REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE YAHIA FARES DE MEDEA



Faculté des Sciences
Département du Mathématique et Informatique
Mémoire de fin d'études de Master
Filière : Informatique
Spécialité : Ingénierie des Systèmes et Technologie du web

Proposition d'un système de recommandation pour la sélection des services web en utilisant Deep Learning

Présentés par :

- Annane mounira
- Hebbache fouzia

Proposé et Dirigé par :

TOUBAL El Bahi

Année universitaire: 2022/2023

Remerciement

Nous remercions Dieu qui nous a accordé l'aide, la patience et le courage pour accomplir ce travail.

Nous tenons à exprimer notre plus profonde gratitude envers Monsieur TOUBAL El Bahi pour son encadrement et ses recommandations qui ont été d'une grande contribution pour nous.

Nous tenons également à adresser nos plus sincères remerciements aux membres du jury pour avoir accepté d'évaluer notre travail.

Merci à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail et qui méritent d'y trouver leur nom.

Nous vous prions d'accepter nos remerciements les plus sincères pour votre précieuse aide et votre contribution à la réussite de notre mémoire. Votre soutien continuera d'avoir un impact durable sur notre parcours académique et professionnel.

Dédicace

je remercie ALLAH qui m'a aidé et donné la patience et le courage durant ces longues années d'étude

Je dédie ce travail,

A Mes chers parents que dieu les gardes et préserve Pour leur patience et leur soutien tout au long de parcours scolaire.

et Toutes Mes sœurs et mon frère pour votre soutien constant et vos encouragements

A toute ma famille ;

A mon binôme dans ce travail Et tous mes amis.

Fouzia…

Dédicace

Je dédie ce travail en signe de respect et de reconnaissance :

A mon cher père :

A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect.

A ma chère mère :

A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse.

A mes très chère tantes Touria et Hanna

quoi que je fasse ou que je dise, je ne saurai point vous remercier comme il se doit.

Votre affection me couvre votre bienveillance me guide et votre présence à mes cotés a toujours été ma source de force pour affronter les différents obstacles.

A ma chère sœur, et mon frère :

Qui n'ont pas cessée de me conseiller, encourager et soutenir tout au long de mes études. Que Dieu les protège et leurs offre la chance et le bonheur.

A ma deuxième sœur

que j'aime beaucoup, mon soutien moral et source de joie et de bonheur, que dieu la garde.

A tous les cousines et les amis, merci pour leurs amours et leurs encouragements.

A mon binôme Fouzia le travail avec toi est un plaisir.

Mounira…

Résumé:

Cette proposition vise à développer un système de recommandation utilisant le deepLearning pour sélectionner des services web adaptés aux préférences et aux besoins des utilisateurs. Le deepLearning est une technique de machine Learning capable d'extraire des modèles complexes à partir de données brutes sans nécessiter une extraction manuelle de fonctionnalités.

Le système de recommandation collectera des données sur les utilisateurs, leurs comportements et les services web disponibles. En utilisant des architectures de réseaux de neurones adaptées, le modèle de deepLearning apprendra à partir de ces données pour générer des recommandations personnalisées.

Cette proposition mettra également l'accent sur des aspects essentiels tels que la collecte et le prétraitement des données, ainsi que l'évaluation de la performance du modèle de recommandation à l'aide de techniques d'évaluation appropriées. De plus, l'implémentation du système de recommandation en tant que service web sera réalisée.

L'objectif final de cette proposition est de fournir un système de recommandation précis et efficace, aidant les utilisateurs à naviguer à travers la diversité des services web disponibles et à trouver ceux qui correspondent le mieux à leurs besoins et à leurs préférences.

Mots clé : Apprentissage automatique, Système de recommandation, Service web, Réseaux de neurones,

Apprentissage en profondeur, RNN,MLP,CNN

Abstract:

This proposal aims to develop a recommendation system using deep learning to select web services that are tailored to users' preferences and needs. Deep learning is a machine learning technique capable of extracting complex patterns from raw data without requiring manual feature extraction.

The recommendation system will collect data on users, their behaviors, and available web services. By utilizing suitable neural network architectures, the deep learning model will learn from this data to generate personalized recommendations.

This proposal will also focus on essential aspects such as data collection and preprocessing, as well as evaluating the performance of the recommendation model using appropriate evaluation techniques. Additionally, the implementation of the recommendation system as a web service will be carried out.

The ultimate goal of this proposal is to provide an accurate and efficient recommendation system, assisting users in navigating the diversity of available web services and finding those that best match their needs and preferences.

Keywords: Machine Learning, Recommendation System, Web service , Neural Networks, Deep Learning, RNN, MLP, CNN

ملخص :

تهدف هذه الاقتراح إلى تطوير نظام توصية باستخدام التعلم العميق لاختيار خدمات الويب المناسبة لتفضيلات واحتياجات المستخدمين. التعلم العميق هو تقنية في مجال تعلم الآلة قادرة على استخراج أنماط معقدة من البيانات الخام دون الحاجة إلى استخلاص الميزات يدويًا.

سيقوم نظام التوصية بجمع البيانات عن المستخدمين، سلوكهم وخدمات الويب المتاحة. باستخدام تصاميم مناسبة للشبكات العصبية، سيتعلم نموذج التعلم العميق من هذه البيانات لتوليد توصيات مخصصة.

ستركز هذه الاقتراح أيضًا على جوانب أساسية مثل جمع البيانات ومعالجتها، وتقييم أداء نموذج التوصية باستخدام تقنيات التقييم المناسبة. بالإضافة إلى ذلك، سيتم تنفيذ نظام التوصية كخدمة ويب.

الهدف النهائي من هذا الاقتراح هو توفير نظام توصية دقيق وفعال يساعد المستخدمين في التنقل في تنوع خدمات الويب المتاحة وإيجاد تلك التي تتوافق بشكل أفضل مع احتياجاتهم وتفضيلاتهم.

الكلمات الرئيسية: التعلم الآلي، نظام التوصية، الشبكات العصبية،خدمة ويب ، التعلم العميق,CNN, RNN,MLP

Sommaire

Titre	Page
Remerciement	
Dédicace	
Résumé	
Sommaire	
Liste des Figures	
Liste des Tableaux	
Liste des abriviations	
Introduction générale	**2**
objectif	**4**
Problématique	**5**
Chapitre 01: systèmes de recommandation	
1. Introduction	**7**
2. Système de recommandation	**8**
2.1.Historique	**8**
2.2. Définition	**9**
2.3.Classification de système de recommandation	**9**
3. Les approches de la recommandation	**10**
3.1. Les méthodes de filtrage basées sur le contenu (CBF)	**10**
3.2. système de recommandation collaboratif	**11**
3.3. Recommandation Hybride	**11**
4. Les étapes principales de la recommandation	**12**
4.1. Collecte d'information	**12**
4.2. Mise en œuvre d'une matrice utilisateur d'informations récoltées	**13**
4.3. Extraction de la liste de recommandations	**14**
5. Les avantages et les inconvénients des systèmes de recommandation	**14**
6. Domaine d'applications des systèmes de recommandations	**15**
6.1. Service de commerce électronique	**15**
6.2. Service de réseau social	**15**
6.3. Service de tourisme	**15**
6.4. Service de l'éducation	**15**
6.5. Service de santé	**16**
6.6. Service d'information académique	**16**
8. Conclusion :	**17**
Chapitre 02:service web	
1.Introduction	**19**
2.Définition des services web	**20**
3.Sélection des services Web	**20**
3.1Sélection à base de qualité de service QoS-Aware	**20**
3.2Sélection à base de vie privée	**20**
4.Qualité de Service (QoS)	**21**
5.Avantage service web	**21**
6.Service web et système de recommandation	**22**
7.Conclusion	**23**
Chapitre 03 : Apprentissage profond	
1. Introduction	**25**
2. Notions de base:	**26**
2.1. L'intelligence artificielle	**26**
2.1.1. Relation entre l'intelligence artificielle et L'apprentissage profond	**26**

