République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique

MEMOIRE
Présenté en vue de l'obtention du diplôme
Master en informatique
Spécialité : Réseaux et Multimedia

THEME

La classification des Maladies via une Analyse Médicale Basée sur l'Apprentissage Profond

Présenté par :
BENSADI Houssem Eddine
BENSEGHIR Aya

Soutenu publiquement le : 20/06/2024
Devant le jury composé de :
Président : M. Dr NOUIOUA Mourad
Examinateur :M. Prof MOSTEFAI Messaoud
Encadreur : Mme. Dr SAIFI Lynda

2023/2024

Dédicace

À mon éternel exemple, mon soutien moral et ma source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, à toi, mon père.

À la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur ; ma mère que j'adore.

Je souhaite adresser un salut particulier à mon collègue **IDIR Yacine**, un compagnon tout au long de mon parcours éducatif, un ami idéal et l'une des élites dans le domaine de l'informatique.

Je tiens également à remercier ma collègue **BENSEGHIR Aya** pour ses efforts incessants et son soutien continu pendant les périodes difficiles tout au long du parcours universitaire.

Aux personnes qui m'ont toujours aidé et encouragé, qui ont toujours été à mes côtés, qui m'ont accompagné tout au long de mon parcours universitaire, mes chers amis, mes camarades de classe, mes frères et sœurs de cœur, à toi BENSEGHIR Aya.

BENSADI Houssem Eddine

Dédicace

Début, je remercie Allah pour la grâce de la force, de la volonté et du courage dont Il m'a doté, et qui m'ont aidé à réaliser cette humble réussite.

Je dédie ce travail à mon cher père, qui a toujours été une source d'inspiration et de soutien pour moi, et qui m'a enseigné que le travail acharné et la loyauté sont la clé du succès dans la vie. Que Dieu ait son âme et l'accueille dans ses vastes paradis, et je n'oublierai pas ses conseils et ses orientations qui m'ont aidé dans mon parcours éducatif.

Et à ma mère, qui est une source de tendresse et de lumière, et qui m'a guidé sur le chemin du succès par ses sacrifices et ses précieux conseils, et son soutien continu dans ma vie.

À ma chère sœur et à ses enfants, à mon cher frère et à mon grand-père, qui m'ont toujours soutenu et encouragé, et qui ont été à mes côtés à chaque étape du chemin.

En conclusion, je tiens à exprimer ma profonde gratitude à tous les enseignants sans exception, pour leurs efforts et leurs contributions à mon parcours éducatif.

BENSEGHIR Aya

Remerciement

Tout d'abord, nous remercions Allah de nous avoir donné la force, la capacité et la volonté de mener à bien ce travail.

Nous adressons nos sincères remerciements à notre estimée encadrante Mme. **SAIFI Lynda**, pour ses efforts remarquables, ses précieux conseils et son dévouement constant à suivre et superviser les étapes de réalisation de ce projet. Nous tenons également à lui exprimer notre profonde gratitude pour le temps et les efforts qu'elle nous a consacrés et pour son soutien continu.

Nous exprimons également nos sincères remerciements aux membres du jury pour leur volonté d'examiner et d'évaluer attentivement notre travail, et pour leurs précieux conseils qui ont contribué à son développement et à son amélioration.

Résumé

L'un des défis majeurs auxquels les médecins sont confrontés est la prise de décisions concernant la maladie liée à l'état du patient. Notre sujet aborde l'un des problèmes liés à cela, qui est le suivant : les données peuvent-elles être davantage utilisées pour améliorer la précision de la prise de décision médicale ? Pour résoudre ce problème, nous proposons un mécanisme de classification des maladies basé sur les données médicales et les symptômes associés au patient, en s'appuyant sur des algorithmes d'apprentissage profond.

La solution proposée à la problématique présentée dépend de la construction de modèles pour les réseaux artificiels profonds standard (ANN) en appliquant différents algorithmes d'optimisation (SGD, ADAM) et en appliquant des algorithmes de réduction de dimension ACP. Les résultats de la comparaison entre les performances des différents modèles ont montré une grande efficacité dans l'application de la réduction dimensionnelle avec l'algorithme d'optimisation SGD en termes de traitement de nouvelles données et de temps nécessaire à l'entraînement.

Nous concluons des résultats à l'efficacité du deep learning pour résoudre les problèmes associés à la classification et à l'exploitation des données.

Mots-clés : Intelligence artificielle , Apprentissage profond ,Classification, Optimisation, Maladies, Symptômes.

Abstract

One of the major challenges faced by doctors is making decisions regarding disease related to the patient's condition. Our topic addresses one of the problems related to this, which is : can data be further used to improve the accuracy of medical decision-making ? To solve this problem, we propose a disease classification mechanism based on medical data and patient-associated symptoms, relying on deep learning algorithms.

The proposed solution to the presented problem relies on building models for standard deep artificial networks by applying different optimization algorithms (SGD, ADAM) and applying PCA for dimensionality reduction. The results of comparing the performance of different models showed high efficiency in applying dimensionality reduction with the SGD optimization algorithm in terms of handling new data and the time required for training.

We conclude from the results the effectiveness of deep learning in solving problems associated with classification and data exploitation.

Keywords : Artificial intelligence, Deep learning, Classification, diseases, Optimisation, Symptoms.

ملخص

يواجه الأطباء تحديات كبرى عند اتخاذ القرارات الطبية المتعلقة بحالة المريض. يتناول موضوعنا أحد هذه المشاكل وهو: هل يمكن استخدام البيانات بشكل أكبر لتحسين دقة اتخاذ القرار الطبي؟ لحل هذه المشكلة، نقترح آلية لتصنيف الأمراض تعتمد على البيانات الطبية والأعراض المرتبطة بالمريض، بالاعتماد على خوارزميات التعلم العميق.

تعتمد الحلول المقترحة للمشكلة المقدمة على بناء نماذج للشبكات العميقة القياسية من خلال تطبيق خوارزميات تحسين مختلفة (النزول الاشتقاقي العشوائي، ادام) وتطبيق خوارزميات تخفيض الأبعاد (تحليل العنصر الاساسي). أظهرت نتائج مقارنة أداء النماذج المختلفة فعالية كبيرة في تطبيق تخفيض الأبعاد مع خوارزمية تحسين ض من حيث معالجة البيانات الجديدة والوقت اللازم للتدريب.

نستنتج من النتائج فعالية التعلم العميق في حل المشاكل المرتبطة بتصنيف واستغلال البيانات.

الكلمات المفتاحية: الذكاء الاصطناعي، التعلم العميق، التصنيف، الأمراض، تحسين ،اعراض.

Table des matières

Liste des abréviations

ACP Analyse en Composantes Principales.

ADAM Adaptive Moment Estimation.

ANN Artificielle Neural Network.

CCE Categorical cross Entropy.

CNN Convolutional Neural Network.

CPU Unité centrale de traitement.

DL Deep Learning.

EMC Évaluation Médicale Compréhensive.

FN Faux Négatifs.

FP Faux Positifs.

GPU Unité de traitement graphique.

IA Intelligance Artificielle.

KNN K Nearest Neighbors.

LR Logistique Regression.

LSTM Long Short-Term Memory.

ML Machine Learning.

PC Personal Computer.

RAM Mémoire vive.

ReLU Rectified Linear Unite.

RNN Recurrent Neural Network.

SGD Stochastic Gradient Descent.

SVM Support Vector Machine.

TPU Unité de traitement tensoriel.

VN Vrais Négatifs.

VP Vrais Positifs.

Table des figures

Liste des tableaux

List of Algorithms

Introduction Générale

Contexte Général

Le domaine médical est l'un des domaines qui suscite un grand intérêt de la part des chercheurs et des scientifiques, alors que le monde cherche à développer des méthodes, notamment des méthodes de prise de décision pour les médecins. L'un des tournants est l'adoption de la technologie de l'intelligence artificielle dans le domaine médical, qui améliorera la qualité des soins médicaux, et l'une des technologies les plus importantes est l'apprentissage profond.
L'apprentissage profond est l'une des techniques de classification et d'analyse des données les plus importantes de l'ère moderne, car il a permis une plus grande expansion de l'intelligence artificielle et peut désormais être exploité pour résoudre de nombreux problèmes auparavant insolubles.

Problématique

Les médecins sont confrontés à de nombreux défis lorsqu'ils prennent des décisions médicales en raison de la grande quantité de données médicales, ce qui entraîne parfois des incohérences dans leurs diagnostics. Cela soulève donc la question suivante :
Les données peuvent-elles être davantage utilisées pour améliorer la précision de la prise de décision médicale ?

Objectif

Notre objectif visent à améliorer la qualité des soins de santé et du diagnostic médical en fournissant un processus de diagnostic plus rapide et plus précis. L'amélioration de la qualité des soins de santé et du diagnostic médical constitue l'un des principaux défis auxquels est confronté le secteur de la santé, car les retards ou les inexactitudes dans le diagnostic peuvent avoir des effets négatifs.

Comme une mauvaise compréhension de la condition médicale, qui affecte les résultats thérapeutiques.

Pour assurer une amélioration continue, nous développerons différents models des réseaux neurons artificiel ANN pour le but de classification des symptômes médicales ensuite comparerons les différents modèles en termes de performances et de résultats.

Cela nous permettra d'identifier les meilleures pratiques et de choisir l'approche optimale pour améliorer la qualité des soins et du diagnostic médical.

Organisation du mémoire

Le mémoire est organisé en suivant le plan suivant :

Le premier chapitre, souligne l'interdépendance cruciale entre le diagnostic, les symptômes et le pronostic médical dans le contexte de la médecine moderne.

Le deuxième chapitre ,présente l'apprentissage automatique afin de pouvoir analyser les données médicales.

Dans **le troisième chapitre**, nous avons présenté l'apprentissage profond (DL) et ses différents types de réseaux de neurones artificiels .

Le quatrième chapitre, nous avons clarifié la stratégie générale de notre travail en exposant clairement nos objectifs. Nous avons décrit la méthodologie adoptée pour mener une étude comparative entre des modèles de classification des symptômes, combinés à différentes méthodes d'optimisation été adopté.

Le cinquième chapitre, présente nos expérimentations visant à atteindre nos objectifs de recherche. Nous y détaillons les outils utilisés tels que le langage de programmation, l'environnement de développement et le matériel, ainsi que le processus de classification utilisant différentes architectures d'apprentissage profond.

Chapitre 1: Domaine médical

1.1 Introduction

L'évaluation médicale est un processus fondamental dans la pratique clinique, visant à évaluer l'état de santé d'un patient et à déterminer les meilleures stratégies de prise en charge. Dans ce contexte, l'Évaluation Médicale Compréhensive (EMC) émerge comme une approche intégrative, cherchant à appréhender la globalité des besoins du patient au-delà de la simple symptomatologie.

1.2 Diagnostic médical

Le diagnostic médical, fruit de la science et de l'expertise, il éclaire le chemin vers la compréhension des symptômes, guidant ainsi le parcours vers la guérison .

1.2.1 Définitions diagnostic

Le diagnostic est le processus d'évaluation d'un état de fonctionnement donné. Si cet état est comparé avec un état de référence, il s'agit d'évaluation de dérive de fonctionnement [1].

1.2.2 Les étapes de diagnostic

- **Collecte d'informations** : Recueillir des données pertinentes liées au problème ou à la situation, que ce soit des symptômes, des données techniques, des antécédents, etc.
- **Identification du problème** : Analyser les informations collectées pour déterminer la nature du problème. Cela peut impliquer la comparaison des données avec des normes ou des critères établis.

- **Élaboration d'hypothèses** : Formuler des hypothèses sur les causes possibles du problème en se basant sur les informations disponibles.
- **Tests et investigations** : Mettre en place des tests ou des investigations pour valider ou invalider les hypothèses formulées. Cela peut inclure des examens médicaux, des tests techniques, des simulations, etc.
- **Analyse des résultats** : Examiner les résultats des tests et des investigations afin de confirmer la cause du problème ou de revoir les hypothèses si nécessaire.
- **Établissement du diagnostic** : Formuler un diagnostic final en identifiant la cause principale du problème ou de la situation, en tenant compte de toutes les informations et des résultats obtenus.
- **Proposition de solutions** : Proposer des solutions ou des recommandations pour résoudre le problème diagnostiqué. Cela peut impliquer des traitements médicaux, des ajustements techniques, des interventions psychologiques, etc.
- **Suivi et évaluation** : Mettre en place un suivi pour évaluer l'efficacité des solutions proposées et ajuster si nécessaire. Cela peut également inclure des mesures préventives pour éviter la récurrence du problème .

1.3 Symptômes médicaux

Les symptômes médicaux sont souvent les signaux précurseurs d'un état de santé sous-jacent, nécessitant une évaluation médicale approfondie pour établir un diagnostic précis et élaborer un plan de traitement adapté.

1.3.1 Définitions les symptômes

Les symptômes se réfèrent à des signes ou manifestations d'une maladie, d'un trouble ou d'une condition médicale. Ce sont des indicateurs observables ou ressentis par le patient, ainsi que détectés par les professionnels de la santé. Les symptômes médicaux peuvent varier en fonction de la maladie ou du trouble et Il est important de noter que les symptômes ne sont pas la maladie elle-même, mais plutôt des signaux qui indiquent la présence possible d'un problème de santé. Ils peuvent être utilisés pour aider les professionnels de la santé à poser un diagnostic et à recommander un plan de traitement approprié.

1.3.2 Exemples de symptômes

- La fatigue est définie comme une sensation d'épuisement survenant durant ou après une activité habituelle. Une cause organique ou psychiatrique est retrouvée dans la majorité des cas.
- Le vomissement est le rejet du contenu de l'estomac par la bouche. Il correspond à un réflexe mécanique de défense de l'organisme destiné à vider l'estomac. Il est possible de vomir des aliments, de la bile ou beaucoup plus rarement, du sang[2].
- La fièvre est une température corporelle anormalement élevée, qui dépasse 38°C [2].
- La toux est l'expiration brusque et sonore de l'air contenu dans les poumons provoquée par une irritation des voies respiratoires [2].
- La douleur thoracique désigne toute douleur ou toute sensation anormale et pénible localisée dans la zone du thorax [2].
- La perte d'appétit est la perte de l'envie de manger, peut aussi être appelée anorexie [3].
- Les céphalées sont un problème très courant. Il existe plusieurs types de céphalées, les céphalées de tension étant les plus fréquentes. Bien qu'en général bénignes, les céphalées peuvent être le symptôme d'une maladie grave [2].

1.4 Pronostic médical

Le pronostic médical, étroitement lié aux symptômes médicaux présentés, offre une perspective éclairante sur l'évolution potentielle de la condition de santé, permettant aux professionnels de la santé de prendre des décisions informées pour un plan de traitement optimal.

1.4.1 Définition Pronostic médical

Le pronostic médical est une évaluation anticipée de l'évolution probable d'une maladie chez un patient, basée sur des données cliniques et des connaissances médicales, afin de prédire les perspectives de guérison, de rémission, de stabilisation ou de progression de la condition.

1.4.2 Type Pronostic médical

- **Pronostic vital** : Il concerne la probabilité de survie d'un patient, souvent exprimée en termes de taux de survie à un certain nombre d'années [4].

- **Pronostic qualité de vie** : Il prend en compte l'impact de la maladie ou du traitement sur la qualité de vie globale du patient [4].
- **Pronostic génétique** : Il se base sur des facteurs génétiques et évalue la probabilité de développement de maladies héréditaires [4].
- **Pronostic fonctionnel** : Il évalue la capacité du patient à maintenir une fonction normale ou à retrouver une fonction normale après un traitement [4].

1.5 Relation entre diagnostic, les symptômes et Pronostic médical

La relation entre le diagnostic, les symptômes et le pronostic médical est cruciale, les symptômes fournissant des indices précieux pour orienter le processus diagnostique, tandis que le diagnostic éclaire le pronostic en permettant une évaluation anticipée de l'évolution probable de la maladie, guider ainsi les choix de traitement et les décisions médicales.

1.6 Conclusion

Cette étude souligne l'interdépendance cruciale entre le diagnostic, les symptômes et le pronostic médical dans le contexte de la médecine moderne. Une compréhension approfondie de ces éléments est essentielle pour assurer des soins de qualité et des résultats positifs pour les patients. Le chapitre suivant présente l'apprentissage automatique afin de pouvoir analyser les données médicales.

Chapitre 2: Apprentissage Automatique

2.1 Introduction

L'apprentissage automatique, en anglais est l'appelé "Machine Learning (ML)". Représente une discipline passionnante au cœur de l'Intelligance Artificielle (IA). Ce domaine cherche à développer des modèles informatiques capables d'apprendre à partir de l'expérience et d'améliorer leurs performances au fil du temps, sans être explicitement programmés.

2.2 Apprentissage Automatique

L'apprentissage automatique est un type d'Intelligance Artificielle, c'est une science qui permet aux ordinateurs d'apprendre sans être explicitement programmés « Arthur Samuel, 1959 ». Plus précisément, l'apprentissage automatique fait référence au développement, l'analyse et l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer et de remplir des tâches associées à une intelligence artificielle grâce à un processus d'apprentissage. Cet apprentissage permet d'avoir un système qui s'optimise en fonction de l'environnement, les expériences et les résultats observés. Dans le domaine médicale, l'apprentissage automatique a été conçu pour réaliser l'analyse de données médicales, surtout lorsque l'évolution numérique a fourni des moyens (capteurs) peu coûteux permettant de recueillir et de stocker des informations importantes liées aux patients et maladies. Par exemple, les algorithmes d'apprentissage sont utiles au médecin lors du diagnostic des patients, afin d'améliorer la vitesse, la précision et la fiabilité de son diagnostic [5].

2.3 Types d'Apprentissage Automatique

Il existe fondamentalement quatre types d'apprentissage automatique : Supervisé, semi-supervisé et non-supervisé et Apprentissage par renforcement. Dans notre étude, nous utilisons l'apprentissage supervisé pour construire des modèles pour la prédiction des maladies. Dans la suite de cette section, nous allons présenter les deux types d'apprentissage les plus utilisés qui sont l'apprentissage supervisé et apprentissage non supervisé.

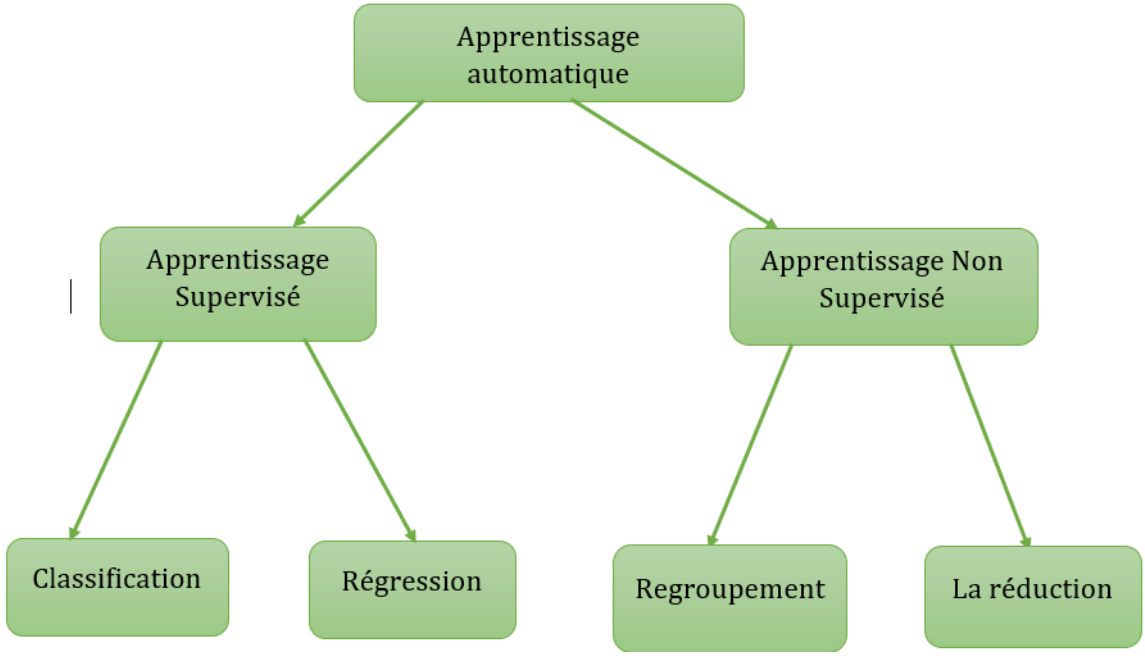

FIGURE 2.1 – Les différentes méthodes d'apprentissage automatique.

