الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة الإخوة منتوري قسنطينة ا

Frères Mentouri Constantine I University

Université Frères Mentouri Constantine I

Faculté des Sciences de la Nature et de la Vie كلية علوم الطبيعة و الحياة

Département de Biologie Appliquée قسم البيولوجيا التطبيقية

Mémoire présenté en vue de l'obtention du diplôme de Master

Domaine : Sciences de la Nature et de la Vie

Filière : Sciences Biologiques

Spécialité : Bioinformatique

N° d'ordre :

N° de série :

Intitulé :

Modèle d'apprentissage profond pour l'identification génomique de la bactérie *Escherichia coli*

Présenté par : BEYOUD Mohamed Borhan Eddine

BOUSSOUF Mohamed Amine

Le 18/06/2023

Président : Dr. MEDJROUBI Mohamed

Encadreur : Dr. DAAS Mohamed Skander

Examinateur : Dr. DJAMAA Ouahiba

Année universitaire 2022 – 2023

REMERCIEMENTS

Nous remercions Dieu tout puissant de nous avoir données la santé, la force et la volonté d'entamer et de terminer ce mémoire de fin d'étude.

Nous souhaitons profiter de cette occasion pour exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de notre mémoire de fin d'études. Ce projet représente un jalon important dans notre parcours en bioinformatique, et nous sommes reconnaissants envers ceux qui nous ont soutenus tout au long de ce voyage.

Tout d'abord, nous tenons à remercier notre encadrant de mémoire Dr. DAAS Mohamed Skander, pour sa guidance précieuse et ses conseils éclairés. Sa passion pour le domaine de la bioinformatique et son expertise nous ont inspirés et ont grandement contribué à l'élaboration de notre projet. Nous sommes honorés d'avoir pu bénéficier de sa supervision et de ses retours constructifs tout au long du processus.

Je tiens également à remercier chaleureusement tous les membres de mon jury de soutenance, Dr. MEDJROUBI Mohamed et Dr. DJAMAA Ouahiba .Votre expertise, votre intérêt pour mon sujet et vos commentaires éclairés ont enrichi ma réflexion et ont été d'une valeur inestimable. Je suis honoré(e) d'avoir pu bénéficier de vos connaissances lors de ma présentation.

Enfin, nous souhaitons exprimer notre gratitude envers nos familles pour leur soutien inconditionnel. Leur amour, leur patience et leurs encouragements ont été une source d'inspiration et de motivation tout au long de cette aventure académique.

Nous sommes conscients que ce projet n'aurait pas été possible sans l'aide et le soutien de toutes ces personnes exceptionnelles. Leur contribution a été essentielle à notre réussite, et nous leur en sommes profondément reconnaissants.

Cordialement,

RÉSUMÉ

Ce mémoire présente une étude sur l'utilisation de modèles d'apprentissage profond pour l'identification des bactéries Escherichia Coli à partir des données bactériennes des séquences ARNr 16S. Les séquences ARNr 16S sont des marqueurs moléculaires couramment utilisés pour identifier et classer les bactéries. L'objectif principal de notre projet est de développer des modèles d'apprentissage profond efficaces capables d'identifier avec précision les bactéries Escherichia Coli. Les principales étapes comprennent la collecte de données, le prétraitement des séquences, la construction du modèle d'apprentissage profond adapté aux données séquentielles, l'entraînement et l'évaluation du modèle. Les performances du modèle sont évaluées à l'aide de plusieurs métriques telles que la précision et la perte.

Les résultats ont démontré que le modèle proposé est capable d'identifier avec précision la bactérie Escherichia Coli.

Mots clés : apprentissage profond, classification génomique, Escherichia coli, réseaux de neurones convolutifs, bioinformatique

ABSTRACT

This work presents a study on the use of deep learning models for the identification of Escherichia Coli bacteria from bacterial 16S rRNA sequence data. 16S rRNA sequences are commonly used molecular markers to identify and classify bacteria. The main objective of our project is to develop efficient deep learning models capable of accurately identifying Escherichia Coli bacteria. The main steps include data collection, sequence pre-processing, construction of the deep learning model fitted to the sequence data, training and evaluation of the model. Model performance is evaluated using several metrics such as Accuracy, Recall, and F-measure. The results demonstrated that the proposed model is able to accurately identify Escherichia Coli bacteria.

Keywords : deep learning, genomic classification, Escherichia coli, convolutional neural networks, bioinformatics.

الملخص

تقدم أطروحة المشروع هذه دراسة حول استخدام نماذج التعلم العميق لتحديد بكتيريا Escherichia Coli من بيانات تسلسل الرنا الريباسي S16 البكتيري. يشيع استخدام متواليات الرنا الريباسي S16 في الواسمات الجزيئية لتحديد وتصنيف البكتيريا. الهدف الرئيسي لمشروعنا هو تطوير نماذج تعلم عميق فعالة قادرة على التعرف بدقة على بكتيريا Escherichia Coli. تشمل الخطوات الرئيسية جمع البيانات والمعالجة المسبقة للتسلسل وبناء نموذج التعلم العميق المناسب لبيانات التسلسل والتدريب وتقييم النموذج. يتم تقييم أداء النموذج باستخدام عدة مقاييس مثل الدقة والاستدعاء والقياس F. أظهرت النتائج أن النموذج المقترح قادر على التعرف بدقة على بكتيريا Escherichia Coli.

الكلمات المفتاحية: التعلم العميق ، التصنيف الجيني ، الإشريكية القولونية ، الشبكات العصبية التلافيفية ، المعلوماتية الحيوية.

LISTES DES FIGURES

LISTE DES TABLEAUX

ACRONYMES

- ADN : Acide Désoxyribonucléique
- aDNA : Ancient DNA (ADN ancien)
- ANN : Artificial Neural Network (Réseau de neurones artificiels)
- API : Application Programming Interface (Interface de programmation d'application)
- ARN : Acide Ribonucléique
- ARNc : Acide Ribonucléique codant
- ARNi : ARN interférent
- ARNm : ARN messager
- ARNnc : ARN non codant
- ARNr : Acide Ribonucléique Ribosomal
- ARNt : ARN de transfert
- ATP : Adénosine TriPhosphate
- CE : Cross-Entropy (Entropie croisée)
- CNN : Convolutional Neural Networks (Réseaux de neurones convolutionnels)
- COLAB : Google Colaboratory (service de cloudcomputing basé sur Jupyter Notebook)
- CONV : Couche de convolution
- DL : Deep Learning (Apprentissage profond)
- E. coli : Escherichia coli
- exp : Exponential (Exponentielle)
- FAD : Flavine Adénine Dinucléotide
- FC : FullyConnected (Entièrement connecté)
- FN : Faux négatif (False Negative)
- FP : Faux positif (False Positive)
- GTP : GuanosineTriPhosphate
- GPU : GraphicsProcessing Unit (Unité de traitement graphique)
- H2S : Sulfure d'hydrogène
- IA : Intelligence Artificielle
- LOSS : Couche de perte
- MLP : Multilayer Perceptron (Perceptron multicouche)
- ML : Machine Learning (Apprentissage automatique)
- MSE : MeanSquaredError (Erreur quadratique moyenne)
- NAD+ : Nicotinamide Adénine Dinucléotide (forme oxydée)

- NLL : Negative Log-Likelihood (Log-vraisemblance négative)
- PCR : Réaction en chaîne par polymérase (Polymerase Chain Reaction en anglais)
- POOL : Couche de pooling
- ReLU : RectifiedLinear Unit (Unité de rectification linéaire)
- RNN : Réseau de neurones récurrents
- SGD : Stochastic Gradient Descent (Descente de gradient stochastique)
- TN : Vrai négatif (TrueNegative)
- TP : Vrai positif (True Positive)
- TPU : TensorProcessing Unit (Unité de traitement de tenseurs)

TABLE DES MATIÈRES

TABLE DES MATIÈRES

Introduction

INTRODUCTION

La classification précise et rapide des entérobactéries revêt une importance cruciale dans de nombreux domaines de la microbiologie, tels que la santé publique, l'agriculture et l'environnement. Les méthodes traditionnelles de classification reposent principalement sur des critères phénotypiques et génétiques, mais elles peuvent être coûteuses, chronophages et sujettes à des erreurs. L'avènement des techniques de séquençage à haut débit a ouvert de nouvelles perspectives en permettant l'utilisation des séquences d'ARNr 16S pour la classification taxonomique des entérobactéries. Cependant, l'interprétation et l'analyse de ces vastes ensembles de données nécessitent des outils bio-informatiques puissants et novateurs.

Dans cette étude, nous utilisons des modèles d'apprentissage profond pour l'identification taxonomique de la bactérie Escherichia Coli à partir de données de séquences d'ARNr 16S. L'apprentissage profond, une branche de l'intelligence artificielle, a connu des avancées majeures ces dernières années grâce à des modèles tels que les réseaux de neurones convolutifs (CNN). Ces modèles ont démontré leur efficacité dans de nombreuses tâches de classification et de prédiction de séquences biologiques.

L'objectif principal de cette étude est de développer et d'évaluer des modèles d'apprentissage profond capables d'identifier avec précision des bactéries Escherichia Coli à partir de leurs séquences d'ARNr 16S. Pour cela, nous mettrons en œuvre une approche en plusieurs étapes, comprenant la collecte et le prétraitement des données, la conception et l'entraînement des modèles d'apprentissage profond, ainsi que l'évaluation des performances obtenues.

Enfin, nous chercherons à interpréter les résultats obtenus pour comprendre les motifs et les caractéristiques utilisés par les modèles d'apprentissage profond pour effectuer l'identification avec précision des bactéries *Escherichia Coli*. Nous discuterons également des implications de ces résultats dans le contexte de cette étude et des perspectives futures pour l'amélioration des méthodes d'identification.

Ce mémoire vise à contribuer à l'identification des bactéries *Escherichia Coli* .en utilisant des approches innovantes basées sur l'apprentissage profond. Les résultats de cette étude pourraient avoir des implications significatives dans des domaines tels que la surveillance épidémiologique, la recherche en santé publique.

PARTIE 1 :

RECHERCHE BIBLIOGRAPHIQUE

CHAPITRE 1 :

Biologie du génome et des bactéries

1. Introduction

La biologie du génome et des bactéries est étroitement liée. Les bactéries possèdent des génomes qui contiennent du matériel génétique, tel que l'ADN ou l'ARN, codant des informations cruciales pour leur croissance et leur fonction. Les génomes bactériens sont constitués de chromosomes circulaires et peuvent inclure des plasmides plus petits. L'étude de la biologie des génomes bactériens consiste à analyser leur structure, leur organisation et leur fonction. Le transfert horizontal de gènes, où le matériel génétique est transféré entre les bactéries, contribue à l'évolution bactérienne. Comprendre la biologie du génome bactérien à de vastes implications dans divers domaines. Ce chapitre présente les concepts de base sur Biologie du génome et la classification des bactéries.

2. Acide désoxyribonucléique (ADN)

L'ADN est une molécule qui stocke l'information génétique dans la plupart des êtres vivants. Sa double hélice a été découverte par James Watson et Francis Crick en 1953 montré dans la Figure1 , pour laquelle ils ont remporté le prix Nobel de physiologie ou de médecine en 1962[1]

La structure de l'ADN se compose de deux brins enroulés autour d'un axe central, chacun composé de nucléotides contenant une base azotée, un sucre désoxyribose et un groupe phosphate. Les quatre bases azotées dans l'ADN sont l'adénine, la thymine, la guanine et la cytosine[2].

