types de données :** On s'assure que chaque colonne est du type attendu : par exemple, l'âge en entier, le BNP et la fraction d'éjection en float, et la classe NYHA en catégorie.

   (d) **Réindexer proprement le DataFrame :** Après suppression de lignes, on réinitialise les index pour garder un tableau propre.

5. **Division du jeu de données :** À cette étape, le jeu de données nettoyé est divisé en deux sous-ensembles : un ensemble d'entraînement et un ensemble de validation (80 % entraînement, 20 % validation). L'ensemble d'entraînement sert à apprendre les paramètres du modèle, tandis que l'ensemble de validation permet d'évaluer les performances du modèle sur des données non vues pendant l'entraînement. Le fichier de test est déjà fourni séparément, il sert à mesurer la performance finale du modèle de façon indépendante, sans aucune influence pendant l'entraînement ou la validation.

6. **Préparation des caractéristiques :** Avant de passer les données dans le modèle ResNet, plusieurs étapes de préparation sont nécessaires afin d'assurer une cohérence des formats et une exploitation optimale des différentes sources d'information.

   (a) **Encodage des labels :** Les étiquettes textuelles associées aux patients (« Normal » et « Malade ») doivent être converties en valeurs numériques, car les modèles de machine learning et deep learning ne peuvent pas traiter directement des chaînes de caractères. Pour cela, on utilise un `LabelEncoder` de la bibliothèque `sklearn`, qui attribue un entier unique à chaque catégorie (Normal encodé en 0 et Malade en 1). Ce procédé garantit que les labels soient au bon format pour la phase d'entraînement et d'évaluation.

   (b) **Normalisation des variables cliniques :** Les cinq variables cliniques sont souvent exprimées dans des échelles très différentes. Pour éviter qu'une variable ayant de grandes valeurs (comme le BNP) n'écrase l'influence des autres variables, on applique une normalisation. Le `StandardScaler` de la bibliothèque `scikit-learn` est utilisé ici pour standardiser les données cliniques : il soustrait la moyenne et divise par l'écart-type de chaque variable, de sorte qu'elles aient toutes une moyenne de 0 et un écart-type de 1, selon la formule suivante :

   $$z = \frac{x - \mu}{\sigma}$$

   Où :

   - $x$ : la valeur d'origine.
   - $\mu$ : la moyenne de la variable.
   - $\sigma$ : l'écart-type de la variable.
   - $z$ : la valeur normalisée.

   (c) **Transformation des images avant ResNet :** Les images ECG doivent également être préparées avant d'être passées dans le modèle ResNet. D'abord, chaque image est redimensionnée en 224×224 pixels pour correspondre à la

taille d'entrée attendue par ResNet, qui a été initialement entraîné sur des images de cette dimension via le dataset ImageNet. Ensuite, les images sont converties en tenseurs PyTorch et normalisées en utilisant les moyennes et écarts-types des canaux RGB calculés sur ImageNet (Moyennes : [0,485, 0,456, 0,406], Écarts-types : [0,229, 0,224, 0,225]). Ces transformations assurent la compatibilité avec le modèle préentraîné et améliorent la qualité des représentations extraites.

## 2.3 Extraction des caractéristiques d'image

Pour extraire des représentations visuelles des images ECG, nous avons choisi d'utiliser un réseau de neurones convolutif pré-entraîné, notamment ResNet18, connu pour son bon compromis entre précision et légèreté. L'idée est d'utiliser ce réseau comme extracteur de caractéristiques en supprimant sa couche de classification finale.

- **Association des caractéristiques aux variables cliniques :** Pour chaque patient, le vecteur de caractéristiques visuelles extrait à partir de l'image ECG est concaténé avec les cinq variables cliniques normalisées. Cette fusion permet de constituer un vecteur multimodal, intégrant à la fois les informations visuelles et les données cliniques.

## 2.4 Construction du graphe

Nous construisons un graphe de similarité entre les patients afin d'entraîner un modèle GCN. Dans ce graphe, chaque nœud représente un patient caractérisé par un vecteur de 517 dimensions : une représentation de 512 dimensions extraite des images ECG à l'aide d'un modèle ResNet, combinée à 5 variables cliniques standardisées (EF, BNP, NYHA, SBP, âge). Les arêtes du graphe encodent la similarité entre patients, déterminée à l'aide de l'algorithme des k plus proches voisins (k-NN) avec $k = 5$, basé sur la distance euclidienne dans l'espace des caractéristiques complètes (images + données cliniques). La distance entre deux patients $x$ et $y$ est calculée par :

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

Où $n = 517$ est le nombre total de caractéristiques utilisées.
À partir de cette distance, on construit une matrice de voisinage binaire avec la fonction `kneighbors_graph` de `Scikit-learn`, où une valeur de 1 indique une connexion entre deux nœuds (patients), et 0 sinon.

Il est important de noter que cet algorithme garantit qu'aucun nœud n'est isolé dans le graphe final : chaque patient est obligatoirement connecté à ses $k = 5$ voisins les plus proches, même si la distance qui les sépare est relativement élevée. Cette propriété permet d'éviter l'exclusion de patients atypiques ou rares dans l'entraînement du modèle, ce qui est essentiel en contexte médical afin de conserver toutes les observations et préserver la diversité des cas représentés.
La matrice de voisinage est ensuite convertie au format `edge_index` compatible avec PyTorch Geometric : une matrice de forme $[2, E]$, où chaque colonne représente une arête dirigée du nœud source vers le nœud cible.
Enfin, un objet `Data` est créé avec les éléments suivants :

- `x` : les caractéristiques des nœuds (patients).
- `edge_index` : les connexions entre patients.
- `y` : les étiquettes cibles (malade, normal).

Ce graphe permet au modèle GCN de propager l'information entre patients similaires, renforçant l'apprentissage par rapport à une approche classique indépendante ou purement tabulaire.

## 2.5 Classification avec GCN et GAT

Dans le cadre de la prédiction de l'insuffisance cardiaque à partir de données multimodales (images ECG + variables cliniques), nous avons exploré deux modèles de graphes neuronaux profonds : le Graph Convolutional Network (GCN) et le Graph Attention Network (GAT). Nous présentons ci-dessous l'architecture détaillée de chacun de ces modèles.

### 1. Architecture du modèle Graph Convolutional Network (GCN)

La Figure 3.12 présente l'architecture de notre GCN. Une explication des différentes couches qui le composent est donnée ci-après :

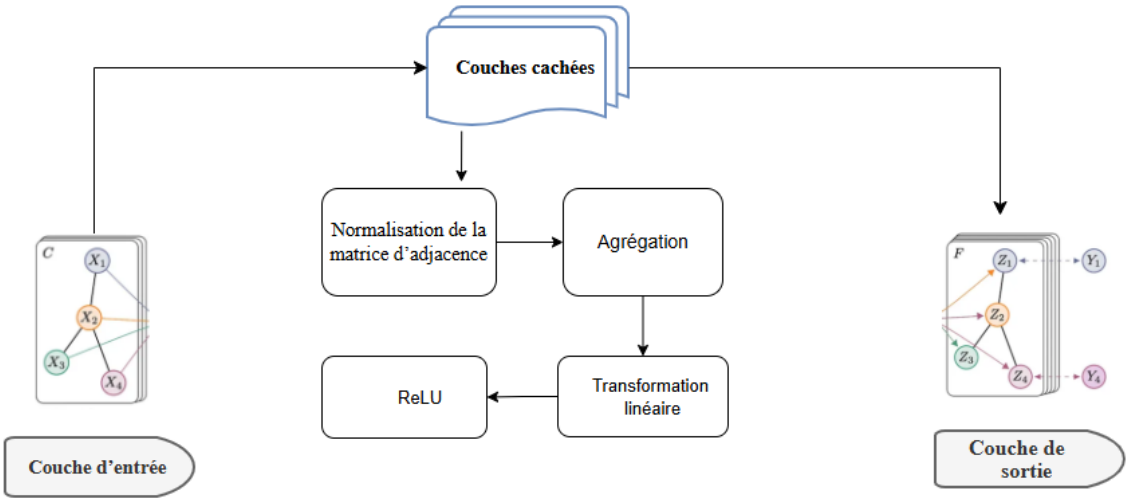



**Figure 3.12 :** Architecture du modèle GCN

1. **Couche d'entrée (Input Layer)**

   Chaque nœud (patient) est représenté par un vecteur de 517 dimensions, obtenu par concaténation de : 512 caractéristiques visuelles extraites d'images ECG à l'aide d'un ResNet18 pré-entraîné, et 5 variables cliniques, préalablement normalisées. Ce vecteur constitue l'entrée de la première couche `GCNConv`.

2. **Couches cachées (Hidden Layers)**

   Cette section détaille le mécanisme de propagation et de transformation des informations à travers les couches du modèle.

   (a) **Normalisation de la matrice d'adjacence**

   Cette étape prépare la matrice d'adjacence $A$ pour la propagation de l'information dans le graphe. On commence par ajouter l'identité $I$ à $A$, pour inclure des self-loops. Cela permet à chaque nœud d'intégrer ses propres caractéristiques lors de l'agrégation. Ensuite, une normalisation symétrique est appliquée à l'aide de la matrice de degré $D$, afin d'équilibrer la contribution des voisins et garantir une diffusion stable :

   $$\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\tilde{\mathbf{A}}\mathbf{D}^{-1/2}$$

   (b) **Première couche GCNConv**

   i. **Agrégation :** Chaque nœud agrège alors les caractéristiques de ses voisins directs à l'aide d'une multiplication matricielle entre la matrice d'adjacence

normalisée $\hat{\mathbf{A}}$ et la matrice des caractéristiques $\mathbf{X}$ :

$$\mathbf{Z} = \hat{\mathbf{A}} \cdot \mathbf{X}$$

Cela permet de combiner l'information provenant du voisinage avec les caractéristiques propres à chaque nœud.

ii. **Transformation linéaire + ReLU :** Les représentations agrégées sont ensuite transformées via une couche linéaire paramétrée par une matrice de poids $\mathbf{W}^{(0)}$, puis activées par la fonction *ReLU*. Un *dropout* avec un taux de 0,5 est également appliqué afin de limiter le surapprentissage. L'opération est formalisée par :

$$\mathbf{H} = \text{ReLU}\left(\hat{\mathbf{A}} \cdot \mathbf{X} \cdot \mathbf{W}^{(0)}\right)$$

Avec :

- $\mathbf{X}$ : matrice des caractéristiques d'entrée (517 dimensions).
- $\mathbf{W}^{(0)}$ : matrice de poids à apprendre.
- *ReLU* (Rectified Linear Unit) : fonction d'activation non-linéaire définie par $f(x) = \max(0, x)$.