2.3.1 Apprentissage Supervisé

Dans l'apprentissage supervisé, l'ordinateur est fourni avec des exemples d'entrées qui sont étiquetés avec les sorties souhaitées. Le but de cette méthode est que l'algorithme puisse « apprendre » en comparant sa sortie réelle avec les sorties « apprises » pour trouver des erreurs et modifier le modèle en conséquence. L'objectif des algorithmes d'apprentissage supervisé est d'apprendre une fonction qui mappe les vecteurs de caractéristiques (entrées) aux étiquettes (sortie), sur la base d'exemples de paires entrée-sortie.

Comme il est illustré dans 2.1 l'apprentissage supervisé peut être utilisé pour deux types de tâches principales : la classification et la régression. Les algorithmes de classification cherchent à prédire la classe ou la catégorie à laquelle appartient une donnée d'entrée tandis que les

algorithmes de régression servent à prédire une valeur numérique continue à partir de variables d'entrée. Dans ce travail, nous nous intéressons aux méthodes de classification. [6]

2.3.2 Apprentissage Non-Supervisé

Dans l'apprentissage non supervisé, les données sont non étiquetées, de sorte que l'algorithme d'apprentissage trouve tout seul des points communs parmi ses données d'entrée. Les données non étiquetées étant plus abondantes que les données étiquetées, les méthodes d'apprentissage automatique qui facilitent l'apprentissage non supervisé sont particulièrement utiles. L'objectif de l'apprentissage non supervisé peut être aussi simple que de découvrir des modèles cachés dans un ensemble de données. Les plus fréquents problèmes connus dans ce type sont : [7]

- Le régroupement qui consiste à regrouper un ensemble d'éléments hétérogènes sous forme de sous-groupes homogènes.
- La réduction de dimension qui consiste à prendre des données dans un espace de grande dimension, et à les remplacer par des données dans un espace de plus petite dimension sans perdre la variance .

2.4 Définition de classification

Un modèle de classification est un modèle d'apprentissage automatique qui attribue des étiquettes ou des catégories à des instances d'entrée en les associant à un ensemble fini de classes prédéfinies. L'objectif est de créer une fonction qui peut généraliser à de nouvelles données et assigner correctement des classes à des observations inconnues en fonction des modèles appris à partir des données d'entraînement. Les classes peuvent être binaires (deux classes, comme oui/non) ou multinomiales (plus de deux classes), et le modèle cherche à discriminer entre ces classes en fonction des caractéristiques des données d'entrée. [8]

2.5 Types de classification

Dans cette partie, nous présentons les types de classification :

2.5.1 Classification binaire

La classification binaire (ou la classification binomiale) est une transformation de données qui vise à répartir les membres d'un ensemble dans deux groupes disjoints selon que l'élément possède ou non une propriété / fonctionnalité donnée, mais peut également être interprété comme vrai et faux, 1 et 0, ou toute autre combinaison de deux valeurs. Par exemple : tests médicaux visant à déterminer si un patient est atteint d'une certaine maladie ou non.

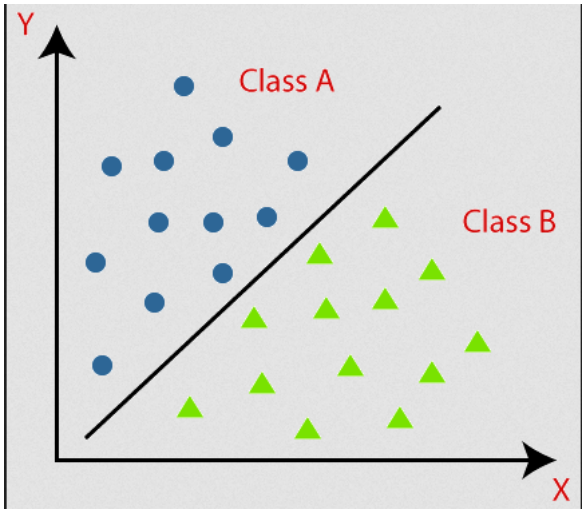

FIGURE 2.2 – Exemple de problème de classification binaire.

L'image 2.2 montre qu'il y a deux classes classe A et classe B.

2.5.2 Classification multi-classe

La classification multi-classe désigne une tâche de classification comportant plus de deux classes, par exemple, La classification des visages, classification des espèces végétales...Ect. Un jeu de données multi-classe n'a qu'une seule classe en sortie, comme dans la classification binaire.

2.5.3 Classification multi-lable

Jusqu'à présent, chaque instance était toujours affectée à une seule classe. Dans certains cas, le classificateur peut produire plusieurs classes pour chaque instance. Par exemple, un classificateur pour reconnaître des types de maladies : que faire s'il reconnaît plusieurs symptômes en même temps ? Bien entendu, il doit noter chaque symptôme associé à une maladie spécifique. Supposons qu'un classificateur ait été formé pour reconnaître trois symptômes : « fatigue, nausée et température élevée ». Ainsi, lorsqu'il apparaît « fatigué et une forte température ». Il devrait produire [1, 0, 1] (c'est-à-dire : fatigué oui, nausée non, fièvre oui.) Un système de classification qui produit plusieurs binaires est appelé système de classification multi-étiquettes. [9]

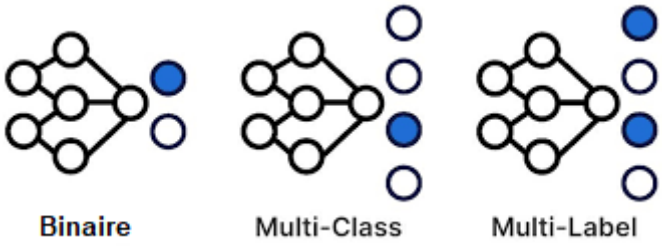

FIGURE 2.3 – Exemple de problème de classification binaire , multi-classe et multi-lable .

2.6 Techniques de Classification

Dans cette partie, nous présentons les algorithmes de classification :

2.6.1 K plus proches voisins

L'algorithme des k plus proches voisins en anglais est l'appelé « K Nearest Neighbors (KNN) » est un algorithme de classification supervisé. Chaque observation de l'ensemble d'apprentissage est représentée par un point dans un espace à n dimensions ou n est le nombre de variables prédictives. Pour prédire la classe d'une observation, on cherche les k points les plus proches de cet exemple. La classe de la variable cible, est celle qui est la plus représentée parmi les k plus proches voisins. Il existe des variantes de l'algorithme ou on pondère les k observations en fonction de leur distance à l'exemple dont on veut classer les observations les plus éloignées de notre exemple seront considérées comme moins importantes. [10] Pour k = 3 la

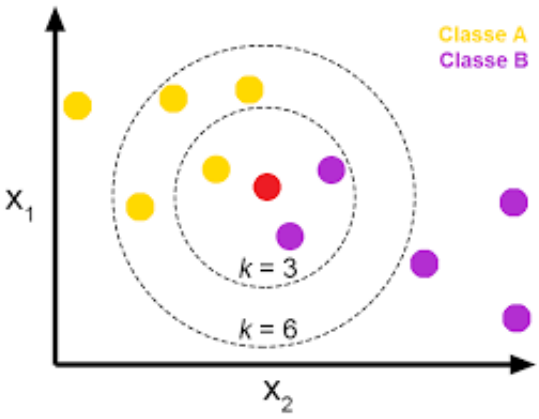

FIGURE 2.4 – Classification K-Nearest Neighbors (KNN)

classe majoritaire du point central est la classe B, mais si on change la valeur du voisinage k = 6 la classe majoritaire devient la classe A .

2.6.2 Régression Logistique

L'analyse de régression LR est souvent utilisée pour faire des prédictions, comprendre les variables indépendantes par rapport à la variable dépendante et étudier la forme de leur relation. Dans des circonstances limitées, l'analyse de régression peut être utilisée pour déduire la relation causale entre la variable indépendante et la variable dépendante. La régression est un algorithme robuste lorsqu'il s'agit de classer des ensembles de problèmes, et a une fonction logistique (fonction sigmoïde) au cœur de celui-ci [11]. Dans cet algorithme, les valeurs d'entrée sont combinées en fonction de coefficients ou de poids pour donner les valeurs de sortie/prédites.

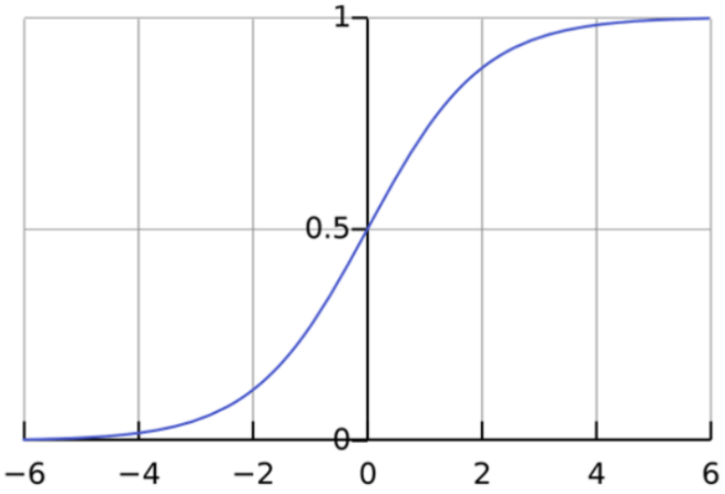

FIGURE 2.5 – Modèle la régression logistique.

2.6.3 Machine à vecteurs de supports

Machine à Vecteurs de Supports en anglais cela l'appelé « Support Vector Machine (SVM) » est l'un des algorithmes d'apprentissage supervisé les plus populaires, utilisé pour les problèmes de classification et de régression. Cependant, il est principalement utilisé pour les problèmes de classification dans l'IA. Le but de l'algorithme SVM est de créer la meilleure ligne ou limite de décision qui peut séparer l'espace à n dimensions en classes afin que nous puissions facilement mettre le nouveau point de données dans la bonne classe à l'avenir. [12]

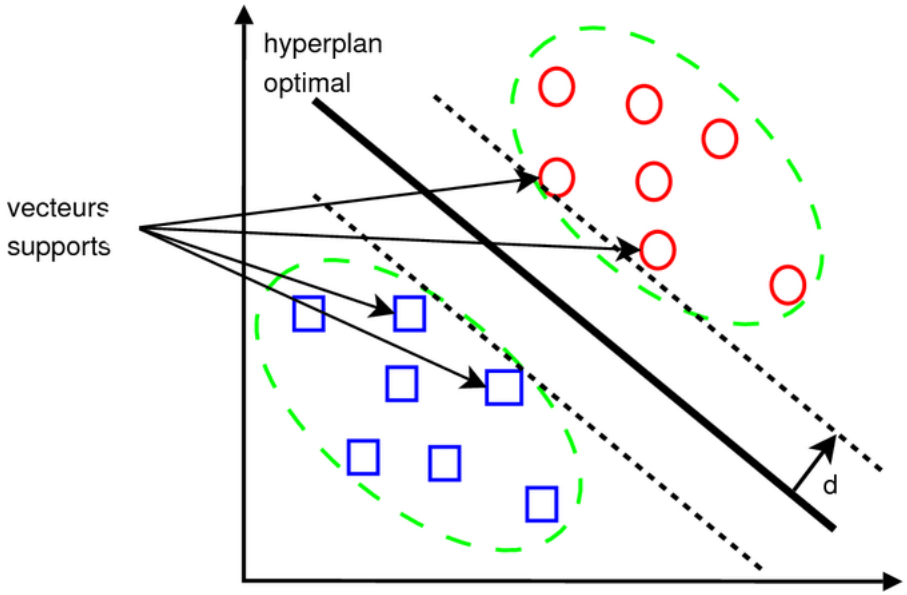

FIGURE 2.6 – Un diagramme d'un hyperplan avec des vecteurs de support dans un espace vectoriel à deux dimensions.

- La figure 2.6 montre un exemple simple d'un hyperplan avec des vecteurs de support. Cette technique peut être utilisée pour séparer deux classes de points dans un espace vectoriel, ce qui est utile pour des tâches d'apprentissage automatique telles que la classification et la régression. L'objectif d'un hyperplan avec des vecteurs de support est de trouver la meilleure séparation possible entre deux classes de points dans un espace vectoriel. Cela peut être utilisé pour des tâches de classification, telles que la reconnaissance d'image ou le traitement du langage naturel.

2.6.4 Apprentissage profond

Apprentissage profond en anglais est l'appelé « Deep Learning (DL) » est un type d'IA dérivé du Deep Learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.

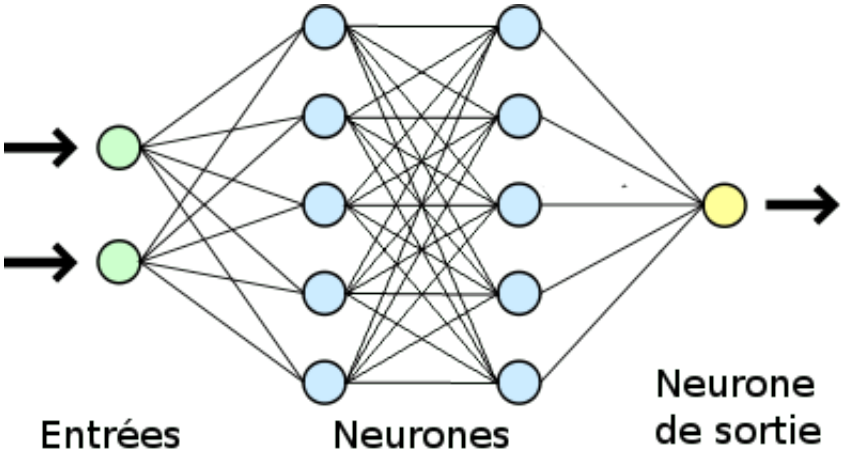

FIGURE 2.7 – Architecture Apprentissage profond.

La figure 2.7 fournie montre une architecture d'apprentissage profond composée de plusieurs couches empilées les unes sur les autres. Chaque couche est composée d'un certain nombre de neurones artificiels qui sont interconnectés. Les neurones sont responsables de traiter les informations et de les transmettre à la couche suivante.

- **Couches d'entrée** :Elles reçoivent les données brutes, comme des images ou du texte.
- **Couches cachées** :Elles traitent les données et extraient des caractéristiques de plus en plus complexes.
- **Couche de sortie** Elle produit la sortie finale, comme une classification ou une prédiction.

2.7 Comparaison entre techniques de classification

Techniques de Classification	Avantages	Inconvénients
KNN	- Simple à concevoir	- Sensible aux bruits. - Pour un nombre de variable prédictives très grands, le calcul de la distance devient très coûteux.
Régression Logistique	- Ses résultats sont faciles à interpréter.	- La phase d'apprentissage peut être longue car l'optimisation des coefficients est complexe. - Sa linéarité empêche la prise en compte des interactions entre les variables.
SVM	- Il permet de traiter des problèmes de classification non linéaire complexe. - Les SVM constituent une alternative aux réseaux de neurones car plus faciles à entraîner.	- Les SVM peuvent être lents à entraîner sur de grands ensembles de données. - Cela peut affecter la précision du modèle et le rendre moins fiable.
Apprentissage profond	- Adaptabilité à Divers Domaines. - Performances Exceptionnelles dans des Tâches Complexes.	- Interprétabilité Limitée. - Besoin d'Expertise Technique Élevée.

2.8 Analyse en Composantes Principales (ACP)

L'Analyse en Composantes Principales (ACP) est une méthode de réduction de dimensionnalité qui cherche à représenter les données dans un espace de dimension inférieure tout en conservant autant d'informations que possible. Mathématiquement, l'ACP peut être définie comme suit [13] :

1. **Centrage des données** : Tout d'abord, nous centrons les données en soustrayant la moyenne de chaque variable. Cela signifie que pour chaque variable, nous soustrayons

la moyenne de cette variable à chaque observation. Cela garantit que les données sont centrées autour de zéro.

La moyenne de chaque variable X_j est calculée comme :

$$\bar{X}_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}$$

où X_{ij} est la valeur de la variable j pour l'observation i.

Ensuite, nous soustrayons la moyenne de chaque variable à toutes les observations :

$$X^*_{ij} = X_{ij} - \bar{X}_j$$

où X^*_{ij} est la valeur centrée de la variable j pour l'observation i.

2. **Calcul de la matrice de covariance** : Ensuite, nous calculons la matrice de covariance des données centrées. La matrice de covariance S est une matrice symétrique $p \times p$ où chaque élément s_{jk} représente la covariance entre les variables X_j et X_k. La formule pour calculer la covariance est :
$$s_{jk} = \frac{1}{n}\sum_{i=1}^{n}(X^*_{ij} \times X^*_{ik})$$
3. **Calcul des vecteurs propres et valeurs propres** : Ensuite, nous calculons les vecteurs propres $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$ et les valeurs propres correspondantes $\lambda_1, \lambda_2, \ldots, \lambda_p$ de la matrice de covariance S. Les vecteurs propres représentent les directions principales de la variance des données, et les valeurs propres indiquent l'importance de ces directions.
4. **Sélection des composantes principales** : Nous sélectionnons les k premiers vecteurs propres qui correspondent aux k plus grandes valeurs propres. Ces vecteurs propres forment la base de l'espace des composantes principales.
5. **Projection des données** : Enfin, nous projetons les données centrées dans l'espace des composantes principales en utilisant les k vecteurs propres sélectionnés. La projection de l'observation i sur la j-ème composante principale est donnée par :

$$Y_{ij} = \mathbf{v}_j^T \times X_i^*$$

où Y_{ij} est la valeur projetée de l'observation i sur la j-ème composante principale, $\mathbf{v}_j^T$ est le vecteur propre j transposé, et X_i^* est l'observation i centrée.

Cette projection transforme les données d'un espace de grande dimension en un espace de dimension inférieure, où les variables (composantes principales) sont non corrélées (orthogonales) et ordonnées selon leur variance, les premières composantes principales capturant la variance maximale des données.

2.9 Conclusion

Dans ce chapitre, nous avons présenté les algorithmes d'apprentissage supervisé. Après avoir présenté les deux principaux types d'apprentissage automatique, une description détaillée de chaque méthode de classification a été fournie, en expliquant le principe de fonctionnement de chaque méthode ainsi que ses avantages et inconvénients. Le chapitre suivant présente les méthodes d'apprentissage profond qui ont été appliquées dans le domaine de la santé.

Chapitre 3: Deep Learning

3.1 Introduction

L'apprentissage profond est un nouveau domaine de recherche du ML, qui a été introduit dans le but de rapprocher le ML de son objectif principal : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

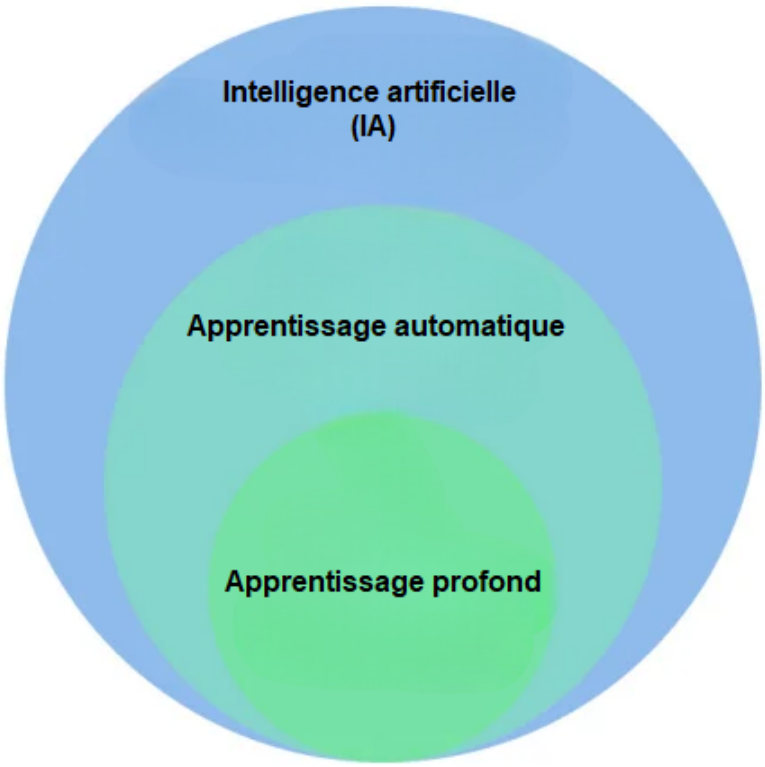

FIGURE 3.1 – La relation entre l'intelligence artificielle, le Machine L earning et le deep learning.