Les bases azotées de chaque brin s'apparient complémentairement, formant des liaisons hydrogène entre l'adénine et la thymine, et entre la guanine et la cytosine. Cette complémentarité permet la réplication de l'ADN lors de la division cellulaire et la transcription de l'information génétique en ARN. La double hélice de l'ADN assure sa stabilité et sa résistance mécanique, protégeant ainsi l'information génétique contre les dommages physiques et chimiques[1].

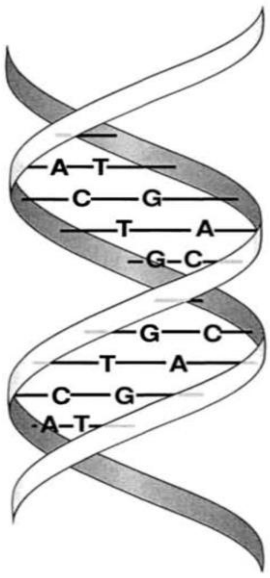

Figure 1 : Modèle d'ADN décrit par watson and crik [3].

2.1. Types d'ADN

L'ADN, acronyme pour acide désoxyribonucléique, est la molécule qui contient les instructions génétiques de tous les organismes vivants. Il existe différents types d'ADN, chacun ayant des caractéristiques spécifiques et remplissant des fonctions différentes[4].

- ADN génomique : c'est l'ADN présent dans le noyau des cellules eucaryotes. Il contient les gènes qui codent pour les protéines et d'autres séquences régulatrices qui contrôlent leur expression.
- ADN mitochondrial : c'est l'ADN présent dans les mitochondries, les organites responsables de la production d'énergie dans les cellules eucaryotes. Il est transmis de la mère à l'enfant et ne subit pas de recombinaison génétique.
- ADN ribosomal : c'est l'ADN présent dans les ribosomes, les complexes de protéines et d'ARN responsables de la synthèse des protéines dans les cellules. Il est souvent utilisé pour étudier les relations évolutives entre les organismes.
- ADN chloroplastique : c'est l'ADN présent dans les chloroplastes, les organites responsables de la photosynthèse chez les plantes et certains organismes photosynthétiques. Il est transmis de manière maternelle chez les plantes et subit une recombinaison génétique limitée.

- ADN nucléaire ancien (aDNA) : c'est de l'ADN extrait de restes fossiles ou archéologiques. Il est souvent fragmenté et altéré par les conditions environnementales, ce qui rend son séquençage difficile.
- ADN extra chromosomique : c'est de l'ADN présent en dehors du noyau des cellules eucaryotes, tel que l'ADN plasmidique chez les bactéries. Il peut contenir des gènes qui confèrent des avantages adaptatifs à l'hôte[5].

3. Acide Ribonucléique (ARN)

L'ARN est l'acronyme pour acide ribonucléique. Il s'agit d'une molécule proche de l'ADN, mais qui remplit des fonctions différentes dans les cellules. L'ARN intervient notamment dans la synthèse des protéines, en transportant les informations génétiques de l'ADN jusqu'aux ribosomes, les usines de production des protéines[6].

3.1. Structure d'ARN

L'ARN (acide ribonucléique) est une molécule biologique qui joue un rôle crucial dans l'expression de l'information génétique dans les cellules vivantes .L'ARN est structurellement similaire à l'ADN (acide désoxyribonucléique), mais il diffère en quelques points clés[7].

L'ARN est généralement simple brin, ce qui signifie qu'il est composé d'une seule chaîne de nucléotides voir dans la Figure 2.

Un nucléotide est une unité de base des acides nucléiques tels que l'ADN et l'ARN. Il se compose d'un groupement phosphate, d'un sucre (désoxyribose pour l'ADN et ribose pour l'ARN) et d'une base azotée. Les bases azotées comprennent l'adénine (A), la thymine (T), la cytosine (C), la guanine (G) et l'uracile (U) voir dans la Figure 2. Les nucléotides se lient entre eux pour former des brins d'ADN ou d'ARN, qui jouent un rôle essentiel dans le stockage, la transmission et l'expression de l'information génétique. [8].

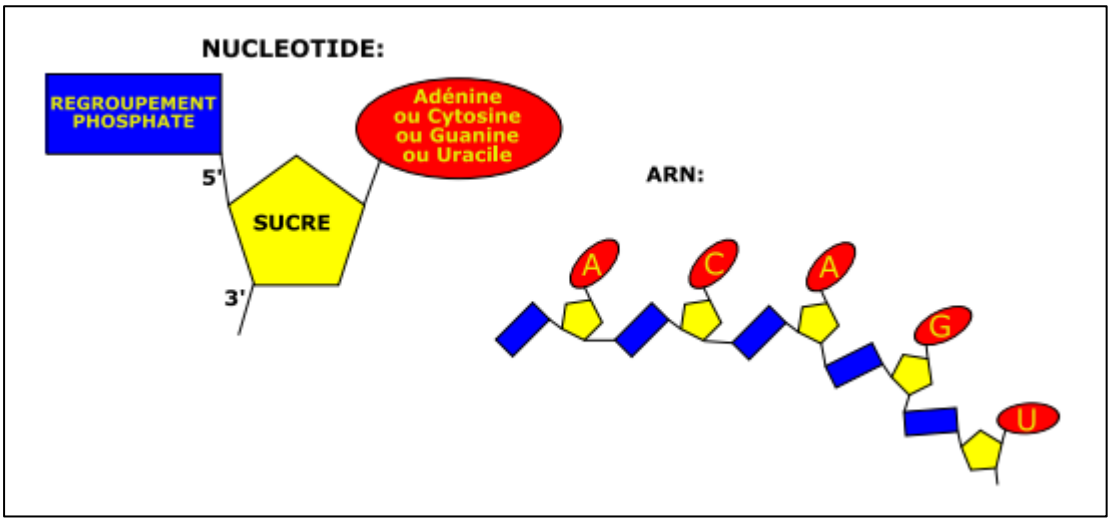

Figure 2 : Sur la gauche, la représentation d'un nucléotide ; `adroite celle d'un ARN (Allali, s. d.).

La structure tridimensionnelle de l'ARN peut varier considérablement en fonction de la séquence des nucléotides et des interactions entre eux. Cependant, il existe des motifs structurels communs trouvés dans l'ARN, notamment[9] :

- **La tige-boucle :** une structure en forme de tige qui est stabilisée par des appariements de bases complémentaires et qui peut former une boucle à l'extrémité.
- **La boucle simple :** une boucle non appariée dans la chaîne d'ARN qui peut jouer un rôle dans l'interaction avec d'autres molécules.
- **Le bulbe :** une structure en forme de bulbe formée par l'appariement de bases complémentaires dans une région spécifique de l'ARN.
- **L'hélice tige-boucle :** une structure en forme d'hélice formée par l'appariement de bases complémentaires dans une région spécifique de l'ARN.
- **Le pseudo-nœud :** une structure complexe en trois dimensions formée par l'appariement de bases non adjacentes dans une région spécifique de l'ARN.
- **Le site actif :** une région spécifique de l'ARN qui peut interagir avec d'autres molécules pour catalyser des réactions chimiques.
- **Les régions régulatrices :** des régions spécifiques de l'ARN qui peuvent réguler l'expression des gènes en interagissant avec d'autres molécules dans la cellule.

L'étude de la structure de l'ARN est un domaine de recherche actif depuis plusieurs décennies, car la compréhension de la structure tridimensionnelle de l'ARN est essentielle pour comprendre sa fonction. Les avancées techniques telles que la cristallographie aux rayons X, la spectroscopie de résonance magnétique nucléaire (RMN) et la microscopie

électronique à cryo ont considérablement élargi nos connaissances sur la structure de l'ARN[10].

3.2. Types d'ARN

Les ARN messagers, abrégés ARNm, sont les types d'ARN les plus connus. Ils sont responsables du transport de l'information génétique codée dans l'ADN pour la production de protéines. La synthèse de protéines à partir de l'ARNm est réalisée par le processus de traduction représenté dans la Figure 3 [11].

- **ARN ribosomaux :** abrégés ARNr, constituent les éléments fondamentaux des ribosomes. Ces derniers se composent de deux sous-unités de ARNr, appelées grande et petite sous-unités, ainsi que de protéines. Les ribosomes jouent un rôle central dans la traduction de l'ARNm en protéines[12].
- **ARN de transfert :** appelés ARNt, interviennent également dans la traduction en collaboration avec les ribosomes. Les ARNt établissent un lien entre les nucléotides de l'ARNm et les acides aminés, les constituants de base des protéines. Ils transportent les acides aminés nécessaires à la synthèse des protéines, permettant ainsi la liaison entre le monde des ARN et celui des protéines[11]. Comme mentionné précédemment, certains ARN tels que les ARNr, les ARNt et certains ARNm ont des rôles moléculaires spécifiques dans la cellule qui sont liés à leur structure spatiale particulière. Les ARN sont donc considérés comme étant structurés, car ils adoptent une conformation spatiale précise qui est spécifique à leur activité. Des motifs de structure ont été établis pour certaines classes d'ARN en fonction de leur fonction moléculaire[13].

.

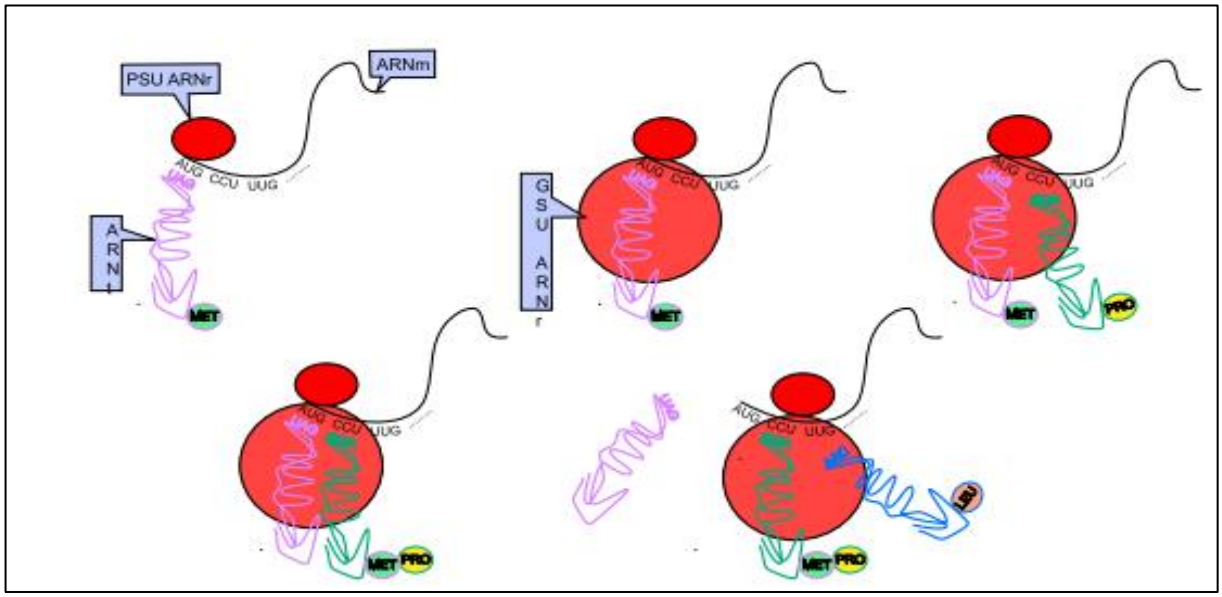

Figure 3 : Présente de manière très simplifiée le mécanisme de traduction et le rôle joue par les ARNm, ARNr et ARNt. Et les différentes étapes de la traduction (Allali, s. d.).