(c) **Deuxième couche GCNConv**

La seconde couche `GCNConv` prend en entrée les embeddings générés précédemment (128 dimensions) et applique le même schéma d'agrégation et de transformation :

- Agrégation via la matrice normalisée $\hat{\mathbf{A}}$.
- Transformation linéaire avec la matrice de poids $\mathbf{W}^{(1)}$.
- Activation *ReLU*.
- Application d'un *dropout* de taux 0,5.

La dimension des représentations reste fixée à 128. L'opération s'écrit :

$$\mathbf{H}^{(l+1)} = \text{ReLU}\left(\hat{\mathbf{A}} \cdot \mathbf{H}^{(l)} \cdot \mathbf{W}^{(l)}\right)$$

Avec :

- $\mathbf{H}^{(l)} \in R^{n \times 128}$ : matrice des représentations des nœuds à l'étape $l$.
- $\mathbf{W}^{(l)}$ : matrice de poids à apprendre pour la couche $l$.

(d) **Couche de sortie (Output Layer)**

Les représentations finales (128 dimensions) sont transmises à une couche fully

connected linéaire (`nn.Linear`) qui projette chaque vecteur dans un espace de 2 dimensions, correspondant aux deux classes :

- 0 : patient sain.
- 1 : patient malade.

La prédiction est supervisée par la fonction de perte `CrossEntropyLoss`, qui applique un softmax implicite pour produire une distribution de probabilités sur les classes. La classe prédite est celle ayant la probabilité maximale.

**Architecture GCNConv et absence de pooling :** Le modèle repose sur deux couches `GCNConv` successives, ce qui permet de :

- Propager efficacement l'information entre les nœuds du graphe.
- Éviter le sur-lissage (*over-smoothing*), un phénomène où les représentations deviennent trop similaires après un trop grand nombre de couches.

Aucune opération de pooling n'est appliquée, car il s'agit d'une tâche de classification au niveau des nœuds (patients). Chaque nœud conserve donc sa représentation individuelle jusqu'à l'étape de prédiction finale, afin de préserver toute l'information locale pertinente.

## 2. Architecture du modèle Graph Attention Network (GAT)

La Figure 3.13 présente l'architecture de notre GAT. Une explication des différentes couches qui le composent est donnée ci-après :

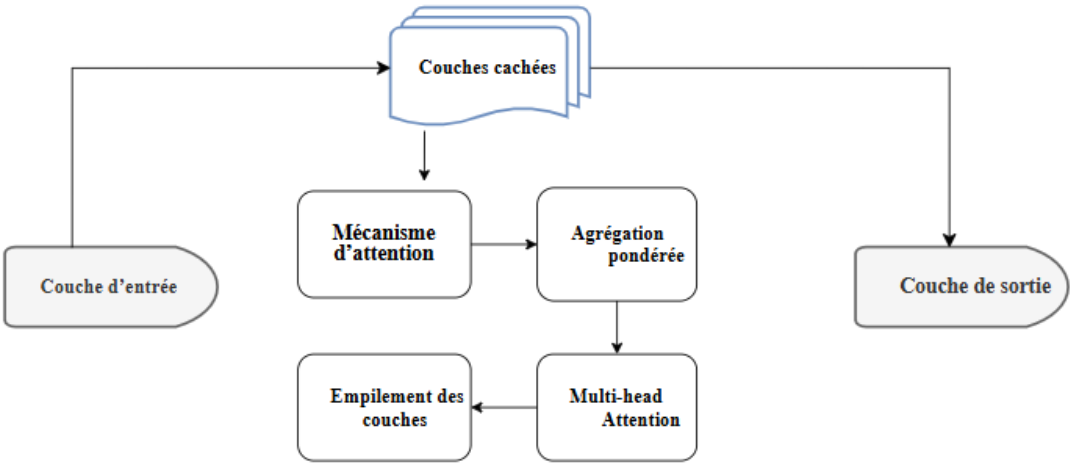



**Figure 3.13 :** Architecture du modèle GAT

1. **Couche d'entrée (Input Layer)**

   Chaque patient est représenté par un vecteur de 517 dimensions : 512 features visuelles extraites d'ECG via ResNet18 et 5 variables cliniques normalisées. Ces vecteurs sont transformés via une couche linéaire partagée en embeddings de 128 dimensions.

2. **Couches cachées (Hidden Layers)**

   Ces couches permettent de propager et d'enrichir les représentations des patients en exploitant les relations avec leurs voisins dans le graphe. Elles se composent des étapes suivantes :

   (a) **Mécanisme d'attention :** Le mécanisme d'attention constitue le cœur du modèle GAT. Il se compose de deux étapes principales : le calcul des scores d'attention et leur normalisation afin de pondérer l'influence de chaque voisin dans la mise à jour des représentations des nœuds.

   - **Calcul des scores d'attention :** Pour chaque paire de nœuds connectés $(i, j)$, un score d'attention est d'abord calculé pour mesurer l'importance du voisin $j$ dans la mise à jour de la représentation du nœud $i$ :

   $$e_{ij} = \text{LeakyReLU}\left(\vec{a}^T \left[Wh_i \,\|\, Wh_j\right]\right)$$

   - **Normalisation des scores d'attention :** Ces scores sont ensuite normalisés localement sur le voisinage de chaque nœud à l'aide d'une fonction softmax :

   $$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})}$$

   Cette normalisation permet d'obtenir des poids d'attention $\alpha_{ij}$ compris entre 0 et 1, et dont la somme vaut 1 pour chaque nœud $i$. Ainsi, les voisins les plus pertinents auront une plus grande influence dans la mise à jour de l'état du nœud.

   (b) **Agrégation pondérée :** Après le calcul et la normalisation des coefficients d'attention $\alpha_{ij}$, chaque nœud met à jour sa représentation en agrégeant les embeddings de ses voisins directs :

   $$h'_i = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} \cdot Wh_j\right)$$

   Où :

- $h'_i \in R^{128}$ : nouvelle représentation du nœud $i$
- $W \in R^{128\times128}$ : matrice de poids partagée
- $\alpha_{ij}$ : poids d'attention représentant l'importance du voisin $j$
- $\sigma$ : fonction d'activation non-linéaire (ici ELU)

Cette fonction ELU introduit de la non-linéarité après l'agrégation, ce qui améliore la capacité du modèle à capturer des relations complexes.

**Remarque :** Les étapes (a) et (b) décrivent ensemble le fonctionnement interne d'une seule tête d'attention.

(c) **Multi-head Attention :** Afin de mieux capturer différentes relations entre les nœuds, le mécanisme d'attention utilise plusieurs têtes d'attention en parallèle. Dans ce travail, 4 têtes sont utilisées, chacune calculant ses propres coefficients d'attention et procédant à une agrégation indépendante. Les représentations obtenues à partir des différentes têtes sont ensuite :

- concaténées dans les couches intermédiaires, ce qui donne :

$$\dim(h'_i) = 4 \times 128 = 512$$

- moyennées ou réduites via un seul head dans la couche finale, selon le besoin.

Ce mécanisme permet au modèle de capturer plusieurs types de relations :

- Des relations basées sur les caractéristiques visuelles des ECG
- D'autres basées sur les variables cliniques associées aux patients

Chaque tête pouvant se spécialiser dans un aspect des données.

(d) **Empilement des couches :** Le modèle est constitué de deux couches GAT successives :

- La première couche exploite les informations des voisins directs de chaque patient (connexion 1-hop).
- La seconde couche récupère les informations des voisins des voisins (connexion 2-hop), permettant à chaque nœud d'intégrer un contexte plus large.

Ce mécanisme d'empilement affine progressivement les représentations des patients en capturant des relations locales et élargies dans le graphe.

3. **Couche de sortie (Output Layer)**

Les embeddings finaux de dimension 128 sont transmis à une couche fully connected linéaire (nn.Linear) qui projette chaque vecteur dans un espace à 2 dimensions correspondant aux deux classes :

- 0 : patient sain
- 1 : patient malade

La prédiction est supervisée par la fonction de perte **CrossEntropyLoss**, qui applique un softmax implicite pour produire des probabilités sur les classes. La classe prédite est celle ayant la probabilité maximale.

## 2.6 Entraînement du modèle

L'entraînement du modèle repose sur une approche solide afin de garantir une convergence stable et une bonne généralisation :

- **Fonction de perte :**
  `CrossEntropyLoss` est utilisée pour la classification multi-classes. Elle compare les logits produits par le modèle avec les étiquettes réelles (`train_data.y`) sur l'ensemble des nœuds du graphe d'entraînement.
- **Optimiseur :**
  L'optimisation est assurée par l'algorithme `Adam`, avec un taux d'apprentissage fixé à 0.001.
- **Nombre d'époques :**
  L'entraînement s'effectue sur 30 époques (`EPOCHS = 30`), avec une évaluation de la performance sur l'ensemble de validation après chaque itération.
- **Rétropropagation :**
  - La perte est calculée sur toutes les sorties du modèle.
  - `loss.backward()` est utilisé pour calculer les gradients.
  - `optimizer.step()` met à jour les paramètres du modèle.
  - Le `Dropout` (`p=0.5`) est activé uniquement pendant l'entraînement (`self.training = True`), afin de réduire le surapprentissage.

Les courbes d'évaluation de l'entraînement du modèle incluent : **accuracy**, **précision**, **rappel** et **F1-score** sur les données d'apprentissage.

# 3 Conclusion

Dans ce chapitre, nous avons appliqué des modèles GNN, notamment GCN et GAT, sur un jeu de données multimodal combinant images ECG et variables cliniques simulées. Après avoir décrit le jeu de données et les étapes de prétraitement, nous

avons extrait les caractéristiques visuelles via ResNet18 et construit un graphe de similarité entre patients. Nous avons ensuite détaillé l'architecture des modèles utilisés et finalisé par l'entraînement et la validation des modèles sur ce graphe. Le chapitre suivant sera consacré au développement et à la mise en œuvre du système proposé, depuis le prétraitement des données jusqu'à la validation expérimentale et la conception de l'interface utilisateur.