2.2. Apprentissage automatique **26**
2.2.1. Les avantages d'apprentissage automatique (machine learning) **27**
2.2.2. L'apprentissage automatique selon la nature des données en entrée **28**
3. Les algorithmes d'apprentissage supervisé **28**
3.1. La méthode des K plus proches voisins **28**
3.1.2. Avantages de la méthode des k plus proches voisins **29**
3.1.3. Inconvénients de la méthode des k plus proches voisins **29**
3.2. La regression logistique **30**
3.2.1. Avantages La regression logistique **30**
3.2.2. Inconvenient La regression logistique **30**
3.3. K-moyen (K-mean) **31**
3.3.1. Avantages du K-moyen **31**
3.3.2. Inconvenient du K-moyen **31**
3.4. Arbres de Décision **31**
3.5. Machine à Vecteurs Supports **31**
3.5.1. Avantages des SVM **32**
3.5.2. Inconvénients des SVM **32**
4. l'apprentissage profond **32**
4.1. Différence entre Machine Learning et Deep Learning **33**
4.2. Historique deep learning **34**
4.3. Avantages de l'apprentissage en profondeur **35**
4.4. Inconvénients du Deep Learning **36**
4.5. Quels sont les différents types de fonctions d'activation utilisées dans le deep learning? **38**
4.6. Comment optimiser un modèle de Deep Learning ? **38**
5. Réseaux de neurones: **39**
5.1. Historique: **39**
5.2. Les réseaux de neurones : **39**
5.3. Architecture **40**
5.4. Fonctionnement des réseaux de neurones artificiels: **40**
5.5. Avantages des réseaux de neurones **41**
5.6. Inconvénients des réseaux de neurones **41**
6. Architectures de réseaux de neurones profonds **42**
6.1. Réseaux de neurones convolutifs (CNN) **42**
6.1.1. Avantages CNN **42**
6.1.2. Désavantages CNN **42**
6.2. Quelques réseaux Convolutifs célèbres **42**
6.3. Réseaux de neurones récurrents (RNN) **44**
6.3.1. Avantages RNN **44**
6.3.2. Désavantages RNN **44**
6.4. Cellules LSTM **44**
6.5. Différence entre CNN ET RNN **45**
6.6. Perceptron Multi-Couches **45**
7.hybride basé sur l'apprentissage profondsystèmes de recommandation **46**
7.1 rnn dans les systèmes de recommandation **47**
7.2 CNN dans les systèmes de recommandation **48**
7.3 MLP dans les systèmes de recommandation **48**
8. Conclusion **50**

Chapitre 04 : Approche proposé

1. Introduction **52**
2. Les démarches de création de la base de données **53**
2.1Diagramme des classes **53**
2.2Algorithme random dataset **54**
3.L'architecture proposée **55**
3.1Prétraitement **56**
3.2CF et CB model **58**
3.3 Hybride modèle **62**
3.3.1 Hybride avec RNN **62**
3.3.2 hybride avec CNN **65**
3.3.3 Hybride avec MLP **68**
4. Algorithme get top N service **71**
5. Conclusion **72**
Chapitre 05 : Expérimentation et Evaluation
1. Introduction **74**
2. Base de donnée **75**
3. Langage de programmation **75**
4. Métriques d'évaluation **75**
4. Résultats et diagnostic **77**
4.1 modèle Simple RNN **77**
4.2 modèle CONV1D **77**
4.3 modèle MLP **78**
4.4 Evaluation des modèles utilisés **78**
5. Get_top_recommandations **79**
6. Conclusion **81**
Conclusion générale **83**
Références **85**

Liste des Figures:

Figures	Page
Figure 1.1:Classification de système de recommandation	**10**
Figure 1.2: Les étapes principales de système de recommandation	**12**
Figure2.1 les relations entre l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond	**26**
Figure 2.2 : les grandes classes d'apprentissage automatique	**27**
Figure 2.3 : visualisation d'un exemple de classification en utilisant le method KNN(K=3)	**29**
Figure 2.4: comparaison entre une régression linéaire et un régression logistique	**30**
Figure 2.5: Exemple d'un SVM avec noyau linéaire	**32**
Figure 2.6 : Neurone artificiel avec une seule sortie	**39**
Figure 2.7 : Le réseau de neurones à un seul niveau	**40**
Figure 2.8: représentation d'un neurone artificielle extraite et adaptée de Data Camp(2018)	**41**
Figure 2.9 :Architecture d'un MLP de R2 dans R à deux couches cachées de tailles h1 " 3 et h2 " 4	**46**
Figure 2.10 : Architecture du modèle HybMLP	**48**
Figure 3.1 : Diagramme de classe proposée	**53**
Figure 3.2 : Algorithme random dataset	**54**
Figure 3.3: Table utilisateur	**54**
Figure 3.4: Table service	**54**
Figure 3.5 : Table type matériel	**54**
Figure 3.6: Table region	**55**
Figure 3.7: Table QOS	**55**
Figure3.8 :Architecture de l'approche.	**56**
Figure3.9 : Architecture de Prétraitement	**57**
Figure3.10 : Algorithme préparation données	**58**
Figure 3.11: Algorithme création model CF et CBF	**59**
Figure3.12: Architecture CF model	**60**
Figure3.13: Architecture CBF model	**61**
Figure3.14 : Algorithme création hybride RNN	**62**
Figure3.15: Architecture Simple RNN model	**64**
Figure 3.16: Algorithmehybride CNN model	**65**
Figure3.17: Architecture CNN model	**67**
Figure 3.18: Algorithme hybride MLP model	**68**
Figure3.19: ArchitectureMLP model	**70**
Figure3.20: Algorithme get topN service	**71**
Figure4.1 : Graph accuracy simplernn	**77**
Figure4.2 :Graph loss SimpleRNN	**77**
Figure4.3: Graph accuracyCNN1D	**77**
Figure4.4: Graph lossCNN1D	**77**
Figure 4.5 : Graph accuracy MLP	**78**

Figure 4.6: Graph Loss MLP **78**
Figure 4.7 : Recommandations Simple RNN **79**
Figure 4.8 : RecommandationsCONV1D **79**
Figure 4.9 : Recommandations MLP **80**

Liste des Tableaux:

Tableau	Page
Tableau 1.1: Exemple fictif de matrice binaire contenant des informations de type " l'utilisateur u a apprécié/n'a pas apprécié l'item i "	**13**
Tableaux 1.2: Les avantages et les inconvénients des systèmes de recommandation	**14**
Tableau 2.1: Les différences entre Machine Learning et Deep Learning	**33**
Tableaux 2.2 : Les différences entre CNN et RNN	**45**
Tableau 4.1: Description des champs BDDSW	**75**
Tableau 4.2: Résultats de précision (accuracy)	**79**

Liste des abriviations

Acronyme	Signification
CBF	Filtrage Basé sur le Contenu
CF	Filtrage Collaboratif
CNN	Réseaux de neurones de convolution
CONV1D	Convolution 1D
DL	Deep Learning
IA	Intelligence Artificielle
KNN	K-Nearest-Neighbors
ML	Machine Learning
MLP	Multi Layer Perceptron
RNN	Réseau de Neurones Récurrents
SdR	Système de recommandation
SVM	Support Vecteur Machine

INTRODUCTION GÉNÉRALE

Introduction générale:

L'essor d'Internet a conduit à une abondance de services web offrant une multitude de fonctionnalités aux utilisateurs. Cependant, avec cette profusion de choix, il devient de plus en plus difficile pour les utilisateurs de trouver les services web les plus pertinents et adaptés à leurs besoins spécifiques. Pour résoudre ce défi, les systèmes de recommandation se sont avérés être des outils précieux en fournissant des suggestions personnalisées.

L'intelligence artificielle (notamment via l'apprentissage automatique et l'apprentissage profond) a connu une forte croissance ces dernières décennies, notamment grâce à l'augmentation exponentielle du volume des données publiées chaque jour sur Internet. Désormais, notre vie quotidienne est significativement impactée par l'utilisation de ces technologies, mises en œuvre dans des applications réelles dont les systèmes de recommandation (SdR) font partie. A cause du flot d'informations qui nous submerge, il devient très difficile pour les utilisateurs de trouver par eux-mêmes les items pertinents au sein de catalogues. Ces items peuvent être des articles à acheter, des films ou des disques intéressants ou encore des voyages, …..etc. Parmi un large catalogue d'items, un SdR possède la capacité d'extraire une courte liste composée des items qui correspondent au mieux aux goûts de chaque utilisateur.

Dans ce mémoire, nous proposons la conception et l'implémentation d'un système de recommandation innovant qui exploite les puissantes capacités du deeplearning.

En l'appliquant au domaine des systèmes de recommandation pour les services web, nous cherchons à améliorer considérablement la précision et la pertinence des recommandations proposées aux utilisateurs. Les SdR sont généralement catégorisés en trois familles, définies par le type de filtrage utilisé : le filtrage collaboratif (CF), le filtrage basé sur le contenu (CBF) et le filtrage hybride qui combine CF et CBF Notre approche se concentre sur l'utilisation de réseaux de neurones convolutifs (CNN) et de réseaux de neurones récurrents (RNN) pour capturer les informations clés des utilisateurs et des services web. Les CNN sont particulièrement adaptés pour extraire des caractéristiques significatives à partir de données structurées, tandis que les RNN sont efficaces pour modéliser des séquences et des interactions temporelles .Nous visons à exploiter au mieux les informations disponibles et à obtenir des recommandations plus précises.

Pour bien structurer notre travail, nous l'avons élaboré en quatre chapitres, comme suit:
-Le chapitre 1 en présente une vue générale sur les systèmes de recommandation avec leurs types ainsi que leurs avantages et inconvénients.

- Le deuxième chapitre on a présents les techniques utiliser dans l'intelligence artificielle comme deepLearning et aussi en incluent les avantages et inconvénients de ces techniques.
- Dans le troisième chapitre nous présentons les étapes de notre approche pour la réalisation de notre modèle.
- Le dernier chapitre est de présenter l'expérimentation et l'évaluation de l'approche proposée dans cette mémoire.

Enfin, nous concluons ce travail en résumant nos principales contributions et les perspectives de ce projet.