3.2 L'apprentissage profond

L'apprentissage profond est une technologie d'intelligence artificielle révolutionnaire qui permet aux machines d'apprendre à partir de données massives, ouvrant ainsi la voie à des avancées majeures. Malgré ses succès, il présente des défis, mais continue de repousser les frontières de l'intelligence artificielle.

3.2.1 Définition l'apprentissage profond

L'apprentissage profond en anglais cet appelée « Deep Learning (DL) » une technique d'apprentissage automatique révolutionnaire qui a permis de réaliser des progrès importants dans de nombreux domaines tels que la vision par ordinateur, la reconnaissance de la parole et la traduction automatique. Grâce à sa capacité à apprendre par elle-même et à résoudre des problèmes complexes, l'apprentissage profond est un outil puissant qui continuera à jouer un rôle majeur dans l'évolution de l'intelligence artificielle. [14]

3.2.2 Domaines d'application de l'apprentissage profond

Le Deep Learning utilisé dans plusieurs domaines différents tels que :

- **Traitement du langage naturel (NLP)** :Le DL est employé dans la compréhension du langage naturel, la traduction automatique, la génération de texte, la classification de texte, et l'analyse des sentiments.
- **Santé** :Le DL est appliqué dans le diagnostic médical, la segmentation d'images médicales, la prédiction de maladies, et la découverte de médicaments.
- **Vision par ordinateur** :Le DL est largement utilisé pour la reconnaissance d'images, la segmentation d'images, la détection d'objets, la classification d'images, et la génération d'images.
- **Robotique** :Le DL est employé dans la perception sensorielle des robots, la planification de mouvements, et l'interaction homme-robot.

3.3 Principes de fonctionnement

D'apprentissage profond sont formés sur la base des structures de données complexes qu'ils rencontrent. Ils construisent des modèles informatiques composés de plusieurs couches (couche d'entrée, couche cachée, couche de sortie) de traitement pour créer plusieurs niveaux d'abstraction pour représenter les données. [14]

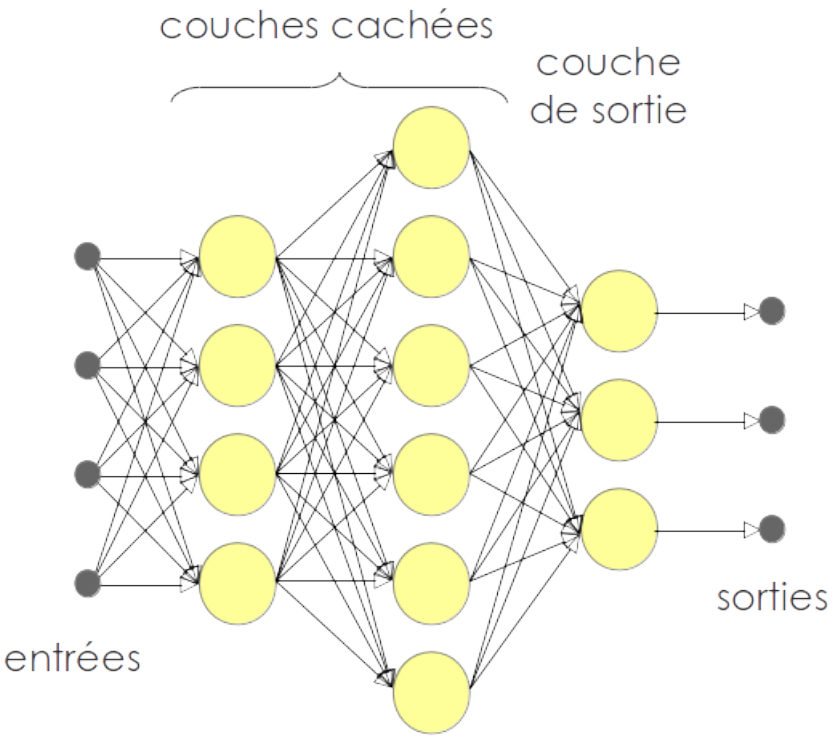

FIGURE 3.2 – les couches d'apprentissage profond.

3.3.1 Les types des couches dans d'apprentissage profond

- **Couche d'entrée** : composée d'un ensemble de neurones qui favoriseront la propagation des informations dans le réseau neuronal. Elle comprend un nombre de neurones généralement égal au nombre de caractéristiques constituant l'enregistrement en entrée (p. ex. une image, une transaction, etc.).
- **Couches intermédiaires** : servant à traiter l'information propagée dans le réseau de neurones pour capturer les caractéristiques de son apprentissage.
- **Couche de sortie** : formée d'un ensemble de neurones représentant les différentes

classes de résultat. Dans une problématique de classification, les classes sont les différentes possibilités que le résultat peut offrir [15].

3.4 Poids

Les poids des flèches sont utilisés pour accorder de l'importance à certaines fonctionnalités par rapport à d'autres, afin d'obtenir les résultats souhaités. La somme pondérée de tous les poids des flèches est calculée pour chaque neurone d'une couche cachée, et chacun de ces neurones exécute une fonction d'activation qui lui est propre.

3.5 Fonction d'activation

Une fonction d'activation, qui associe à chaque valeur agrégée une unique valeur de sortie dépendant du seuil.

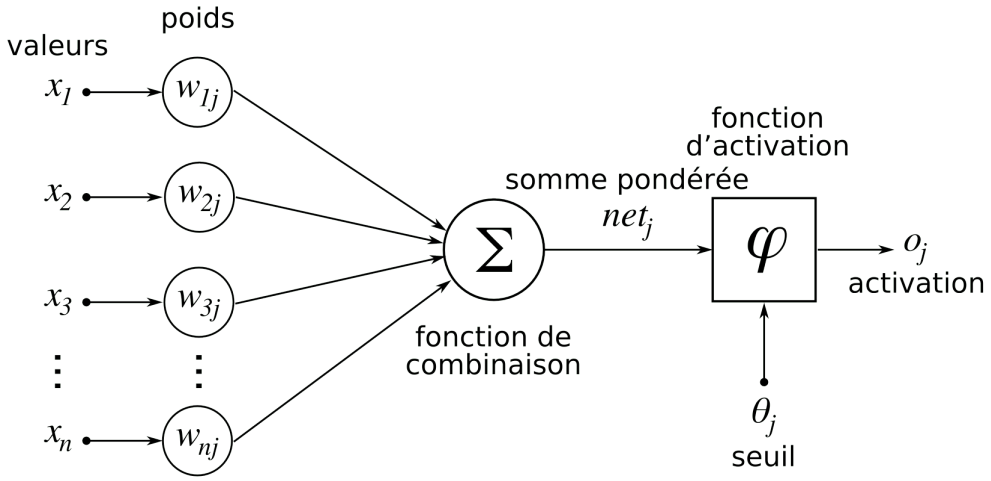

FIGURE 3.3 – Fonction d'activation.

3.5.1 Fonction Relu

La fonction Unité Linéaire Rectifiée en anglais, appelée Rectified Linear Unite (ReLU), est la fonction d'activation la plus couramment utilisée en Deep Learning. Elle est définie comme suit :

$$\text{ReLU}(x) = \max(x, 0)$$

Cette fonction renvoie x si x est supérieur à 0, et 0 sinon. Autrement dit, elle calcule le maximum entre x et 0 [16].

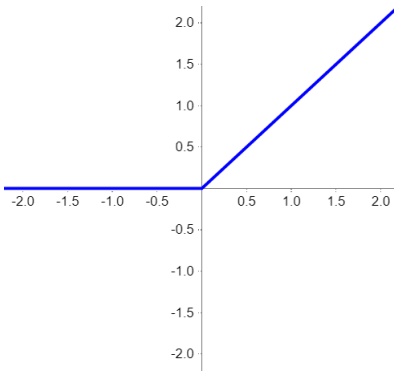

FIGURE 3.4 – La fonction Relu.

Cette fonction permet d'appliquer un filtre en sortie de couche. Elle laisse passer les valeurs positives dans les couches suivantes et bloque les valeurs négatives. Ce filtre permet alors au modèle de se concentrer uniquement sur certaines caractéristiques des données, les autres étant éliminées.

3.5.2 Fonction d'un softmax

La fonction Softmax est la fonction d'activation utilisée en dernière couche d'un réseau de neurones construit pour effectuer une tâche de classification multi-classes. Pour chaque sortie, Softmax donne un résultat entre 0 et 1. De plus, si l'on additionne ces sorties entre elles, le résultat donne 1.

La fonction Softmax est définie mathématiquement comme suit :

$$\text{Softmax}(x) = \frac{e^x}{\sum_i e^{x_i}}$$

La fonction Softmax, grâce à sa caractéristique de produire des résultats qui, additionnés, donnent 1, respecte les lois de probabilité. Elle est donc le noyau dur d'un réseau de neurones construit pour effectuer une tâche de classification multi-classes. [17]

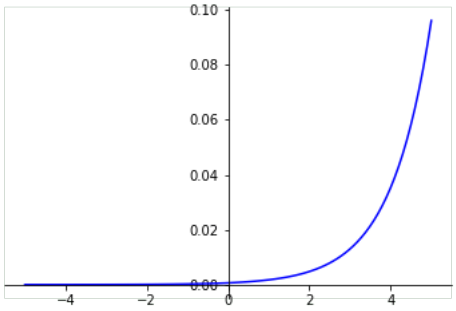

FIGURE 3.5 – La fonction d'un softmax.

3.5.3 Fonction sigmoïde

La fonction sigmoïde est la fonction d'activation utilisée en dernière couche d'un réseau de neurones construit pour effectuer une tâche de classification binaire. Elle donne une valeur entre 0 et 1 [18].

La fonction sigmoïde est définie mathématiquement comme suit :

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$

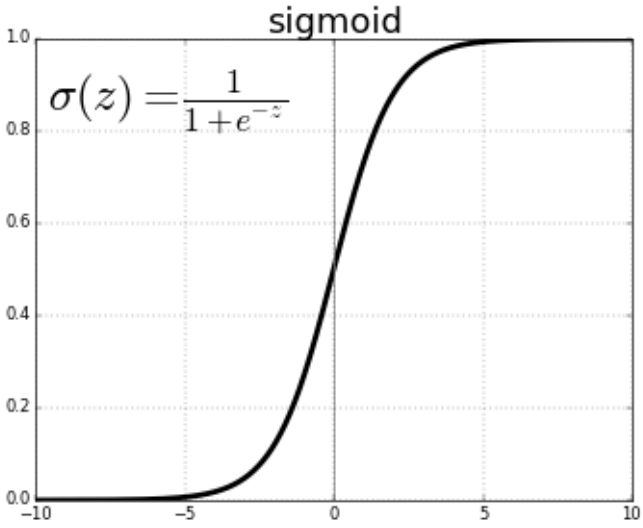

FIGURE 3.6 – La fonction sigmoïde.

Cette valeur peut être interprétée comme une probabilité. Dans une classification binaire, la fonction d'activation sigmoïde permet alors d'obtenir, pour une donnée, la probabilité d'appar-

tenir à une classe. Dans cet exemple 3.6, plus le résultat de la sigmoïde est proche de 1, plus le modèle considère que la critique est positive. Inversement, plus le résultat de la sigmoïde est proche de 0, plus le modèle considère que la critique est négative. La fonction d'activation sigmoïde permet donc d'obtenir un résultat ambivalent, donnant une indication sur deux classes à la fois.

3.6 Différents types de model deep Learning

On va présenter dans cette section les modèles de deep Learning utilisés dans notre proposition à savoir les ANN, CNN, RNN et LSTM.

3.6.1 Réseaux de neurones artificiel

En anglais cela l'appelé « Artificielle Neural Network (ANN) » , c'est l'architecture standard global de l'apprentissage profond, une structure constituée de suite successive de couches de nœuds et qui permet de définir une fonction de transformation non linéaire des vecteurs d'entrées (composés dans le cas de classification des mots pondérés de leur poids) en vecteur de catégories. La disposition des neurones dans le réseau ainsi que le nombre de couches utilisées ont une influence sur le résultat de classification. Comparés aux autres méthodes de classification par apprentissage supervisé, les réseaux de neurones artificiels sont habituellement utilisés pour des tâches de classification. Par analogie avec la biologie, ces unités sont appelées neurones formels [19] . Un neurone formel est caractérisé par :

- Le type des entrées et des sorties.
- Une fonction d'entrée.
- Une fonction de sortie.

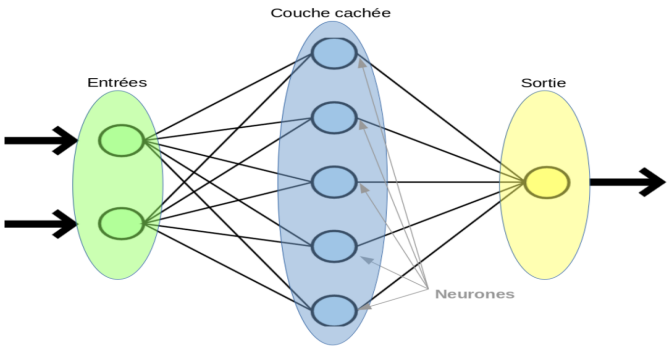

FIGURE 3.7 – Architecture de réseau de neurones artificiels.

3.6.2 Réseau de neurone convolutif

Le nom ‘Réseau de neurones à convolution’ en anglais cela l’appelé « Convolutional Neural Network (CNN) » indique que le réseau emploi une opération mathématique appelée la convolution. Les réseaux de convolution sont un type spécialisé de réseaux neuronaux qui utilisent la convolution à la place de la multiplication matricielle générale dans au moins une de leurs couches. Les CNN sont l’un des meilleurs algorithmes d’apprentissage pour faire l’opération de convolution qui aide à l’extraction de fonctionnalités utiles à partir de points de données corrélés localement. La sortie des noyaux convolutifs est ensuite affectée à l’unité de traitement non linéaire (fonction d’activation), qui non seulement aide à apprendre les abstractions, mais intègre également la non-linéarité dans l’espace des fonctionnalités. [14]

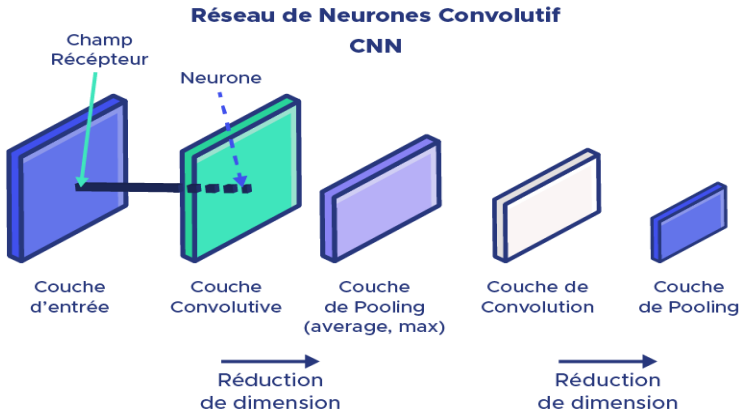

FIGURE 3.8 – Structure générale d’un réseau CNN.

3.6.3 Réseau de neurones récurrents

Un réseau de neurones récurrent (Recurrent Neural Network (RNN)) est un type de réseau de neurones artificiels principalement utilisé dans la reconnaissance vocale et le traitement automatique du langage naturel et la traduction automatique. Les RNN sont conçus de manière à reconnaître les caractéristiques séquentielles et les modèles d’utilisation des données requis pour prédire les scénarios faisant intervenir le contexte dans la prédiction d’un résultat . [14]

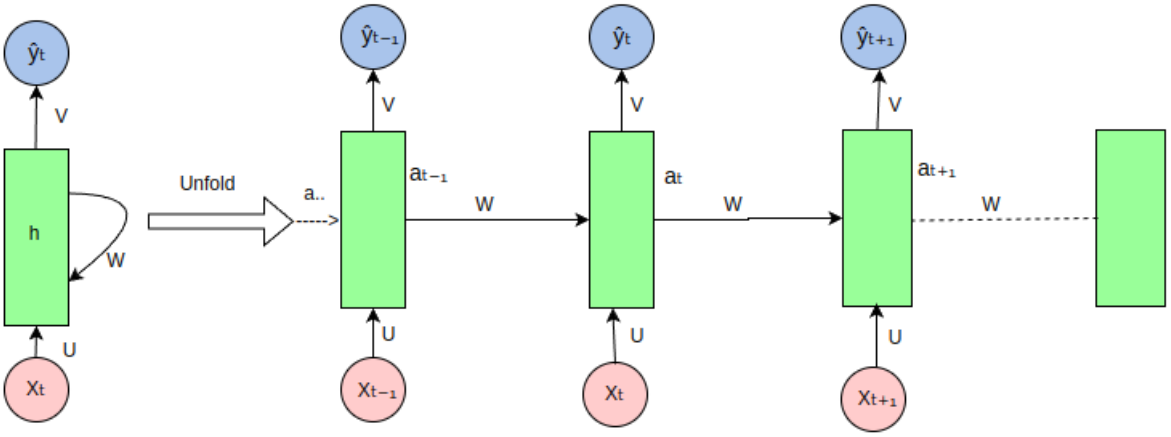

FIGURE 3.9 – Architecture de RNN.

3.6.4 Réseaux mémoire à long terme

Les réseaux mémoire à long terme (Long Short-Term Memory (LSTM), en anglais) sont des dérivés de RNN. Ils peuvent apprendre et mémoriser des dépendances sur une longue durée. Les LSTM conservent ainsi les informations mémorisées sur le long terme. Ils sont particulièrement utiles pour prédire des séries chronologiques, car ils se rappellent des entrées précédentes. Outre ce cas d'utilisation, les LSTM sont également utilisés pour composer des notes de musique et reconnaître des voix. [20]

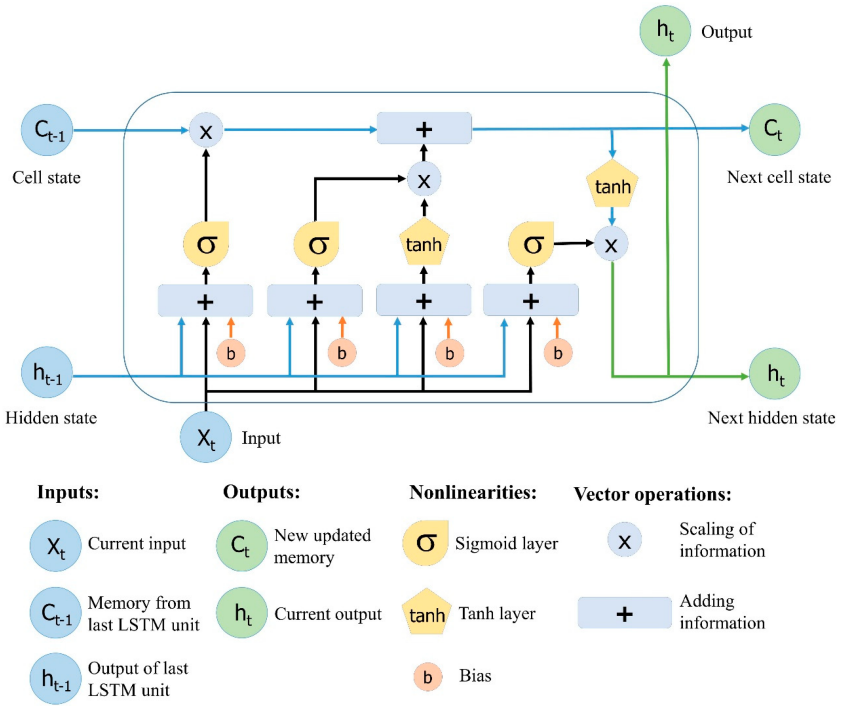

FIGURE 3.10 – Le module répétitif dans un LSTM.

3.7 Conclusion

Dans ce chapitre nous avons présenté l'apprentissage profond DL et ses différents types de réseaux de neurones artificiels : les réseaux de neurones artificiels (ANN), les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (RNN), y compris les réseaux à mémoire longue durée (LSTM).