Et Il existe plusieurs autres types d'ARN, chacun ayant une fonction spécifique dans la cellule. Voici une liste des principaux types d'ARN avec une petite définition :

- **ARN messager (ARNm) :** L'ARNm est une copie de l'ADN qui est utilisée pour synthétiser des protéines dans la cellule. Il transporte l'information génétique du noyau de la cellule vers les ribosomes, où la traduction de l'information en protéines a lieu[14].
- **ARN ribosomal (ARNr) :** L'ARNr est une composante des ribosomes, qui sont les structures cellulaires responsables de la synthèse des protéines. L'ARNr forme la majorité des ribosomes et joue un rôle crucial dans la lecture de l'ARNm pour la synthèse de protéines.
- **ARN de transfert (ARNt) :** L'ARNt est un petit ARN qui transporte les acides aminés nécessaires à la synthèse des protéines jusqu'aux ribosomes. Chaque ARNt est spécifique pour un acide aminé particulier et contient un triplé de nucléotides appelé anticodon qui se lie à un codon spécifique de l'ARNm.
- **ARN interférent (ARNi) :** L'ARNi est un ARN double brin qui régule l'expression des gènes en bloquant la traduction de l'ARNm cible ou en dégradant l'ARNm cible. Il est utilisé en recherche pour éteindre spécifiquement des gènes et ainsi étudier leur fonction.
- **ARN non codant (ARNnc) :** Les ARNnc sont des ARN qui ne sont pas traduits en protéines. Ils peuvent jouer un rôle régulateur dans la cellule en interagissant avec d'autres

molécules d'ARN ou en régulant l'expression des gènes. Les exemples d'ARNnc incluent les microARN, les piARN et les lncARN[13].

4. Entérobactéries

Les Entérobactéries constituent un groupe diversifié de bactéries Gram négatives qui font partie de la famille des Enterobacteriaceae. Ce groupe comprend des espèces telles que *Escherichia coli, Klebsiellapneumoniae, Salmonella enterica* et *Yersinia pestis*, entre autres. Les Enterobactéries sont des micro-organismes ubiquitaires, qui colonisent des environnements variés, tels que le sol, l'eau, les plantes et les animaux, y compris les humains. De plus, certaines espèces sont des pathogènes opportunistes ou des agents pathogènes importants pour l'homme et les animaux[15].

De plus, l'étude de la génomique des Enterobactéries a permis de mieux comprendre leur génétique moléculaire, leur évolution, leur adaptation et leur virulence. Des techniques telles que le séquençage de nouvelle génération ont permis d'identifier des gènes associés à la résistance aux antibiotiques, à la virulence et à la fitness des Enterobactéries[15].

4.1. Présentation des principaux genres

- ***Escherichia :*** *Escherichia* est une bactérie commune que l'on retrouve normalement dans l'intestin de l'homme et des animaux. C'est l'espèce aérobie la plus abondante dans le tube digestif.
- ***Shigella :*** *Les shigelles* sont des bactéries strictement présentes chez les humains et ne font pas partie de la flore intestinale normale. Elles se trouvent uniquement chez les personnes malades, en convalescence ou chez quelques porteurs sains. Elles sont responsables de la célèbre "dysenterie bacillaire" qui avait des effets dévastateurs sur les armées en campagne.
- ***Klebsiella :*** Parmi les entérobactéries, les bactéries du genre *Klebsiella* se distinguent par leur immobilité constante et leur tendance à se regrouper en diplobacilles généralement encapsulés. Plusieurs espèces sont identifiées, mais Klebsiellapneumoniae est la plus fréquemment observée chez les patients cliniques.
- ***Proteus-Providencia :*** Le groupe *Proteus-Providencia* au sein de la famille des Enterobacteriaceae se caractérise principalement par la présence du tryptophane désaminase et par son invasion constante de la gélose nutritive. Ce sont des hôtes normaux de l'intestin chez l'homme et les animaux, mais dans certains cas, ils peuvent devenir

pathogènes et causer diverses infections telles que des entérites, des cystites, des otites et des méningites. On observe une augmentation de l'incidence de ces infections.

- ***Salmonella :*** Les Salmonelles se trouvent dans l'eau et divers aliments et sont pathogènes, soit exclusivement pour l'homme *(Salmonella typhi),* soit exclusivement pour l'animal *(Salmonella abortusovis).* Chez l'homme, elles sont responsables de la fièvre typhoïde et de gastro-entérites. Morphologiquement, elles sont similaires aux autres entérobactéries, bien que certaines souches qui sont normalement mobiles puissent apparaître immobiles lorsqu'isolées[16].

4.2. Génomique des Entérobactéries

La génomique des Entérobactéries est un domaine de recherche en constante évolution qui vise à comprendre la structure, la fonction et l'évolution des génomes des bactéries de la famille des Enterobacteriaceae. Les avancées technologiques en matière de séquençage de nouvelle génération ont permis de séquencer un grand nombre de génomes d'Entérobactéries, ce qui a permis de mieux comprendre leur diversité génétique et leur adaptation à différents milieux.

La comparaison des génomes d'Entérobactéries a permis d'identifier des caractéristiques génomiques communes, telles que des gènes de virulence, des systèmes de sécrétion et des gènes de résistance aux antibiotiques. Par exemple, l'étude de la génomique comparative *d'Escherichia coli* a permis de mettre en évidence des gènes de résistance aux antibiotiques présents chez les souches pathogènes, ce qui a des implications pour la surveillance et la prévention des infections nosocomiales [17].

La génomique des Entérobactéries a également permis de mieux comprendre l'écologie et l'évolution de ces bactéries. Par exemple, des études ont montré que l'acquisition de gènes de résistance aux antibiotiques par les bactéries du sol pouvait se propager à des pathogènes humains, ce qui souligne l'importance de la surveillance environnementale pour la prévention de la propagation de la résistance aux antibiotiques[18].

5. Méthodes de classification des entérobactéries

Plusieurs méthodes de classification ont été développées pour les Enterobacteriaceae, notamment les tests biochimiques, le typage sérologique, les méthodes moléculaires, les tests de sensibilité aux antimicrobiens et le typage phagique[19].

5.1. Classification phénotypique

La classification phénotypique des Enterobacteriaceae est une méthode utilisée pour identifier et classer ces bactéries en fonction de leurs caractéristiques morphologiques, biochimiques et physiologiques. Cette méthode est importante pour la détection et l'identification précise des espèces d'Enterobacteriaceae, y compris celles qui sont pathogènes pour l'homme. La classification phénotypique des Enterobacteriaceae est basée sur les caractéristiques morphologiques, biochimiques et physiologiques des bactéries. Les caractéristiques phénotypiques sont souvent utilisées pour identifier et classer les bactéries en groupes taxonomiques, tels que les genres et les espèces[20].

Cependant, il est important de noter que la classification phénotypique seule peut ne pas être suffisante pour identifier et caractériser précisément toutes les espèces d'Enterobacteriaceae.

5.1.1. Tests biochimiques

Les tests biochimiques sont largement utilisés pour l'identification et la classification des bactéries, en particulier pour les entérobactéries, qui sont des bactéries gram-négatives comprenant des espèces comme *Escherichia coli, Salmonella* et *Shigella.* Ces tests se basent sur la capacité des bactéries à utiliser différents substrats pour produire des enzymes spécifiques ou des métabolites détectables à l'aide de réactifs adaptés[21].

Les tests biochimiques sont simples, rapides et peu coûteux, ce qui les rend utiles pour une identification préliminaire des bactéries en laboratoire clinique ou de recherche. Toutefois, la classification phénotypique peut parfois être limitée, et pour une identification plus précise [22].

5.1.2. Tests physiologiques

Les tests physiologiques sont des méthodes qui mesurent les réponses des bactéries à des conditions environnementales spécifiques. Ces tests peuvent être utilisés pour déterminer la capacité des bactéries à fermenter différents sucres, à produire des enzymes, à métaboliser

différents substrats, à résister à certains antibiotiques, et à d'autres caractéristiques. Les résultats de ces tests permettent d'établir une identification précise de la bactérie et de la classer dans une catégorie spécifique[23].

Les tests physiologiques sont souvent utilisés en combinaison avec d'autres méthodes d'identification, telles que la morphologie des colonies, la coloration de Gram, et les tests biochimiques pour obtenir une identification plus précise des bactéries[24].

5.1.3. Tests sérologiques

Les tests sérologiques sont largement utilisés pour la classification des bactéries de la famille des Enterobacteriaceae en différentes souches ou sérotypes. Cette classification est importante pour des raisons épidémiologiques, diagnostiques et taxonomiques. Les tests sérologiques sont basés sur la détection d'antigènes spécifiques à la surface des cellules bactériennes à l'aide d'anticorps spécifiques. Les antigènes O, H et K sont les plus couramment utilisés pour la classification des Enterobacteriaceae[25].

Le test sérologique le plus couramment utilisé est le test de sérotypage des antigènes O, qui permet de différencier les souches de bactéries en fonction de leur composition en antigènes O. Les antigènes O sont des polysaccharides présents à la surface de la paroi cellulaire bactérienne[26].

D'autres tests sérologiques, tels que les tests de sérotypage des antigènes H et K, peuvent également être utilisés pour classer les Enterobacteriaceae en différents sérotypes. Les antigènes H sont des flagelles présents à la surface des bactéries, tandis que les antigènes K sont des capsules bactériennes[27].

5.2. Classification génétique

La classification génétique des Enterobacteriaceae est basée sur l'analyse de l'ADN des bactéries. Cette classification a été développée à partir de techniques moléculaires avancées telles que l'analyse de l'empreinte génétique, la PCR (réaction en chaîne de la polymérase), la séquençage de l'ADN, l'hybridation génomique comparative, etc.[28].

L'analyse de l'ADN permet de classer les Enterobacteriaceae en différents groupes, en fonction de leur similitude génétique. Les techniques de classification génétique sont souvent utilisées pour identifier des groupes spécifiques de bactéries associés à des maladies particulières ou à des environnements spécifiques. Par exemple, les analyses génétiques ont permis de classer les Escherichia coli en différentes lignées, telles que les E. coli

entéropathogènes, entérohémorragiques, entéro-invasifs, etc. Cette classification est utile pour déterminer la source des infections et pour élaborer des stratégies de prévention et de traitement[29].

En résumé, la classification génétique des Enterobacteriaceae est une méthode puissante pour étudier la diversité de cette famille de bactéries et pour élucider les relations évolutive entre les différentes espèces et genres.

6. Différents niveaux de classification taxonomique

La classification taxonomique est la méthode utilisée pour organiser les organismes vivants en groupes hiérarchiques basés sur leurs caractéristiques physiques, leur évolution et leur histoire[30]. Les différents niveaux de classification sont les suivants, du plus général au plus spécifique[31].

- **Domaine :** Le domaine est le niveau le plus général de la classification des êtres vivants. Il y a trois domaines reconnus : les bactéries, les archées et les eucaryotes.
- **Règne :** Le règne est le deuxième niveau de classification, qui regroupe les organismes en fonction de leurs caractéristiques fondamentales. Il y a cinq règnes reconnus : les bactéries, les archées, les protistes, les champignons et les animaux et les plantes.
- **Phylum :** Le phylum est un niveau de classification qui regroupe des organismes ayant des caractéristiques anatomiques et physiologiques similaires.
- **Classe :** La classe est un niveau de classification qui regroupe des organismes ayant des caractéristiques anatomiques et physiologiques similaires à celles du phylum, mais plus spécifiques.
- **Ordre :** L'ordre est un niveau de classification qui regroupe des organismes ayant des caractéristiques anatomiques et physiologiques similaires à celles de la classe, mais encore plus spécifiques.
- **Famille :** La famille est un niveau de classification qui regroupe des organismes ayant des caractéristiques anatomiques et physiologiques similaires à celles de l'ordre, mais encore plus spécifiques.
- **Genre :** Le genre est un niveau de classification qui regroupe des organismes ayant des caractéristiques anatomiques et physiologiques similaires à celles de la famille, mais encore plus spécifiques.