Objectifs :

L'objectif principal de ce mémoire est de concevoir et d'implémenter un système de recommandation innovant qui exploite les capacités du deeplearning. Nous cherchons à améliorer la précision et la pertinence des recommandations proposées aux utilisateurs en exploitant les informations clés des utilisateurs et des services web.

Plus spécifiquement, nos objectifs sont les suivants :

- Concevoir une architecture de système de recommandation basée sur des réseaux de neurones Convolutifs (CNN) et des réseaux de neurones récurrents (RNN) pour capturer les informations importantes des utilisateurs et des services web.
- Exploiter les caractéristiques structurées des données à l'aide des CNN et modéliser les séquences et les interactions temporelles à l'aide des RNN afin d'obtenir des recommandations plus précises.
- Évaluer et comparer les performances de notre approche avec les approches traditionnelles de filtrage collaboratif et de filtrage basé sur le contenu, en utilisant des mesures telles que la précision, le rappel et le taux de satisfaction des utilisateurs.

Problématique :

La profusion de services web disponibles sur Internet a créé un défi majeur pour les utilisateurs : comment trouver les services les plus pertinents et adaptés à leurs besoins spécifiques parmi cette abondance de choix ? Les systèmes de recommandation se sont révélés être des outils précieux pour résoudre cette problématique en fournissant des suggestions personnalisées.

Cependant, malgré les progrès réalisés dans le domaine des systèmes de recommandation, il existe toujours des lacunes à combler. Les approches traditionnelles, telles que le filtrage collaboratif et le filtrage basé sur le contenu, présentent certaines limites en termes de précision et de pertinence des recommandations.

Dans ce contexte, une question se pose : comment exploiter les avancées récentes de l'intelligence artificielle, en particulier le deeplearning, pour améliorer considérablement les performances des systèmes de recommandation pour les services web ?

CHAPITRE 01 :
SYSTÈMES DE RECOMMANDATION

1. Introduction:

Dans le monde moderne, nous sommes constamment bombardés par une surcharge d'informations et de choix. Que ce soit lors de nos sessions de navigation sur Internet, de nos interactions avec les plateformes de streaming ou même de nos séances de shopping en ligne, nous sommes confrontés à une multitude d'options parmi lesquelles choisir. Face à cette abondance d'opportunités, il devient essentiel de disposer d'outils capables de nous guider et de nous recommander les éléments les plus pertinents et personnalisés.

Les Systèmes de Recommandation (SR) se sont imposés comme des technologies incontournables dans notre quotidien. Ils sont présents dans une variété de domaines, tels que la musique, le cinéma, la littérature, le commerce électronique, les réseaux sociaux et bien d'autres. Grâce à leur capacité à analyser de vastes ensembles de données et à extraire des informations pertinentes, les SR sont en mesure de prédire et de suggérer des éléments qui répondent aux préférences et aux besoins spécifiques de chaque individu.

2. Système de recommandation:

2.1. Historique:

La capacité des ordinateurs pour faire des recommandations à des utilisateurs a été reconnue assez tôt dans l'histoire de l'informatique. Grundy [1], un système bibliothécaire, était une première étape vers des systèmes de recommandation automatiques.

Ce système était assez primitif. Il classait les utilisateurs en "stéréotypes" en se basant sur une courte interview, et utilisait ces stéréotypes pour produire des recommandations de livres.

Ce travail constituait une première tentative intéressante dans le domaine des systèmes de recommandation. Cependant, son utilisation est restée très limitée. Au début des années 1990, le filtrage collaboratif apparait comme une solution pour faire face à la surcharge d'information. L'année 1992 voit l'apparition du système de recommandation de documents Tapestry[2], ainsi que la création du laboratoire de recherche GroupLens, qui travaille explicitement sur le problème de la recommandation automatique dans le cadre des forums de news de Usenet. Tapestry avait pour but de recommander à des groupes d'utilisateurs des documents issus des newsgroups susceptibles de les intéresser.

L'approche utilisée était de type "plus proches voisins" à partir de l'historique de l'utilisateur. On parle alors de filtrage collaboratif manuel, comme une réponse au besoin d'outils pour le filtrage de l'information énoncé à la même époque. La recommandation résulte d'une action collaborative des utilisateurs qui recommandent à d'autres utilisateurs des documents en leur attribuant des notes d'intérêt selon certains critères. Les systèmes de filtrage collaboratif automatiques apparaissent ensuite. GroupLens[3]utilise cette technique pour identifier les articles de Usenet susceptibles d'être intéressants pour un utilisateur donné. Les utilisateurs doivent seulement attribuer des notes ou effectuer d'autres opérations observables (par exemple, lire un article) ; le système combine alors ces données avec les notes ou les actions d'autres utilisateurs pour fournir des résultats personnalisés. Avec ces systèmes, les utilisateurs n'ont aucune connaissance directe des opinions des autres utilisateurs, ni des articles présents dans le système. Au cours de ces dernières années, les systèmes de recommandation deviennent un sujet d'un intérêt croissant dans les domaines de l'interaction homme-machine, de l'apprentissage automatique ainsi que la recherche d'information. En 1995 apparaissent successivement Ringo[4], un système de recommandation de musique, basé sur les appréciations des utilisateurs [5], un système de recommandation de vidéos. La même année, GroupLens crée la société Net Perceptions dont

le premier client a été Amazon. De nos jours, les systèmes de recommandation sont devenus des composantes incontournables pour la plupart des sites du e-commerce.[6]

2.2. Définition :

Les systèmes de recommandation (ou de recommandation) (RS) sont des techniques ou des logiciels qui visent à fournir des suggestions précises ou utiles de certains éléments à un utilisateur[7],[8]. Utilisateur est le terme général pour désigner l'entité à laquelle la recommandation est fournie, tandis qu'Article étiquette le produit recommandé. Selon le domaine d'application du système, l'utilisateur est généralement un individu, mais ce terme peut aussi désigner un groupe de personnes, une entreprise. . . tandis que les articles recommandés peuvent être aussi variés qu'un ordinateur à acheter, un film à regarder, une chanson à écouter, une actualité à lire, etc.

Une fois que l'utilisateur a navigué sur le RS ou accepté une recommandation, un retour (une note par exemple) est donné au système concernant l'article acheté ou consommé. Afin de suggérer des recommandations plus personnalisées et appropriées lors des futures apparitions de cet utilisateur, le RS utilisera toutes les interactions passées et les commentaires de l'utilisateur. Si disponible, le système peut potentiellement utiliser comme entrée d'autres sources de données diverses, telles que des informations sur l'utilisateur ou l'élément. Par exemple, les informations démographiques fournies par l'utilisateur pour remplir son profil, comme son âge, son sexe, sa profession, peuvent être utilisées comme modèle d'utilisateur, tandis que des informations sur l'élément comme le genre d'un film ou son réalisateur et ses acteurs peuvent être utilisées pour construire un modèle d'article[9]

2.3. Classification de système de recommandation :

- Classification classique : Cette classification est reconnue par trois types de filtrage, un filtrage collaboratif (CF) utilise les opinions des utilisateurs similaires à l'utilisateur actif, un filtrage basé sur le contenu (CBF) utilise uniquement les préférences de l'utilisateur actif et le filtrage hybride combine le filtrage basé sur le contenu et le filtrage collaboratif pour exploiter au mieux les avantages de chacun.[10]
- La Classification de : Su et al, 2009 : elle est utilisée dans les systèmes collaboration. Ils proposent une sous-classification qui comprend les techniques hybrides[11].
- La classification de Rao et Talwar :C'est une classification en fonction de la source d'information utilisée. Rao et Talwar ont répertorié un vaste ensemble de systèmes de recommandation pour différents domaines applicatifs, dans des contextes académiques et industriels. Ils peuvent être classés de manière générale en six catégories en fonction de

l'information utilisée (filtrage collaboratif, filtrage à base de contenu, filtrage hybride, filtrage démographique, filtrage communautaire, filtrage à base de connaissance).[4]