Le chapitre suivant se concentre sur l'expérimentation de la construction de modèles de réseaux de neurones artificiels avec compression de données utilisant l'Analyse en Composantes Principales (ACP). Nous appliquerons également différents types d'algorithmes d'optimisation.

Chapitre 4: Méthodologie

4.1 Introduction

Ce chapitre présente nos expérimentations visant à atteindre nos objectifs de recherche. Nous y détaillons les outils utilisés tels que le langage de programmation, l'environnement de développement et le matériel, ainsi que le processus de classification utilisant différentes architectures d'apprentissage profond. Nous fournissons une analyse détaillée des résultats obtenus et de leur interprétation, suivie d'une discussion approfondie des conclusions. Enfin, nous clôturons ce chapitre par une synthèse des observations et une conclusion.

4.2 Notre Objectif

Cette étude vise à mettre en évidence l'importance de l'utilisation des données basé sur un étude statistique préalable dans la construction des modèles de prédiction supervisée. Nous menons une analyse comparative pour évaluer l'impact de la compression des données sur les algorithmes de classification profonde, ainsi que l'étendue de l'impact des algorithmes d'optimisation. Pour ce faire, nous avons utilisé un ensemble de données sur lesquels nous avons réalisé ces tests.

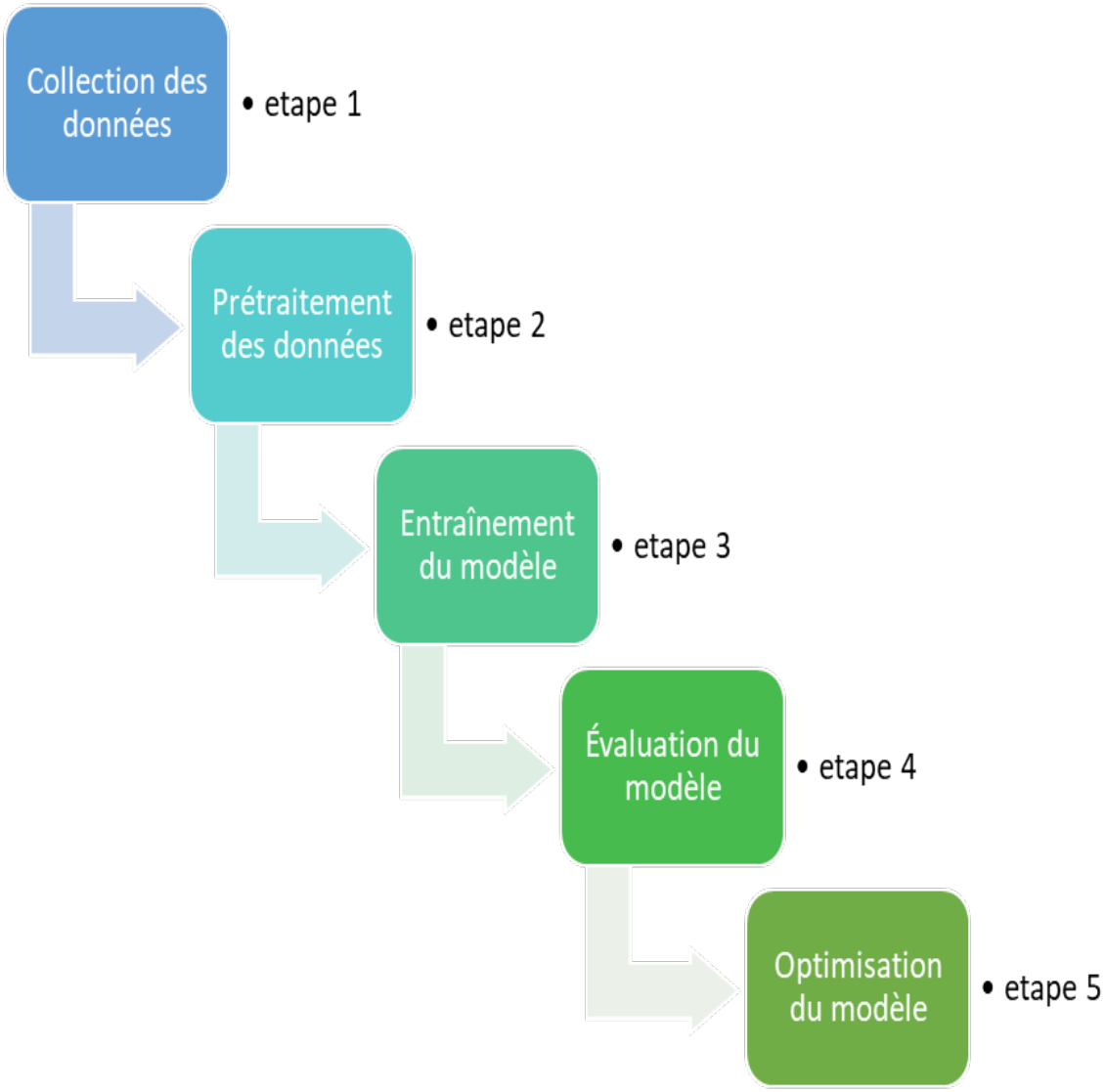

FIGURE 4.1 – Le cycle de vie d'un modèle d'apprentissage automatique

4.3 Prétraitement

Le prétraitement des données, également connu sous le nom de nettoyage des données, représente une phase cruciale dans l'analyse de données. Cette étape englobe la transformation des données brutes en données exploitables en éliminant les erreurs, les valeurs manquantes, les doublons, et en les standardisant dans un format approprié. Les étapes suivantes illustrent le processus de prétraitement des données que nous avons effectué :

1. **Collection des données** : Pour cette étude, nous avons exploré et choisi un jeu de données approprié sur la plateforme **Kaggle**, comprenant environ 5 000 enregistrements en tenant compte de sa pertinence par rapport aux objectifs spécifiques de notre cas d'étude. Après avoir récupéré le jeu de données, nous avons effectué des ajustements pour l'adapter aux exigences et aux paramètres définis dans le cadre de notre étude.
2. **Importation des données** : Pour l'importation de données dans l'environnement **Python**, il est courant de recourir à la bibliothèque **Pandas**, renommée pour sa capacité à gérer divers formats de fichiers de données. Plus précisément, la fonction **read_csv()** de pandas est souvent utilisée pour importer des données à partir de fichiers au format **CSV.** Cette fonctionnalité permet de charger les données contenues dans un fichier CSV dans une structure de données appelée **DataFrame** , offrant ainsi une manipulation et une analyse aisées des données dans Python.
3. **Répartition des données entraînement et test** : Dans notre étude, nous avons effectué une étape cruciale de division des données de l'ensemble d'entraînement et de test. Nous avons attribué 80% des données à l'ensemble d'entraînement, tandis que les 20% restants ont été réservés à l'ensemble de test. Cette répartition nous a permis de disposer d'un ensemble suffisamment large pour entraîner nos modèles tout en conservant un ensemble distinct pour évaluer leur performance. Cela garantit une évaluation fiable de la capacité de généralisation de nos modèles sur des données qu'elles n'ont pas encore vues.
4. **Sélectionner les colonnes nécessaires** : Pour cette étude, nous avons sélectionné un ensemble de données considérable en fonction de sa pertinence pour notre cas d'étude portant sur la prédiction des maladies basée sur les symptômes. Après avoir récupéré le jeu de données, nous avons utilisé la fonction **drop()** pour éliminer les paramètres (colonnes) non significatifs pour notre modèle, dans le but de sélectionner les symptômes

les plus significatifs pour la prédiction des maladies. Ces symptômes ont été choisis avec soin en tenant compte de leur corrélation avec les maladies cibles et de leur importance dans la précision des diagnostics prédictifs. Cette étape a été réalisée en collaboration avec des experts médicaux et des médecins afin d'assurer la pertinence clinique et la validité des symptômes sélectionnés.

5. **Suppression des documents contenant des valeurs nulles** : La tâche appliquée initialement pour vérifier la présence de valeurs nulles dans un DataFrame Python. Ensuite, nous avons utilisé la méthode **isna().sum()** de la bibliothèque Pandas, permettant de calculer le nombre de valeurs nulles par colonne ou par ligne dans le DataFrame. Après avoir identifié les colonnes ou les lignes contenant des valeurs nulles, nous avons procédé à la suppression de ces entrées à l'aide de la méthode **dropna()** de Pandas. Cette opération a permis d'éliminer les lignes ou les colonnes contenant des valeurs nulles, assurant ainsi la qualité des données utilisées dans l'analyse subséquente.
6. **Compression des colonnes (la méthode de ACP)** : Dans ce projet, nous avons utilisé la méthode ACP (voir 2.8) de la bibliothèque **scikit-learn**, avec l'argument **n_components=20**. Cette étape de réduction de dimensionnalité avec ACP _ncomponents=20 a pour l'objectif de comprimer les données vers un espace à deux dimensions. En fixant le nombre de composantes principales à 2, nous avons cherché à réduire la dimensionnalité du jeu de données tout en préservant au mieux sa structure et ses informations essentielles. Cette approche permet une représentation plus concise des données, ce qui facilite la visualisation, l'analyse et l'interprétation des résultats.
7. **Encodage des caractéristiques**
   - **Encodage multi-étiquette** : nous avons réalisé un encodage multi-étiquette sur les variables relatives aux symptômes dans le jeu de données. Cela signifie que chaque symptôme présent dans les données a été transformé en une série de variables binaires, où chaque variable représente la présence ou l'absence d'un symptôme spécifique. Cette approche permet de traiter les symptômes comme vecteur des caractéristiques.
   - **Encodage de label** : En ce qui concerne la variable cible, qui représente le pronostic de la maladie, nous avons utilisé un encodage de label en utilisant la fonction **LabelEncoder()** de la bibliothèque **scikit-learn**. Cela signifie que les différentes classes de pronostic de maladie ont été encodées avec des labels numériques uniques. Cette

méthode est couramment utilisée pour traiter les variables cibles dans les problèmes de classification, où chaque classe est représentée par un nombre entier unique, facilitant ainsi l'entraînement des modèles d'apprentissage profond. Cette étape appelé la normalisation de label .

4.4 Consultation médicale

4.4.1 But du Consultation

En planifiant des rendez-vous avec plusieurs médecins pour collecter des informations sur les symptômes, nous pouvons créer un modèle de classification des symptômes à l'aide de techniques d'apprentissage profond. Cela permet une évaluation objective et automatisée des symptômes, ce qui peut être utile pour le diagnostic et la prise en charge médicale.

4.4.2 Etapes de Consultation

1. **Prendre rendez-vous avec plusieurs médecins** :Prendre rendez-vous avec plusieurs médecins spécialisés dans différents domaines de la médecine pour discuter des symptômes. Le but est d'obtenir une évaluation complète et variée des symptômes.
2. **Rassembler des informations sur les symptômes** :À ce point, nous avons mené des discussions sur le concept de chaque symptôme, ainsi que sur sa relation avec les maladies qui lui sont associées et les raisons de son apparition.
3. **Classification des symptômes** :Nous avons classé les symptômes en catégories spécifiques. Cela implique d'identifier les similitudes et les différences entre les symptômes signalés par différents médecins et de les regrouper en catégories ou modèles. Par exemple : si vous ressentez des maux de tête, des nausées et des étourdissements, les médecins voudront peut-être déterminer si ces symptômes sont liés à des problèmes neurologiques, gastro-intestinaux ou autres.
4. **Processus électoral** :Après avoir discuté avec les médecins, nous appliquons le processus d'élection (voir le section 5.11) pour atteindre les résultats finaux et déterminer les données sélectionnées pour l'étape suivante.
5. **Utilisation du Deep Learning** :Une fois que nous avons recueilli les informations détaillées sur symptômes auprès de plusieurs médecins, nous avons utiliser des techniques

de deep learning pour analyser de grandes quantités de données et extraire des modèles complexes.

4.5 Représentation des Données

4.5.1 La Représentation en Vecteur de Caractéristiques Binaires

La représentation de vecteur de caractéristiques binaires est utilisée pour représenter les symptômes des patients. Chaque symptôme est représenté par une variable binaire, où 1 indique la présence du symptôme et 0 indique son absence. Par exemple, si nous avons un ensemble de symptômes comprenant "fièvre", "toux", "maux de tête", "fatigue", et qu'un patient présente seulement la fièvre et la toux, sa représentation en vecteur de caractéristiques binaires serait [1, 1, 0, 0]. Cette représentation permet de traiter les symptômes comme des variables catégorielles dans nos modèles d'apprentissage profond, facilitant ainsi l'analyse et la prédiction des diagnostics de maladies en fonction des symptômes présentés par les patients.

4.6 Méthodologie de Travail

Dans le cadre de notre étude, nous avons entrepris une évaluation comparative des performances de modèles d'apprentissage profond, notamment des réseaux de neurones artificiels ANN, avec et sans l'utilisation de l'Analyse en Composantes Principales ACP, sur un ensemble de données composé de 4962 enregistrements. Notre objectif principal était de déterminer le modèle offrant les meilleures performances en termes de précision, de rappel, de score F1 et d'exactitude. Dans la phase de prétraitement des données, différentes techniques ont été appliquées. Pour coder les variables liées aux symptômes, nous avons opté pour la représentation en vecteur de caractéristiques binaires, tandis que les variables cibles ont été encodées en utilisant l'encodage de label. De plus, notre processus de prétraitement comprenait une phase de sélection des caractéristiques significatives et l'élimination des valeurs nulles, les données étant nettoyées selon cette méthode avant d'être utilisées pour l'entraînement et le test des modèles. Les résultats obtenus ont ensuite été analysés et comparés afin de déterminer le modèle présentant les performances les plus prometteuses pour la tâche de classification considérée.

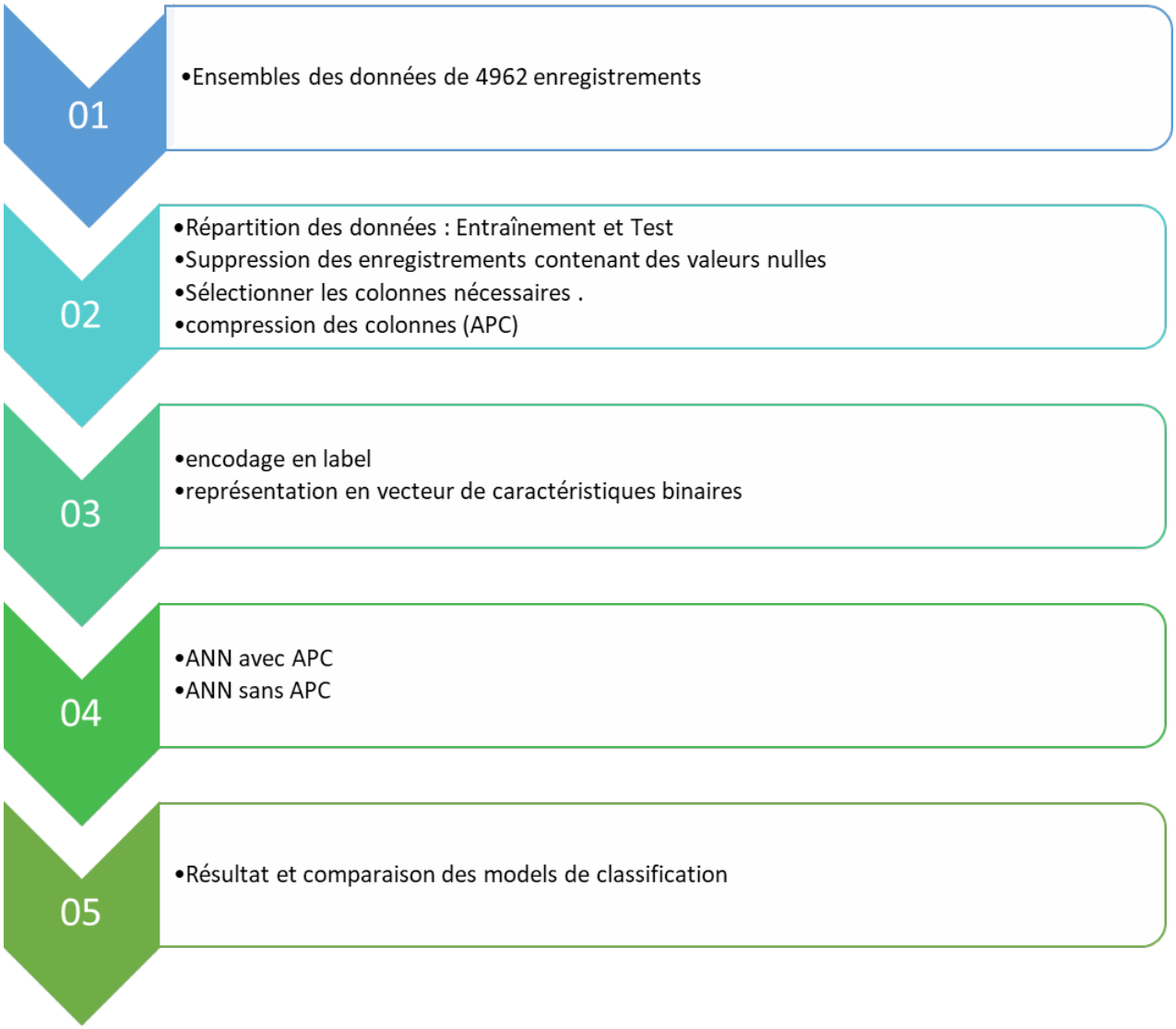

FIGURE 4.2 – Processus de classification en apprentissage automatique

4.7 Evaluation des modèles de classification

Dans notre étude, nous avons sélectionné trois mesures de performance essentielles pour évaluer les modèles : la précision, le rappel et l'exactitude (Accuracy en anglais). Ces mesures ont été choisies pour fournir une évaluation holistique des performances des modèles dans la classification des données médicales. En plus de ces métriques, nous avons également pris en compte le score F1

		réelle		
		Positive	Negative	Total
prédite	Positive	VP	FP	$VP+FP$
	Negative	FN	VN	$FN+VN$
	Total	$VP+FN$	$FP+VN$	

TABLE 4.1 – La matrice de confusion

4.7.1 Rappel

Le rappel **R** est une métrique qui mesure à quelle fréquence un modèle identifie correctement les instances positives (vrais positifs) parmi tous les échantillons positifs réels dans l'ensemble de données.

$$R = \frac{VP}{VP+FN}$$

4.7.2 Précision

La précision **P** mesure la proportion d'échantillons correctement identifiés comme positifs parmi tous les échantillons identifiés comme positifs par le modèle.

$$P = \frac{VP}{VP+FP}$$

4.7.3 Exactitude

L'exactitude **E** est une mesure qui évalue la proportion totale d'échantillons correctement classés parmi tous les échantillons dans l'ensemble de données. C'est-à-dire qu'elle mesure la capacité globale du modèle à prédire correctement les classes de tous les échantillons, qu'ils soient positifs ou négatifs.

$$E = \frac{VP+VN}{VP+VN+FP+FN}$$

4.7.4 Score f1

Le score **F1** est une métrique de performance qui combine à la fois la précision et le rappel d'un modèle en une seule valeur. Il offre une mesure globale de la capacité du modèle à classer

correctement les échantillons positifs tout en minimisant à la fois les faux positifs et les faux négatifs.

$$F1 = \frac{2 * Prcision * Rappel}{Prcision + Rappel} = \frac{2 * VP}{2 * VP + FP + FN}$$

4.7.5 Temps d'entraînement

Le temps d'entraînement **T** du modèle fait référence à la durée nécessaire pour entraîner un modèle d'apprentissage automatique sur un ensemble de données donné.

$$\mathrm{T} = t_{\text{fin}} - t_{\text{début}}$$

Où :

- $t_{\text{début}}$ est le moment où l'entraînement du modèle commence.
- t_{fin} est le moment où l'entraînement du modèle se termine.

4.8 Optimisation des modeles de classification

4.8.1 Estimation de Moment Adaptatif

Ce sont les étapes principales de l'algorithme ADAM pour mettre à jour les poids pendant le processus d'entraînement. L'algorithme Adam combine la prise en compte des moments 1 et 2 ainsi que le taux d'apprentissage pour mettre à jour les poids 3.4, ce qui lui permet d'améliorer la stabilité et la vitesse du processus d'entrainement dans de nombreux cas. Pour accéder aux poids et aux biais de notre modèle, on peut utiliser les fonctions **model.weights** et **model.bias.numpy()**.