# Chapitre 4

# Implémentation

## 1 Introduction

Dans le chapitre précédent, nous avons proposé une solution de prédiction de l'insuffisance cardiaque basée sur les Graph Neural Networks, et plus précisément sur un Graph Convolutional Network (GCN). Notre approche exploite des images ECG ainsi que cinq facteurs de risque cliniques.

Dans ce chapitre, nous procéderons au développement de cette proposition. Nous commençons par présenter l'environnement de développement et les outils utilisés, tels que Python, PyTorch Geometric, Scikit-learn et Streamlit. Ensuite, nous décrivons la base d'apprentissage, suivie par le prétraitement des données, comprenant le chargement et le nettoyage initial des images, la division des données en ensembles d'entraînement et de validation, ainsi que la préparation des features à travers l'encodage des labels, la normalisation des variables cliniques et le prétraitement des images. Nous poursuivons avec l'extraction des caractéristiques d'image, puis la fusion des données images et cliniques, avant de procéder à la construction du graphe de similarité.

Par la suite, nous présentons les résultats expérimentaux, en abordant la comparaison entre ResNet18 et ResNet50, l'influence du paramètre $k$ dans le graphe k-NN, l'évaluation comparative des approches unimodale, tabulaire et multimodale, les résultats comparatifs des méthodes testées, ainsi que la visualisation des performances du modèle GCN multimodal.

Enfin, nous présentons l'interface interactive du système, permettant à l'utilisateur de soumettre une image ECG et cinq paramètres cliniques pour obtenir une prédiction instantanée.

## 2 Environnement de développement et outils utilisés

Dans cette section, nous présentons les différents langages, plateformes et bibliothèques utilisés tout au long de notre travail.

- **Python :** Python est un langage de programmation interprété de haut niveau, conçu pour être facile à lire et à apprendre. Il prend en charge de nombreux paradigmes, y compris la programmation orientée objet. Python est largement utilisé pour le prototypage rapide d'applications et comme langage de liaison entre des composants logiciels existants. Très populaire parmi les développeurs, il est disponible gratuitement sous licence open source, avec une bibliothèque standard compatible avec toutes les principales plateformes[59].

- **Jupyter Notebook :** Jupyter Notebook, anciennement connu sous le nom d'IPython Notebook avant d'être renommé en 2014, est une application web interactive qui permet de créer et de partager des documents informatiques. Ce logiciel entièrement open source est gratuit et prend en charge plus de 40 langages de programmation, tels que Python, R ou Scala[19].

- **Google Colab :** Google Colab (ou Colaboratory) est un service cloud gratuit proposé par Google, basé sur l'environnement Jupyter Notebook et destiné à la formation et à la recherche en apprentissage automatique. Il permet d'exécuter du code Python directement dans le cloud, sans aucune installation locale[39].

- **PyTorch :** PyTorch est un projet libre et gratuit pour l'apprentissage automatique, basé sur le langage de programmation Python et reposant sur la bibliothèque Torch (qui utilise le langage Lua). Il est utilisé dans des domaines tels que la reconnaissance d'images et le traitement du langage. PyTorch se distingue par sa prise en charge des unités de traitement graphique (GPU) et par la rétropropagation automatique, ce qui facilite la création de graphes computationnels[50].

- **PyTorch Geometric :** PyTorch Geometric est une bibliothèque libre et gratuite ajoutée à PyTorch pour faciliter le traitement des réseaux de neurones graphiques (GNN). Elle comprend différentes méthodes d'apprentissage profond appliquées aux graphes et aux structures non régulières[60].

- **Scikit-learn :** Scikit-learn, également connue sous le nom de sklearn, est une bibliothèque open source dédiée à l'apprentissage automatique et à la modélisation des données en Python. Elle propose de nombreux algorithmes de classification, de régression et de clustering, tels que les machines à vecteurs de support (SVM), les forêts aléatoires, le gradient boosting, k-means ou encore DBSCAN. Elle est conçue

pour s'intégrer facilement avec d'autres bibliothèques de l'écosystème Python, notamment NumPy et SciPy[20].

- **Pandas :** Pandas est une bibliothèque Python open source utilisée pour les tâches liées à la manipulation des données, à l'analyse des données et à la préparation des données pour l'apprentissage automatique. Elle est construite sur une autre bibliothèque appelée NumPy, qui prend en charge les tableaux multidimensionnels. Pandas s'intègre bien avec les autres bibliothèques de l'écosystème Python dédiées à l'analyse de données, et est souvent incluse dans certaines distributions Python, ainsi que dans des distributions commerciales, telles qu'ActivePython d'ActiveState[2].

- **Matplotlib :** Matplotlib est une bibliothèque Python utilisée pour le tracé de données et la visualisation graphique, notamment les histogrammes, les nuages de points et les diagrammes en barres. Elle s'appuie sur NumPy pour la gestion des données numériques et constitue une alternative open source à MATLAB. Les développeurs peuvent utiliser des interfaces de programmation (API) pour intégrer des graphiques dans des applications avec interface utilisateur graphique (GUI)[2].

- **NumPy :** NumPy est la bibliothèque de base pour le calcul scientifique en Python. Elle fournit un objet central appelé tableau multidimensionnel, ainsi que des objets dérivés comme les tableaux masqués. Elle propose également un ensemble de routines permettant d'effectuer rapidement diverses opérations sur les tableaux : opérations mathématiques, logiques, transformations de forme, etc[49].

- **Plotly :** Plotly est une bibliothèque de visualisation interactive, open source et basée sur le navigateur. Elle est développée en Python et repose sur la bibliothèque JavaScript plotly.js. Elle prend en charge plus de 30 types de graphiques, incluant des graphiques scientifiques, 3D, statistiques, financiers, ainsi que des cartes interactives en SVG, et bien plus encore[21].

- **Streamlit :** Streamlit est une bibliothèque Python open source, facile à utiliser, qui permet de créer des applications web interactives pour l'apprentissage automatique et les sciences des données. Elle permet aux développeurs de concevoir des applications organisées et esthétiques avec seulement quelques lignes de code[18].

# 3 Évaluation du modèle

L'évaluation du modèle est réalisée sur l'ensemble de test (`test_mask`) afin de garantir une estimation honnête de la performance générale. Les métriques suivantes ont été calculées [56] :

1. **Accuracy (Précision globale)** L'*accuracy* est une mesure qui calcule la fréquence à laquelle un modèle d'apprentissage automatique prédit correctement un résultat. Elle est calculée en divisant le nombre de prédictions correctes (vrais positifs + vrais négatifs) par le nombre total de prédictions effectuées. Elle est définie par :

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

   Où :

   - TP = Vrais Positifs
   - TN = Vrais Négatifs
   - FP = Faux Positifs
   - FN = Faux Négatifs

2. **Recall (Rappel)** Le *recall*, également appelé *sensibilité*, est une mesure qui évalue l'efficacité d'un modèle d'apprentissage automatique à identifier correctement les véritables cas positifs parmi l'ensemble des cas positifs réels. Il est défini par :

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

   Où :

   - TP = Vrais Positifs
   - FN = Faux Négatifs

3. **Precision (Précision)** La *précision* est une mesure qui détermine la fréquence à laquelle un modèle d'apprentissage automatique prédit correctement les cas positifs parmi tous les cas qu'il a prédits comme positifs. Elle est définie par :

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

   Où :

   - TP = Vrais Positifs
   - FP = Faux Positifs

4. **F1-score** Le *F1-score* est la moyenne harmonique de la précision et du rappel ; il fournit une mesure unique qui équilibre ces deux indicateurs. Il est défini par :

$$\text{F1-score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

5. **Matrice de confusion** La matrice de confusion est un tableau utilisé pour résumer les performances d'un modèle de classification en comparant ses prédictions aux valeurs réelles d'un ensemble de données de test [22].

# 4 Base d'apprentissage

Dans ce travail, nous avons utilisé un jeu de données multimodal combinant des images ECG et cinq variables cliniques. Cette structure bimodale permet de tirer parti à la fois des informations visuelles (images ECG) et des données tabulaires (les variables cliniques) pour une meilleure prédiction de l'insuffisance cardiaque.
Nous avons effectué quelques modifications sur cette base, notamment le nettoyage des images (suppression des fichiers non valides ou trop petits) et la normalisation des variables cliniques, afin de l'adapter à notre pipeline d'expérimentation.

# 5 Prétraitement de données

## 5.1 Chargement et nettoyage initial des images

La première étape de notre pipeline consiste à charger les images ECG en vérifiant leur validité. À l'aide d'un script Python, chaque image a été contrôlée pour s'assurer qu'elle est bien présente, qu'elle n'est pas corrompue, et qu'elle possède une taille minimale de 100×100 pixels. Les images ne répondant pas à ces critères ont été automatiquement écartées, et un message explicite a été généré (Figure 4.1). Ce filtrage garantit que seules les images exploitables soient conservées pour la suite du traitement

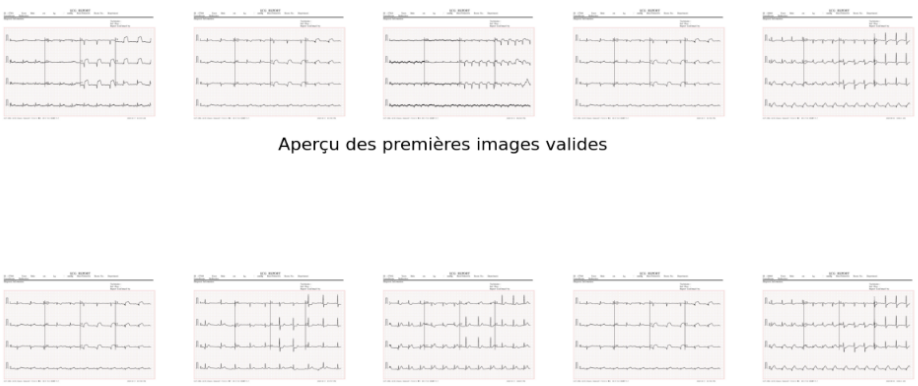



**Figure 4.1 :** Extrait de quelques images ECG valides parmi l'ensemble filtré

Notre dataset initial souffrait d'un déséquilibre marqué entre les classes *« Malade »* et *« Normal »*. Pour résoudre ce problème, nous avons utilisé la technique d'augmentation

de données.
En appliquant des transformations aléatoires (rotations, flips horizontaux, zooms, translations) via le module `ImageDataGenerator` de la bibliothèque `Keras`, nous avons généré artificiellement de nouvelles images pour la classe minoritaire.
Ces images augmentées ont ensuite été enregistrées et ajoutées au dataset initial, résultant en un jeu de données équilibré (Figure 4.2).