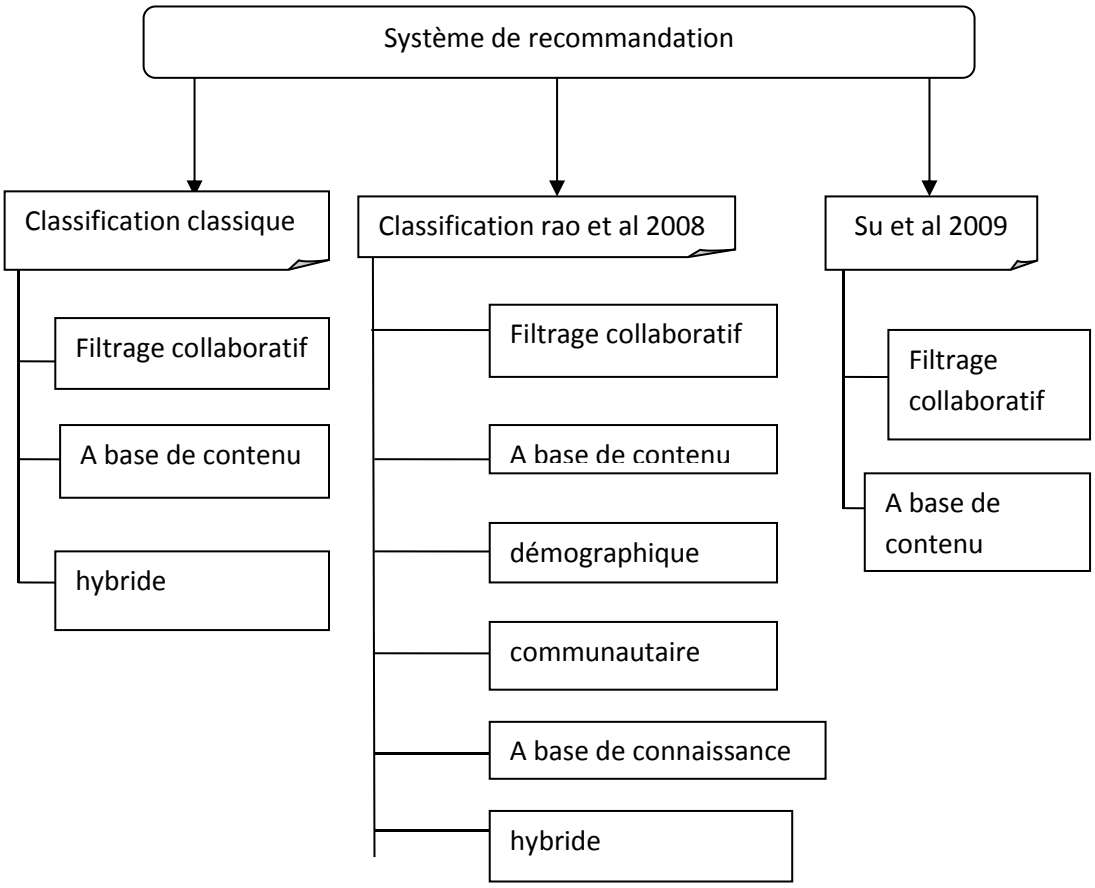


Figure1.1: Classification de système de recommandation

3. Les approches de la recommandation :

Les systèmes de recommandation permettent de générer des suggestions sur de nouveaux items ou de prédire l'utilité d'un item pour un utilisateur donné. Trois types d'approches sont principalement utilisés: le filtrage basé sur le contenu (FBC), le filtrage collaboratif (FC), et le filtrage hybride, a pour objectif d'atténuer les insuffisances de chacune des deux approches précédentes, en les combinant de différentes manières .La figure synthétise les différentes approches de recommandation)

3.1. Les méthodes de filtrage basées sur le contenu (CBF) :

sont basées sur la description des éléments et un profil de la préférence de l'utilisateur [2]Ces algorithmes tentent de recommander des éléments similaires à ceux qu'un utilisateur a aimés dans le passé ou examine dans le présent. En particulier, divers items candidats sont comparés à des items préalablement notés par l'utilisateur et les articles les mieux assortis sont recommandés. Cette approche trouve ses racines dans la recherche en recherche d'information. fondamentalement, ces méthodes utilisent un profil d'article caractérisant les éléments au sein

du système. Le système crée une base de contenu(CB) profil des utilisateurs basé sur un vecteur pondéré de caractéristiques de l'article. Les poidsindiquent l'importance de chaque caractéristique pour l'utilisateur et peuvent être calculées à partir de vecteurs de contenu évalués individuellement en utilisant une variété de techniques. Des approches simples utilisent les valeurs moyennes du vecteur d'item évalué tandis que d'autres des méthodes sophistiquées utilisent des techniques d'apprentissage automatique telles que Bayesian Classificateurs, analyse de clusters, arbres de décision et réseaux de neurones artificiels dans afin d'estimer la probabilité que l'utilisateur va privilégier l'item.

La norme des éléments recommandés proposés par les méthodes (CB)a des explications plus claires sur le fonctionnement du système de recommandation, car ils peuvent être fournis en énumérant explicitement les caractéristiques ou les descriptions du contenu. Ceux les fonctionnalités sont des indicateurs à consulter pour décider de faire confiance à une recommandation. A l'inverse, Les méthodes (CB) sont capables de recommander des éléments qui n'ont pas encore été notés par n'importe quel utilisateur. En conséquence, ils ne souffrent pas du "démarrage à froid".[12]

3.2. Système de recommandationcollaboratif:

Le filtrage collaboratif (CF) est le processus de filtrage d'informations ou de modèles à l'aide de techniques qui impliquent la collaboration entre divers agents, points de vue, sources de données, etc. [5]

CF peut être utilisé pour faire des prédictions automatiques sur les intérêts d'un utilisateur en collectant des informations sur les préférences ou les goûts de nombreux utilisateurs grâce à la collaboration. L'approche CF suppose que si un personne X a la même opinion qu'une personne Y sur une question, X est plus susceptible de avoir l'opinion de Y sur une question 'a' différente de celle d'avoir l'opinion sur 'a' d'un personne choisie au hasard. Par exemple, un système de recommandation CF pour les goûts des ordinateurs portables pourrait faire des prédictions sur la marque d'ordinateur portable qu'un utilisateur devrait comme étant donné une liste partielle des goûts de cet utilisateur (aime ou n'aime pas).[12]

3.3. Recommandation Hybride :

Les techniques que nous avons présentées jusqu'à présent peuvent être combinées dans un système de recommandation hybride.

En effet, ce faisant, les faiblesses et les lacunes de chaque technique peuvent annuler les uns des autres et donc les performances sont également améliorées[4]. Par exemple, Collaboratif Les techniques de filtrage doivent faire face au problème des nouveautés, c'est-à-dire qu'elles ne sont pas en mesure de recommander articles qui n'ont pas encore été évalués.

Cependant, les nouvelles fonctionnalités des articles (description) sont généralement disponible et peut être utilisé avec des méthodes basées sur le contenu.

En effet, un système de recommandation comme celui opérant sur Amazon combine plusieurs techniques de recommandation, conduisant à des prévisions plus efficaces et plus précises.[13]

4. Les étapes principales de la recommandation :

Un système de recommandation requiert généralement trois(03) étapes, et qui sont illustré dans la figure suivante :

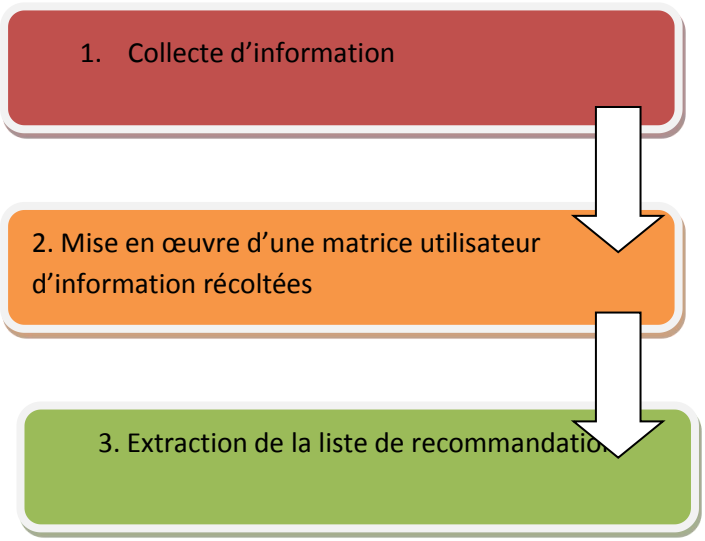


Figure1.2: Les étapes principales de système de recommandation

4.1. Collecte d'information :

Pour être pertinent, un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur. Une distinction peut être faite entre 2 formes de collecte de données :

Collecte de données explicites - Filtrage actif : La collecte repose sur le fait que l'utilisateur indique explicitement au système ses intérêts. Exemple : Demander à un utilisateur de commenter, taguer/étiqueter, noter, liker ou encore ajouter comme favoris des contenus (objets, articles...) qui l'intéressent. On utilise souvent une échelle de ratings allant de 1 étoile (je n'aime pas du tout) à 5 étoiles (j'aime beaucoup) qui sont ensuite transformées en valeurs numériques afin de pouvoir être utilisées par les algorithmes de recommandation. Avantage : Capacité à reconstruire l'historique d'un individu et capacité à éviter d'agréger

une information qui ne correspond pas à cet unique utilisateur (plusieurs personnes sur un même poste).