1. **Initialisation** :Les poids du modèle sont initialisés de manière aléatoire.
2. **Calcul des Gradients** :Les gradients des poids du modèle par rapport à la fonction de coût sont calculés à l'aide de la technique de la rétropropagation du gradient.
3. **Mise à jour des moments 1 et 2** :Les moments 1 et 2 sont mis à jour en utilisant les formules suivantes :

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \text{gradient}_t$$

$$V_t = \beta_2 \cdot V_{t-1} + (1 - \beta_2) \cdot \text{gradient}_t^2$$

(t est le temps actuel et β_1 et β_2 sont des paramètres qui contrôlent l'importance des moments.)

4. **Correction des biais 1 et 2** :Les biais 1 et 2 sont corrigés pour compenser les biais initiaux :

$$m_{t_{\text{corr}}} = m_t(1-\beta_1^t)$$

$$V_{t_{\text{corr}}} = V_t(1-\beta_2^t)$$

5. **Mise à jour des poids** :Les poids sont mis à jour en utilisant la formule suivante :

$$W_{t+1} = W_t - \left(lr \cdot \left(\sqrt{v_{t_{\text{corr}}} + \varepsilon}\right) \cdot m_{t_{\text{corr}}}\right)$$

(lr est le taux d'apprentissage et ε est une petite valeur pour éviter la division par zéro.)

6. **Répétition** :Les étapes 2 à 5 sont répétées jusqu'à ce qu'une condition d'arrêt spécifiée soit remplie.

4.8.2 Descente de gradient stochastique

Ce sont les principales étapes de l'algorithme SGD pour mettre à jour les poids pendant le processus d'entraînement. Les poids sont mis à jour en fonction du gradient calculé et du taux d'apprentissage. Les étapes sont appliquées à chaque échantillon ou lot de données dans le cas de l'entraînement par mini-lots (mini-batch training) ou à chaque donnée dans le cas de l'entraînement par lots complets (batch training). Le processus est répété jusqu'à ce que les conditions spécifiées, telles que le nombre de cycles ou l'atteinte de la précision souhaitée, soient remplies

1. **Initialisation** :Les poids du modèle sont initialisés de manière aléatoire.
2. **Calcul des gradients** :Les gradients des poids du modèle par rapport à la fonction de coût sont calculés en utilisant la technique de la rétro propagation du gradient.
3. **Mise à Jour des Poids** :Les poids sont mis à jour en utilisant la formule suivante :

$$W_{t+1} = W_t - lr \cdot \text{gradient}(t)$$

(où W_t représente les poids actuels, W_{t+1} est le poids mis à jour, et lr est le taux d'apprentissage.)

4. **Répétition** :Les étapes 2 et 3 (calcul des gradiant + mise à jour de poids) sont répétées pour chaque donnée d'entraînement ou pour un nombre spécifié de cycles (epochs) jusqu'à ce qu'un critère d'arrêt spécifié soit atteint

4.9 Pertes des modèle de classification

Dans les modèles de classification, l'optimiseur utilise les informations de la fonction de perte pour ajuster les poids du modèle. Son objectif est de réduire l'erreur de prédiction au fil du temps. En somme, la fonction de perte fournit une mesure de performance que l'optimiseur utilise pour guider le processus d'optimisation vers la solution optimale.

4.9.1 Entropie Croisée Catégorielle

La fonction de perte Categorical cross Entropy (CCE) est largement utilisée dans les problèmes de classification où les étiquettes sont fournies sous forme de vecteurs one-hot. Elle mesure la différence entre les prédictions du modèle et les étiquettes réelles associées aux données. La formule mathématique de la fonction de perte entropie croisée catégorielle est la suivante :

$$\text{CCE}(y,t) = -\sum_i t_i \cdot \log(y_i)$$

- **CCE** représente l'entropie croisée catégorielle.
- y est la sortie (prédiction) produite par le modèle, un vecteur de probabilités sur les classes.
- t est la valeur réelle cible (étiquette), un vecteur one-hot où t_i est 0 ou 1.
- log représente le logarithme naturel.
- La somme est effectuée sur toutes les classes i.

Cette fonction de perte mesure la dissimilarité entre la distribution de probabilité prédite par le modèle (y) et la distribution de probabilité réelle (t) pour chaque exemple dans l'ensemble de données. Elle est souvent utilisée dans les tâches de classification avec plusieurs classes.

Chapitre 5: Implémentation et résultats

5.1 Introduction

Au cours de ce chapitre, nous expliquerons les étapes d'implémentation pour construire et entraîner les différents modèles de réseaux de neurones artificiels spécifiques, en plus d'étudier les résultats de performance de chaque modèle et de faire une comparaison entre eux.
Enfin,la présentation de l'application mobile liée avec les modèles ANN développé.

5.2 Outils matériels et logiciels

5.2.1 Configuration matérielle

Ce travail a été réalisé sur deux machines ordinateur (PC) présentant les caractéristiques suivantes :

Composant	**Machine 1**	**Machine 2**
CPU	Intel Core i3-6006U @ 2.00GHz (1.99GHz)	Intel Core i5-8250U @ 1.60GHz (1.80GHz)
GPU	Intel HD Graphics 5500	Intel UHD Graphics 620
Système d'exploitation	Windows 10 Pro 64-bit	Windows 10 Pro 64-bit
RAM	6 Go	8 Go

TABLE 5.1 – Spécifications des machines

5.2.2 Environnement logiciel

Pour le développement de Nos modèles, nous avons opté pour les outils suivants :

- **Python** s'impose comme le langage de programmation open source le plus utilisé au sein de la communauté informatique. Sa popularité découle de sa richesse en outils et bibliothèques dédiés à une variété d'applications, allant de la gestion d'infrastructures à l'analyse de données en passant par le développement logiciel. Cette préférence pour Python repose sur plusieurs atouts intrinsèques. En effet, sa syntaxe concise et expressive libère les développeurs des contraintes syntaxiques souvent rencontrées dans les langages plus anciens, favorisant ainsi un processus de développement plus rapide et efficace. De surcroît, Python offre une vaste panoplie d'outils et de bibliothèques spécifiquement dédiés à l'apprentissage profond, conférant aux développeurs des moyens considérables pour la conception et l'entraînement de modèles avec une efficacité accrue.
- **Flutter** est un framework open-source développé par Google qui permet de créer des applications nativement compilées pour mobile (iOS et Android), web et desktop à partir d'une base de code unique.
- **Anaconda** est une distribution de logiciels open source largement adoptée dans le domaine de l'informatique scientifique, notamment en langage Python. Cette plateforme offre un environnement complet pour le développement et l'exécution de programmes Python, comprenant un gestionnaire de paquets, un environnement de développement intégré (IDE) et une multitude de packages scientifiques préinstallés. Sa popularité découle principalement de sa polyvalence et de sa richesse fonctionnelle, qui en font un choix privilégié pour les professionnels travaillant dans les domaines de la science des données, de l'apprentissage automatique et de l'analyse statistique. En fournissant une plateforme puissante et conviviale, Anaconda facilite la mise en œuvre de projets de programmation complexes et la manipulation de données à grande échelle.
- **Jupyter Notebook** est un environnement de développement interactif qui permet aux utilisateurs de créer et de partager des documents contenant du code exécutable, des visualisations, des textes narratifs et des médias. Il est particulièrement populaire dans les domaines de la science des données, de la recherche académique et de l'éducation en raison de sa facilité d'utilisation et de sa polyvalence.
- **Google Colab** également connu sous le nom de Colaboratory, est un environnement de notebook basé sur le cloud offrant aux utilisateurs la possibilité d'écrire et d'exécuter du code Python dans leur navigateur sans nécessiter d'installation locale. Populaire dans

les domaines de l'apprentissage automatique, de l'analyse de données et de la recherche, Colab permet la création de notebooks contenant du code exécutable, des visualisations et des textes narratifs, ainsi que la collaboration en temps réel avec d'autres utilisateurs. Intégré à Google Cloud Platform et Google Drive, Colab offre un accès gratuit à des GPU (Unité de traitement graphique) et des TPUs (Unité de traitement tensoriel) pour des calculs intensifs, faisant de lui un outil puissant et polyvalent pour le développement et l'exécution de projets Python dans le cloud.

- **GitHub** est une plateforme de développement collaboratif basée sur Git, offrant aux développeurs un espace pour héberger, gérer et collaborer sur des projets logiciels. Elle permet le suivi des modifications apportées au code source, la gestion des demandes de tirage, le suivi des problèmes et bugs, ainsi que d'autres fonctionnalités liées au contrôle de version. GitHub est largement utilisé dans la communauté open source et par de nombreuses entreprises pour faciliter le développement de logiciels en équipe.

5.3 Base de données utilisée

Il s'agit d'un ensemble d'enregistrements liés au diagnostic de maladies associées à plusieurs symptômes pour un groupe d'individus réels. La base de données est un type de fichier qui contient un ensemble de lignes et un ensemble de colonnes. Les colonnes représentent l'ensemble des symptômes et les lignes représentent l'ensemble des enregistrements liés à chaque patient. La base de données contient 133 colonnes. Là où 132 colonnes représentent les symptômes et la dernière colonne représente la maladie, elle contient également 4920 enregistrements pour 42 maladies différentes.

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scu
3969	0	0	0	0	0	0	0	0	0	0	...	0	
3970	0	0	0	0	0	0	1	0	0	0	...	0	
3971	0	0	0	0	0	0	0	0	0	0	...	0	
3972	0	0	0	0	0	0	0	0	0	0	...	0	
3973	0	1	0	0	0	0	0	0	0	0	...	1	

5 rows × 133 columns

FIGURE 5.1 – Un aperçu sur la base de données utilisée

5.3.1 Ensemble des colonnes

Grâce à la base de données approuvée, la base de données contient 132 colonnes, où les colonnes représentent l'ensemble des symptômes médicaux possibles, qui représentent les « caractéristiques » dans la construction du modèle. Ils sont les suivants, selon ce qui est mentionné dans la base de données :

itching	skin_rash	nodal_skin_eruptions	continuous_sneezing
shivering	chills	joint_pain	...

TABLE 5.2 – Liste de symptômes

5.3.2 Ensemble des classes

La colonne n° 133 représente le diagnostic des symptômes du patient, appelé « pronostic ». Elle contient au total 42 maladies ou « classes» différentes qui sont les suivantes :

Fungal infection	Allergy	GERD	Chronic cholestasis
Drug Reaction	Peptic ulcer disease	AIDS	...

TABLE 5.3 – Liste de maladies

5.4 Prétraitement des données

Les données sont sujettes à un certain nombre de modifications et de changements proportionnés à la nature du modèle à réaliser. Les modifications sont apportées selon les directives des experts et des médecins, voir section 4.4

5.4.1 Répartition des données

Cette étape se fait en important les données, en les collectant en un seul groupe, puis en les divisant en 3969 (80%) lignes pour l'entraînement et 993 (20%) pour les tests :

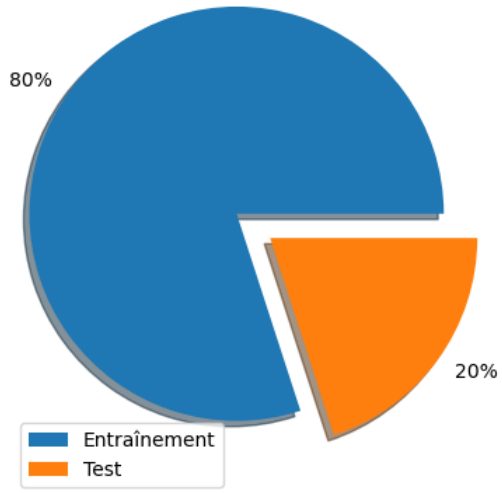

FIGURE 5.2 – La distribution des données d'entraînement et de test

Le processus a été complété selon l'algorithme suivant :

Algorithm 1: Lecture des données et division en ensembles d'entraînement et de test

Data: Train_dataset et Test_dataset

Result: Ensembles de données d'entraînement et de test

```
Importer pandas;
Train_dataset ← pd.read_csv('symptoms/Training.csv');
Test_dataset ← pd.read_csv('symptoms/Testing.csv');
full_data ← pd.concat([Train_dataset, Test_dataset], ignore_index=True);
train_size ← int(0.8 * len(full_data));
train_data ← full_data[ :train_size];
test_data ← full_data[train_size :];
```

5.4.2 Nettoyage des données

Lors de cette étape, nous vérifions qu'il n'y a pas de valeurs nulles dans la base de données, et si elles sont présentes, nous les supprimerons. Lors du traitement des données, nous avons constaté que la colonne 134 a toutes les valeurs nulles, et donc nous supprimons la colonne entière.

Algorithm 2: Suppression d'une colonne inutile et vérification des valeurs manquantes

Data: train_df et test_df
Result: DataFrames d'entraînement et de test sans la colonne inutile et vérification des valeurs manquantes

```
train_df ← train_df.drop(['Unnamed : 133'], axis=1);
test_df ← test_df.drop(['Unnamed : 133'], axis=1);
print(test_df.isna().sum());
print(train_df.isna().sum());
```

5.4.3 Sélection des colonnes

Cela a été fait en utilisant les conseils d'experts (voir section 4.4), concernant les critères de sélection des symptômes cliniques appropriés, tout en supprimant les colonnes non pertinentes. Suppression également de la colonne "prognosis" de la liste des caractéristiques en tant que colonne de notes. Les colonnes suivantes ont été supprimé :
['altered_sensorium', 'blackheads', 'brittle_nails', 'bruising', 'coma', 'dischromic _patches', 'drying_and_tingling_lips', 'fluid_overload', 'hip_joint_pain', 'inflammatory_nails', 'loss_of_balance', 'muscle_weakness', 'pain_during_bowel_movements', 'painful_walking', 'passage_of_gases', 'phlegm', 'receiving_blood_transfusion', 'receiving_unsterile_injections', 'red_sore_around_nose', 'rusty_sputum', 'scurring', 'silver_like_dusting', 'small_dents_in_nails', 'spinning_movements', 'unsteadiness', 'visual_disturbances', 'weakness_in_limbs', 'prognosis'].

5.4.4 Traitement des classes

Lors de ce processus, les catégories de la colonne « prognosis» sont codées en chiffres à l'aide du l'algorithm 3 suivant :

Algorithm 3: Encodage des labels de classe

Data: train_df et test_df
Result: Labels encodés pour les ensembles d'entraînement et de test

```
le ← LabelEncoder();
Catégories ← le.fit_transform(train_df['prognosis']);
y_train ← pd.DataFrame(Catégories);
Catégories_test ← le.fit_transform(test_df['prognosis']);
y_test ← pd.DataFrame(Catégories_test);
```

et selon le dictionnaire 5.4 suivent :

Label	Maladie
0	(vertigo) Paroymsal Positional Vertigo
1	AIDS
2	Acne
3	Alcoholic hepatitis
4	Allergy
5	Arthritis
6	Bronchial Asthma
7	Cervical spondylosis
8	Chicken pox
9	Chronic cholestasis
10	Common Cold
11	Dengue
12	Diabetes
13	Dimorphic hemmorhoids(piles)
14	Drug Reaction
15	Fungal infection
16	GERD
17	Gastroenteritis
18	Heart attack
19	Hepatitis B
20	Hepatitis C
21	Hepatitis D
22	Hepatitis E
23	Hypertension
24	Hyperthyroidism
25	Hypoglycemia
26	Hypothyroidism
27	Impetigo
28	Jaundice
29	Malaria
30	Migraine
31	Osteoarthristis
32	Paralysis (brain hemorrhage)
33	Peptic ulcer disease
34	Pneumonia
35	Psoriasis
36	Tuberculosis
37	Typhoid
38	Urinary tract infection
39	Varicose veins
40	Hepatitis A

TABLE 5.4 – Correspondance entre les labels encodés et les maladies

5.5 Réduction des dimensions (ACP)

Lors du modèle dans lequel la réduction de dimension a été appliquée, le code suivant a été utilisé pour compresser les colonnes de 105 à 20 colonnes afin de réduire la période d'apprentissage du modèle et de déterminer ses effets.

Algorithm 4: Réduction de dimension avec l'analyse en composantes principales (PCA)

Data: x_train et x_test
Result: Données transformées après PCA pour les ensembles d'entraînement et de test

```
pca ← PCA(n_components=20, svd_solver='full');
pca.fit(x_train);
x_train ← pca.transform(x_train);
pca.fit(x_test);
x_test ← pca.transform(x_test);
```

5.6 Construction des models

Lors de la construction des modèles, le modèle ANN Séquentiel à 4 couches a été adopté selon le code (5) suivant :

Algorithm 5: Définition du modèle séquentiel avec Keras

Result: Modèle séquentiel défini

```
model ← Sequential([;
Dense(units=15, activation='relu'),;
Dense(units=100, activation='relu'),;
Dense(units=50, activation='relu'),;
Dense(units=42, activation='softmax');
]);
```

et les paramètres de chaque architecture est la suivante 5.5 :

Modèle	Optimisation	ACP	Paramètres
1	ADAM	Oui	9,107
2	ADAM	Non	10,382
3	SGD	Oui	9,107
4	SGD	Non	10,382

TABLE 5.5 – Tableau des paramètres des modèles

5.7 Représentation des poids

Lors de l'entraînement des modèles, les poids sont constamment modifiés pour obtenir les meilleurs résultats. Les poids sont affichés selon l'algorithme 6 suivant :

Algorithm 6: Chargement des modèles et affichage des biais et poids

Data: Liste des noms de fichiers des modèles
Result: Affichage des biais et poids pour chaque couche de chaque modèle

```
models ← ['sgd_best_model.h5', 'sgd_best_pca_model.h5', 'adam_best_model.h5',
  'adam_best_pca_model.h5'];
for modl in models do
    model ← load_model(modl);
    print("Modèle : ", modl);
    print("Nombre de couches : ", len(model.layers));
    for x in range(len(model.layers)) do
        print("Couche ", x, " biais : ", model.layers[x].bias.numpy());
    end
    for x in range(len(model.layers)) do
        print("Couche ", x, " poids : ", model.layers[x].weights);
    end
end
```

5.8 Résultats et évaluations des différents modèles.

Les résultats de classification pour les différents modèles sont présentés comme suit :

5.8.1 Matrice de confusion

Les résultats des matrices de confusion pour chaque modèle, les valeurs sont présentées dans les figures suivantes :