- **Espèce :** L'espèce est le niveau de classification le plus spécifique, regroupant des organismes ayant des caractéristiques anatomiques et physiologiques similaires et capables de se reproduire entre eux pour produire une descen dances fertile.

CHAPITRE 2 :

Deep learning

1. Intelligence artificielle

L'intelligence artificielle (IA) représente une branche de l'informatique dont l'objectif est de développer des systèmes capables d'exécuter des tâches pour lesquelles une intelligence humaine est normalement requise. Ces tâches incluent la reconnaissance de la parole, la vision par ordinateur, la traduction automatique, la prise de décision et bien d'autres. Pour cela, l'IA utilise des algorithmes et des modèles mathématiques pour assimiler des données et améliorer ses performances au fil du temps. L'IA est aujourd'hui omniprésente dans de nombreux domaines tels que la médecine, la finance, la sécurité, les transports, l'agriculture, l'énergie et est une pièce maîtresse de la révolution numérique actuelle. Les applications de l'IA sont diverses, allant des assistants virtuels, des chatbots, des recommandations personnalisées aux voitures autonomes (Figure 4) [32].

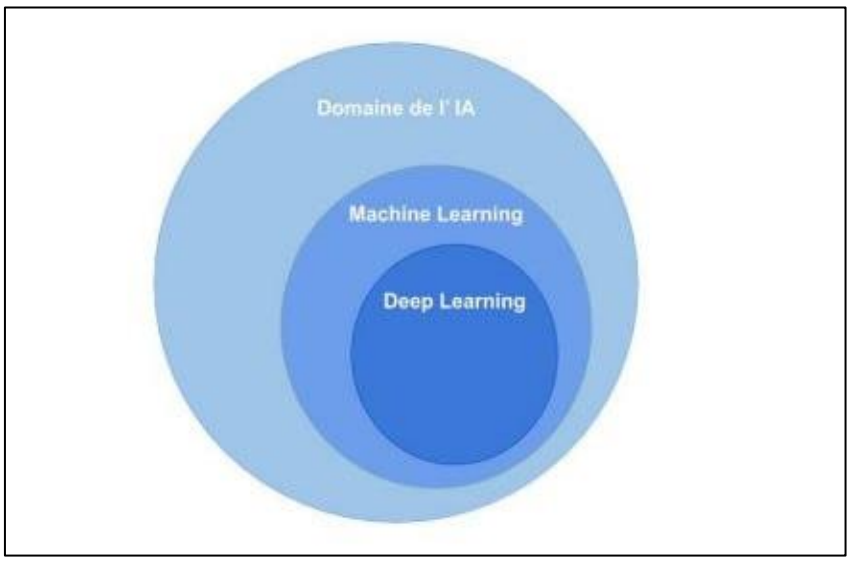

Figure 4 : Schématisation du rapport entre IA, ML et DL.

2. Deep learning

Depuis l'année 2006, le domaine de la recherche en apprentissage machine a été enrichi par l'avènement du deep learning, également connu sous le nom d'apprentissage hiérarchique ou structuré profond. Cette nouvelle approche a apporté des avancées significatives dans la capacité des machines à apprendre à partir de données et à effectuer des tâches complexes.

Le deep learning fait référence à une série d'algorithmes utilisés en apprentissage machine, qui visent à apprendre à plusieurs niveaux d'abstraction en utilisant des réseaux de neurones artificiels.

Le deep learning est une catégorie de techniques d'apprentissage automatique qui utilisent de nombreuses couches de traitement non linéaire pour extraire et transformer des caractéristiques de données supervisées ou non supervisées. Ces techniques sont également utilisées pour l'analyse et la classification de modèles de données (Tableau 1) [33].

Tableau 1 : Différence entre le deep learning et la programmation classique.

	Deep Learning	**Programmation classique**
Entrée	Data sets	Données
Exécution	Réseaux de neurones	Algorithmes
Sorties	Modèle	Programme

3. Apprentissage automatique

L'apprentissage automatique, également connu sous le nom de machine learning, est un domaine de l'intelligence artificielle qui consiste à entraîner des algorithmes à partir de données pour qu'ils puissent effectuer des tâches spécifiques sans être explicitement programmés. Plus précisément, l'apprentissage automatique utilise des modèles mathématiques qui identifient des motifs ou des structures dans les données, permettant ainsi de prendre des décisions ou de prédire des résultats pour de nouvelles données[34].

L'apprentissage supervisé et l'apprentissage non supervisé sont deux types d'algorithmes d'apprentissage automatique utilisés pour extraire des informations à partir de données :

3.1. Apprentissage supervisé

L'apprentissage supervisé implique l'utilisation d'un ensemble de données d'entraînement étiquetées pour apprendre à prédire les résultats pour de nouvelles données. En d'autres termes, l'algorithme reçoit des entrées et les sorties correspondantes, puis apprend à prédire les sorties pour de nouvelles entrées. Les exemples courants d'apprentissage supervisé incluent la classification (par exemple, la classification d'images) et la régression (par exemple, la prédiction du prix d'une maison en fonction de diverses caractéristiques)[35].

3.2. Apprentissage non supervisé

L'apprentissage non supervisé, en revanche, ne nécessite pas de données étiquetées pour apprendre à partir des données. L'algorithme recherche des modèles dans les données en utilisant des techniques telles que la réduction de dimensionnalité et le clustering. Les exemples courants d'apprentissage non supervisé incluent la segmentation d'images, la détection d'anomalies et l'analyse de texte[36].

4. Réseaux de neurones

Les réseaux de neurones sont des modèles informatiques qui s'inspirent du fonctionnement du cerveau pour résoudre des tâches complexes. Ils sont composés de plusieurs couches de neurones connectés qui traitent l'information en effectuant des opérations mathématiques sur les entrées et les poids des connexions entre les neurones affiché dans les Figures 5 et 6 [37].

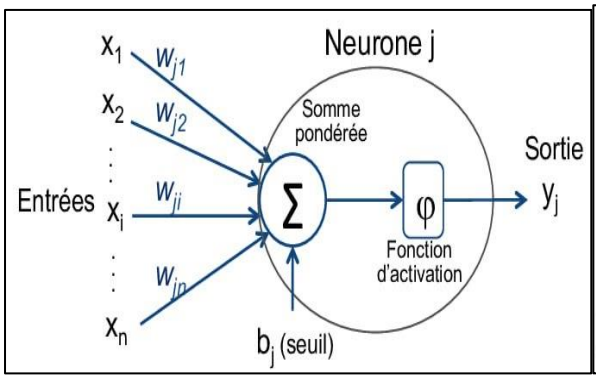

Figure 5 : Structure d'un neurone artificielle*[38]*.

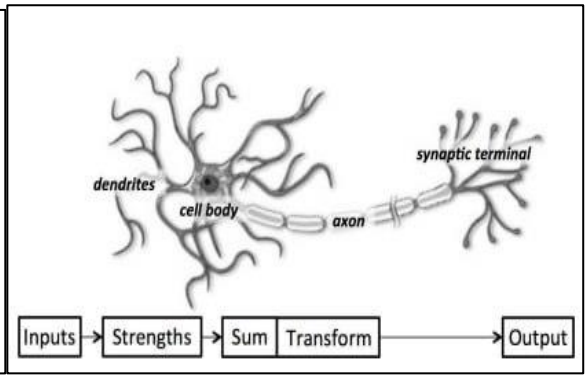

Figure 6 : Description fonctionnelle de la Structure d'un neurone*[39]*.

4.1. Types des Réseaux de neurones

- Perceptrons

Les perceptrons sont des réseaux de neurones artificiels à une seule couche qui ont été créés dans les années 1950 et 1960 pour résoudre des problèmes de classification binaire tels que la détection de spam ou la reconnaissance de caractères. Ils se basent sur un modèle linéaire simple et utilisent une fonction de seuil pour produire une sortie montré dans la Figure 7. Bien qu'ils aient une capacité limitée à résoudre des problèmes complexes, les perceptrons ont servi de base pour le développement de réseaux de neurones multicouches[40].

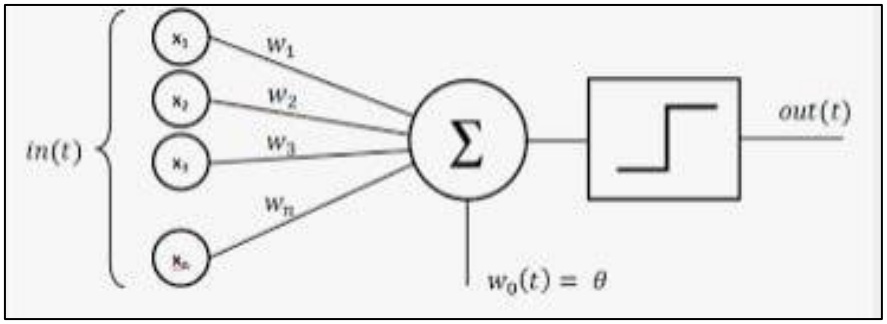

Figure 7 : Modèle de perceptrons sous forme graphique (rédac 2021).

- Réseaux de neurones multicouches

Les réseaux de neurones multicouches, aussi connus sous le nom de réseaux de neurones profonds, sont des réseaux de neurones artificiels qui comportent plusieurs couches cachées illustré dans le schéma de la Figure 8. Ils ont vu le jour dans les années 1980 et ont permis de résoudre des problèmes de classification plus complexes tels que la reconnaissance d'images ou la traduction automatique. Les réseaux de neurones multicouches sont capables de saisir des caractéristiques plus complexes et abstraites des données d'entrée grâce à l'utilisation de couches cachées. Ils sont devenus l'un des outils les plus couramment utilisés pour le deeplearning[41].

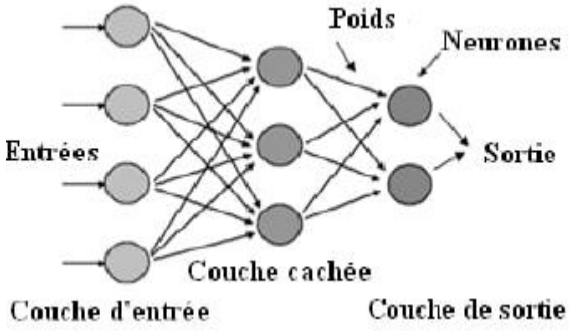

Figure 8 : Réseaux de neurones multicouches(« Figure 2.

Diagramme Schématique Du Réseau de Neurone MLP » s. d.).

- Réseaux de neurones récurrents

Les réseaux de neurones récurrents sont des réseaux de neurones qui ont des connexions récurrentes, ce qui signifie que la sortie d'un neurone peut être utilisée comme entrée pour le même neurone à l'étape suivante. Ils sont utilisés pour modéliser des séquences temporelles, telles que des séries temporelles de données ou des séquences de texte. Les réseaux de neurones récurrents sont capables de capturer des dépendances à long terme dans les données d'entrée grâce à leur mécanisme de rétroaction. Ils ont été utilisés pour des applications telles que la prédiction de la parole et la génération de texte[43].

- Réseaux de neurones convolitifs

Les réseaux de neurones convolutionnels (ou CNN en anglais pour Convolutional Neural Networks) sont une famille de réseaux de neurones artificiels spécialement conçus pour le traitement de données structurées telles que les images, les vidéos ou les signaux sonores.