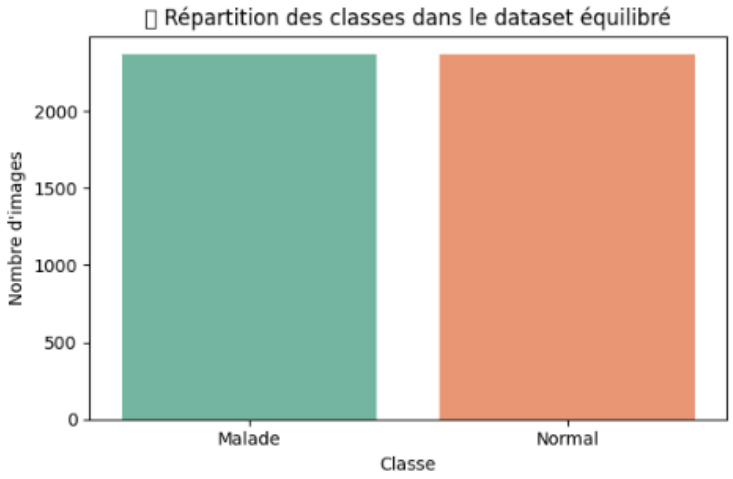



**Figure 4.2 :** Répartition des classes après équilibrage

Pour enrichir le dataset et assurer une multimodalité des caractéristiques, nous avons généré cinq variables cliniques pour chaque patient : l'Ejection Fraction (EF), le BNP, la classification NYHA, la Pression artérielle systolique (SBP), et l'âge. Ces valeurs ont été simulées à l'aide de distributions statistiques différenciées selon le label du patient, tout en respectant des plages physiologiquement plausibles (Figure 4.3). Les nouvelles variables ont été intégrées dans un DataFrame final, chaque ligne contenant désormais l'image, son label, et les cinq variables cliniques associées.

| image_path | label | ef | bnp | nyha | sbp | age |
|---|---|---|---|---|---|---|
| /content/drive/MyDri | Malade | 0.334 | 463.331 | 3 | 132 | 70 |
| /content/drive/MyDri | Malade | 0.298 | 580.505 | 1 | 104.3 | 75 |
| /content/drive/MyDri | Malade | 0.382 | 640.537 | 2 | 109.8 | 73 |
| /content/drive/MyDri | Malade | 0.4 | 133.121 | 2 | 96.9 | 58 |
| /content/drive/MyDri | Malade | 0.531 | 576.904 | 3 | 117.7 | 60 |
| /content/drive/MyDri | Malade | 0.302 | 776.536 | 3 | 106.1 | 65 |
| /content/drive/MyDri | Malade | 0.486 | 374.236 | 2 | 102.4 | 57 |
| /content/drive/MyDri | Malade | 0.406 | 561.009 | 3 | 115.9 | 89 |

**Figure 4.3 :** La structure de l'ensemble de données combinées

Avant d'entamer la phase d'entraînement, un nettoyage final a été effectué. Les entrées avec des valeurs manquantes ou invalides ont été supprimées. Les doublons ont été éliminés, et les types de données ont été vérifiés et corrigés si nécessaire. L'index du DataFrame a été réinitialisé pour assurer une structure propre et cohérente.

```
Total initial: 4734
Total après nettoyage: 4734
Lignes supprimées: 0

Distribution des labels:
label
Malade    2367
Normal    2367
Name: count, dtype: int64
```

**Figure 4.4 :** Nettoyage final de l'ensemble de données combinées

## 5.2 Division des données en ensembles d'entraînement et de validation

Le jeu de données nettoyé est ensuite divisé en deux sous-ensembles : 80 % pour l'entraînement et 20 % pour la validation, via `train_test_split`. Les données de test sont gérées à part, fournies séparément. Cette division permet de mesurer les performances du modèle de manière fiable sur des données non vues pendant l'entraînement.

```
Train : 3976 images
Validation : 758 images

Répartition des labels dans Train :
label
Normal    1988
Malade    1988
Name: count, dtype: int64

Répartition des labels dans Validation :
label
Malade    379
Normal    379
Name: count, dtype: int64
```

**Figure 4.5 :** Division des données

## 5.3 Préparation des caractéristiques

1. **Encodage des labels** Les étiquettes textuelles *Normal* et *Malade* sont encodées numériquement avec `LabelEncoder` (*Normal* = 0, *Malade* = 1). Cela permet de rendre les labels exploitables par les modèles d'apprentissage automatique.

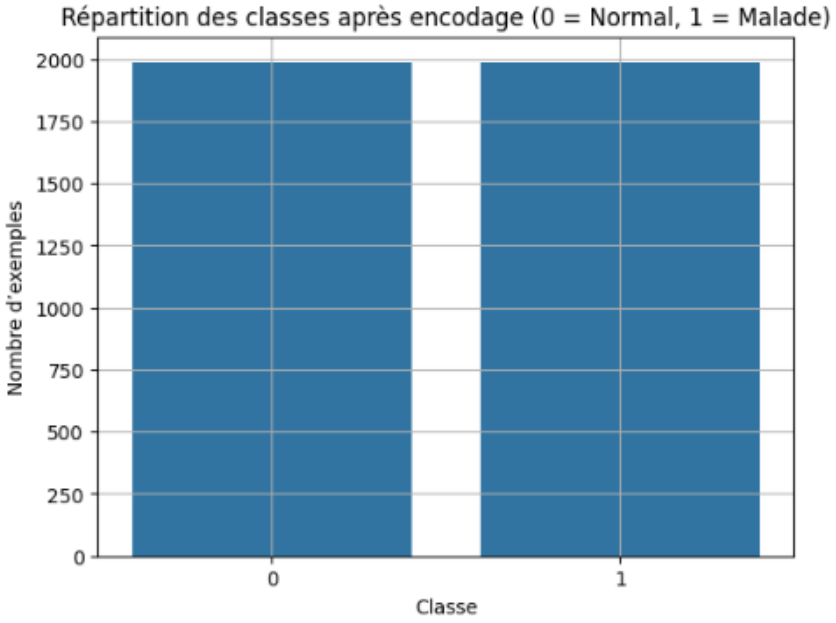



**Figure 4.6 :** Répartition des classes (après Label Encoding)

2. **Normalisation des variables cliniques**

   Les cinq variables cliniques (`ef`, `bnp`, `nyha`, `sbp`, `age`) sont standardisées avec `StandardScaler` de `scikit-learn`. Cette transformation garantit que chaque variable possède une moyenne nulle et un écart-type de 1, ce qui évite que certaines variables dominent l'apprentissage.

| | ef | bnp | nyha | sbp | age |
|---|---|---|---|---|---|
| 0 | 1.387369 | -0.823828 | 0.745633 | -0.207670 | -0.378676 |
| 1 | 0.760972 | -0.787688 | -1.219655 | -0.189465 | 0.468699 |
| 2 | -1.766726 | -0.244759 | -1.219655 | -1.045102 | 1.993974 |
| 3 | -1.243500 | -0.368229 | 0.745633 | 0.423438 | 0.977124 |
| 4 | 0.031403 | 0.008442 | 0.745633 | 0.684377 | 1.993974 |
| 5 | 1.262090 | 0.058442 | -1.219655 | 2.650522 | -0.209201 |
| 6 | 0.915729 | -0.491728 | 0.745633 | -1.348519 | -1.819214 |
| 7 | -2.032023 | 2.330656 | 0.745633 | -0.238012 | 0.129749 |
| 8 | 1.247351 | -0.841823 | -0.237011 | 1.188050 | -0.209201 |
| 9 | -1.280347 | 1.826629 | -1.219655 | -0.474677 | 1.485549 |

**Figure 4.7 :** Aperçu des variables cliniques après normalisation

3. **Prétraitement des images**

   Les images ECG sont redimensionnées à $224 \times 224$ pixels pour correspondre aux exigences du modèle ResNet. Ensuite, elles sont converties en tenseurs PyTorch et normalisées avec les moyennes et écarts-types RGB d'ImageNet : Moyennes : $[0.485, 0.456, 0.406]$, Écarts-types : $[0.229, 0.224, 0.225]$.

## 5.4 Extraction des caractéristiques d'image

Après la préparation des données, l'étape suivante consiste à extraire des caractéristiques visuelles à partir des images ECG à l'aide d'un modèle ResNet pré-entraîné. Nous commençons par charger un modèle ResNet18 dont la dernière couche de classification est supprimée, car l'objectif n'est pas de prédire des classes ImageNet, mais d'extraire des vecteurs de caractéristiques représentatifs des images.
Les images, qui ont été normalisées et redimensionnées à des dimensions compatibles avec ResNet, sont ensuite converties en tenseurs et passées dans ce modèle. À la sortie, chaque image est transformée en un vecteur numérique résumant ses caractéristiques visuelles de taille 512. Ces vecteurs sont ensuite récupérés et stockés.

## 5.5 Fusion des données visuelles et cliniques

Pour chaque patient, le vecteur visuel de 512 dimensions est concaténé avec les cinq variables cliniques normalisées. Cette opération génère un vecteur multimodal de 517 dimensions, combinant données cliniques et représentations d'image. Ces vecteurs sont utilisés pour représenter les nœuds dans le graphe.

## 5.6 Construction du graphe de similarité

Un graphe est ensuite construit à partir des vecteurs multimodaux des patients. Chaque nœud représente un patient, et les connexions sont établies via l'algorithme k-plus proches voisins (k-NN), avec $k = 5$. La distance utilisée est la distance euclidienne dans l'espace des 517 features. Cette structure garantit que chaque patient est connecté à ses voisins les plus proches, sans nœuds isolés. La matrice de voisinage est générée avec `kneighbors_graph` de `scikit-learn` et convertie en `edge_index` pour PyTorch Geometric.

# 6 Résultats expérimentaux

Nous présentons ici les expériences réalisées pour évaluer notre approche, en comparant différents modèles, paramètres et modalités de données.