Collecte de données explicites - Filtrage actif : La collecte repose sur le fait que l'utilisateur indique explicitement au système ses intérêts. Exemple : Demander à un utilisateur de commenter, taguer/étiqueter, noter, liker ou encore ajouter comme favoris des contenus (objets, articles...) qui l'intéressent. On utilise souvent une échelle de ratings allant de 1 étoile (je n'aime pas du tout) à 5 étoiles (j'aime beaucoup) qui sont ensuite transformées en valeurs numériques afin de pouvoir être utilisées par les algorithmes de recommandation. Avantage : Capacité à reconstruire l'historique d'un individu et capacité à éviter d'agréger une information qui ne correspond pas à cet unique utilisateur (plusieurs personnes sur un même poste).[14]

4.2. Mise en œuvre d'une matrice utilisateur d'informations récoltées :

La mise en œuvre d'une matrice appelée " matrice utilisateur " ou bien " modèle utilisateur " incluant les informations concernant les utilisateurs récoltées durant la précédente étape qu'est la collecte d'informations. On peut la représenter comme un tableau qui contient des données recueillies sur l'utilisateur associées aux produits disponibles sur le site web

	Item1	Item2	Item3	Item4	Item5	…
User1	☺			☺		…
User2		☹	☺			…
User3		☺				…
User4			☹	☺	☹	…
…	…	…	…	…	…	…

Tableau1.1 : Exemple fictif de matrice binaire contenant des informations de type " l'utilisateur u a apprécié/n'a pas apprécié l'item i "

Le tableau ci-dessus présente un exemple fictif de matrice binaire contenant des informations de type " l'utilisateur u a apprécié/n'a pas apprécié l'item i ". Ces informations peuvent également être " a achète/n'a pas achète ", "a consulté/n'a pas consulté ", etc. Elles peuvent également se mesurer sur un nombre plus élève de classes : " a mis 1/2/3/4/5 étoiles " etc. Un autre point important est comment le temps influence le profil de l'utilisateur. Les intérêts des utilisateurs, évoluent généralement au cours du temps. Les données du modèle utilisateurs devraient donc constamment être réajustées pour rester conformes aux nouveaux centres d'intérêts de l'utilisateur.[14]

4.3. Extraction de la liste de recommandations :

Pour extraire une liste de suggestions c.-à-d. de recommandation à partir d'une matrice utilisateur, les algorithmes utilisent la notion de mesure de similarité entre objets ou personnes décrits par le modèle utilisateur. La similarité a pour but de donner une valeur ou un nombre (au sens mathématique du terme) à la ressemblance entre 2 choses.

Plus la ressemblance est forte, plus la valeur de la similarité sera grande. A l'inverse, plus la ressemblance est faible, et plus la valeur de la similarité sera petite.[14]

5. Les avantages et les inconvénients des systèmes de recommandation:

Dans cette section, nous essayons de collecter les avantages et les inconvénients de chaque méthode de recommandation.

Types	Avantage	Inconvénients
Les approches basées sur le contenu	- Pas besoin de données sur les autres utilisateurs. -Il est possible de faire des recommandations à des utilisateurs avec des goûts « uniques ». - Les données relatives aux autres utilisateurs sont inutiles. -Il est possible de recommander des produits qui ne sont pas populaires ou nouveaux	- Trouver les bons "features" n'est pas toujours facile. - Il n'existe pas d'historique, lorsqu'un nouvel utilisateur commence à utiliser le système. - Comment créer un profil pour les nouveaux utilisateurs ?
Les approches basées sur le filtrage collaboratif	-Utiliser les scores d'autres utilisateurs pour évaluer l'utilité des éléments. -Pas besoin d'une large communauté d'utilisateurs pour pouvoir effectuer des recommandations. -La qualité croit avec le temps. -La qualité de la recommandation peut être évaluée.	-Nécessite du profil d'utilisateur. -Trouver des utilisateurs ou groupes d'utilisateurs similaires est difficile. - Problème de confidentialité. - Démarrage a froid. - Nouvel Item. - Nouvelutilisateur.

Tableaux 1.2: Les avantages et les inconvénients des systèmes de recommandation

Les systèmes de recommandation hybrides combinent les techniques de recommandation dites pures pour atteindre une meilleure performance, en tirant profit des avantages des approches employées, et limitant les problèmes qui leurs sont liés.[15]

6. Domaine d'applications des systèmes de recommandations :

6.1. Service de commerce électronique :

Dans le passé, les items tels que les vêtements, la nourriture et les livres étaient principalement consommés par les utilisateurs via des magasins hors ligne. Cependant, ces dernières années, avec le développement des plateformes numériques tels que le web et les applications. La forme de consommation des items a changé à travers les plateformes de commerce électronique tels qu'Amazon, eBay et Alibaba. Le commerce électronique offre aux consommateurs de nombreux articles et diverses options dans l'environnement en ligne et offre aux vendeurs un moyen facile de vendre. En particulier, les consommateurs n'ont pas pu sortir en raison des mesures de confinement dues au COVID-19 et, par conséquent, ils n'ont pas pu utiliser les magasins hors ligne. Par conséquent, la consommation via les plateformes numériques a augmenté de façon exponentielle. Les catégories d'articles vendus sur les plateformes numériques ont également commencé à se diversifier .[16]

6.2. service de réseau social :

Les réseaux sociaux fournissent à la tâche de recommandation un nouveau paramètre qui est le cercle social de l'utilisateur. De ce fait, un système de recommandation social peut proposer à l'utilisateur des items qui sont populaires dans son cercle social [9]. Les systèmes de recommandation classique considèrent que les utilisateurs sont indépendants alors que les systèmes de recommandation sociale prennent en considération que l'utilisateur est connecté de plusieurs manières : amitié ou relation de confiance[17].

6.3. service de tourisme :

Les systèmes de recommandation existants dans le e-tourisme acquièrent les besoins et les désirs de l'utilisateur, explicitement (en demandant) ou implicitement (en explorant l'activité en ligne de l'utilisateur), et suggérer des destinations à visite, points d'intérêts, événements/activités ou tourisme complet paquets. L'objectif principal des SR de voyage est de faciliter le processus de recherche d'informations pour le voyageur et de le convaincre (persuader) de la pertinence des services proposés.[18]

6.4. service de l'éducation :

Les systèmes du e-learning utilisent les nouvelles technologies informatiques pour améliorer la qualité de l'éducation et aussi pour assurer un bon suivi de l'activité d'apprentissage de l'apprenant. Ces systèmes ont été largement utilisé dans l'éducation durant

la pandémie de COVID-19 et le confinement. Des fois les apprenants nécessitent des conseils auprès de leurs enseignants pour les guider durant leur apprentissage de choisir par exemple des cours, exercices ou travaux dirigés. Afin de répondre à ce besoin, de nombreuses différentes stratégies d'information et de recommandation ont été développé. Les systèmes de recommandation ont été intégré dans les environnements de e-learning, ils se basent sur les caractéristiques et les préférences des apprenant où des enseignants. En outre, ils peuvent baser sur les caractéristiques des autres utilisateurs ayant les mêmes données. Ces systèmes de recommandation collectent les informations et les activités des apprenants et recommande des items (documents, les pages Web, les cours, les personnes plus proche, etc) liées aux caractéristiques pédagogiques et aux intérêts des apprenants.[19]

6.5. service de santé :

Les systèmes de recommandation de santé (HRS) offrent le potentiel de motiver et d'inciter les utilisateurs à changer leur comportement en partageant de meilleurs choix et des connaissances exploitables basées sur le comportement observé des utilisateurs.[20]

6.6. service d'information académique :

En raison de l'augmentation exponentielle de la quantité d'information académiques , les chercheurs universitaires doivent consacres beaucoup de temps et d'efforts à la recherche d'information académiques dans le domaine lié à leur recherche. L'objectif principal de la recherche sur les systèmes de recommandation dans la domaine de l'information académiques et de recommander et de fournir des informations académiques adaptés à divers utilisateurs, y compris les communautés scientifique , les institues de recherche et les praticiens du développement ainsi que de soutenir la recherche elle-même. Par conséquent, un certain nombre d'études sur la recommandation d'information académique utilisant le modèle de recommandation de filtrage basé sur le contenu et le modèle de recommandation hybride utilisant des techniques de textmining ont été menées[16].

7. Conclusion :

Nous avons évoqué dans ce chapitre, les principes des systèmes de recommandation (SdR), outils efficaces qui facilitent la découverte d'items pouvant potentiellement nous intéresser, dans le contexte de la surcharge d'information auquel on doit faire face. Un SdR dispose de la capacité de prédire les goûts d'un utilisateur en fonction de son historique d'interaction avec le système. La prédiction peut se baser sur les goûts de l'utilisateur focal (approche CBF) ou sur les goûts d'utilisateurs similaires (approche CF). Des approches hybrides ont été également proposées afin de bénéficier des avantages de chaque type d'approche et pallier leurs inconvénients.

CHAPITRE02 :SERVICE WEB

1. Introduction :

Dans ce chapitre, nous décrivons les Services Web en mettant en évidence la définition, l'architecture SOA (Service-Oriented-Architecture), ainsi les technologies des Services Web., et les Principaux standards utilisé dans les services Web, enfin les avantages et les inconvénients des services web. Nous mettons aussi l'accent sur l'aspect non fonctionnel des services Web en présentant quelques paramètres de QoS. Et nous terminantnotrechapitre par une conclusion.