Ces réseaux sont basés sur des opérations de convolution qui permettent de capturer des motifs locaux dans les données d'entrée. Les couches de convolution sont généralement suivies de couches de pooling qui réduisent la taille de la représentation en conservant les informations les plus importantes [44]. Les couches des CNN sont :

- La couche de convolution (CONV) qui traite les données d'un champ récepteur.
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la
- taille de l'image intermédiaire (souvent par sous-échantillonnage).
- La couche de correction (ReLU), souvent appelée par abus 'ReLU' en référence à la

- fonction d'activation (Unité de rectification linéaire).
- La couche "entièrement connectée" (FC), qui est une couche de type perceptron.
- La couche de perte (LOSS).

4.2. Fonctions de perte et leur minimisation

Lors de l'entraînement d'un réseau de neurones, il est crucial de choisir une fonction de perte appropriée qui mesure la différence entre la sortie prédite et la valeur réelle[45]. L'objectif de l'apprentissage du réseau de neurones est de minimiser cette fonction de perte en ajustant les poids des connexions entre les neurones pour améliorer les prédictions futures. Il existe plusieurs fonctions de perte couramment utilisées dans les réseaux de neurones, notamment la fonction de perte quadratique ou MSE (MeanSquaredError) est utilisée pour les tâches de régression, la fonction de perte d'entropie croisée ou CE (Cross-Entropy) est utilisée pour les tâches de classification et la fonction de perte de log-vraisemblance négative ou NLL (Negative Log-Likelihood) est également utilisée pour la classification également[45].

4.3. Fonctions d'activation

Les fonctions d'activation sont des fonctions mathématiques utilisées dans les réseaux de neurones pour introduire de la non-linéarité dans le modèle. Voici quelques exemples de fonctions d'activation couramment utilisées, avec une brève description et une référence pour en savoir plus :

- Fonction d'activation sigmoïde

La fonction sigmoïde est un outil couramment utilisé dans les réseaux de neurones pour classifier des données en deux catégories distinctes. Elle prend une valeur numérique en entrée et produit une valeur comprise entre 0 et 1 en sortie. La formule de la fonction sigmoïde est f(x) = 1 / (1 + exp(-x)) voir la Figure 9 [46].

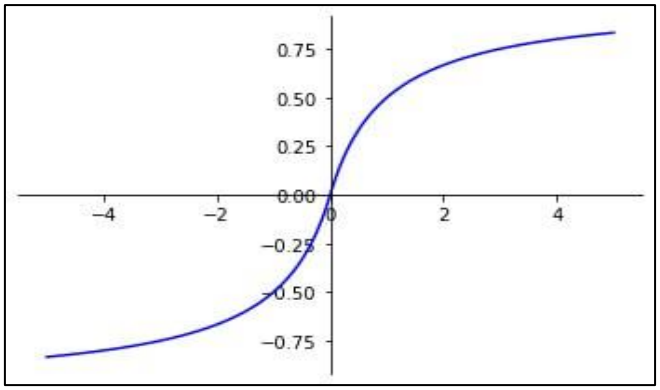

Figure 9 : Courbe la fonction sigmoid (« fonctions d’activation sigmoid - Google Search » s. d.).

- Fonction d'activation ReLU

La fonction ReLU est une fonction mathématique qui est largement utilisée dans les réseaux de neurones profonds. Elle est simple et non linéaire. Elle prend une valeur numérique en entrée et renvoie la même valeur si elle est positive, mais si elle est négative ou égale à zéro, elle renvoie zéro. La fonction ReLU est définie comme suit : f(x) = max(0,x) voir la Figure 10 [47].

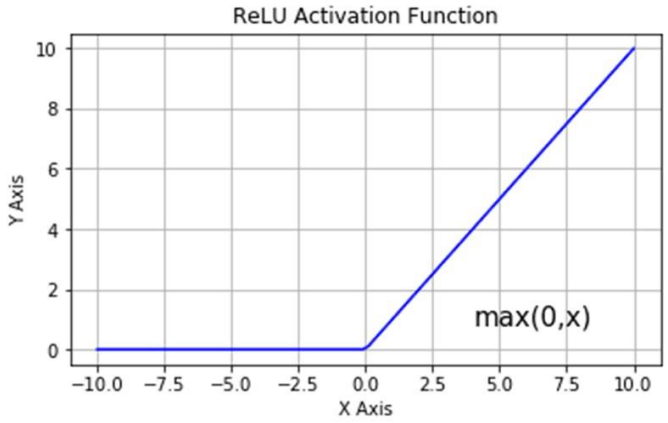

Figure 10 : Courbe la fonction ReLu (Naveen 2022).

- Fonction d'activation tanh

La fonction tanh est souvent employée comme fonction d'activation dans les réseaux de neurones et ressemble à la fonction sigmoïde, mais ses valeurs se situent entre -1 et 1. Sa formule est la suivante : f(x) = (exp(x) - exp(-x)) / (exp(x) + exp(-x)) voir la Figure 11 [48].

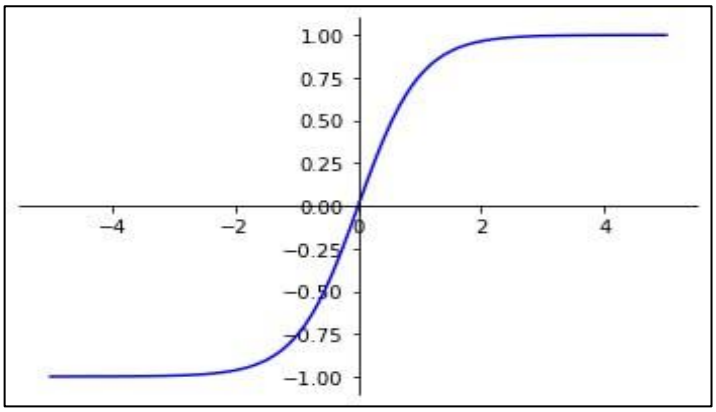

Figure 11 : Courbe la fonction tanh (Keldenich 2021a).

- Fonction d'activation softmax

La fonction softmax est couramment employée dans les réseaux de neurones pour effectuer la classification de données dans plusieurs catégories. Elle transforme un vecteur de valeurs réelles en une distribution de probabilités normalisées représentant les différentes classes. La formule de la fonction soft max est la suivante : pour chaque élément du vecteur d'entrée, on calcule son exponentielle, on divise cette valeur par la somme des exponentielles de tous les éléments du vecteur d'entrée, ce qui permet d'obtenir une valeur normalisée entre 0 et 1 représentant la probabilité de cette classe. f(x_i) = exp(x_i) / sum(exp(x_j)) voir la Figure 12.

Il existe de nombreuses autres fonctions d'activation, chacune ayant ses propres avantages et inconvénients[49].

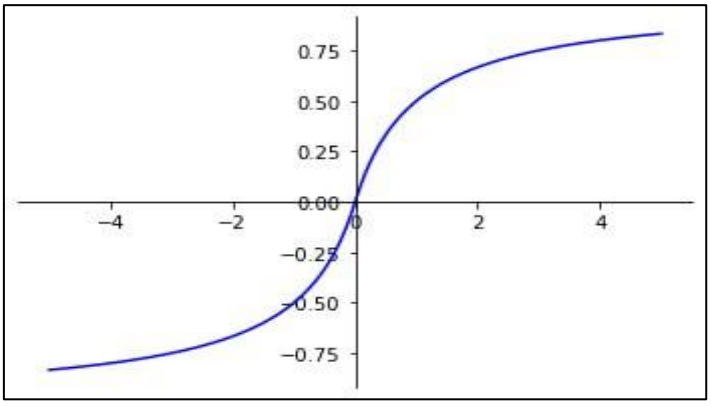

Figure 12 : Courbe la fonction softmax (Keldenich 2021b).

4.4. Hyperparamètres

Les réseaux de neurones sont des modèles d'apprentissage en profondeur (deeplearning) qui peuvent avoir des millions de paramètres. Les hyper paramètres des réseaux de neurones sont des variables qui affectent le processus d'apprentissage, mais qui ne sont pas directement appris à partir des données. Les hyper paramètres doivent être réglés de manière appropriée pour obtenir les meilleures performances de l'algorithme [50].

Voici quelques exemples d'hyper paramètres courants pour les réseaux de neurones :

- **Le nombre de couches cachées**
- **Le nombre de neurones dans chaque couche**
- **Le taux d'apprentissage**
- **La fonction d'activation**
- **Le nombre d'itérations d'entraînement**

Il existe de nombreux autres hyper paramètres qui peuvent être ajustés pour optimiser les performances des réseaux de neurones, tels que le taux de régularisation, le type d'optimiseur, la taille de lot (batch size), etc. [50].

PARTIE 2 :

MATÉRIEL ET MÉTHODES

1. MATÉRIEL

1.1. Données biologiques

La présente étude se concentre sur l'identification de la bactérie *Escherichia coli* en utilisant des modèles d'apprentissage profond. Pour ce faire, nous avons utilisé un ensemble de données récupéré depuis la banque de données **SILVA [https://www.arb-silva.de/]**. Cet ensemble comprend 10 000 séquences d'ARNr 16S d'entérobactéries, dont 5000 séquences d'*Escherichia coli* et d'autres types d'*Escherichia*, et 5000 séquences d'autres types d'entérobactéries. Nous avons utilisé les séquences ARNr 16S car ils sont efficaces pour la classification taxonomique des bactéries.

1.2. Outils et bibliothèques

Nous décrivons brièvement les outils utilisés pour réaliser ce travail dans le tableau 2.

Tableau 2 : Principaux outils utilisés.

Outil	**Description**
Python	Python est un langage de programmation polyvalent et puissant, qui offre une syntaxe claire et concise.
Jupyter	Jupyter est une application web open-source qui permet de créer et de partager des documents interactifs appelés "notebooks". Ces notebooks permettent d'exécuter du code en temps réel, d'afficher des graphiques et des visualisations, et de documenter les étapes de votre travail.
Notepad	programme informatique simple destiné à la création et à la modification de fichiers texte. C'est un éditeur de texte basique, souvent inclus dans les systèmes d'exploitation Window
Google Colab	Google Colab est un service de Cloud Computing gratuit basé sur Jupyter Notebook. Il permet d'exécuter du code Python, y compris des bibliothèques telles que TensorFlow, sur des machines virtuelles hébergées par Google. Colab offre également un accès gratuit aux GPU et aux TPU (TensorProcessingUnits) pour accélérer les calculs liés à l'apprentissage automatique.

Les bibliothèques utilisées dans ce travail sont décrites brièvement dans le tableau 3.

Tableau 3 : Différents bibliothèques python utilisées.

Bibliothèque	**Description**
numpy	Bibliothèque pour le calcul numérique en Python. Elle fournit des structures de données et des fonctions pour manipuler des tableaux multidimensionnels et effectuer des opérations mathématiques sur ces tableaux de manière efficace.
tensorflow	Bibliothèque open-source d'apprentissage automatique et de calcul numérique développée par Google. Elle permet de construire, d'entraîner et de déployer des modèles d'apprentissage automatique, y compris des réseaux de neurones profonds.
sklearn	Bibliothèque open-source d'apprentissage automatique en Python. Elle propose divers outils pour la préparation des données, le prétraitement, la sélection de modèles, l'évaluation des performances, etc. Dans le code donné, la fonction "train_test_split" de scikit-learn est utilisée pour diviser les données en ensembles d'apprentissage et de test.
matplotlib	Bibliothèque pour la création de visualisations en Python. Elle permet de générer des graphiques, des diagrammes et des figures pour afficher les résultats, les métriques et les tendances. Dans le code fourni, matplotlib est utilisé pour tracer les courbes de perte et de précision du modèle d'apprentissage.
biophython	bibliothèque open-source et un ensemble d'outils de programmation en Python conçus pour faciliter l'analyse bioinformatique. Elle fournit une vaste gamme de modules et de fonctionnalités permettant la manipulation, la visualisation et l'interprétation des données biologiques.