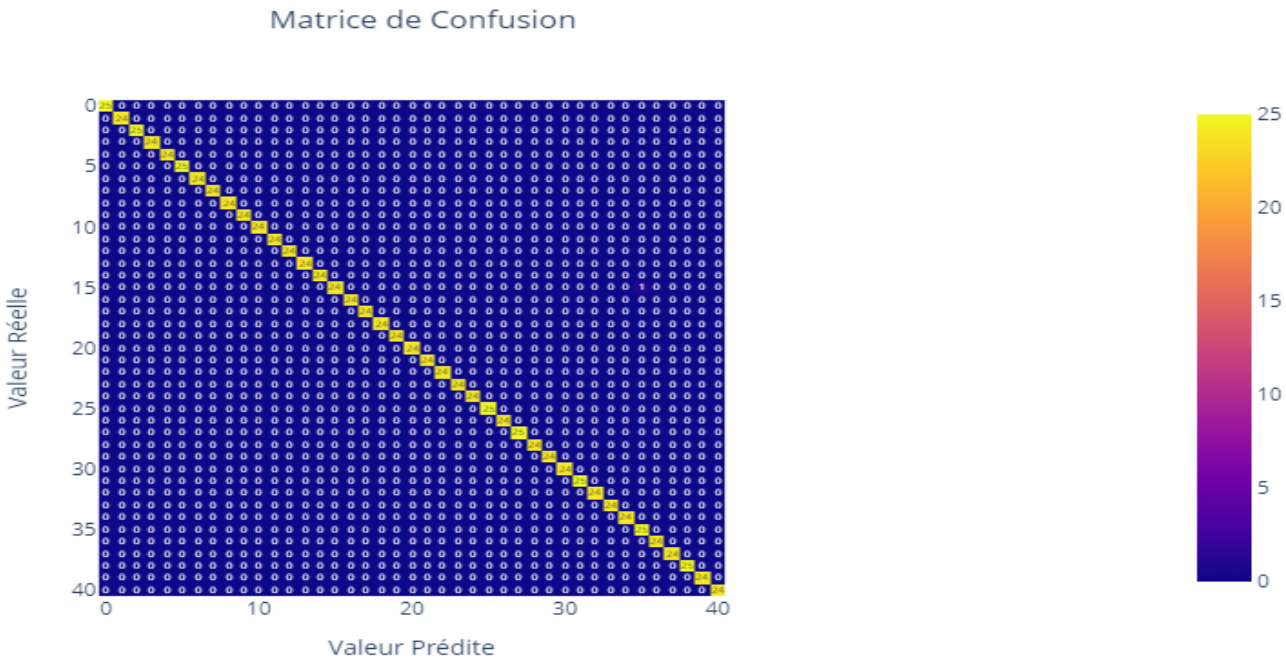

FIGURE 5.3 – Matrice de confusion de modèle ANN avec ADAM sans ACP

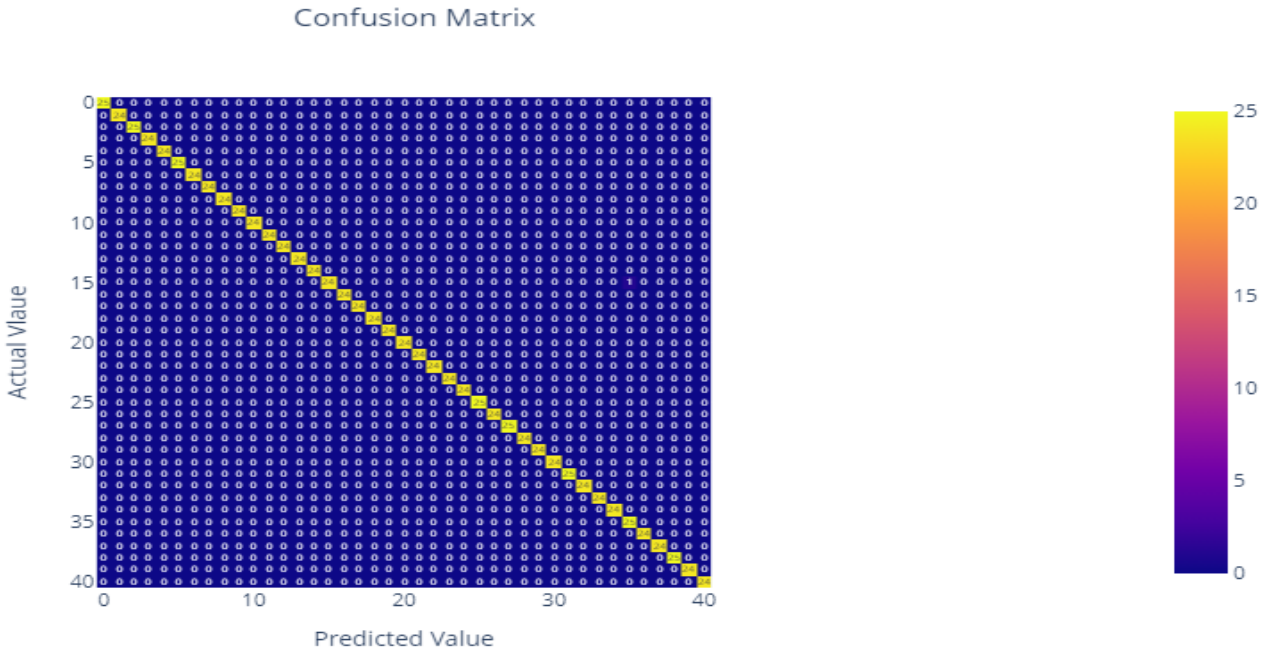

FIGURE 5.4 – Matrice de confusion de modèle ANN avec SGD sans ACP

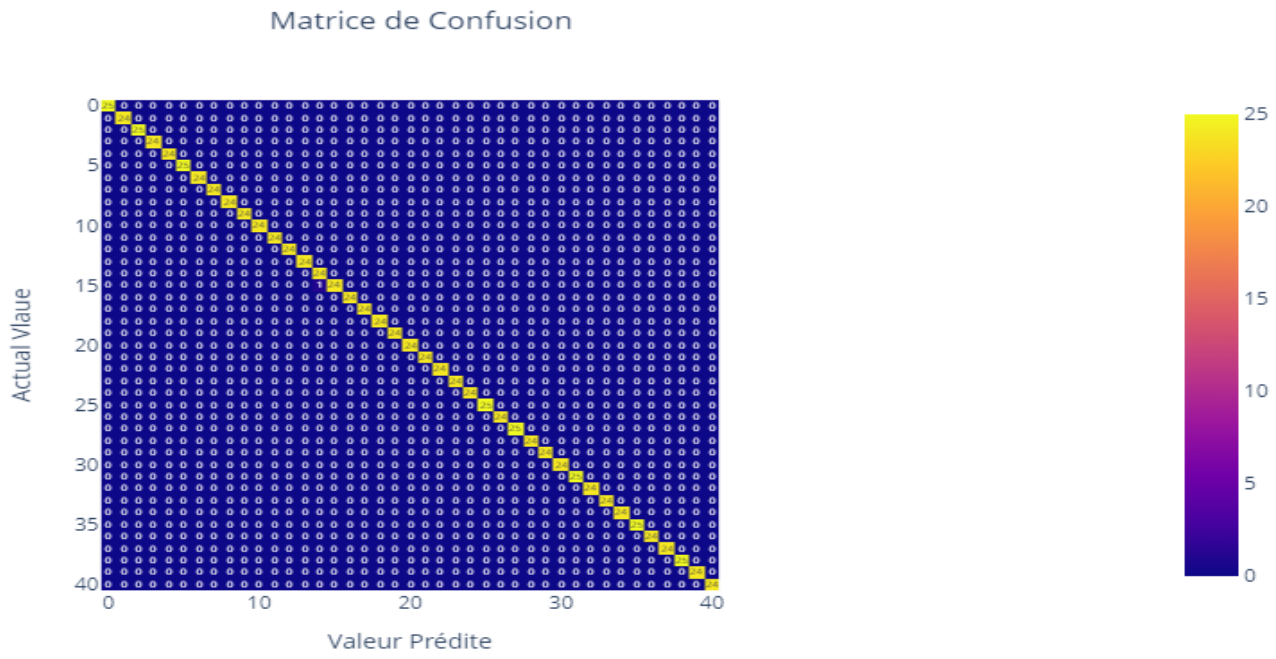

FIGURE 5.5 – Matrice de confusion de modèle ANN avec ADAM avec ACP

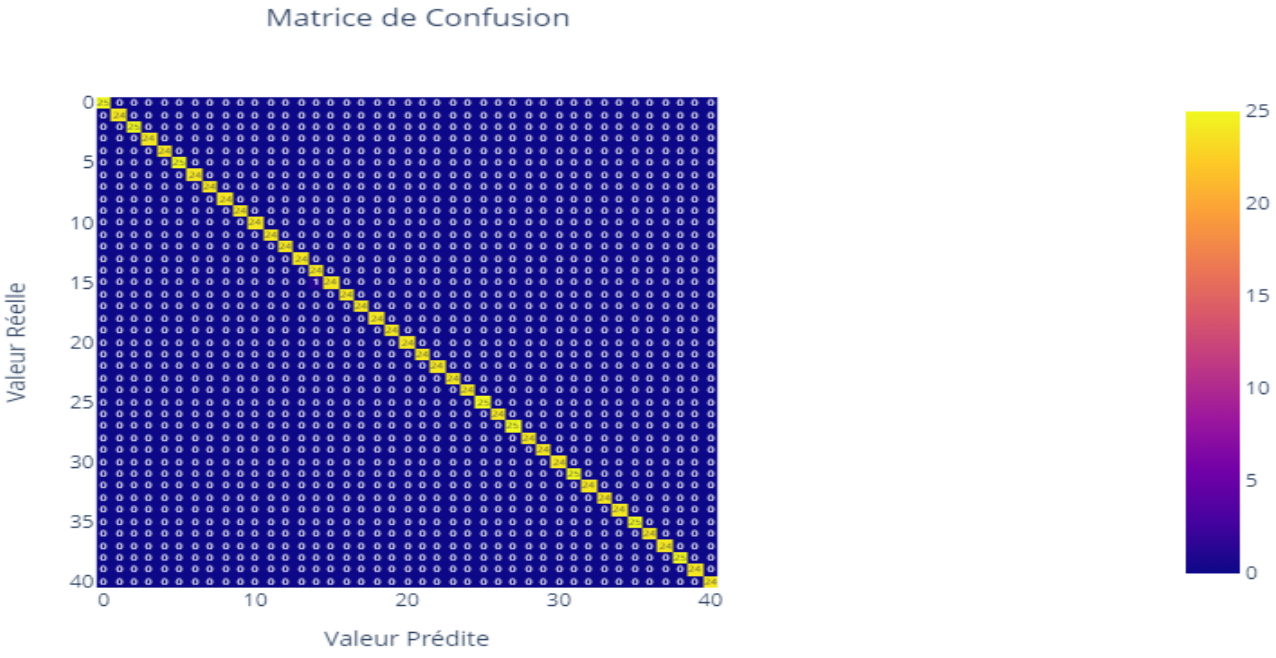

FIGURE 5.6 – Matrice de confusion de modèle ANN avec SGD avec ACP

5.8.2 Rappel

nous calculons la mesure d'évaluation "Rappel" de chaqu emodel. Les résultats sont présentés dans le tableau suivent :

Modèle	**Optimisation**	**ACP**	**Rappel** (10^{-3})
1	ADAM	Oui	0.998
2	ADAM	Non	0.998
3	SGD	Oui	1.000
4	SGD	Non	0.998

TABLE 5.6 – Tableau des résultats de rappel

5.8.3 Précision

Modèle	**Optimisation**	**ACP**	**Précision** (10^{-3})
1	ADAM	Oui	0.999
2	ADAM	Non	0.999
3	SGD	Oui	1.000
4	SGD	Non	0.999

TABLE 5.7 – Tableau des précisions

5.8.4 Score f1

Modèle	**Optimisation**	**ACP**	**Score F1** (10^{-3})
1	ADAM	Oui	0.998
2	ADAM	Non	0.998
3	SGD	Oui	1.000
4	SGD	Non	0.998

TABLE 5.8 – Tableau des scores F1

5.8.5 Exactitude

Modèle	**Optimisation**	**ACP**	**Exactitude** (10^{-3})	**Perte** (10^{-3})
1	ADAM	Oui	0.998	0.004
2	ADAM	Non	0.999	0.003
3	SGD	Oui	0.999	0.002
4	SGD	Non	0.998	0.006

TABLE 5.9 – Tableau des mesures d'évaluation

5.8.6 Temps d'entrainement

Modèle	Optimisation	ACP	Temps (secondes)
1	ADAM	Oui	16.55
2	ADAM	Non	22.77
3	SGD	Oui	3.20
4	SGD	Non	12.04

TABLE 5.10 – Tableau des temps d'exécution

5.8.7 Évolution des Performances d'apprentissage des Modèles

Au cours du processus d'entrainer les modèles, nous avons suivre les changements dans les performances du modèles dans chaque Epoch en termes de l'accuracy de validation et l'accuracy d'entainement et le perte de validation et le perte d'entainement, et nous avons enregistré les changements dans les graphes suivant :