2. Définition des services web :

- Le groupe de W3C[21]qui travaille sur les services Web a défini le service Web comme « un système logiciel destine à supporter l'interaction ordinateur-ordinateur sur le réseau. Il a une interface décrite en un format traitable par l'ordinateur (ex. WSDL). Autres systèmes réagissent réciproquement avec le service web d'une façon prescrite par sa description en utilisant des messages SOAP, typiquement transmis avec le protocole http et une sérialisation XML, en conjonction avec d'autres standards relatifs
au web ».[22]
- Selon [23] : « Un service web est une agrégation de fonctionnalités publiées pour être utilisées. Il utilise internet comme conduit pour réaliser une tâche. Il est semblable à un processus métier virtuel qui définit des interactions au niveau application ».[22]
- Selon IBM[24] : « Les services web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto descriptive qui peuvent être publiées, localisées et invoquées depuis le web. Les services web effectuent des actions allant de simples requêtes à des processus complexes. Une fois qu'un service Web est déployé, d'autres applications peuvent le découvrir et l'invoquer ».[22]

3. Sélection des services Web :

La migration que suivent les entreprises vers les services Web et le développement de ce mode de service a induit à une croissance très prononcée dans le nombre de services proposé. Dans le dilemme du choix qui se pose aux utilisateurs des moyens de comparaison et de sélection doivent les guider. La sélection des services doit répondre aux besoins des utilisateurs fonctionnels (à base de qualité) ou non-fonctionnels soit elle (à base de critère de vie privée). Nous citons donc ces deux bases de sélection [25].

3.1 Sélection à base de qualité de service QoS-Aware :

La sélection à base de qualité de service dans les services Web œuvre pour satisfaire les différents critères non-fonctionnels des utilisateurs (exemple : temps, prix, marque, origine…)[25].qualités de service définissent les capacités de service à fonctionner dans de bonnes conditions en termes de disponibilité, performance, coût d'invocation, fiabilité, etc.[22]

3.2 Sélection à base de vie privée :

La sélection de service Web à base de vie privée touche aussi le côté des besoins non fonctionnels des utilisateurs. Les exploitations de données de vie privée sans avoir de permission des utilisateurs des services doit être résolu pour fournir un niveau acceptable de confidentialité. La sélection à base de vie privée consiste à prendre en charge les exigences et politiques de vie privée lors de la composition des services[25].

4. Qualité de Service (QoS) :

La QoS est un ensemble de propriétés et caractéristiques d'une entité ou d'un service qui lui confèrent l'aptitude à satisfaire des besoins déclarés ou implicites. Ces besoins peuvent être liés à des paramètres tels que l'accessibilité, la disponibilité, le temps de réponse, le coût, la fiabilité, etc. Ces paramètres peuvent être alors considérés comme un critère de choix lorsqu'on a affaire à sélectionner parmi plusieurs services web découverts ceux qui respectent les contraintes de temps imposées.[22]

Dans ce qui suit, nous présentons l'ensemble des attributs de QoS génériques. Les attributs les plus communs sont décrits par les paramètres liés à la performance.

- Le Débit : Le nombre de requêtes servies par unité de temps.
- Le Temps de Réponse : Le temps séparant l'émission de la requête et la réception des résultats.
- La Fiabilité : La capacité d'un service d'exécuter correctement ses fonctions.
- La Scalabilité : La capacité du service de traiter le plus grand nombre d'opérations ou de transactions pendant une période donnée tout en gardant les mêmes performances.
- La Robustesse : La probabilité qu'un service peut réagir proprement à des messages d'entrées invalides, incomplets ou conflictuels.
- La Disponibilité : La probabilité d'accessibilité d'un service.
- La Réputation : C'est une mesure de la crédibilité du service. Elle dépend principalement des expériences d'utilisateurs finaux.
- La sécurité : C'est un regroupement d'un ensemble de qualités à savoir : la confidentialité, le cryptage des messages et le contrôle d'accès.
- Coût : également appelé financière ou le cours, concerne le coût et les frais liés à un service. Le coût de la demande et l'utilisation de chaque service est le prix de service Web. Le prix des services est affecté par la valeur de la fonctionnalité. Fournir des fonctions plus complexes augmente le coût du service.[22]

5. Avantage service web :

Un des grands avantages des services Web c'est qu'il se repose sur les standards de l'internet et notamment le protocole http. De ce fait l'utilisation des Web Service n'est pas bloquée par proxy et Firewall, L'implémentation des différentes spécifications sur la majorité des langages de programmation à pour avantage de rendre extrêmement simple l'utilisation des Web Service[26].

6. Service web et système de recommandation :

Chaque propriétaire d'entreprise souhaite atteindre davantage de clients intéressés par ce qu'il propose. D'un autre côté, les utilisateurs ont du mal à trouver le contenu pertinent qu'ils recherchent parmi la multitude de services web créés chaque seconde. Il est utile de développer des systèmes qui aident les clients à trouver des services web. Le profil utilisateur, qui résume les préférences et les aversions d'un utilisateur, peut être pris en compte par ces systèmes. Le profil est généralement utilisé pour la personnalisation en ligne, une stratégie Internet visant à adapter les sites web à des utilisateurs spécifiques[27].

7. Conclusion :

les services web jouent un rôle essentiel dans la connectivité et l'échange d'informations sur Internet. Leur utilisation offre de nombreux avantages, tels que la facilité d'intégration, la réutilisabilité des services et la compatibilité multiplateforme. La sélection minutieuse des services web et la garantie d'une qualité de service optimale sont des facteurs clés pour répondre aux besoins des entreprises et offrir une expérience utilisateur satisfaisante.

CHAPITRE03 :
APPRENTISSAGE PROFOND

1. Introduction:

Depuis plusieurs années, l'intelligence artificielle (IA) connaît une très forte croissance de popularité aussi bien dans le milieu de la recherche scientifique qu'auprès des grandes compagnies des technologies de l'information comme Google, Amazon, Microsoft ou Facebook. Cette technologie s'invite aujourd'hui dans de nombreux domaines tels que les moteurs de recherche sur le Web.

Cette popularité est due aux récents développements d'un domaine de l'intelligence artificielle : l'apprentissage profond (deeplearning en anglais).

Dans ce chapitre nous allons présenter tout d'abord les notions en relation avec l'apprentissage profond.

2. Notions de base:

2.1. L'intelligence artificielle:

L'intelligence artificielle est née dans les années 50, quand une poignée de pionniers de l'informatique naissante ont commencé à se demander si les ordinateurs pouvaient être conçus pour « penser ». L'intelligence artificielle est donc un domaine général qui englobe l'apprentissage automatique et l'apprentissage profond, mais qui comprend également de nombreuses autres approches qui n'impliquent aucun apprentissage.[28]

Pendant très longtemps, de nombreux experts ont cru qu'une intelligence artificielle équivalente à celle de l'homme pouvait être atteinte en faisant en sorte que les programmeurs fabriquent un ensemble suffisamment large de règles explicites pour manipuler les connaissance.[28]

2.1.1. Relation entre l'intelligence artificielle et L'apprentissage profond :

Le Deep Learning est une sous-discipline de intelligence artificielle(IA), IA est un domaine général qui se concentre sur la création de systèmes intelligents capables de simules certaine capacités humaines telles que la perception , le raisonnement, l'apprentissage et la prise de décision.

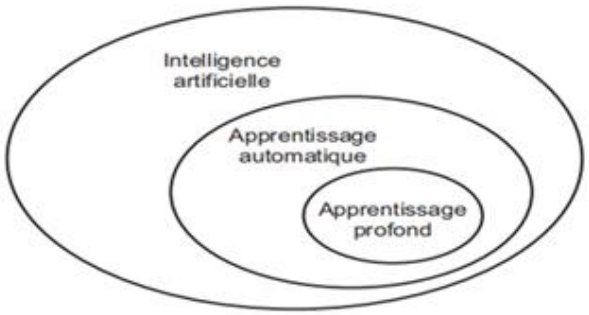


Figure2.1 Les relations entre l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond

2.2. Apprentissage automatique:

Dans le cas d'un programme informatique, qui est celui qui nous intéresse dans cet ouvrage, on parle d'apprentissage automatique, ou machine Learning, quand ce programme a la capacité d'apprendre sans que cette modification ne soit explicitement programmée. On peut ainsi opposer un programme classique, qui utilise une procédure et les données qu'il reçoit en entrée pour produire en sortie des résultats , à un programme d'apprentissage automatique, qui utilise les données et les résultats afin de produire le modèle qui permet d'obtenir les secondes à partir des premières.[29]

Afin que la machine converge vers la meilleure solution, la performance est mesurée généralement à l'aide d'une fonction d'analyse .Ensuite, une fois la phase d'apprentissage terminée, la machine est confrontée à des données qu'elle n'a jamais vues et, grâce aux règles déterminées pendant la phase d'apprentissage, elle sera à même de faire des prédictions [30]

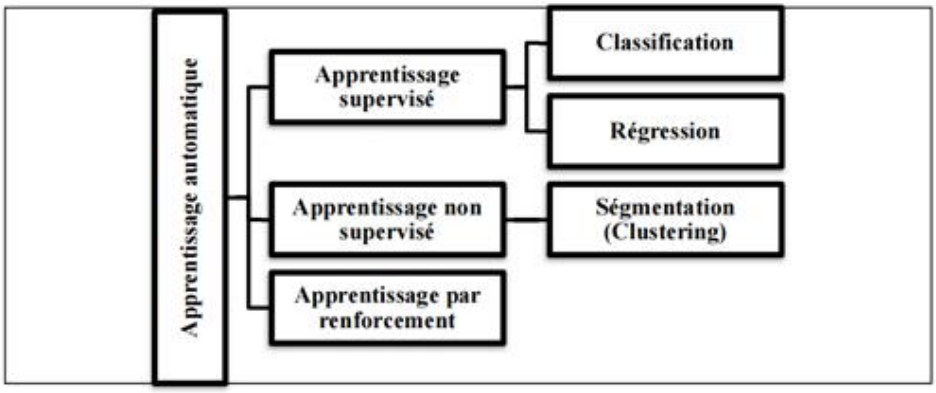


Figure 2.2 : Les grandes classes d'apprentissage automatique

2.2.1. Les avantages d'apprentissage automatique (machine learning) :

• Identifie facilement les tendances et les modèles :