2. MÉTHODES

2.1. Prétraitement de données

Après le téléchargement des séquences, on fait une série de manipulation :

Tout d'abord, on utilise la bibliothèque Biopython qui doit être installée via la commande "pip install biopython". Biopython est une bibliothèque Python utilisée pour traiter des données biologiques, notamment des séquences d'ARNr 16s.

Ensuite, on ouvre un fichier d'entrée au format FASTA contenant des séquences d'ARNr 16s. On ouvre également un fichier de sortie où les séquences d'ARNr 16s traitées seront écrites.

On parcourt chaque séquence d'ARNr 16s dans le fichier d'entrée et on écrit la séquence dans le fichier de sortie, en la convertissant en chaîne de caractères. Chaque séquence est écrite sur une nouvelle ligne dans le fichier de sortie.

Ensuite, on ouvre le fichier de sortie précédent et on lit toutes les lignes contenant les séquences d'ARNr 16s.

On limite la longueur de chaque séquence à 1300 caractères en enlevant les caractères excédentaires.

Les séquences modifiées sont écrites dans un nouveau fichier texte

On complète les séquences plus courtes avec le gap "-" pour atteindre la longueur maximale.

On ajoute un espace entre chaque élément de chaque séquence.

La catégorie des séquences *d'Escherichia coli* et les autres types *d'Escherichia* sont notées par 1. Les autres catégories d'entérobactéries sont notées par 0.

Le code ci-dessous (Figure13) montre les bibliothèques utilisées et comment lire un fichier contenant les séquences.

```
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns

# Charger les données à partir du fichier
data_file = '/content/fichier final des seq.txt'

sequences = []
labels = []

with open(data_file, 'r') as file:
    for line in file:
        line = line.strip().split()
        seq = line[:-1]
        label = int(line[-1])
        sequences.append(seq)
        labels.append(label)
```

Figure 13 : bibliothèques et lecture du fichier contenant les séquences.

2.2. Encodage des séquences

En utilisant la fonction *encode_sequence ()*, les données sont converti en représentation One-Hot. Cette méthode prend chaque séquence en entrée et l'encode en utilisant l'encodage One-Hot. Chaque base (A, U, C, G) est représentée par un vecteur de taille 5, où chaque élément représente la présence ou l'absence de la base correspondante (Figure 14).

```
encoded_sequences = [encode_sequence(seq) for seq in sequences]

# Fonction pour encoder les séquences avec one-hot encoding
def encode_sequence(seq):
    encoded_seq = np.zeros((len(seq), 5))
    mapping = {'A': 0, 'U': 1, 'C': 2, 'G': 3}
    for i, char in enumerate(seq):
        if char in mapping:
            encoded_seq[i, mapping[char]] = 1
        else:
            encoded_seq[i, 4] = 1
    return encoded_seq
```

Figure 14 : Encodage des séquences.

2.3. Division des données en ensembles d'apprentissage

Cette méthode permet de diviser les données encodées (encoded_sequences) et les étiquettes (labels) en ensembles d'apprentissage et de test à l'aide de la fonction *train_test_split()*. Les données d'apprentissage représentent 80% de l'ensemble de données, et les données de test représentent 20%. L'argument test_size spécifie la taille de l'ensemble de test (Figure 15).

```
train_sequences, test_sequences, train_labels, test_labels = train_test_split(encoded_sequences, labels, test_size=0.2, random_state=42)
```

Figure 15 : Division des données.

2.4. Construction du modèle

Nous avons commencé à définir le modèle d'apprentissage profond à l'aide de la fonction *tf.keras.Sequential()*. Puis, nous avons ajouté des couches au modèle, notamment une couche de convolution (Conv1D), une couche de max pooling (MaxPooling1D), une couche de mise à plat (Flatten), une couche dense (Dense), et une couche de sortie avec activation softmax pour la classification binaire (Figure 16).

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv1D(128, 5, activation='relu', input_shape=(1300, 5)),
    tf.keras.layers.MaxPooling1D(2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Figure 16 : Définition du modèle d'apprentissage profond et Ajout des couches au modèle.

2.5. Compilation du modèle

La fonction *model.compile()* est utilisée pour spécifier l'optimiseur (adam), la fonction de perte (binary_crossentropy pour la classification binaire.), et les métriques d'évaluation (accuracy) (Figure 17).

```
# Compiler le modèle
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Figure 17 : Compilation du modèle.

2.6. Entraînement du modèle

L'entraînement du modèle est effectué à l'aide de la fonction *model.fit()* en fournissant les données d'apprentissage (train_sequences et train_labels), le nombre d'époques, la taille des lots, et les données de validation (test_sequences et test_labels) (Figure 18).

```
history=model.fit(train_sequences, train_labels, epochs=40, batch_size=32, validation_data=(test_sequences, test_labels))
```

Figure 18 : Entraînement du modèle.

2.7. Extraction des métriques d'entraînement et de validation

- *train_loss :* C'est la valeur de la fonction de perte lors de l'entraînement du modèle à chaque époque.
- *val_loss :* C'est la valeur de la fonction de perte lors de la validation du modèle à chaque époque.
- *train_accuracy :* C'est la précision du modèle sur l'ensemble d'entraînement à chaque époque. La précision est une mesure de la performance du modèle qui indique la proportion des cas correctement classés par rapport au nombre total des cas dans l'ensemble d'entraînement.
- *val_accuracy :* C'est la précision du modèle sur l'ensemble de validation à chaque époque. La précision de validation est similaire à la précision d'entraînement, mais elle est calculée sur l'ensemble de validation (Figure 19).

```
train_loss = history.history['loss']
val_loss = history.history['val_loss']
train_accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']
```

Figure 19 : Extraire des métriques d'entraînement et de validation de l'historique.

2.8. Évaluation du modèle sur l'ensemble de test

- *test_loss, test_accuracy = model.evaluate(test_sequences, test_labels)* : Cette ligne évalue le modèle en utilisant les données de test test_sequences et les étiquettes de test

correspondantes test_labels. La méthode evaluate calcule la perte (loss) et la précision (accuracy) du modèle sur cet ensemble de test.

- *print("Test Loss:", test_loss)* : Cette ligne affiche la perte (loss) obtenue par le modèle lors de l'évaluation sur l'ensemble de test. La perte est une mesure de l'erreur du modèle, où des valeurs plus faibles indiquent de meilleures performances.
- *print("Test Accuracy:", test_accuracy)* : Cette ligne affiche la précision (accuracy) obtenue par le modèle lors de l'évaluation sur l'ensemble de test. La précision est une mesure de la justesse du modèle, c'est-à-dire la proportion de prédictions correctes parmi toutes les prédictions effectuées (Figure 20).

```
test_loss, test_accuracy = model.evaluate(test_sequences, test_labels)

print("Test Loss:", test_loss)
print("Test Accuracy:", test_accuracy)
```

Figure 20 : Évaluation du modèle sur l'ensemble de test.

2.9. Affichage des résultats d'entraînements (perte et précision)

Le code affiche les résultats d'entraînement en accédant aux valeurs enregistrées dans l'objet history (Figure 21). Les lignes suivantes sont utilisées pour cela :

```
print('training loss:' , history.history['loss'])
print('validation loss:' , history.history['loss'])
print('training accuracy:' , history.history['accuracy'])
print('validation accuracy:' , history.history['accuracy'])
```

Figure 21 : Affichage des résultats d'entraînements.

2.10. Affichage des courbes de perte et de précision

Nous avons tracé les courbes de perte pour l'apprentissage et la validation en utilisant la méthode *"plot_loss(history)"* (Figure 22).

```
plt.plot(train_loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training and Validation Loss')
plt.legend()
plt.show()
```

Figure 22 : Affichage des courbe de perte et de précisionMéthode *"plot_loss(history)"*.

La méthode *"plot_accuracy(history)"* trace les courbes de précision pour l'apprentissage et la validation (Figure 23).

```
plt.plot(train_accuracy, label='Training Accuracy')
plt.plot(val_accuracy, label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()
plt.show()
```

Figure 23 : Affichage des courbe de perte et de précision Méthode *"plot_accuracy(history)"*.

2.11. Affichage de la matrice de confusion

- *predictions = model.predict(test_sequences) :* Cette ligne de code utilise le modèle (model) pour faire des prédictions sur les données de test (test_sequences). Les test_sequences doivent être pré-traitées et transformées en séquences appropriées pour le modèle (Figure 24).
- *predictions = np.round(predictions).flatten() :* Les prédictions initiales retournées par le modèle peuvent être des valeurs continues. Cette ligne de code arrondit ces valeurs à la valeur la plus proche (0 ou 1) et les aplatit en un tableau à une seule dimension (Figure 24).
- *confusion_mtx = confusion_matrix(test_labels, predictions) :* Cette ligne de code utilise la fonction confusion_matrix de la bibliothèque sklearn.metrics pour calculer la matrice de

confusion. La matrice de confusion est une table qui montre le nombre de prédictions correctes et incorrectes pour chaque classe (Figure 24).

- *plt.figure(figsize=(8, 6)):* Cette méthode de code crée une nouvelle figure pour afficher la matrice de confusion. Les dimensions de la figure sont spécifiées avec la taille (8, 6) (Figure 24).
- *sns.heatmap(confusion_mtx, annot=True, fmt='d', cmap='Blues') :* Cette méthode de code utilise la fonction heatmap de la bibliothèque seaborn pour créer une représentation graphique de la matrice de confusion. La matrice de confusion est passée en tant qu'argument, et *annot=True* permet d'afficher les valeurs numériques dans chaque cellule. *fmt='d'* spécifie que les valeurs doivent être affichées en tant qu'entiers. cmap='Blues' définit le schéma de couleurs utilisé pour le graphique (Figure 24).
- *plt.xlabel('Prédictions') et plt.ylabel('Vraies étiquettes') :* Ces méthodes de code définissent les étiquettes des axes x et y du graphique (Figure 24).

```
# Faire des prédictions sur l'ensemble de test
predictions = model.predict(test_sequences)
predictions = np.round(predictions).flatten()

# Calculer la matrice de confusion
confusion_mtx = confusion_matrix(test_labels, predictions)

# Afficher la matrice de confusion sous forme graphique
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_mtx, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Prédictions')
plt.ylabel('Vraies étiquettes')
```

Figure 24 : Affichage de la matrice de confusion.

PARTIE 3 :

RÉSULTATS ET DISCUSSION

1. Résultats et discussions de l'historique de l'entrainement

L'entraînement a été effectué pendant 40 époques (epochs). Une époque correspond à une itération complète sur l'ensemble des données d'entraînement. Pour chaque époque, nous avons des informations sur le temps d'exécution, la valeur de la fonction de perte (loss), et la précision (accuracy) du modèle. Au début de l'entraînement, la précision est d'environ 76% et la perte est relativement élevée. Cela suggère que le modèle ne fait pas encore de bonnes prédictions (Figure 25).