## 6.1 Comparaison entre ResNet18 et ResNet50

Pour évaluer l'efficacité de l'extraction des caractéristiques visuelles, nous avons comparé deux architectures populaires : ResNet18 et ResNet50. Les performances obtenues sur notre dataset sont présentées ci-dessous :

| **Metric** | **ResNet18** | **ResNet50** |
|---|---|---|
| Accuracy | 0.97 | 0.97 |
| F1-score (Malade) | 0.98 | 0.98 |
| F1-score (Normal) | 0.94 | 0.94 |
| Recall (Normal) | 0.96 | 0.93 |
| Precision (Normal) | 0.91 | 0.95 |

**Tableau 4.1 :** Comparaison entre ResNet18 et ResNet50

Nous remarquons que les deux modèles, ResNet18 et ResNet50, atteignent une accuracy globale identique de 97 % sur le dataset. Cependant, en regardant de plus près :

- ResNet18 offre un meilleur rappel (recall) sur la classe *Normal* (0.96 contre 0.93 pour ResNet50), ce qui signifie qu'il reconnaît mieux les patients sains et réduit le risque de faux diagnostics de maladie.
- ResNet50 est légèrement plus précis sur la classe *Normal*, mais au prix d'un rappel plus faible, ce qui veut dire qu'il rate plus de cas sains, ce qui est moins acceptable en contexte médical.
- F1-scores et accuracy sont équivalents pour les deux.

Par conséquent, ResNet18 est le meilleur choix dans ce contexte, offrant une meilleure performance clinique (moins de faux positifs), ce qui est essentiel en contexte médical, tout en étant plus simple à déployer et plus rapide à exécuter.

## 6.2 Influence du paramètre $k$ dans le graphe k-NN

Nous avons testé deux valeurs de $k$ (5 et 10) pour construire le graphe, afin d'évaluer l'impact du nombre de voisins sur les performances. Les résultats ci-dessous montrent que $k = 5$ offre de meilleures performances, en particulier pour la détection des cas normaux :

| **Valeur de $k$** | **Accuracy** | **F1-score « Normal »** | **F1-score « Malade »** |
|---|---|---|---|
| $k = 5$ | 0.97 | 0.94 | 0.98 |
| $k = 10$ | 0.93 | 0.87 | 0.96 |

**Tableau 4.2 :** Comparaison entre les valeurs de $k$

## 6.3 Résultats expérimentaux et comparaisons

Afin d'évaluer l'apport des approches multimodales et des architectures de graphes, plusieurs modèles ont été implémentés et testés dans cette étude. Les modèles développés sont les suivants :

### 1. Modèle unimodal GCN appliqué sur les images ECG

En complément des expériences multimodales, un modèle unimodal basé uniquement sur les images ECG a été testé. Les vecteurs de 512 dimensions extraits par ResNet18 ont été utilisés comme entrée dans un classifieur dense simple ainsi qu'un modèle de Graph Convolutional Network (GCN) appliqué uniquement sur ces représentations visuelles. Cette approche visait à établir un baseline visuel, sans inclure les données cliniques. Bien que les performances obtenues aient été acceptables (accuracy 50 %), elles sont restées systématiquement inférieures à celles des modèles multimodaux, confirmant ainsi l'intérêt de la fusion d'informations cliniques et visuelles.

### 2. Modèles unimodaux GCN et GAT appliqués sur les données tabulaires

Nous avons testé deux modèles de Graph Neural Networks (GNN) utilisant uniquement les données tabulaires sur le jeu *Heart Failure Clinical Records* obtenu sur Kaggle, qui contient les dossiers médicaux de 5000 patients atteints d'insuffisance cardiaque, collectés pendant leur suivi. Chaque patient est décrit par 13 variables cliniques, comme l'âge, le sexe, et divers indicateurs médicaux et biologiques.
Le même pipeline de prétraitement que celui appliqué au jeu multimodal a été conservé, à savoir :

- **Nettoyage des données :** Traitement des valeurs manquantes par la moyenne, suppression des valeurs aberrantes via z-score.
- **Normalisation des variables numériques :** Avec `StandardScaler`.
- **Rééquilibrage de la distribution de classes :** Avec une combinaison SMOTE (sur-échantillonnage) et ENN (sous-échantillonnage).
- **Construction d'un graphe de similarité :** Basé sur les k-plus-proches-voisins ($k = 5$), à partir des vecteurs cliniques normalisés.

Les performances obtenues sur le jeu de test sont résumées dans le tableau suivant :

| **Modèle** | **Accuracy** | **F1-score global (pondéré)** | **F1-score « Heart Failure »** |
|---|---|---|---|
| GCN (tabulaire) | 0.92 | 0.92 | 0.87 |
| GAT (tabulaire) | 0.89 | 0.89 | 0.82 |

**Tableau 4.3 :** Comparaison entre GCN et GAT

Ces résultats montrent que le modèle GCN dépasse le GAT, notamment pour la prédiction des cas d'insuffisance cardiaque (classe minoritaire). Bien que le GAT utilise un mécanisme d'attention multi-têtes censé mieux capter les relations entre nœuds, il ne parvient pas à dépasser les performances du GCN. Cela s'explique probablement par la taille réduite du jeu de données et la simplicité des relations entre patients dans le graphe k-NN.

### 3. Modèle GCN appliqué aux données multimodales

Le modèle GCN a été appliqué sur des données multimodales, résultant de la concaténation des descripteurs d'images ECG extraits via ResNet18 et des variables cliniques normalisées. Il commence avec la couche *GCNConv(in_channels=517, hidden_channels=128).* Elle prend en entrée 517 features par nœud (512 d'images + 5 cliniques) et produit 128 features par nœud, en agrégeant les informations des nœuds voisins via la topologie du graphe (`edge_index`). Cela permet de capturer les dépendances locales entre patients.
Ensuite, la sortie passe par `ReLU`, une fonction d'activation non linéaire, qui introduit de la complexité dans l'apprentissage, suivie d'un *dropout* à 50 %, actif seulement pendant l'entraînement, pour éviter le surapprentissage.
La seconde couche *GCNConv(hidden_channels=128, hidden_channels=128)* reprend les 128 features par nœud en entrée et en sortie, et continue à propager et affiner les informations contextuelles au sein du graphe. On applique encore une fois une `ReLU` pour maintenir la non-linéarité.
Enfin, la couche finale *Linear(hidden_channels=128, out_channels=2)* prend chaque vecteur de 128 dimensions et le transforme en 2 valeurs (logits), correspondant aux classes de sortie (*Normal* vs *Malade*).

### 4. Modèle GAT appliqué aux données multimodales

Le modèle GAT a été appliqué sur des données multimodales, résultant de la concaténation des descripteurs d'images ECG extraits via ResNet18 et des variables cliniques normalisées. Il commence avec la couche *GATConv(in_channels=517, out_channels=128, heads=4, dropout=0.6).* Elle prend en entrée 517 features par nœud (512 extraites des images ECG via ResNet18 et 5 données cliniques) et produit $128 \times 4 = 512$ features par

nœud, grâce à l'utilisation de 4 têtes d'attention. Ce mécanisme permet d'attribuer dynamiquement des poids aux informations provenant des nœuds voisins en fonction de leur importance relative. Une activation `ELU`, suivie d'un `dropout` à 50% (actif uniquement pendant l'entraînement), est appliquée pour introduire de la non-linéarité et limiter le surapprentissage.
La seconde couche *GATConv(in_channels=512, out_channels=128, heads=1, concat=False, dropout=0.6)* reprend ces 512 features et les projette en 128 features par nœud à l'aide d'une tête d'attention unique (sans concaténation des résultats des têtes). Cette couche poursuit la propagation et l'ajustement des informations contextuelles dans le graphe, tout en affinant les relations entre patients. Une activation `ELU` et un `dropout` sont également appliqués.
Enfin, la couche finale `Linear(in_features=128, out_features=2)` prend les vecteurs de 128 dimensions issus de la seconde couche et les transforme en 2 valeurs (logits) correspondant aux deux classes possibles : Normal et Malade.

- Les performances des différents modèles implémentés ont été comparées afin d'évaluer l'impact de la nature des données (images seules, données cliniques seules ou combinaison des deux) ainsi que de l'architecture choisie (GCN vs GAT). Les résultats obtenus sont présentés dans le tableau suivant :

| **Modèle** | **Précision (Accuracy)** | **Précision (Malade)** | **Rappel (Malade)** | **F1-Score (Malade)** | **Précision (Normal)** | **Rappel (Normal)** | **F1-Score (Normal)** |
|---|---|---|---|---|---|---|---|
| GCN Multimodal (Images + Cliniques) | 0.97 | 0.99 | 0.97 | 0.98 | 0.91 | 0.96 | 0.94 |
| GAT Multimodal | 0.91 | 0.94 | 0.94 | 0.94 | 0.81 | 0.83 | 0.82 |
| GCN uniquement images | 0.50 | – | – | – | 0.50 | 1.00 | 0.67 |
| GAT uniquement cliniques | 0.89 | 0.94 | 0.89 | 0.92 | 0.77 | 0.86 | 0.82 |
| GCN uniquement cliniques | 0.92 | 0.95 | 0.94 | 0.95 | 0.86 | 0.89 | 0.87 |

**Tableau 4.4 :** Résultats de performance des différents modèles selon les métriques classiques

Le tableau 4.4 présente les performances comparées de différents modèles de classification entre patients *Malade* et *Normal*, selon les types de données utilisées (images, cliniques, ou les deux) et l'architecture (GCN ou GAT). On observe une supériorité marquée du modèle GCN multimodal (images + données cliniques), qui atteint une précision globale de 0,97 et des scores F1 très élevés pour les deux classes : 0,98 pour les malades et 0,94 pour les normaux. Il détecte presque parfaitement les cas malades (précision : 0,99, rappel : 0,97), tout en maintenant de très bonnes performances sur les cas normaux.

Le GAT multimodal, avec une précision globale de 0,91, reste compétent mais montre des limites, notamment pour la détection des cas normaux (F1 : 0,82), ce qui suggère une moins bonne exploitation des relations multimodales.

Le GCN basé uniquement sur les images échoue complètement, avec une précision de 0,50 et une incapacité à identifier les patients malades (rappel nul). Ce résultat met en évidence que les images, seules, ne suffisent pas à une classification fiable dans ce contexte.