L'apprentissage automatique peut examiner de gros volumes de données et découvrir des tendances et des schémas spécifiques qui ne seraient pas évidents pour l'homme. Par exemple, pour un site Web de commerce électronique comme Amazon, il permet de comprendre les comportements de navigation et les historiques d'achat de ses utilisateurs afin d'aider à proposer les bons produits, offres et rappels pertinents. Il utilise les résultats pour leur révéler des publicités pertinentes.[31]

• Amélioration continue :

Comme Algorithmes de ML (machine Learning) acquérir de l'expérience, ils continuent à améliorer en précision et en efficacité. Au fur et à mesure que la quantité de données dont vous disposez augmente, vos algorithmes apprennent à faire des prévisions plus précises plus rapidement.[31]

• Traitement de données multidimensionnelles et variétales :

Les algorithmes d'apprentissage automatique permettent de traiter des données multidimensionnelles et variées, et ce, dans des environnements dynamiques ou incertains.[31]

• Applications larges :

Vous pourriez être un détaillant en ligne ou un fournisseur de soins de santé et faire en sorte que ML (machine Learning) travaille pour vous. Le cas échéant, il est capable de fournir une expérience beaucoup plus personnelle aux clients tout en ciblant les bons clients.[31]

2.2.2. L'apprentissage automatique selon la nature des données en entrée:

a) Apprentissage non supervise:

L'apprentissage non supervisé est la forme la moins courante d'apprentissage. En effet, dans cette forme d'apprentissage, il n'y a pas de résultat attendu. On utilise cette forme d'apprentissage pour faire du clustering : on a un ensemble de données, et on cherche à déterminer des classes de faits.[32]

b) Apprentissage par renforcement :

Dans le cadre de l'apprentissage par renforcement, le système d'apprentissage peut interagir avec son environnement et accomplir des actions. En retour de ces actions, il obtient une récompense, qui peut être positive si l'action était un bon choix, ou négative dans le cas contraire. La récompense peut parfois venir après une longue suite d'actions ; c'est le cas par exemple pour un système apprenant à jouer au go ou aux échecs. Ainsi, l'apprentissage consiste dans ce cas à définir une politique, c'est-à-dire une stratégie permettant d'obtenir systématiquement la meilleure récompense possible. Les applications principales de l'apprentissage par renforcement se trouvent dans les jeux (échecs, go, etc) et la robotique.[29]

c) Apprentissage supervisé:

L'apprentissage supervisé est sûrement le plus courant. Il est utilisé pour des tâches d'estimation, de prévision, de régression ou de classification.[32]

Dans l'apprentissage supervisé, un ensemble d'exemples est fourni à l'algorithme d'apprentissage. Celui-ci va comparer la sortie obtenue Avec le Supervised Learning on peut développer des modèles pour résoudre 2 types de problèmes :

• Les problèmes de Régression

• Les problèmes de Classification

Dans les problèmes de régression, on cherche à prédire la valeur d'une variable continue, c'est-à-dire une variable qui peut prendre une infinité de valeurs le réseau avec la sortie attendue.[33]

Dans un problème de classification, on cherche à classer un objet dans différentes classes, c'est-à-dire que l'on cherche à prédire la valeur d'une variable discrète (qui ne prend qu'un nombre fini de valeurs).[33]

3. Les algorithmes d'apprentissage supervisé:

3.1. La méthode des K plus proches voisins:

En abrégé KNN, de l'anglais 'k-nearestneighbors', est un algorithme de reconnaissance des formes qui peut être utilisé autant pour la classification que pour la régression. C'est l'une des techniques non paramétriques fréquemment utilisée en prédiction financière non linéaire. Cette préférence est dû principalement à deux raisons : [34]

• premièrement, la simplicité algorithmique de la méthode comparée aux autres méthodes globales telles que les réseaux de neurones ou les algorithmes génétiques. [34]

• deuxièmement, la méthode KNN a démontré empiriquement une importante capacité de prédiction. [34]

L'idée de la méthode est de prédire le futur d'une série temporelle en analysant comment elle a évolué dans une situation similaire dans le passé. Ainsi, pour faire une prédiction on prend les données historiques les plus récentes disponibles et on cherche parmi ces données, les K plus proches instances appelés aussi les K plus proches vecteurs.[34]

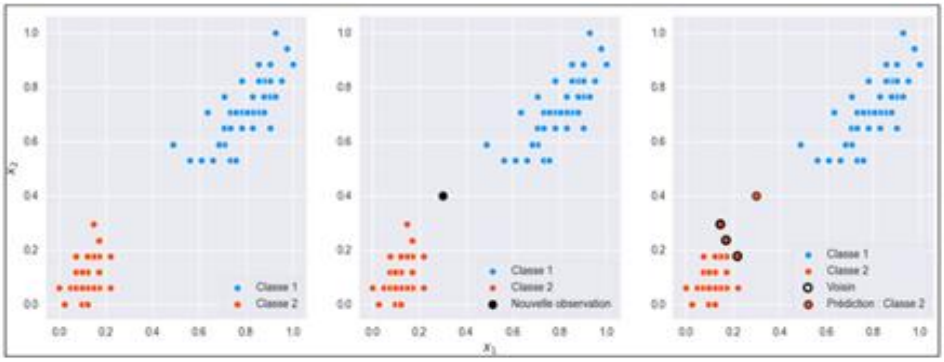


Figure 2.3 : Visualisation d'un exemple de classification en utilisant le method KNN(K=3)

3.1.2. Avantages de la méthode des k plus proches voisins:

La méthode des k plus proches voisins représente des avantages tels que :

1) L'algorithme KNN est robuste envers des données bruitées.
2) La méthode des k plus proches voisins est efficace si les données sont larges et incomplètes.
3) Cette méthode est l'une des plus simples de tous les algorithmes d'apprentissage automatique.[35]

3.1.3. Inconvénients de la méthode des k plus proches voisins:

La méthode des k plus proches voisins comporte des inconvénients tels que:

1) Le besoin de déterminer la valeur du nombre des plus proches voisins (le paramètre k).
2) Le temps de prédiction est très long puisqu'on doit calculer la distance de tous les exemples
3) Cette méthode est gourmande en espace mémoire car elle utilise une grande capacité de stockage pour le traitement des corpus[35]

3.2. La regression logistique:

La régression logistique est un modèle prédictif qui cherche à trouver des associations entre un vecteur de variables explicatives aléatoires, de nature numérique ou catégoriques, (X^1, $X^2 \ldots, X^K$), et une variable dépendante catégorique binaire ou multinomiale Y. Il s'agit tout simplement une transformation non linéaire de la régression linéaire, où on cherche à prédire une classe à la place d'une valeur numérique continue. Pour ce faire, on utilise une fonction logistique appelé aussi 'Sigmoïde', pour retourner les probabilités qui sont utilisées pour séparer les classes à prédire[34]

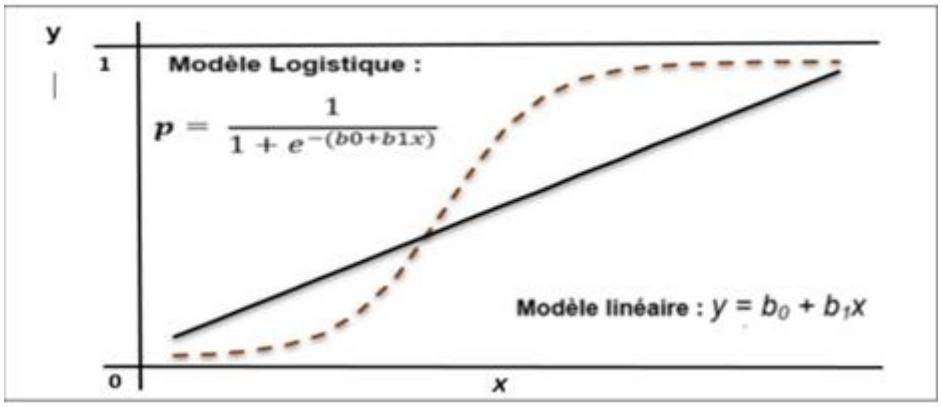


Figure 2.4: Comparaison entre une régression linéaire et un régression logistique

3.2.1. Les Avantages La regression logistique :

1) La régression logistique est plus facile à mettre en œuvre, à interpréter et très efficace à former.

2) Il ne fait aucune hypothèse sur les distributions des classes dans l'espace des caractéristiques.

3) Il peut facilement s'étendre à plusieurs classes (régression multinomiale) et une vue probabiliste naturelle des prédictions de classe.