```
Epoch 1/40
268/268 [==============================] - 46s 168ms/step - loss: 0.5503 - accuracy: 0.7681 - val_loss: 0.3543 - val_accuracy: 0.8349
Epoch 2/40
268/268 [==============================] - 42s 156ms/step - loss: 0.3203 - accuracy: 0.8509 - val_loss: 0.3206 - val_accuracy: 0.8485
Epoch 3/40
268/268 [==============================] - 42s 156ms/step - loss: 0.2680 - accuracy: 0.8814 - val_loss: 0.2754 - val_accuracy: 0.8943
Epoch 4/40
268/268 [==============================] - 53s 198ms/step - loss: 0.2451 - accuracy: 0.8950 - val_loss: 0.2507 - val_accuracy: 0.8891
Epoch 5/40
268/268 [==============================] - 43s 160ms/step - loss: 0.2276 - accuracy: 0.9032 - val_loss: 0.2496 - val_accuracy: 0.8784
```

Figure 25 : Figure montre les résultats au début de l'entraînement.

À mesure que l'entraînement progresse, la précision augmente progressivement, atteignant finalement environ 95% à la fin des 40 époques. En même temps, la perte diminue régulièrement (Figure 26).

```
Epoch 35/40
268/268 [==============================] - 42s 156ms/step - loss: 0.1123 - accuracy: 0.9537 - val_loss: 0.2106 - val_accuracy: 0.9341
Epoch 36/40
268/268 [==============================] - 41s 153ms/step - loss: 0.1078 - accuracy: 0.9549 - val_loss: 0.1965 - val_accuracy: 0.9457
Epoch 37/40
268/268 [==============================] - 42s 157ms/step - loss: 0.1060 - accuracy: 0.9559 - val_loss: 0.2065 - val_accuracy: 0.9443
Epoch 38/40
268/268 [==============================] - 41s 154ms/step - loss: 0.1052 - accuracy: 0.9571 - val_loss: 0.2136 - val_accuracy: 0.9322
Epoch 39/40
268/268 [==============================] - 42s 157ms/step - loss: 0.1053 - accuracy: 0.9578 - val_loss: 0.2144 - val_accuracy: 0.9308
Epoch 40/40
268/268 [==============================] - 42s 155ms/step - loss: 0.1028 - accuracy: 0.9561 - val_loss: 0.2537 - val_accuracy: 0.9191
```

Figure 26 : Figure montre les résultats de fin de l'entraînement.

Les résultats des performances de la perte et de la précision sont alors comme suit :

- Test Loss: **0.25371822714805603** : Cela représente la perte (loss) obtenue lors de l'évaluation du modèle sur l'ensemble de test. Plus la valeur est faible, mieux c'est, car cela indique une meilleure capacité du modèle à prédire correctement les classes des entérobactéries.
- Test Accuracy: **0.9190832376480103** : Il s'agit de la précision (accuracy) obtenue lors de l'évaluation du modèle sur l'ensemble de test. C'est le pourcentage de prédictions correctes

par rapport au nombre total d'échantillons dans l'ensemble de test. Une valeur élevée de précision indique une bonne performance du modèle.

- Training Loss et Validation Loss :

  Ce sont les valeurs de perte (loss) obtenues lors de l'entraînement du modèle sur l'ensemble d'apprentissage et la validation à chaque époque. La perte d'entraînement représente la performance du modèle sur les données d'apprentissage, tandis que la perte de validation est une mesure de la performance sur les données de validation. L'objectif est de réduire la perte à chaque époque pour améliorer les performances du modèle.

- Training Accuracy et Validation Accuracy :

  Ce sont les valeurs de précision (accuracy) obtenues lors de l'entraînement du modèle sur l'ensemble d'apprentissage et la validation à chaque époque. La précision d'entraînement représente la performance du modèle sur les données d'apprentissage, tandis que La précision de validation mesure la performance sur les données de validation. L'objectif est d'augmenter La précision à chaque époque pour améliorer les performances du modèle.

- Les courbes de perte dans la Figure 27 montrent la variation de la perte d'entraînement et de la perte de validation au fil des époques. Idéalement, nous remarquons que les courbes de perte diminuent progressivement au fur et à mesure de l'entraînement, indiquant une convergence du modèle.

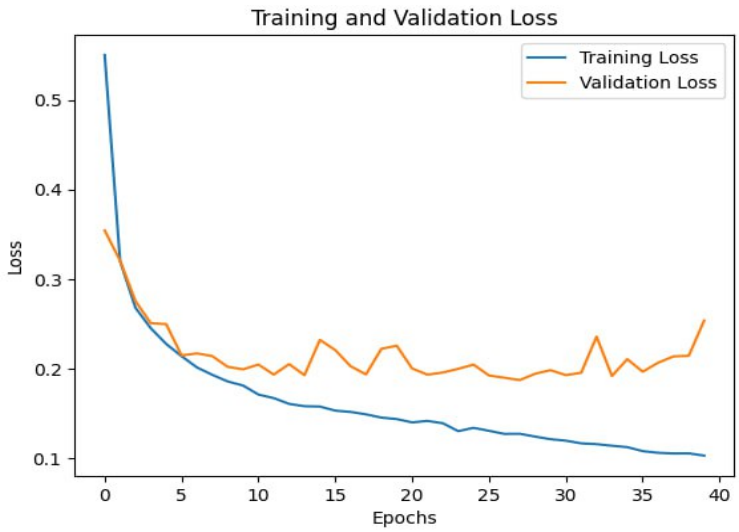

Figure 27 : Courbes de perte (Loss).

- Les courbes de précision dans la Figure 28 montrent la variation de la précision d'entraînement et de la précision de validation au fil des époques. Nous remarquons que les courbes de précision augmentent progressivement au fur et à mesure de l'entraînement, indiquant une amélioration de la performance du modèle.

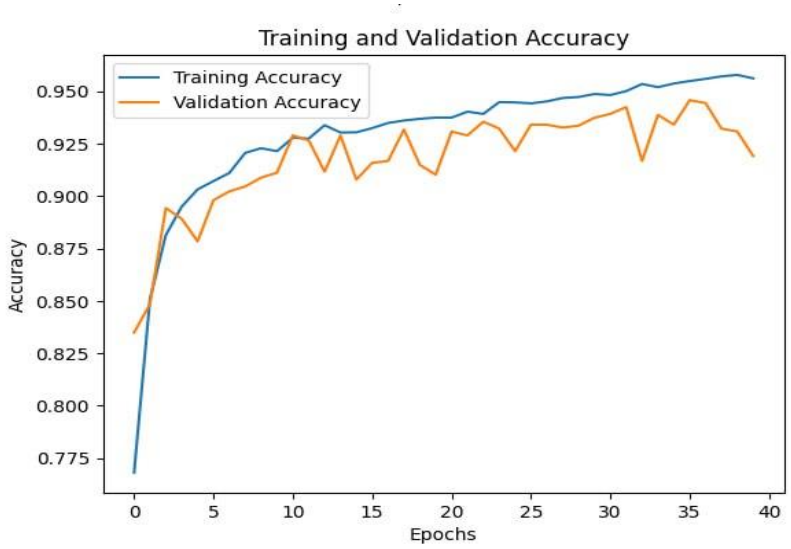

Figure 28 : Courbes de précision (Accuracy).

1.2. Résultats et discussion de la matrice de confusion

La matrice de confusion est un outil de visualisation qui permet d'évaluer les performances d'un modèle de classification en comparant les prédictions du modèle avec les valeurs réelles. Elle est généralement représentée sous forme de tableau avec deux classes, dans notre cas les classes sont 0 et 1.

- Les vraies négatives (VN) : sont les échantillons qui sont réellement de la classe négative (0) et qui ont été prédites correctement comme étant de la classe négative. Dans notre cas, il y a 1267 échantillons qui sont réellement de la classe négative et qui ont été prédites correctement comme telles.
- Les vraies positives (VP) : sont les échantillons qui sont réellement de la classe positive (1) et qui ont été prédites correctement comme étant de la classe positive. nous avons 698 échantillons qui sont réellement de la classe positive et qui ont été prédites correctement comme telles.
- Les faux positifs (FP) : sont les échantillons qui sont réellement de la classe négative (0), mais qui ont été incorrectement prédites comme étant de la classe

positive (1). Dans notre cas, il y a 37 échantillons qui sont réellement de la classe négative, mais qui ont été prédites comme étant de la classe positive.

- Les faux négatifs (FN) : sont les échantillons qui sont réellement de la classe positive (1), mais qui ont été incorrectement prédites comme étant de la classe négative (0). Nous avons 136 échantillons qui sont réellement de la classe positive, mais qui ont été prédites comme étant de la classe négative.

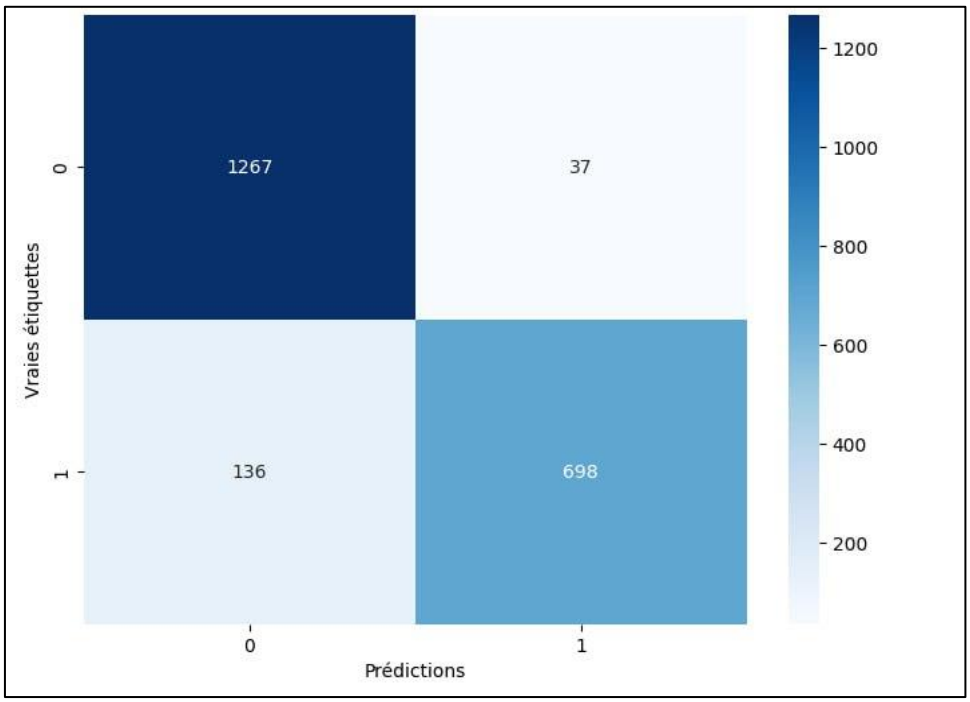

Figure 29 : Matrice de confusion.

En résumé, la matrice de confusion indique que le modèle a correctement prédit 1267 échantillons de la classe 0 et 698 échantillons de la classe 1. Cependant, il a également fait des erreurs en prédisant 136 échantillons de la classe 0 comme étant de la classe 1 et en prédisant 37 échantillons de la classe 1 comme étant de la classe 0.

2. CONCLUSION

Les résultats obtenus pour le modèle d'apprentissage profond utilisé dans la classification taxonomique des bactéries pour identifier la bactérie *Escherichia coli* à partir des données génomiques sont hautement prometteurs.

. Les performances sur les ensembles d'entrainement et de test attestent de la capacité du modèle à généraliser efficacement à de nouvelles données. Les résultats de la matrice de confusion démontrent également que le modèle a réussi à identifier de manière précise les séquences *d'Escherichia coli* par rapport autres entérobactéries .Ces résultats soutiennent l'efficacité du DL en tant qu'approche pour la classification taxonomique des bactéries à partir de données génomiques, notamment les séquences ARNr 16s. Ces résultats ouvrent la voie à de futures applications dans la classification taxonomique des bactéries.

RÉFÉRENCES

[1] « biochemistry-stryer-5th-ed.pdf ». Consulté le: 24 mars 2023. [En ligne]. Disponible sur: https://biokamikazi.files.wordpress.com/2013/10/biochemistry-stryer-5th-ed.pdf

[2] http://library.lol/main/56FA3219E1BDC434C3179298C960A81F (Consulté le 24 mars 2023).

[3] R. R. Sinden, *DNA structure and function*. San Diego: Academic Press, 1994.

[4] N. CHAFFEY, « Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P. Molecular biology of the cell. 4th edn. », *Annals of Botany*, vol. 91, nº 3, p. 401, févr. 2003, doi: 10.1093/aob/mcg023.

[5] O. L. de Weck, « Case 4: DNA Sequencing », in *Technology Roadmapping and Development : A Quantitative Approach to the Management of Technology*, O. L. De Weck, Éd., Cham: Springer International Publishing, 2022, p. 521-533. doi: 10.1007/978-3-030-88346-1_18.

[6] « ALBERTS 6°, Molecular Biology of the Cell [PDF] | Online Book Share ». https://epage.pub/doc/alberts-6-molecular-biology-of-the-cell-35jorjno5y (consulté le 4 avril 2023).