À l'inverse, les modèles fondés uniquement sur les données cliniques (GAT et GCN cliniques) offrent de meilleures performances, avec un avantage significatif pour le GCN clinique (précision : 0,92, F1-malade : 0,95), confirmant que les données cliniques sont

plus informatives que les images seules.
En conclusion, le modèle GCN multimodal s'impose comme la solution la plus efficace et la plus équilibrée. Il surpasse tous les autres en exploitant la combinaison des images médicales et des données cliniques, ce qui en fait le choix optimal pour ce type de tâche.

# 7 Visualisation des performances du modèle GCN multimodal

Afin de suivre le comportement du modèle pendant l'entraînement, nous avons tracé l'évolution de la *loss* (fonction de coût) et de l'*accuracy* pour les ensembles d'entraînement et de validation.

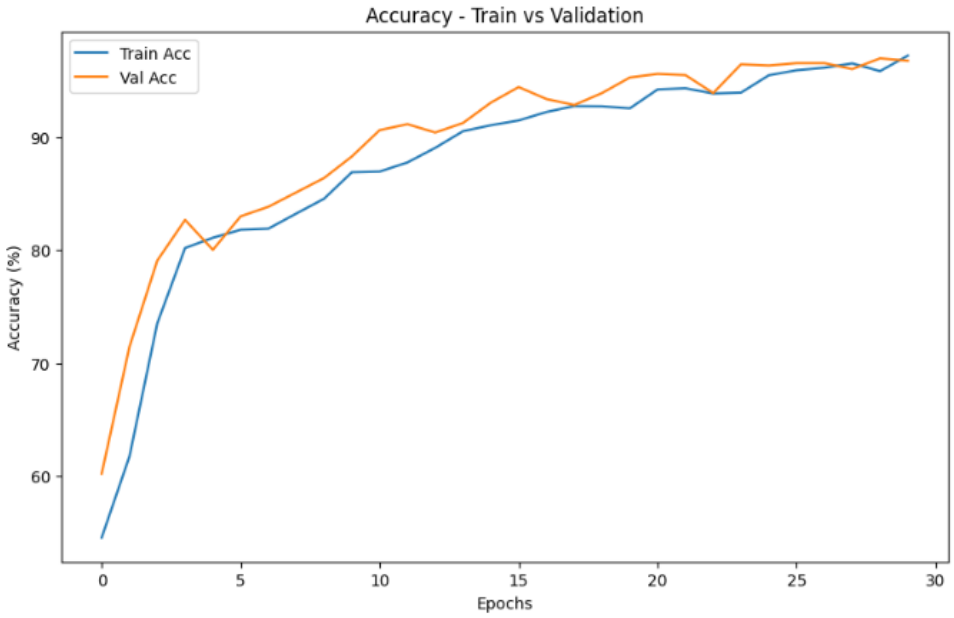



**Figure 4.8 :** Évolution de la précision (accuracy) sur l'entraînement et la validation

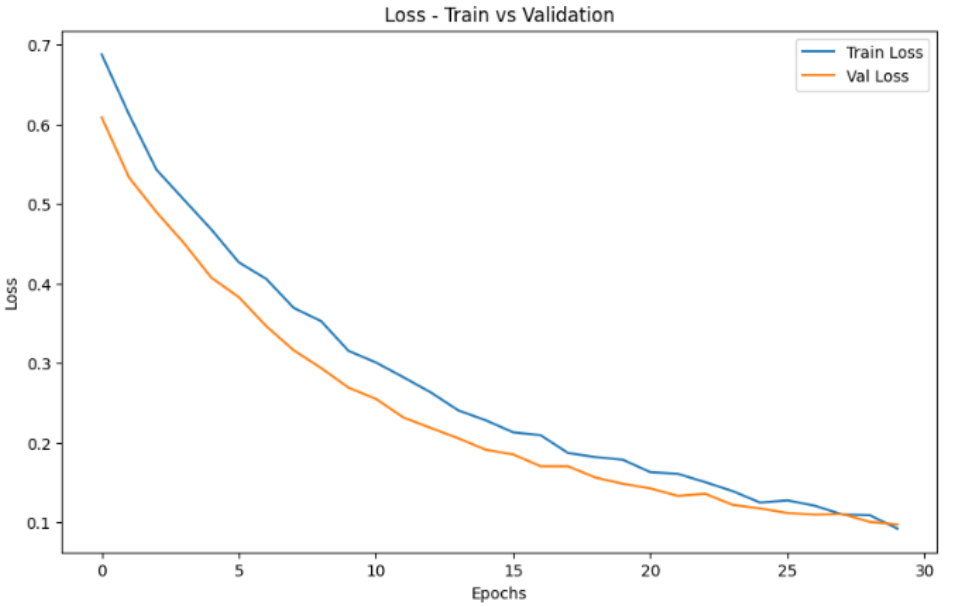



**Figure 4.9 :** Évolution de la loss sur l'entraînement et la validation au fil des époques

Les courbes d'*accuracy* et de *loss* montrent un comportement très stable et bien équilibré entre l'entraînement et la validation, ce qui indique un apprentissage efficace sans surapprentissage (*overfitting*).

1. **Accuracy (Train vs Validation) :** La courbe d'*accuracy* indique que le modèle apprend de manière progressive et équilibrée. La précision augmente de façon régulière pour atteindre environ 97 % sur les ensembles d'entraînement et de validation à la fin des 30 époques. La courbe de validation reste proche, voire légèrement supérieure à celle de l'entraînement à certains moments. Cela montre que le modèle arrive non seulement à bien reconnaître les données qu'il a apprises, mais aussi à faire des prédictions fiables sur des données jamais vues.

2. **Loss (Train vs Validation) :** La courbe de perte (*loss*) diminue de manière continue et régulière au fil des époques, atteignant des valeurs très faibles autour de 0,1. Cette tendance est observée aussi bien sur les données d'entraînement que sur celles de validation, ce qui indique une convergence stable du modèle. Le faible écart entre les deux courbes tout au long de l'apprentissage confirme que le modèle ne souffre pas de surapprentissage (*overfitting*) : il apprend les représentations utiles sans simplement mémoriser les exemples d'entraînement.

3. **Matrice de confusion**

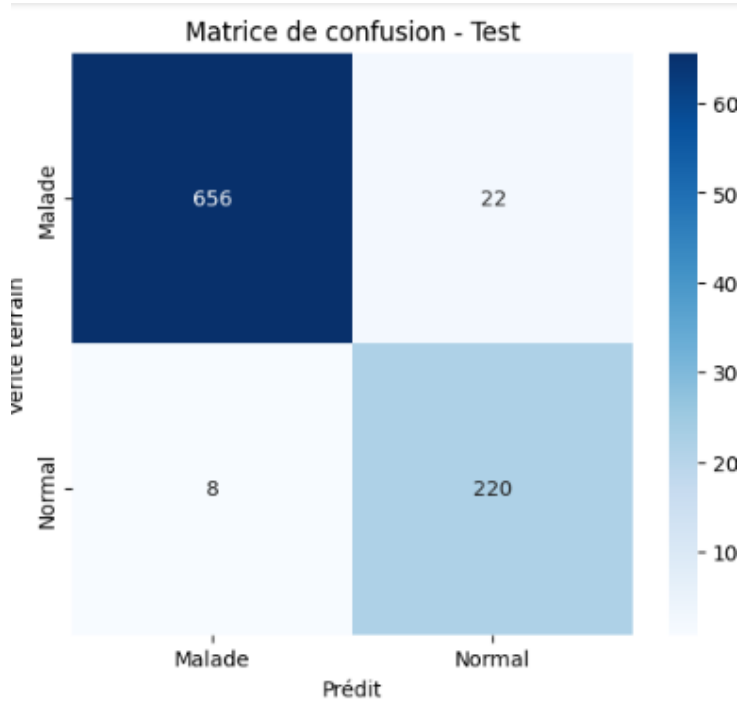



**Figure 4.10 :** Matrice de confusion

La matrice de confusion montre que le modèle distingue très efficacement les patients malades des patients normaux :

- **Très bonne détection des malades :** 656 sur 678 cas sont correctement identifiés (rappel $\approx$ 96,75 %), ce qui est crucial dans un contexte médical où rater un patient malade peut être grave.
- **Faible taux de faux positifs :** Seulement 8 patients normaux sont classés à tort comme malades, ce qui témoigne d'une bonne précision globale et d'une bonne spécificité.
- **Les 220 vrais négatifs sur 228 cas** montrent que le modèle est aussi fiable pour reconnaître les patients normaux (spécificité $\approx$ 96,5 %).

Le modèle offre un excellent compromis entre sensibilité et spécificité. Il est donc fiable pour un usage clinique, avec un taux d'erreur minimal.

# 8 Interface du système

Nous avons développé une interface interactive permettant à l'utilisateur d'entrer une image d'électrocardiogramme (ECG) du patient ainsi qu'un ensemble de variables cliniques :

- Fraction d'éjection (EF)

- Brain Natriuretic Peptide (BNP)
- New York Heart Association (NYHA)
- Pression artérielle systolique (SBP)
- Âge


**Figure 4.11 :** Interface de notre système

Après avoir renseigné ces données et cliqué sur le bouton "Lancer la prédiction", l'interface envoie ces informations à un modèle GCN multi-modal qui combine les caractéristiques extraites de l'image ECG et les données cliniques. Le modèle traite ces entrées et renvoie un diagnostic prédictif : *Normal* ou *Malade*.
Le résultat s'affiche de manière claire à l'utilisateur avec une couleur indiquant l'état du patient. L'image ECG sélectionnée est également affichée sous la prédiction pour permettre une vérification visuelle rapide. (Figure 4.12)


**Figure 4.12 :** Résultats de prédiction

L'exemple suivant présente un cas où l'utilisateur sélectionne une image ECG et renseigne les données cliniques. Après lancement de la prédiction, le modèle indique que le patient est *Normal* et affiche l'image ECG correspondante.

# 9 Conclusion

Dans ce chapitre, nous avons mis en œuvre une solution complète et multimodale pour la détection de l'insuffisance cardiaque, reposant sur l'intégration d'images ECG et de variables cliniques simulées dans un modèle de type Graph Neural Network, plus précisément un Graph Convolutional Network (GCN).