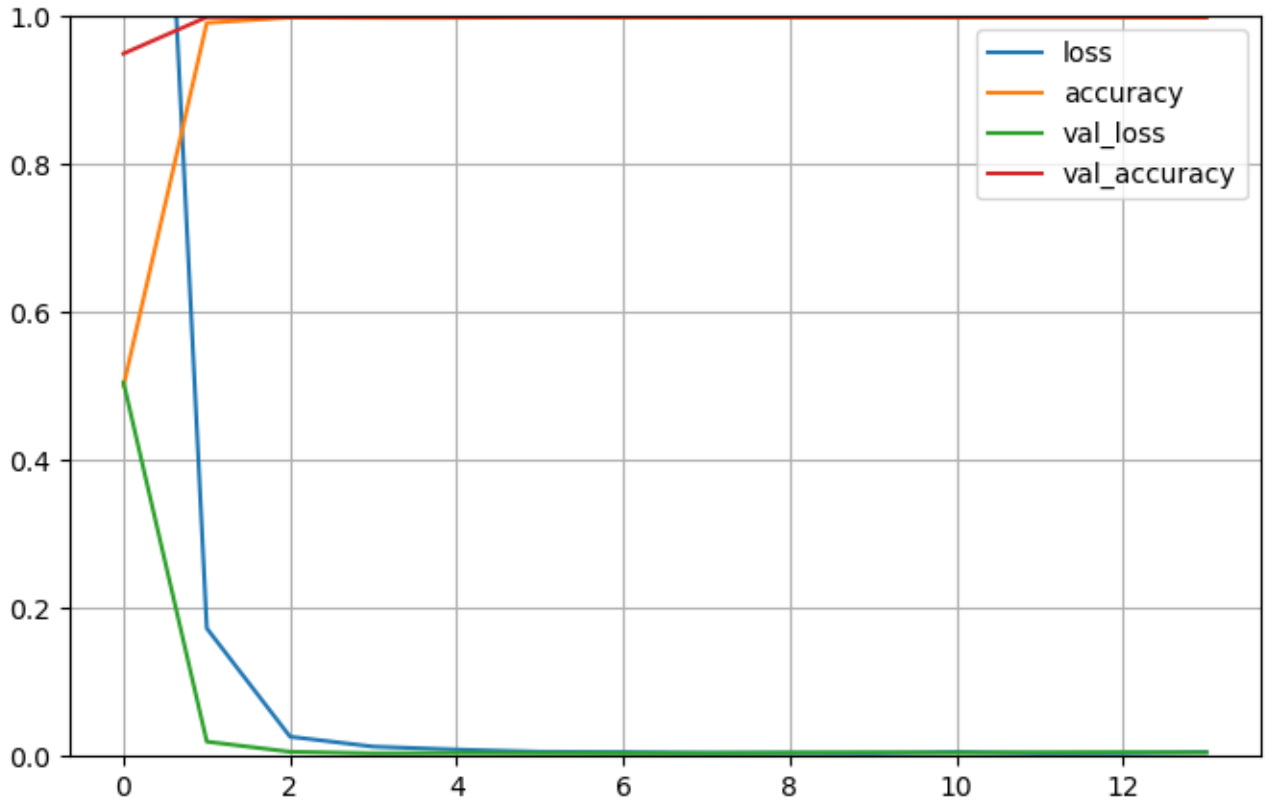

FIGURE 5.7 – Évolution des Performances de modèle ANN avec ADAM sans ACP

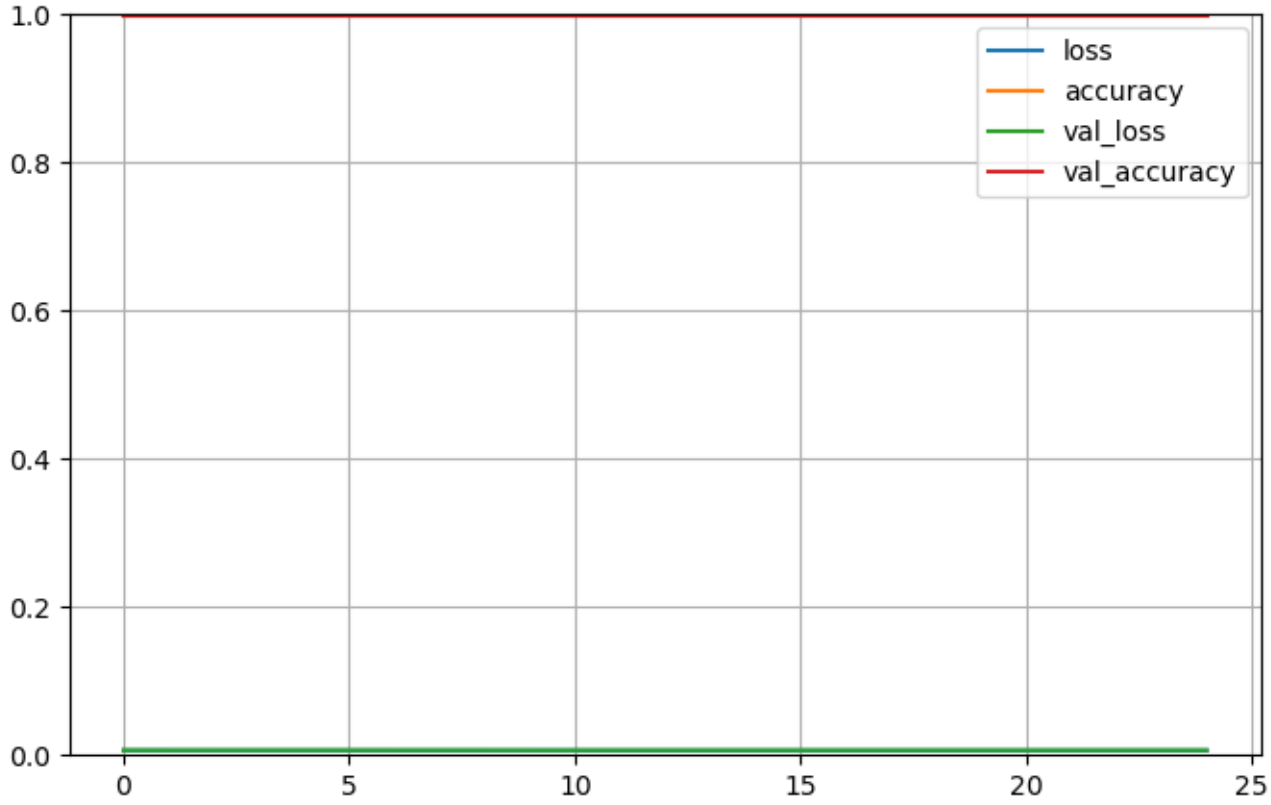

FIGURE 5.8 – Évolution des Performances de modèle ANN avec SGD sans ACP

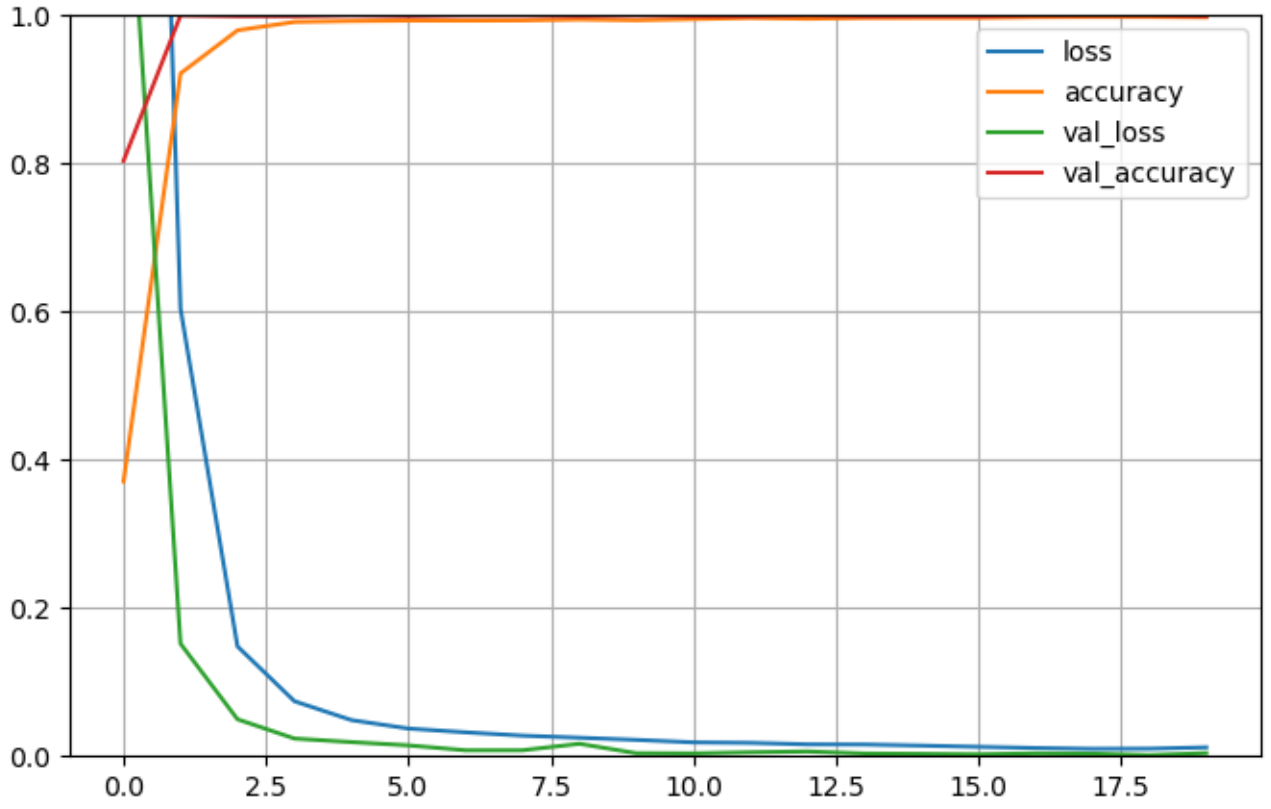

FIGURE 5.9 – Évolution des Performances de modèle ANN avec ADAM avec ACP

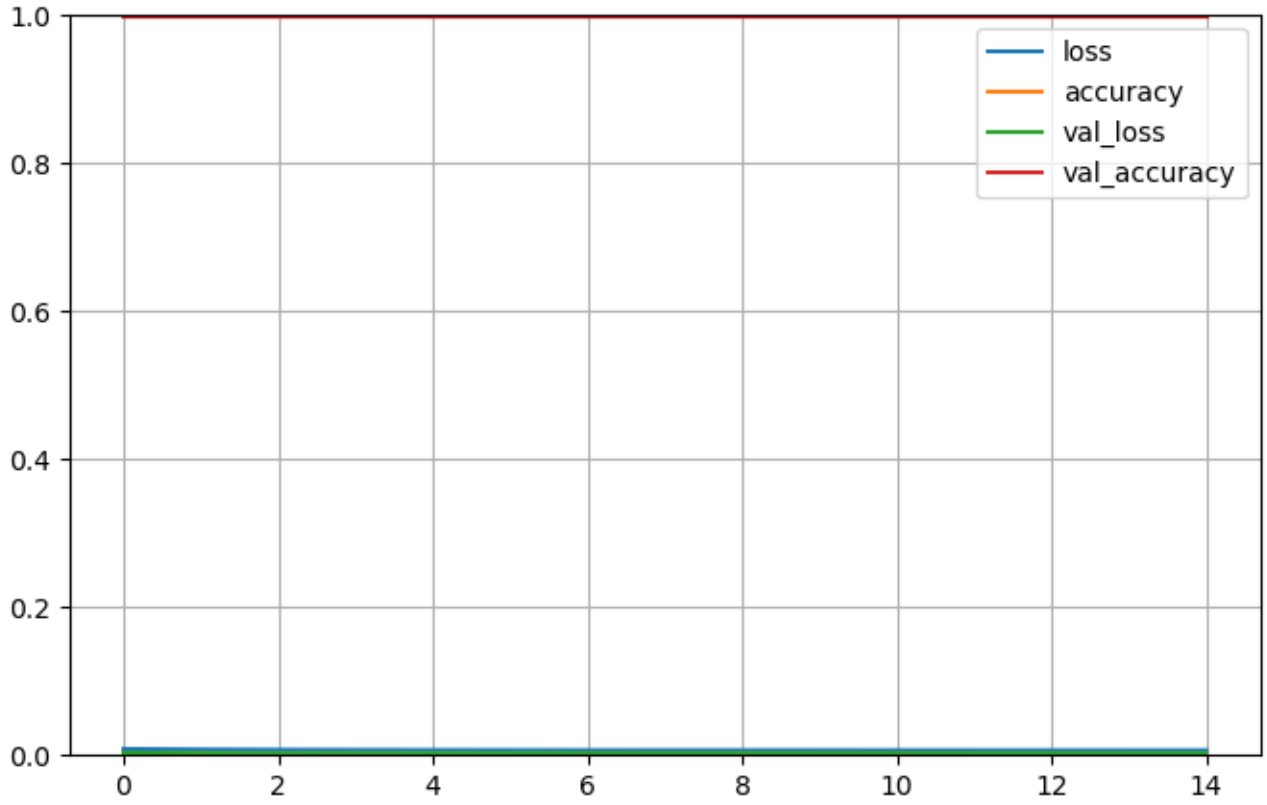

FIGURE 5.10 – Évolution des Performances de modèle ANN avec SGD avec ACP

5.9 Discussion

Les résultats de nos expériences fournissent un aperçu complet des performances des différents modèles d'apprentissage automatique que nous avons évalués. En examinant les mesures d'évaluation telles que le rappel, la précision, le score F1 et l'exactitude, nous constatons que tous les modèles ont montré des performances impressionnantes, indiquant leur capacité à bien généraliser sur les données qu'ils n'ont pas rencontrées pendant leur entraînement.

En particulier, le modèle utilisant l'optimiseur SGD avec l'ACP a montré des résultats remarquables, atteignant une précision et un score F1 parfaits de 1.000. Cette performance exceptionnelle suggère que l'utilisation de l'ACP en conjonction avec l'optimiseur SGD peut conduire à des résultats très fiables dans notre tâche de classification de symptômes médicaux. D'autre part, bien que le modèle utilisant l'optimiseur ADAM sans l'ACP ait également affiché de très bonnes performances, il était légèrement en deçà de celui utilisant l'ACP avec SGD.

Cela suggère que l'optimiseur ADAM peut produire des résultats précis mais peut nécessiter plus de données ou d'itérations pour atteindre un niveau de précision similaire.

Concernant le temps d'entraînement, les modèles utilisant l'optimiseur SGD étaient nettement plus rapides que ceux utilisant l'optimiseur ADAM. De plus, l'ajout de l'ACP a considérablement réduit le temps d'entraînement pour les deux optimiseurs, indiquant que l'ACP peut être une stratégie efficace pour accélérer le processus d'apprentissage tout en maintenant des performances élevées.

En examinant la relation entre l'exactitude (accuracy) d'entraînement et de validation ainsi que la perte d'entraînement et de validation, nous observons une corrélation étroite. Lorsque l'accuracy d'entraînement augmentait, l'accuracy de validation tendait également à augmenter, indiquant une bonne capacité du modèle à généraliser sur des données non vues. De manière similaire, une diminution de la perte d'entraînement entraînait également une diminution de la perte de validation, suggérant que le modèle apprenait efficacement les motifs présents dans les données d'entraînement et pouvait les généraliser aux données de validation. Cette relation entre l'accuracy et la perte d'entraînement et de validation est essentielle pour évaluer la capacité d'un modèle à apprendre à partir des données et à généraliser à de nouvelles données.

Enfin, en tenant compte des paramètres des modèles, nous observons que les modèles avec l'optimisation ACP ont moins de paramètres à entraîner que ceux sans ACP, ce qui peut contribuer à la réduction du temps d'entraînement. Cette considération supplémentaire renforce l'idée que l'utilisation de l'ACP peut être bénéfique pour accélérer le processus d'apprentissage tout en maintenant des performances élevées.

5.10 Présentation de l'application

5.10.1 Introduction à l'application

L'évolution du domaine du développement d'applications a permis d'étendre les possibilités de faciliter la vie quotidienne des individus et de se tourner vers les services numériques, en particulier dans le domaine médical.

iHealth est une application Android pour smartphones que nous avons développée dans le but d'offrir des services de santé intelligents. Notre idée repose essentiellement sur l'exploitation des modèles d'intelligence artificielle existants et leur utilisation de manière efficace et conviviale.

FIGURE 5.11 – le logo de l'application iHealth

5.10.2 Fonctionnalités principales

- Parcourir des articles liés au domaine médical.
- Explorer les maladies et les symptômes et peut également être enregistré et copié.
- Imprimer des rapports médicaux.
- Parcourir les rapports médicaux enregistrés, les symptômes et les maladies préférés.
- La possibilité pour nos utilisateurs de formuler leurs avis et préoccupations.
- Effectuer un diagnostic médical en temps réel.

5.10.3 Interface utilisateurs

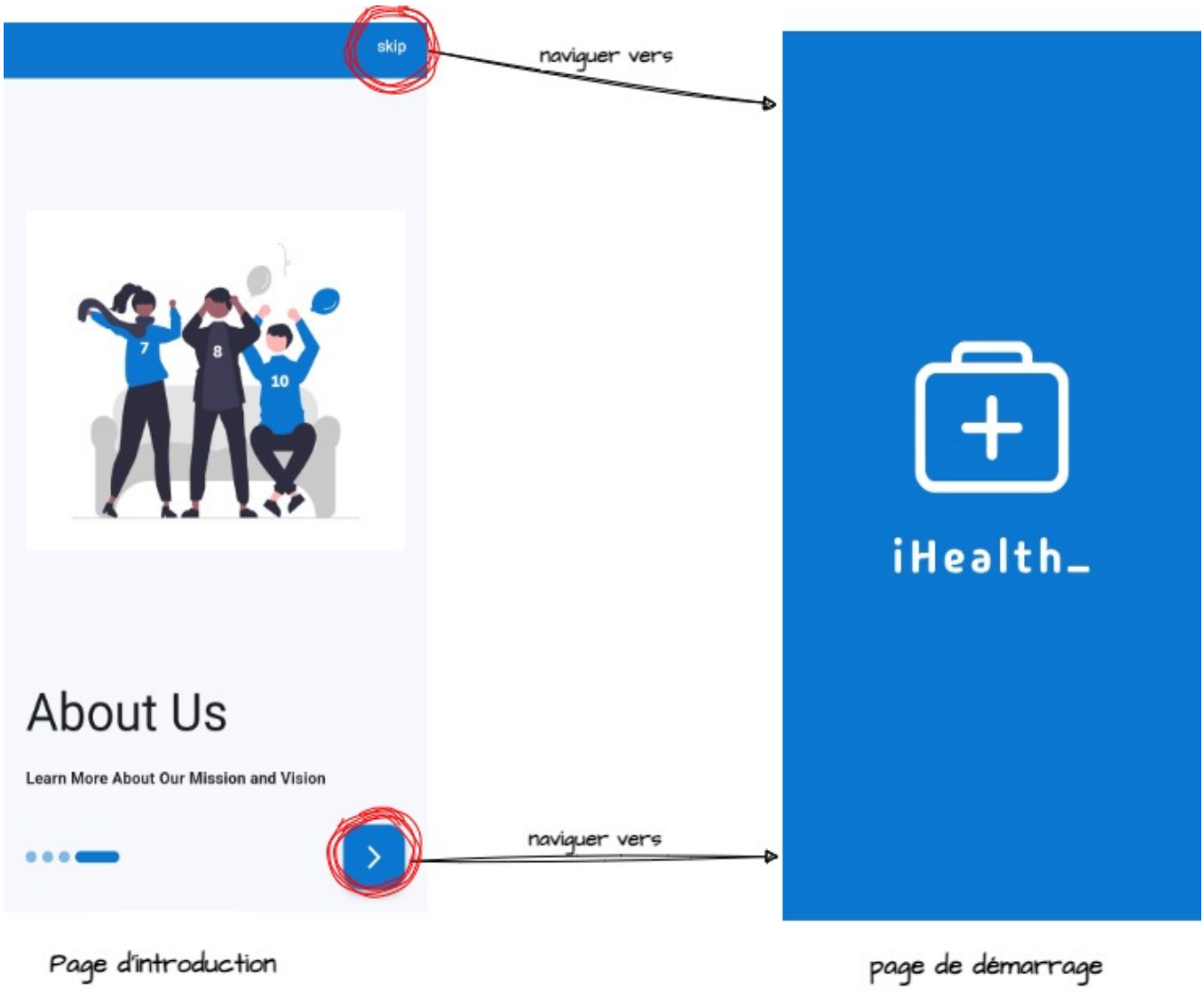

FIGURE 5.12 – La page d'introduction et page de démarrage

La page d'introduction permet la présentation des fonctionnalités fournies par l'application. Lorsque l'on clique sur le bouton Suivant ou "Skip", on passe à la page de démarrage, qui représente l'entrée de la page principale de l'application, où une connexion Internet est nécessaire pour se déplacer. à la page principale, sinon ce sera en attente.

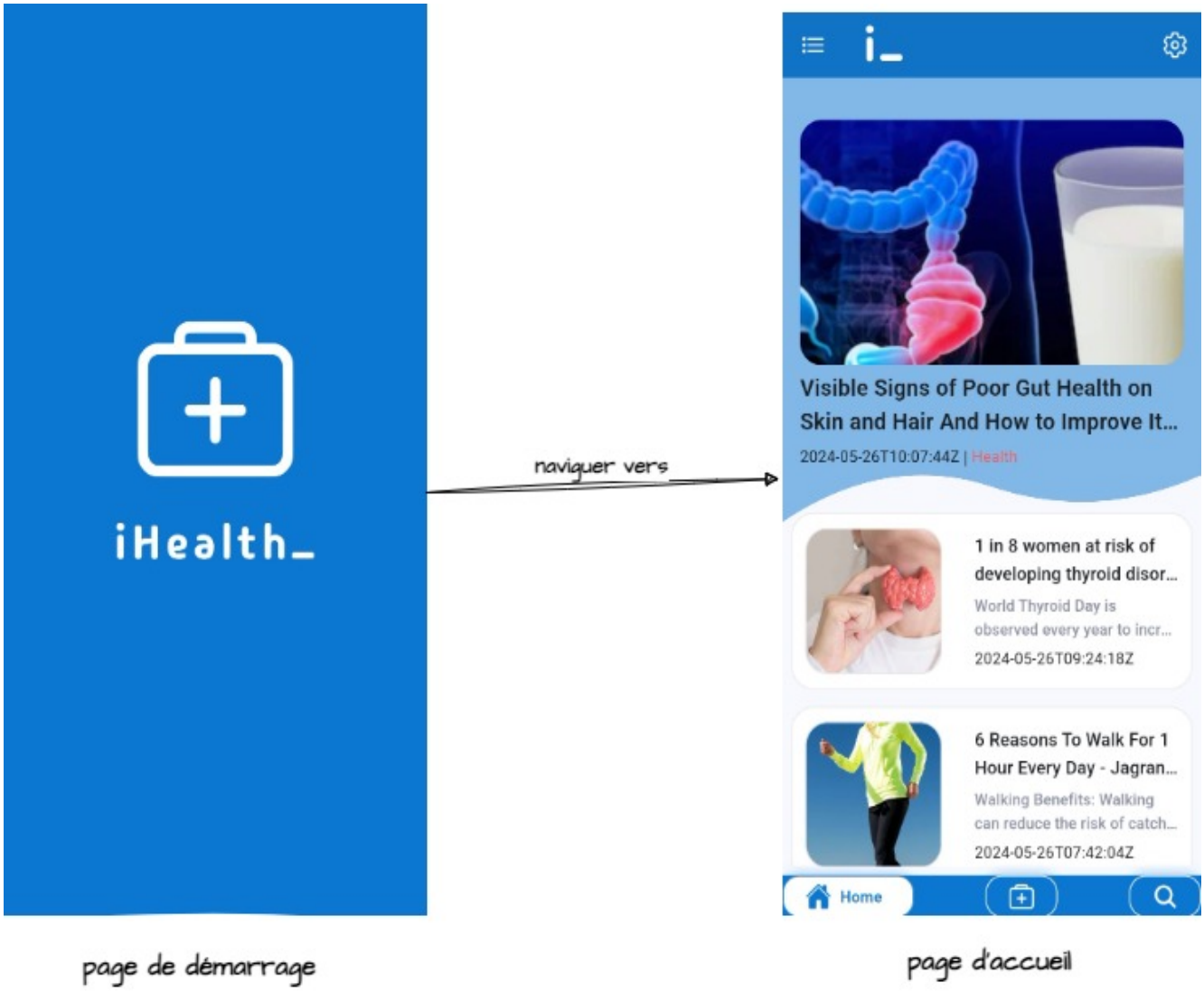

FIGURE 5.13 – La page d'accueil

la page d'accueil fournit un ensemble des articles liée au domain de santé, ses article sont accessible par un click sur l'article cible .