[7] K. M. Weeks, « Advances in RNA structure analysis by chemical probing », *Current Opinion in Structural Biology*, vol. 20, nº 3, p. 295-304, juin 2010, doi: 10.1016/j.sbi.2010.04.001.

[8] « ch6.pdf ». Consulté le: 4 avril 2023. [En ligne]. Disponible sur: https://www2.nsysu.edu.tw/wzhlab/ch6.pdf

[9] M. Thellier, « Plant Memory vs. Animal and Human Memory », in *Plant Responses to Environmental Stimuli: The Role of Specific Forms of Plant Memory*, M. Thellier, Éd., Dordrecht: Springer Netherlands, 2017, p. 55-57. doi: 10.1007/978-94-024-1047-1_6.

[10] S. J. Dupas, D. Gussakovsky, A. Wai, M. J. F. Brown, G. Hausner, et S. A. McKenna, « Predicting human RNA quadruplex helicases through comparative sequence approaches and helicase mRNA interactome analyses », *Biochem Cell Biol*, vol. 99, nº 5, p. 536-553, oct. 2021, doi: 10.1139/bcb-2020-0590.

[11] R. R. Gutell, N. Larsen, et C. R. Woese, « Lessons from an evolving rRNA: 16S and 23S rRNA structures from a comparative perspective. », *Microbiol Rev*, vol. 58, nº 1, p. 10-26, mars 1994.

[12] C. R. Woese *et al.*, « Secondary structure model for bacterial 16S ribosomal RNA: phylogenetic, enzymatic and chemical evidence. », *Nucleic Acids Res*, vol. 8, nº 10, p. 2275-2293, mai 1980.

[13] « The kink-turn: a new RNA secondary structure motif ». https://www.embopress.org/doi/epdf/10.1093/emboj/20.15.4214 (consulté le 4 avril 2023).

[14] A.-C. Gingras, B. Raught, et N. Sonenberg, « Regulation of translation initiation by FRAP/mTOR », *Genes Dev.*, vol. 15, nº 7, p. 807-826, avr. 2001, doi: 10.1101/gad.887201.

[15] E. Masson, « Entérobactéries », *EM-Consulte*. https://www.em-consulte.com/article/60989/enterobacteries (consulté le 28 mars 2023).

[16] « Chapitre_4.pdf ».

[17] L. W. Riley, « Distinguishing Pathovars from Nonpathovars: Escherichia coli », *Microbiol Spectr*, vol. 8, nº 4, p. 8.4.1, déc. 2020, doi: 10.1128/microbiolspec.AME-0014-2020.

[18] K. J. Forsberg, A. Reyes, B. Wang, E. M. Selleck, M. O. A. Sommer, et G. Dantas, « The Shared Antibiotic Resistome of Soil Bacteria and Human Pathogens », *Science*, vol. 337, nº 6098, p. 1107-1111, août 2012, doi: 10.1126/science.1220761.

[19] A. Delétoile *et al.*, « Phylogeny and Identification of Pantoea Species and Typing of Pantoea agglomerans Strains by Multilocus Gene Sequencing », *Journal of Clinical Microbiology*, vol. 47, nº 2, p. 300-310, févr. 2009, doi: 10.1128/JCM.01916-08.
[20] S. T. Cowan, *Cowan and Steel's Manual for the Identification of Medical Bacteria*. Cambridge University Press, 1993.
[21] Z. I. Tahseen, M. H. Edham, et A. S. Karomi, « Study of some Virulence Factors for Clostridium perfringens isolatedfrom Clinical Samples and Hospital Environment and showing their Sensitivity to Antibiotics », *ijpqa*, vol. 11, nº 02, p. 253-256, juin 2020, doi: 10.25258/ijpqa.11.2.11.
[22] Z. S. Ulhaq, T. H. Hendyatama, F. Hameed, et D. Santosaningsih, « Antibacterial activity of Citrus hystrix toward Salmonella spp. infection », *Enfermedades infecciosas y microbiologia clinica (English ed.)*, vol. 39, nº 6, p. 283-286, juin 2021, doi: 10.1016/j.eimce.2020.05.016.
[23] S. T. Cowan, *Cowan and Steel's Manual for the Identification of Medical Bacteria*. Cambridge University Press, 1993.
[24] R. Cruickshank, J. P. Duguid, B. P. Marmion, et R. H. A. Swain, « Medical microbiology: the pratice of medical microbiology », in *Medical microbiology: the pratice of medical microbiology*, 1975, p. 587-587. Consulté le: 28 avril 2023. [En ligne]. Disponible sur: https://pesquisa.bvsalud.org/portal/resource/pt/biblio-1073604
[25] P. B. Smith, K. M. Tomfohrde, D. L. Rhoden, et A. Balows, « API System: a Multitube Micromethod for Identification of Enterobacteriaceae », *Appl Microbiol*, vol. 24, nº 3, p. 449-452, sept. 1972.
[26] S. C. Edberg, S. Pittman, et J. M. Singer, « Esculin hydrolysis by Enterobacteriaceae. », *J Clin Microbiol*, vol. 6, nº 2, p. 111-116, août 1977.
[27] « Test indole : Principe, procédure, résultats et interprétation ». https://microbiologie-clinique.com/test-indole.html (consulté le 1 mai 2023).
[28] O. Clermont, J. K. Christenson, E. Denamur, et D. M. Gordon, « The Clermont *Escherichia coli* phylo-typing method revisited: improvement of specificity and detection of new phylo-groups: A new *E. coli* phylo-typing method », *Environmental Microbiology Reports*, vol. 5, nº 1, p. 58-65, févr. 2013, doi: 10.1111/1758-2229.12019.
[29] O. Clermont, D. Gordon, et E. Denamur, « Guide to the various phylogenetic classification schemes for Escherichia coli and the correspondence among schemes », *Microbiology*, vol. 161, nº 5, p. 980-988, mai 2015, doi: 10.1099/mic.0.000063.
[30] E. Mayr, *Principles of Systematic Zoology*. Scientific Publishers, 2015.
[31] A. MYERS, « SIMPSON, G. G. Principles of animal taxonomy. Columbia University Press, New York: 1990. Pp xii, 247; illustrated. Price: US$ 21.50. ISBN: 0-231-09650-X (paperback reprint). », *Archives of Natural History*, vol. 19, nº 1, p. 124-124, févr. 1992, doi: 10.3366/anh.1992.19.1.124.
[32] G. Brewka, « Artificial intelligence—a modern approach by Stuart Russell and Peter Norvig, Prentice Hall. Series in Artificial Intelligence, Englewood Cliffs, NJ. », *The Knowledge Engineering Review*, vol. 11, nº 1, p. 78-79, mars 1996, doi: 10.1017/S0269888900007724.
[33] L. Deng et D. Yu, « Deep Learning: Methods and Applications », *SIG*, vol. 7, nº 3–4, p. 197-387, juin 2014, doi: 10.1561/2000000039.
[34] A. Burkov, *The Hundred-Page Machine Learning Book*. Leanpub, 2018. Consulté le: 5 avril 2023. [En ligne]. Disponible sur: https://leanpub.next/theMLbook
[35] A. C. Mu, « Introduction to Machine Learning with Python ».
[36] A. C. Mu, « Introduction to Machine Learning with Python ».
[37] M. Nielsen, « Neural Networks and Deep Learning ».

[38] « Structure d'un neurone artificiel - Google Search ». https://www.google.com/search?sxsrf=APwXEdf0UQ07aNtvO2L5JGEcYfdIpHE-KA:1682369853255&q=Structure+d%27un+neurone+artificiel&tbm=isch&sa=X&ved=2ahUKEwiAte-LtMP-AhUGXaQEHVfdDxsQ0pQJegQICxAB&biw=1366&bih=625&dpr=1#imgrc=0dchbkln4CBY_M&vwlns=WyIwQ0JJUWg2Y0dhaGNLRXdpZ2c2aU90TVAtQWhVQUFBQUFIUUFBQUFBUUJBIl0=&lns=W251bGwsbnVsbCxudWxsLG51bGwsbnVsbCxudWxsLG51bGwsIkVrY0tKREppTnpaaFltUXhMVEl4TVRZdE5ETTRPUzFpTXpneExXUmhOREpsTXpnMllURTVZaElmTkRSNVZWbzBReTFCZVVsa05FbGhORUZXWjJGWFdVSlVOeTA1VVdWNFp3PT0iXQ== (consulté le 24 avril 2023).
[39] N. Buduma et N. Locascio, *Fundamentals of Deep Learning: Designing Next-generation Machine Intelligence Algorithms*. O'Reilly Media, 2017.
[40] F. Rosenblatt, « The perceptron: A probabilistic model for information storage and organization in the brain », *Psychological Review*, vol. 65, p. 386-408, 1958, doi: 10.1037/h0042519.
[41] D. E. Rumelhart, G. E. Hinton, et R. J. Williams, « Learning representations by back-propagating errors », *Nature*, vol. 323, nº 6088, Art. nº 6088, oct. 1986, doi: 10.1038/323533a0.
[42] « Figure 2. Diagramme schématique du réseau de neurone MLP », *ResearchGate*. https://www.researchgate.net/figure/Diagramme-schematique-du-reseau-de-neurone-MLP_fig2_262451664 (consulté le 25 avril 2023).
[43] S. Hochreiter et J. Schmidhuber, « Long Short-Term Memory », *Neural Computation*, vol. 9, nº 8, p. 1735-1780, nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[44] Y. Lecun, L. Bottou, Y. Bengio, et P. Haffner, « Gradient-based learning applied to document recognition », *Proceedings of the IEEE*, vol. 86, nº 11, p. 2278-2324, nov. 1998, doi: 10.1109/5.726791.
[45] I. Goodfellow, Y. Bengio, et A. Courville, *Deep Learning*. MIT Press, 2016.
[46] « Learning representations by back-propagating errors | Nature ». https://www.nature.com/articles/323533a0 (consulté le 31 mars 2023).
[47] V. Nair et G. E. Hinton, « Rectified Linear Units Improve Restricted Boltzmann Machines ».
[48] « Deep learning | Nature ». https://www.nature.com/articles/nature14539 (consulté le 31 mars 2023).
[49] *Pattern Recognition and Machine Learning*. Consulté le: 31 mars 2023. [En ligne]. Disponible sur: https://link.springer.com/book/9780387310732
[50] K. G. Kim, « Book Review: Deep Learning », *Healthc Inform Res*, vol. 22, nº 4, p. 351, 2016, doi: 10.4258/hir.2016.22.4.351.