Nous avons d'abord présenté l'environnement de développement et les outils utilisés, comprenant notamment Python, PyTorch Geometric, Scikit-learn et Streamlit. Ensuite, nous avons décrit la base d'apprentissage, suivie par le prétraitement des données incluant le chargement et le nettoyage initial des images, la division des données en ensembles d'entraînement et de validation, ainsi que la préparation des caractéristiques à travers l'encodage des labels, la normalisation des variables cliniques et le prétraitement des images. Cette étape a été suivie par l'extraction des caractéristiques d'image, puis la fusion des données visuelles et cliniques, avant de procéder à la construction du graphe de similarité.

Nous avons ensuite présenté les résultats expérimentaux en abordant successivement la comparaison entre ResNet18 et ResNet50, l'influence du paramètre $k$ dans le graphe k-NN, l'évaluation comparative des différentes approches (unimodale, tabulaire et multimodale), les résultats comparatifs des approches testées, et la visualisation des performances du modèle GCN multimodal.
Enfin, nous avons présenté l'interface du système, permettant une utilisation interactive du modèle, en saisissant une image ECG et cinq paramètres cliniques, pour obtenir une prédiction instantanée.

# Conclusion Générale

La prédiction de l'insuffisance cardiaque représente l'un des défis majeurs dans le domaine médical en raison de la gravité de cette pathologie et de sa large propagation. Il devient donc indispensable de développer des solutions intelligentes et efficaces. Dans ce travail, nous avons abordé cette problématique en nous appuyant sur des techniques d'apprentissage profond, notamment les réseaux de neurones graphiques (GNN), en adoptant une approche multimodale combinant des données d'images et des données cliniques.

Les résultats obtenus ont montré que le modèle GCN a offert les meilleures performances avec une précision de 97 %, surpassant le modèle GAT. Les expériences ont également démontré que la combinaison de différents types de données (images et valeurs cliniques) améliore considérablement la performance des modèles par rapport à l'utilisation d'un seul type de données.

Sur le plan scientifique, ce travail contribue à renforcer les approches récentes basées sur l'intégration des données graphiques, visuelles et numériques au sein d'un même modèle intelligent. Ce projet constitue également une base de référence qui pourra être exploitée dans des travaux futurs en diagnostic médical assisté par intelligence artificielle, notamment en ce qui concerne l'utilisation des réseaux de neurones graphiques avec des données cliniques et des images médicales.

Des recherches de ce type représentent une étape importante dans l'amélioration de la qualité des soins de santé, car elles permettent d'accélérer et d'affiner le diagnostic, de fournir un soutien aux équipes médicales dans la prise de décision et de réduire les erreurs médicales qui peuvent avoir des conséquences graves sur la vie des patients. Malgré les résultats prometteurs, cette étude présente certaines limites, notamment l'utilisation de données simulées ou limitées en taille et en diversité, ce qui pourrait affecter la capacité du modèle à généraliser et à s'adapter à des cas réels plus complexes. De plus, les expériences se sont limitées aux modèles GCN et GAT sans explorer d'autres architectures avancées des réseaux de neurones graphiques.

Les perspectives futures qui peuvent être envisagées dans le prolongement de ce travail consistent à utiliser des données médicales réelles et plus variées provenant d'établissements hospitaliers, à Augmenter la taille des bases de données et tester le modèle sur des cas médicaux diversifiés, ainsi qu'à explorer d'autres modèles GNN plus performants

tels que GraphSAGE et GIN. Ces perspectives permettront de renforcer la fiabilité des modèles et de favoriser leur utilisation en toute sécurité dans un environnement médical réel.

# Bibliographie


[1] ActiGraph Blog. *Demystifying AI : A Review of the Fundamentals and Applications in Drug Development*. https://blog.theactigraph.com/blog/demystifying-ai-drug-development. Consulté le 15 juin 2025. Mai 2025.

[2] ActiveState. *What is Pandas in Python ? Everything You Need to Know*. https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/. Accessed : 2025-05-14. 2025.

[3] Almoosa Health Group. *Quelles sont les maladies que détecte l'électrocardiogramme ?* Consulté le 4 juin 2025. n.d. URL : https://almoosahealthgroup.org.

[4] American Heart Association. *Classes and Stages of Heart Failure*. https://www.heart.org/en/health-topics/heart-failure/what-is-heart-failure/classes-of-heart-failure. Accessed : 2025-06-14. 2025.

[5] WALID AMEUR et BOUZIANE BENHAOUCHE. "Vers une approche de recommandation sociale à base de préférence". Thèse de doct. Université ibn khaldoun-Tiaret, 2024.

[6] Atomcamp. *Deep Learning vs Machine Learning – What's The Difference ?* https://www.atomcamp.com/difference-machine-learning-deep-learning/. Consulté le 15 juin 2025. 2024.

[7] A. Baccouche et al. "Ensemble deep learning models for heart disease classification : A case study from Mexico". In : *Information* 11.4 (2020), p. 207.

[8] Asma Baccouche et al. "Ensemble deep learning models for heart disease classification : A case study from Mexico". In : *Information* 11.4 (2020), p. 207.

[9] Muhammet Balcilar et al. "Bridging the gap between spectral and spatial domains in graph neural networks". In : *arXiv preprint arXiv :2003.11702* (2020).

[10] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In : *arXiv preprint arXiv :1806.01261* (2018).

[11] Khaled BENSID et al. "Heart failure analysis using Artificial Intelligence". Thèse de doct. UNIVERSITY OF KASDI MERBAH OUARGLA, 2024.

[12] Uzair Aslam Bhatti et al. “Deep learning with graph convolutional networks : An overview and latest applications in computational intelligence”. In : *International Journal of Intelligent Systems* 2023.1 (2023), p. 8342104.

[13] Heloisa Oss Boll et al. “Graph Neural Networks for Heart Failure Prediction on an EHR-Based Patient Similarity Graph”. In : *arXiv preprint arXiv :2411.19742* (2024).

[14] A. Bourazana et al. “Artificial intelligence in heart failure : Friend or foe ?” In : *Life* 14.1 (2024), p. 145.

[15] Groupe Insuffisance Cardiaque et Cardiomyopathies (GICC) de la SFC. *Plaidoyer pour une prise en charge de l'insuffisance cardiaque et des cardiomyopathies*. Français. Publié le 23 décembre 2022. GICC, 2021.

[16] Edward Choi et al. “Using recurrent neural network models for early detection of heart failure onset”. In : *Journal of the American Medical Informatics Association* 24.2 (2017), p. 361-370.

[17] Cleverstory. *What is Artificial Intelligence (AI)? Everything You Need to Know*. Consulté le 25 avril 2025. 2021. url : https://experiences.cleverstory.io/media/what-is-artificial-intelligence-ai-everything-you-need-to-know.

[18] Data and Beyond. *Streamlit*. https://medium.com/data-and-beyond/streamlit-d357935b9c. Medium, Accessed : 2025-05-14. 2025.

[19] Domino.ai. *Jupyter Notebook*. https://domino.ai/data-science-dictionary/jupyter-notebook. Accessed : 2025-05-14. 2025.

[20] Domino.ai. *Scikit-learn*. https://domino.ai/data-science-dictionary/sklearn. Accessed : 2025-05-14. 2025.

[21] Domino.ai. *What is Plotly ?* https://domino.ai/data-science-dictionary/plotly. Accessed : 2025-05-14. 2025.

[22] E. Elia. *Drawing Architecture : Building Deep Convolutional GAN's In PyTorch*. https://towardsdatascience.com/drawing-architecure-building-deepconvolutional-gans-in-pytorch-5ed60348d43c. Medium, Accessed : 2025-04-14. 2020.

[23] evilspirit05. *ECG Images Dataset of Cardiac Patients*. https://www.kaggle.com/datasets/evilspirit05/ecg-analysis. Consulté en juin 2025. 2024.

[24] Fédération Française de Cardiologie. *Principaux examens pour dépister, diagnostiquer et surveiller en cardiologie*. Consulté le 3 juin 2025. Paris, 2020. URL : https://www.cite-sciences.fr/fileadmin/fileadmin_CSI/fichiers/au-programme/lieux-ressources/cite-de-la-sante/_documents/Ressources/Brochures/2020-EXAM-EN-CARDIO.pdf.

[25] GeeksforGeeks. *Support Vector Machine (SVM) Algorithm*. Accessed : 2025-06-15. 2023. URL : https://www.geeksforgeeks.org/support-vector-machine-algorithm/.

[26] Aditya Grover et Jure Leskovec. "node2vec : Scalable feature learning for networks". In : *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, p. 855-864.

[27] Chiradeep Gupta et al. "Cardiac Disease Prediction using Supervised Machine Learning Techniques." In : *Journal of physics : conference series*. T. 2161. 1. IOP Publishing. 2022, p. 012013.

[28] Kaiming He et al. "Deep residual learning for image recognition". In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 770-778.

[29] IA School. *L'intelligence artificielle, un outil de diagnostic médical au service de la santé*. Consulté le 3 juin 2025. Déc. 2024. URL : https://www.intelligence-artificielle-school.com/actualite/intelligence-artificielle-outil-diagnostic-medical-service-sante/.

[30] IBM. *Semi-Supervised Learning*. Consulté le : 25 avril 2025. 2025. URL : https://www.ibm.com/think/topics/semi-supervised-learning.

[31] IBM. *Supervised Learning*. Consulté le : 25 avril 2025. 2025. URL : https://www.ibm.com/think/topics/supervised-learning.

[32] Bo Jin et al. "Predicting the risk of heart failure with EHR sequential data modeling". In : *Ieee Access* 6 (2018), p. 9256-9261.

[33] Asifullah Khan et al. "A survey of the recent architectures of deep convolutional neural networks". In : *Artificial intelligence review* 53 (2020), p. 5455-5516.

[34] Bharti Khemani et al. "A Review of Graph Neural Networks : Concepts, Architectures, Techniques, Challenges, Datasets, Applications, and Future Directions". In : *Journal of Big Data* 11.1 (2024), p. 18. DOI : 10.1186/s40537-023-00876-4. URL : https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00876-4.

[35] Thomas N Kipf et Max Welling. "Semi-supervised classification with graph convolutional networks". In : *arXiv preprint arXiv :1609.02907* (2016).

[36] Thomas N. Kipf. *Graph Convolutional Networks*. https://tkipf.github.io/graph-convolutional-networks/. Consulté le 15 juin 2025. 2016.

[37] Surenthiran Krishnan, Pritheega Magalingam et Roslina Ibrahim. "Hybrid deep learning model using recurrent neural network and gated recurrent unit for heart disease prediction." In : *International Journal of Electrical & Computer Engineering (2088-8708)* 11.6 (2021).