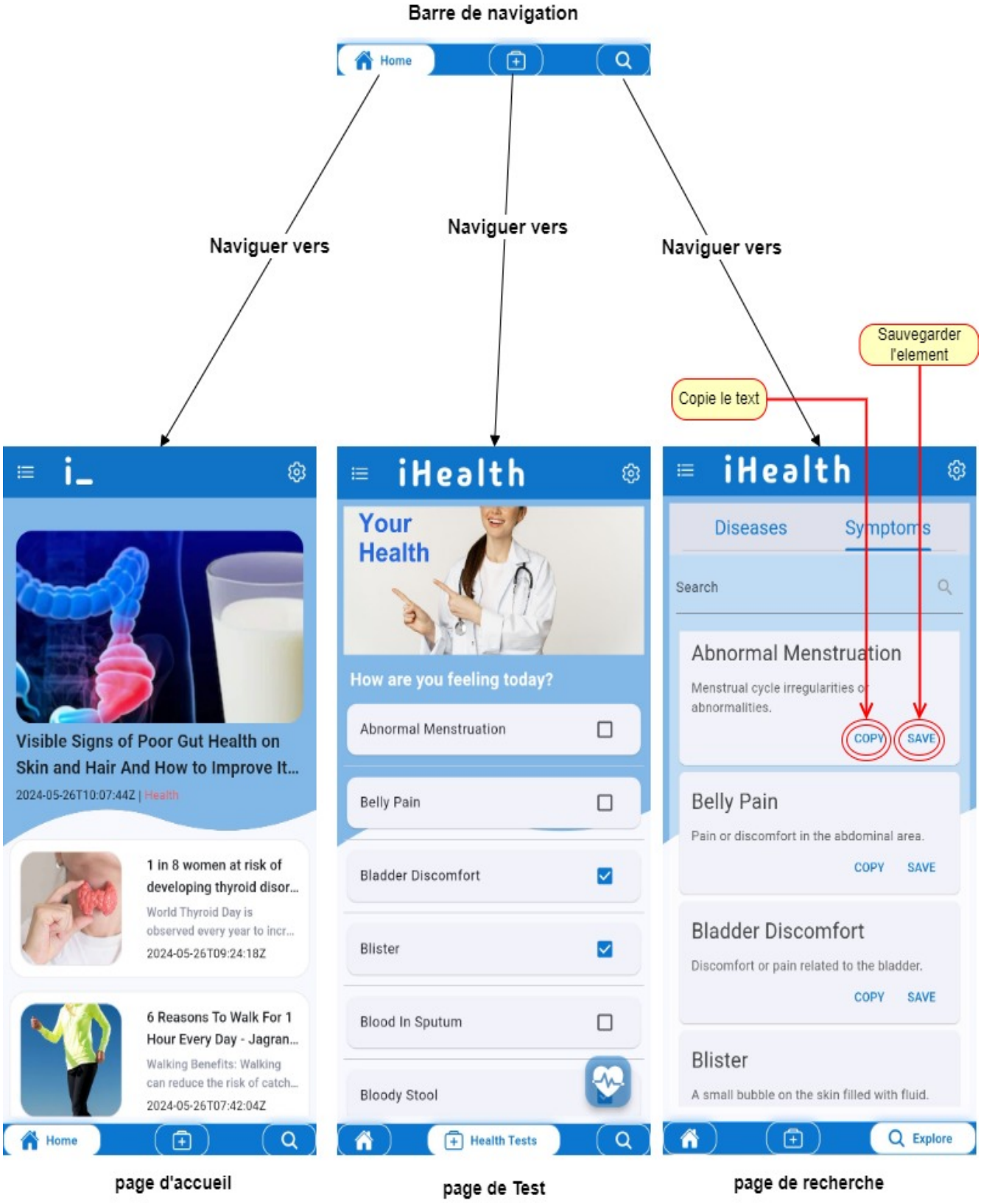

FIGURE 5.14 – Le navigation par le barre de navigation

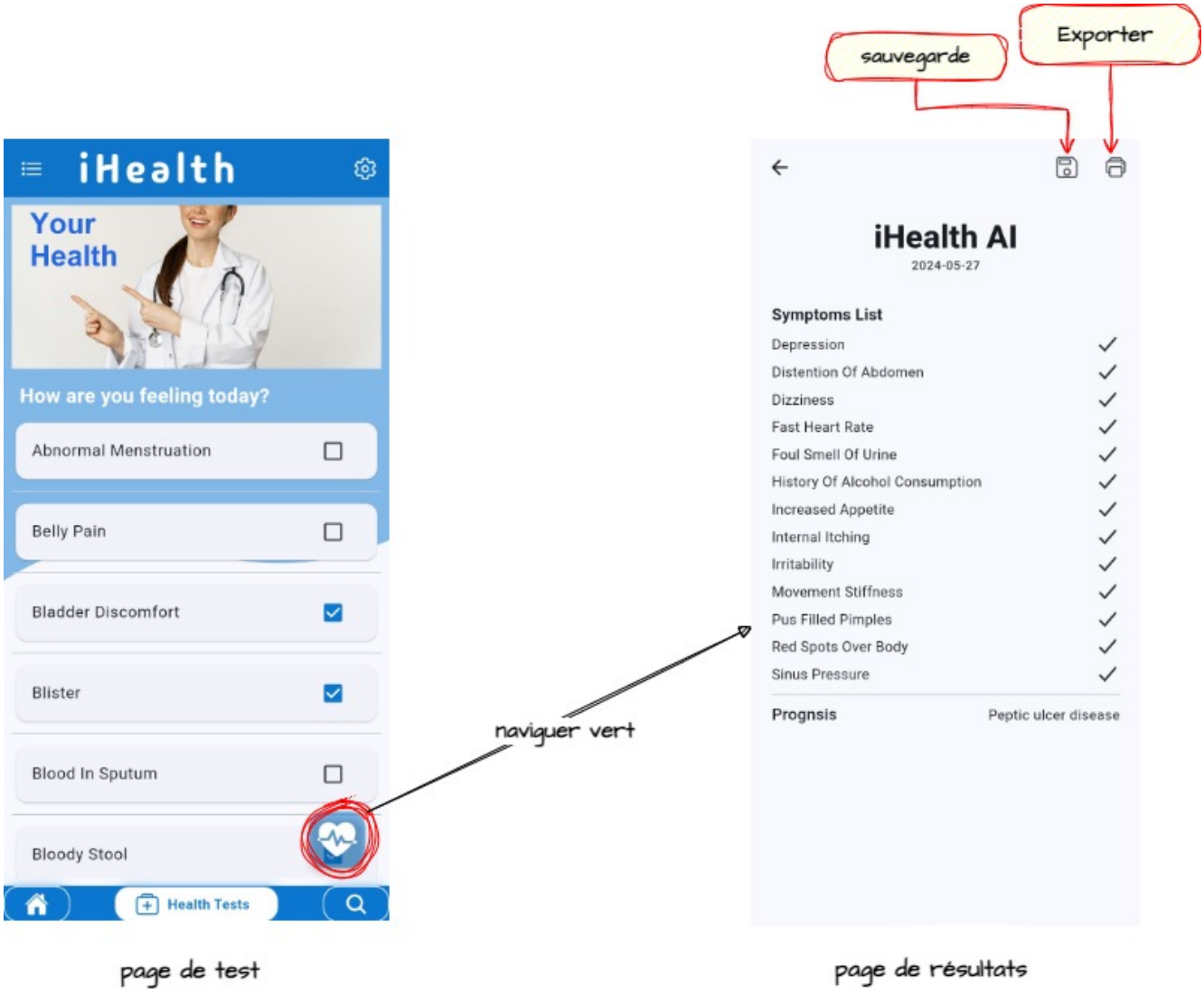

FIGURE 5.15 – La page de test et la page de rapport de resultats

La page de test permet d'effectuer un test médical immédiat en soumettant des symptômes médicales et en attendant le résultat sur la page Rapport des résultats de test. La page offre la fonctionnalité de sauvegarde et d'impression du rapport.

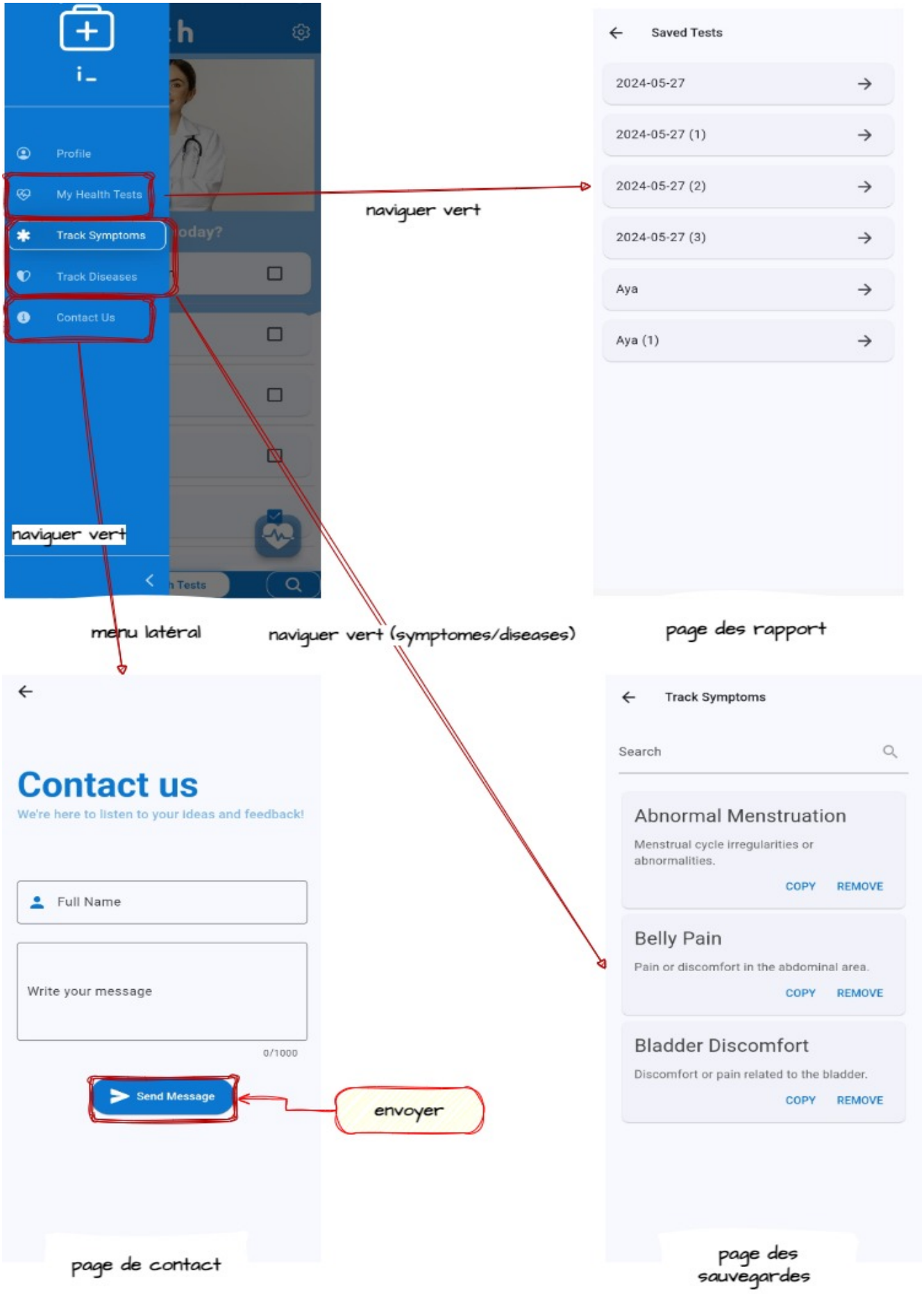

FIGURE 5.16 – Le navigation par le menu litéral

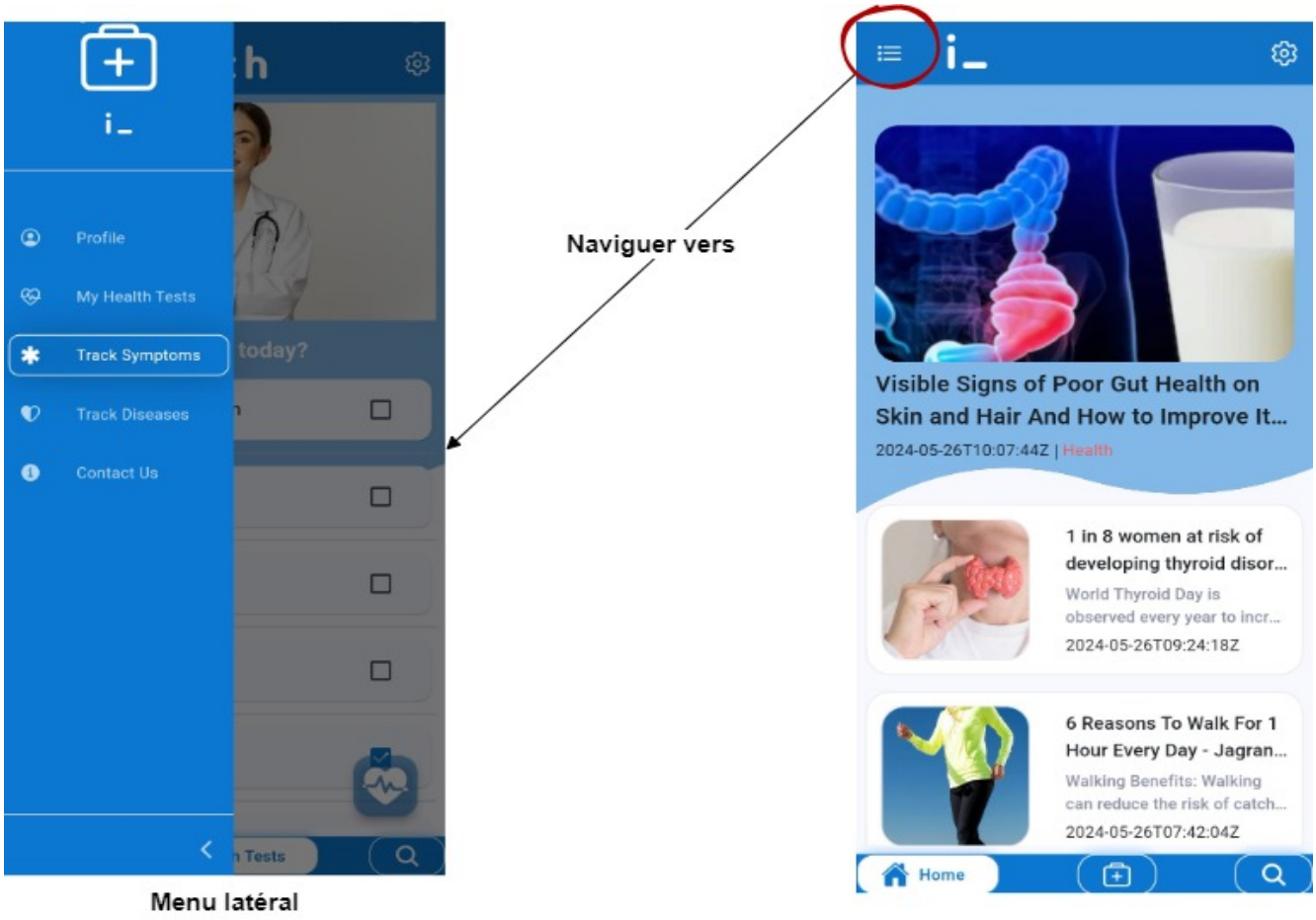

FIGURE 5.17 – Ouvrir le menu litéral

Le menu latéral permet de naviguer vers la page des rapports de résultats de tests enregistrés et permet de naviguer vers les symptômes et maladies enregistrés par l'utilisateur. Il amène également l'utilisateur à la page de contact si l'utilisateur a des opinions ou des messages qui nous sont adressés en tant que propriétaires de l'application.

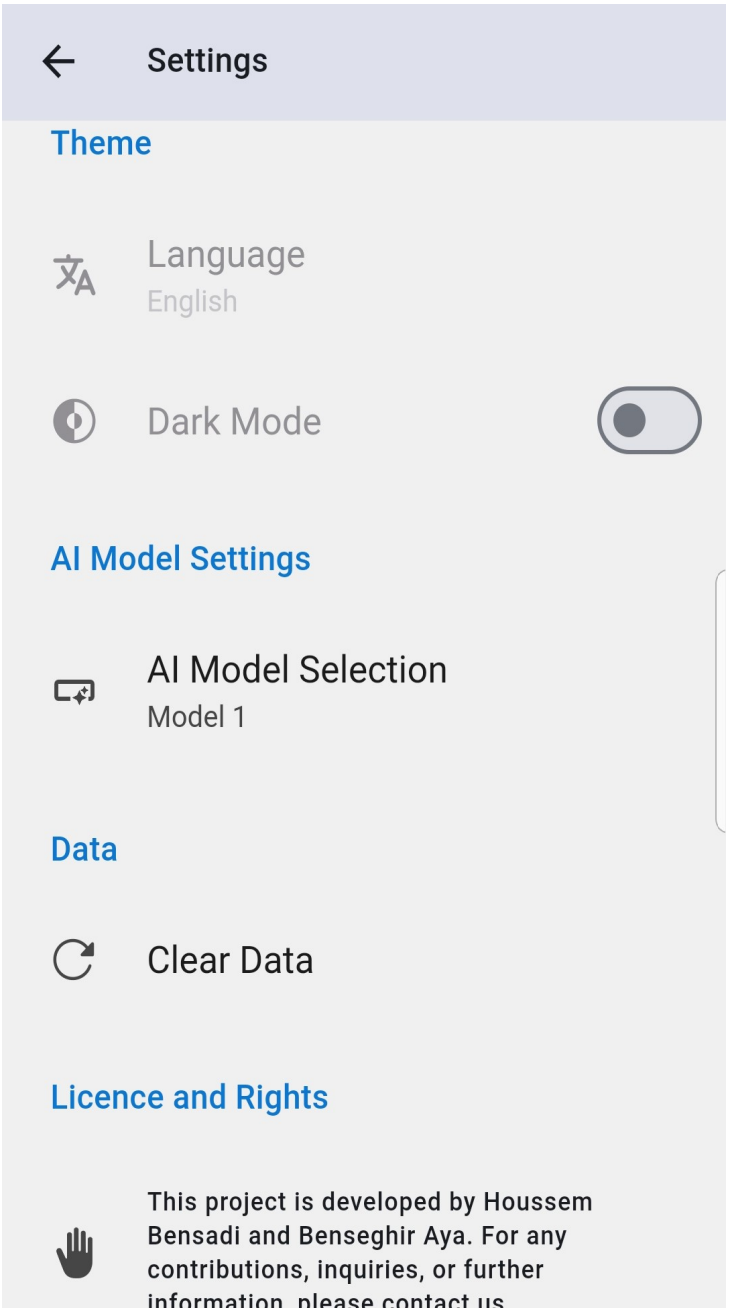

FIGURE 5.18 – La page des paramètres

La page des paramètres permet de contrôler les paramètres généraux de l'application, notamment changer la langue de l'application, activer et désactiver le mode nuit. Elle permet également de choisir le modèle d'intelligence artificielle à adopter pour réaliser des tests médicaux, ainsi que l'option de suppression des données utilisateur est également disponible.

5.11 Conclusion

Dans ce chapitre, nous avons exposé les étapes que pour concevoir et implémenter notre système de classification des symptômes médicaux. Par la suite, nous avons analysé en détail les performances des différents modèles en utilisant diverses mesures d'évaluation. Les résultats obtenus ont démontré l'efficacité de notre approche en tant qu'outil technologique fiable pour exploiter les données électroniques dans le domaine médical. Nous avons ensuite créé une application mobile pour démontrer comment les modèles créés peuvent être utilisés et employés dans des scénarios efficaces, et il faut souligner que cette approche vise à compléter le travail du médecin plutôt qu'à le remplacer.

Conclusion générale

Travaux réalisés

Au terme de notre engagement sur la problématique abordée, nous avons réussi à concevoir des modèles visant à accompagner les agents de santé dans leurs décisions,

Nous avons collecté autant d'informations que possible et des médecins spécialisés ont été consultés pour garantir leur intégrité.

Nous avons ensuite construit plusieurs modèles de réseaux de neurone avec différentes méthodes et enregistré les résultats de chaque modèle.

Nous avons appliqué l'optimisation de ADAM et SGD , et la méthode d'analyse en composantes principales comme une technique de réduction des dimensions ensuite nous avons enregistré les résultats de chaque modèle.

Ensuite, nous avons comparé les résultats obtenus, étudié l'effet des techniques adoptées et déterminé l'efficacité de chaque modèle.

Enfin, nous avons créé une application mobile qui propose des tests médicaux basés sur des modèles construits.

Les résultats obtenus prouvent que l'application de techniques avancées telles que les algorithmes de neuronaux artificiels et les techniques de réduction de dimension contribue à atteindre ces objectifs. Ces techniques nous permettent de faire une analyse des données de manière plus efficace et d'extraire rapidement les informations pertinentes, ce qui améliore la précision et l'efficacité du processus de diagnostic.

Perspectives

L'exploration des applications du deep learning dans le domaine médical ouvre de larges horizons et soulève d'importantes questions sur l'avenir alors que nous aspirons à développer une plateforme électronique dédiée aux soins de santé, basée sur l'idée qui a été développée.
En tant que projets futurs, les portes sont encore ouvertes pour développer des modèles plus complets en collectant un plus grand volume de la base de données actuellement disponible afin d'élargir le nombre de catégories cibles et également de fournir un plus grand nombre de fonctionnalités.
Parmi les projets proposés au cours de la période à venir figurent la construction de modèles du mécanisme d'attention et leur application dans le domaine médical.

Références

[1] F. Vanderhaegen, D. Jouglét, S. Duriez, O. Delville, Y. Picco, Y. Colmant, C. Faure, and A. Moulard, "E-diagnostic," *Rapport de Projet*, responsable du projet et équipe pédagogique, Equipe pédagogique, Equipe technique, Médiatisation du cours en ligne, Equipe technique, Equipe technique, Développement du simulateur et de l'éditeur de scénarios pour le simulateur, Equipe technique, coordination, Equipe technique, coordination, Equipe technique, coordination.

[2] National Health Service (NHS). (2021) Vomiting in adults. NHS. Londres. Consulté le 23 juin 2021. [Online]. Available : https://www.nhs.uk/conditions/vomiting-adults/

[3] J. Smith, *Understanding Medical Symptoms : A Comprehensive Guide*. New York : Oxford University Press, 2020.

[4] F. Scott D. C. Stern, MD, F. Adam S. Cifu, MD, and F. Diane Altkorn, MD, *Symptom to Diagnosis : An Evidence-Based Guide*. City : Publisher, 2010.

[5] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[6] C. M. Bishop, "Pattern recognition and machine learning," *Springer*, 2006.

[7] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," *Springer*, 2009.

[8] K. P. Murphy, *Machine Learning : A Probabilistic Perspective*. MIT Press, 2012.

[9] G. Tsoumakas and I. Katakis, "Multi-label classification : An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[10] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[11] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013.

[12] C. J. Burges, *A tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery, 1998.

[13] I. T. Jolliffe, "Principal component analysis," *International Encyclopedia of Statistical Science*, pp. 1094–1096, 2016.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] J. Schmidhuber, "Deep learning in neural networks : An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th international conference on machine learning (ICML-10)*, vol. 10, pp. 807–814, 2010.

[17] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," *Neurocomputing*, vol. 6, pp. 227–236, 1990.

[18] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," *Neural networks : Tricks of the trade*, vol. 7700, pp. 9–48, 2012.

[19] S. Haykin, *Neural networks and learning machines*, 3rd ed. Pearson, 2009.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

Annexe A: La sélection des symptômes par éléction

TABLE 1.1 – Comparaison des symptômes et des décisions des médecins

Symptômes	Dr.bendjeddou	Dr.siouda	Dr.Univ	med4	est similaire ?	décision
abnormal_menstruation	0	1	1		0	1
altered_sensorium	0	0	0		1	0
belly_pain	1	1	1		1	1
blackheads	1	0	0		0	0
bladder_discomfort	1	0	1		0	1
blister	1	0	1		0	1
blood_in_sputum	1	0	1		0	1
bloody_stool	1	0	1		0	1
blurred_and_distorted_vision	1	0	1		0	1
brittle_nails	0	0	1		0	0
bruising	1	0	0		0	0
chest_pain	1	0	1		0	1
coma	1	0	0		0	0
congestion	1	0	1		0	1
continuous_feel_of_urine	1	0	1		0	1
cramps	0	1	1		0	1
depression	1	1	0		0	1
dischromic _patches	0	0	0		1	0
distention_of_abdomen	1	1	1		1	1
dizziness	1	0	1		0	1
drying_and_tingling_lips	1	0	0		0	0
enlarged_thyroid	1	1	1		1	1
excessive_hunger	1	1	1		1	1
extra_marital_contacts	1	0	1		0	1
family_history	1	1	1		1	1
...	...	...	...	...	...	...