[38] FatimaEzzahra Laghrissi et al. "Intrusion detection systems using long short-term memory (LSTM)". In : *Journal of Big Data* 8.1 (2021), p. 65.

[39] Le Data Scientist. *Google Colab : Le Guide Ultime*. https://medium.com/le-data-scientist/google-colab-le-guide-ultime-ca6464bbdc59. Medium, Accessed : 2025-05-14. 2025.

[40] Yann LeCun, Yoshua Bengio et Geoffrey Hinton. "Deep learning". In : *nature* 521.7553 (2015), p. 436-444.

[41] Lenes, Kjetil (Ekko). *PLAX M-mode echocardiogram (parasternal long-axis view)*. https://fr.wikipedia.org/wiki/Fichier:PLAX_Mmode.jpg. Image domaine public, public domain ; consultée le 15 juin 2025.

[42] Dengao Li et al. "A deep learning system for heart failure mortality prediction". In : *Plos one* 18.2 (2023), e0276835.

[43] R. Auxilia Anitha Mary et T. Ramaprabha. "Predicting Heart Disease Algorithm Using DNN & MNN in Deep Learning". In : *International Research Journal on Advanced Engineering Hub (IRJAEH)* 2.4 (avr. 2024), p. 783-792. issn : 2584-2137. doi : 10.47392/IRJAEH.2024.0110. url : https://irjaeh.com/index.php/journal/article/view/131.

[44] Pamela McCorduck et Cli Cfe. *Machines who think : A personal inquiry into the history and prospects of artificial intelligence*. AK Peters/CRC Press, 2004.

[45] E. Mertens et O. Barthélemy. "Quelle prise en charge des SCA en 2022 ?" In : *Réalités Cardiologiques* 370 (mars 2022).

[46] Kathleen H Miao et Julia H Miao. "Coronary heart disease diagnosis using deep neural networks". In : *international journal of advanced computer science and applications* 9.10 (2018).

[47] Clive Neal-Sturgess. "Assessing Mortality of Blunt Trauma with Co-morbidity". In : *arXiv preprint arXiv :1711.08056* (2017).

[48] Neo4j Inc.. *GraphSAGE - Neo4j Graph Data Science*. Consulté le 7 juin 2025. 2025. url : https://neo4j.com/docs/graph-data-science/current/machine-learning/node-embeddings/graph-sage/.

[49] NumPy Developers. *What is NumPy ?* https://numpy.org/doc/stable/user/whatisnumpy.html. Accessed : 2025-05-14. 2025.

[50] NVIDIA Corporation. *PyTorch*. https://www.nvidia.com/en-us/glossary/pytorch/. Accessed : 2025-05-14. 2025.

[51] Oracle. *Unsupervised Learning*. Consulté le : 25 avril 2025. 2025. url : https://www.oracle.com/ee/artificial-intelligence/machine-learning/unsupervised-learning/.

[52] Organisation mondiale de la Santé. *Maladies cardiovasculaires (CVDs)*. Consulté le 27 avril 2025. 2024. url : https://www.who.int/fr/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[53] Rythmo Paris Ouest. *L'électrocardiogramme*. https://rythmo-parisouest.fr/lelectrocardiogramme/. Consulté le 15 juin 2025.

[54] PasseportSanté. *L'analyse du peptide natriurétique de type B et de ses résultats*. https://www.passeportsante.net/fr/Maux/analyses-medicales/Fiche.aspx?doc=analyse-peptide-natriuretique-B-sang. Accessed : 2025-06-14. 2024.

[55] Seyedamin Pouriyeh et al. "A comprehensive investigation and comparison of machine learning techniques in the domain of heart disease". In : *2017 IEEE symposium on computers and communications (ISCC)*. IEEE. 2017, p. 204-207.

[56] David MW Powers. "Evaluation : from precision, recall and F-measure to ROC, informedness, markedness and correlation". In : *arXiv preprint arXiv :2010.16061* (2020).

[57] ALSON PRASAI. "DETECTING MISINFORMATION ON SOCIAL MEDIA : A GNN-BASED ENSEMBLE LEARNING METHODOLOGY". In : (2023).

[58] Primo Medico. *Diagnostic cardiaque*. Consulté le 3 juin 2025. 2025. url : https://www.primomedico.com/fr/cure/diagnostic-cardiaque/.

[59] Python Software Foundation. *Python Blurb*. https://www.python.org/doc/essays/blurb/. Accessed : 2025-05-14. 2025.

[60] PyTorch Geometric Team. *PyTorch Geometric Documentation*. https://pytorch-geometric.readthedocs.io/en/latest/. Accessed : 2025-05-14. 2025.

[61] Atta Ur Rahman et al. "Enhancing heart disease prediction using a self-attention-based transformer model". In : *Scientific Reports* 14.1 (2024), p. 514.

[62] Fatima Zohra Rahmani. "Une Approche basée TextING pour la classification des services web". Thèse de doct. Université Ibn Khaldoun-Tiaret-, 2022.

[63] P Ramprakash et al. “Heart disease prediction using deep neural network”. In : *2020 international conference on inventive computation technologies (ICICT)*. IEEE. 2020, p. 666-670.

[64] Lakshmi Sravanthi Rangisetti et al. “Heart Disease Detection Using Graph Neural Networks (GNNs)”. In : *Proceedings of the 2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2024, p. 1436-1441. doi : 10.1109/ICESC57971.2024.1234567. url : https://ieeexplore.ieee.org/document/1234567.

[65] WJ Remme et K Swedberg. “Recommandations pour le diagnostic et le traitement de l’insuffisance cardiaque chronique”. In : *Groupe de travail pour le diagnostic et le traitement de l’insuffisance cardiaque chronique. Société européenne de cardiologie. Arch Mal Cœur2009* 95 (), p. 5-53.

[66] Rythmo. *Fraction d’éjection*. https://www.rythmo.fr/glossary/fraction-dejection/. Consulté en juin 2025. n.d.

[67] Mohammed Saber et al. “An optimized spectrum sensing implementation based on SVM, KNN and TREE algorithms”. In : *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE. 2019, p. 383-389.

[68] Oluwarotimi Williams Samuel et al. “An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction”. In : *Expert systems with applications* 68 (2017), p. 163-172.

[69] Franco Scarselli et al. “The graph neural network model”. In : *IEEE transactions on neural networks* 20.1 (2008), p. 61-80.

[70] Sermo. *L’IA au service du diagnostic médical et de la prédiction des résultats obtenus par les patients*. Consulté le 18 avril 2025. 2023. url : https://www.sermo.com/fr/resources/lia-au-service-du-diagnostic-medical-et-de-la-prediction-des-resultats-obtenus-par-les-patients/.

[71] Oleksandr Shchur et al. “Pitfalls of graph neural network evaluation”. In : *arXiv preprint arXiv :1811.05868* (2018).

[72] S Irin Sherly et G Mathivanan. “An efficient honey badger based Faster region CNN for chronc heart Failure prediction”. In : *Biomedical Signal Processing and Control* 79 (2023), p. 104165.

[73] Alex Sherstinsky. “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network”. In : *Physica D : Nonlinear Phenomena* 404 (2020), p. 132306.

[74] B. Sibilia. “Modèle pronostique de machine learning supervisé, incluant des facteurs environnementaux, pour prédire la sévérité intra-hospitalière chez les patients admis pour insuffisance cardiaque aiguë”. Thèse pour le diplôme d’État de docteur en médecine. Université Paris Cité, 2024. URL : https://dumas.ccsd.cnrs.fr/dumas-04741070v1.

[75] VK Sudha et D Kumar. “Hybrid CNN and LSTM network for heart disease prediction”. In : *SN Computer Science* 4.2 (2023), p. 172.

[76] Vivienne Sze et al. “Efficient processing of deep neural networks : A tutorial and survey”. In : *Proceedings of the IEEE* 105.12 (2017), p. 2295-2329.

[77] Claude Touzet. *les réseaux de neurones artificiels, introduction au connexionnisme*. Ec2, 1992.

[78] Ultralytics. *Graph Neural Network (GNN)*. https://www.ultralytics.com/fr/glossary/graph-neural-network-gnn. Consulté le 16 juin 2025. n.d.

[79] Paul Valensi. “Cœur et diabète”. In : *Actualité et Dossier en Santé Publique (ADSP)* 63 (juin 2008). Consulté dans le cadre de l’étude sur les pathologies cardiovasculaires, p. 42-49.

[80] Petar Veličković et al. “Graph attention networks”. In : *arXiv preprint arXiv :1710.10903* (2017).

[81] Rakhi Wajgi et al. “Heart Disease Prediction using Graph Neural Network”. In : *International Journal of Intelligent Systems and Applications in Engineering* 12.12s (2024), p. 280-287. URL : https://ijisae.org/index.php/IJISAE/article/view/4514.

[82] Kemas Rahmat Saleh Wiharja et al. “Early Detection of Heart Disease with Graph Neural Network”. In : *2024 12th International Conference on Information and Communication Technology (ICoICT)*. IEEE. 2024, p. 405-410.

[83] World Health Organization. *Cardiovascular Diseases*. Consulté en avril 2025. Fév. 2022. URL : https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1.

[84] Zonghan Wu et al. “A comprehensive survey on graph neural networks”. In : *IEEE transactions on neural networks and learning systems* 32.1 (2020), p. 4-24.

[85] XenonStack. *Graph Neural Network Applications and its Future*. Consulté le 7 juin 2025. Août 2024. URL : https://www.xenonstack.com/blog/graph-neural-network-applications.

[86] Keyulu Xu et al. “How powerful are graph neural networks ?” In : *arXiv preprint arXiv :1810.00826* (2018).

[87] Heng-Kai Zhang et al. “HONGAT : graph attention networks in the presence of high-order neighbors”. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 38. 15. 2024, p. 16750-16758.

[88] Si Zhang et al. “Graph convolutional networks : a comprehensive review”. In : *Computational Social Networks* 6.1 (2019), p. 1-23.

[89] Jie Zhou et al. “Graph neural networks : A review of methods and applications”. In : *AI open* 1 (2020), p. 57-81.

[90] Marinka Zitnik, Monica Agrawal et Jure Leskovec. “Modeling polypharmacy side effects with graph convolutional networks”. In : *Bioinformatics* 34.13 (2018), p. i457-i466.