4) Il est très rapide pour classer les enregistrements inconnus.[36]

3.2.2. Inconvenient La regression logistique :

1) Si le nombre d'observations est inférieur au nombre d'entités, la régression logistique ne doit pas être utilisée, sinon cela peut entraîner un surajustement.

2) Il construit des frontières linéaires.

3) La principale limitation de la régression logistique est l'hypothèse de linéarité entre la variable dépendante et les variables indépendantes.

4) Les problèmes non linéaires ne peuvent pas être résolus avec la régression logistique car elle a une surface de décision linéaire. Les données linéairement séparables sont rarement trouvées dans les scénarios du monde réel.[36]

3.3. K-moyen (K-mean) :

La méthode k-moyen a été introduite par MacQueen en 1967. En 1965 Forgy a publié un algorithme similaire, c'est pour cette raison qu'elle est parfois référée à Forgy. C'est une méthode efficace qui permet de diviser un ensemble de données en k classes homogènes .La méthode K-moyen est utilisée dans l'apprentissage non supervisé. Elle est itérative c'est-à-dire qu'elle converge vers une solution quelque soit son point de départ.[35]

Dans un premier temps, l'algorithme choisit le centre des k clusters à k objets. Par la suite, on calcule la distance entre les objets et les k centres et on affecte les objets aux centres les plus proches. Ensuite, les centres sont redéfinis à partir des objets qui ont été affectés aux différents clusters. Puis, les objets sont assignés en fonction de leur distance aux nouveaux centres et ainsi de suite. L'algorithme se répète donc jusqu'à ce qu'il y ait convergence[35]

3.3.1. Les Avantages du K-moyen :

La méthode du K-moyen comporte des avantages, comme par exemple:

1) L'assimilation de cette méthode est rapide.
2) La méthode simple à appliquer.
3) Son adaptation à de larges bases de données.[35]

3.3.2. Inconvénients du K-moyen :

La méthode du K-moyen comporte les inconvénients suivants:

1) La difficulté de comparer la qualité des clusters obtenus
2) La performance de l'algorithme dépend fortement de l'estimation initiale de la partition.
3) Le choix du paramètre k influence les résultats. [35]

3.4. Arbres de Décision :

Les arbres de décision sont une classe d'algorithme d'apprentissage souvent utilisés en classification. Un arbre est un graphe non orienté et sans cycle, composé d'un nœud racine, de nœuds internes et enfin de feuilles.[37]

3.5. Machine à Vecteurs Supports :

Les machines à vecteurs supports (Support Vector Machine - SVM) ont été proposées par Vapnik au milieu des années 90 [Vapnik, 1995]. L'hypothèse principale sur laquelle repose ces classifieurs binaires est la séparabilité des données de manière linéaire. L'apprentissage consiste à trouver l'hyperplan séparateur "optimal" qui va maximiser la distance minimale entre l'hyperplan séparateur et les exemples de la base d'apprentissage.

Malheureusement, les données sont rarement linéairement séparables. Pour répondre à cette problématique, il est possible de projeter les données d'apprentissage dans un espace de

plus grande dimension dans lequel il sera possible de trouver un hyperplan séparateur entre les classes.[37]

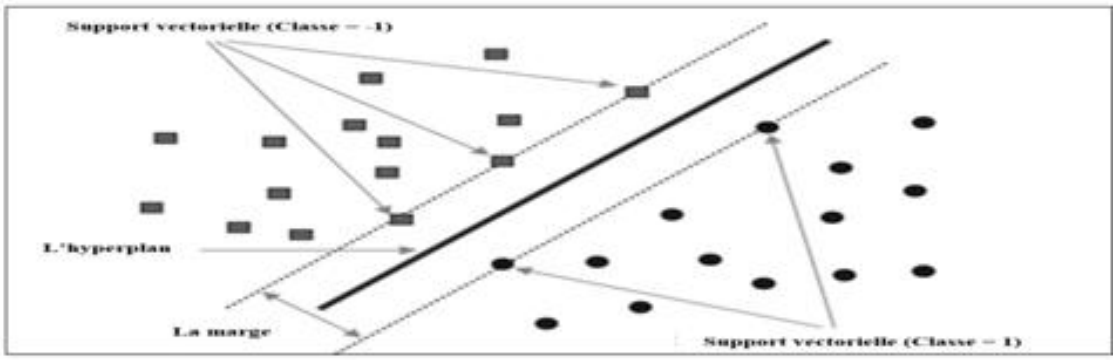


Figure 2.5: Exemple d'un SVM avec noyau linéaire

3.5.1. Avantages des SVM :

Les SVM représentent plusieurs avantages, notamment ceux-ci :

1) Elles ont une base théorique solide.
2) Les SVM sont efficaces dans les espaces de grande dimension.
3) Différentes fonctions noyau peuvent être spécifiées.[35]

3.5.2. Inconvénients des SVM

Malgré leurs performances, les SVM représentent aussi des faiblesses, notamment celles-ci :

1) Elles utilisent des fonctions mathématiques complexes pour la classification.
2) Les machines à support de vecteurs demandent un temps énorme durant les phases de test. [35]

4. l'apprentissage profond :

L'apprentissage profond est un sous-domaine spécifique de l'apprentissage automatique : c'est une nouvelle approche de l'apprentissage des représentations à partir des données qui met l'accent sur l'apprentissage de couches (layers) successives de représentations qui sont de plus en plus significatives L'adjectif « profond » de l'apprentissage profond ne fait pas référence à une forme de compréhension plus approfondie réalisée par l'approche mise en œuvre ; il représente plutôt l'idée de couches successives de représentations. Le nombre de couches qui contribuent à un modèle de données s'appelle la profondeur (depth) du modèle D'autres noms appropriés pour l'apprentissage profond auraient pu être l'apprentissage des représentations en couches (layeredrepresentationslearning) et l'apprentissage des représentations hiérarchiques (hierarchicalrepresentationslearning). [28]

L'apprentissage profond moderne comprend souvent des dizaines, voire des centaines, de couches successives de représentations celles-ci apprennent toutes automatiquement à partir de l'exposition aux données d'apprentissage. Cependant, d'autres approches de l'apprentissage automatique tendent à privilégier l'apprentissage d'une ou deux couches de représentations de données ; c'est pourquoi elles sont parfois nommées apprentissages non profonds (shallowlearning)[28]

En apprentissage profond, ces représentations en couches sont (presque toujours) apprises au moyen de modèles appelés réseaux de neurones (neural networks), structurés en couches littéralement superposées les unes sur les autres. L'expression réseau de neurones est une référence à la neurobiologie mais, bien que certains concepts fondamentaux de l'apprentissage profond aient été développés en partie en s'inspirant de notre compréhension du cerveau, les modèles d'apprentissage profond ne sont pas des modèles du cerveau.

Rien n'indique que le cerveau mette en œuvre des mécanismes d'apprentissage semblables à ceux utilisés dans les modèles modernes d'apprentissage profond.[28]

Pour nos besoins ici, l'apprentissage profond est un cadre mathématique pour l'apprentissage des représentations à partir de données.

4.1. Différence entre Machine Learning et Deep Learning :

Apprentissageautomatique	L'apprentissage en profondeur
Appliquez des algorithmes statistiques pour apprendre les modèles cachés et les relations dans l'ensemble de données.	Utilise une architecture de réseau de neurones artificiels pour apprendre les modèles et les relations cachés dans l'ensemble de données.
Peut travailler sur la plus petite quantité de jeu de données	Nécessite un plus grand volume d'ensembles de données par rapport à l'apprentissage automatique
Mieux pour la tâche à faible étiquette.	Mieux pour les tâches complexes comme le traitement d'images, le traitement du langage naturel, etc**.**
Prend moins de temps pour former le modèle.	Prend plus de temps pour former le modèle
Un modèle est créé par des caractéristiques pertinentes extraites manuellement des images pour détecter un objet dans l'image.	Les caractéristiques pertinentes sont automatiquement extraites des images. C'est un processus d'apprentissage de bout en bout

TABLEAU 2.1: Les différences entre Machine Learning et Deep Learning

4.2. Historique deep learning:

L'histoire de Deep Learning remonte à 1943, lorsque Walter Pitts et Warren McCulloch avaient créé un modèle de calcul pour réseaux de neurones basé sur des mathématiques et des algorithmes, appelé logique à seuil - ou thresholdlogic- pour imiter le processus de pensée du cerveau humain.[38]

En 1958, Frank Rosenblatt a créé le perceptron , un algorithme de reconnaissance de formes basé sur un réseau de neurones à deux couches.[38]

Les premiers efforts dans le développement d'algorithmes d'apprentissage profond remontent à 1965, lorsque Ivakhnenko et al. utilisaient des modèles avec des fonctions d'activation polynomiales (équations complexes).[38]

Au cours des années 70, l'intelligence artificielle a fait l'objet d'un bref recul. Le manque de financement a limité la recherche en apprentissage en profondeur et en intelligence artificielle. Cependant, cetrains ont poursuivi la recherche sans financement pendant ces années difficiles. Kunihiko Fukushima a utilisé les premiers réseaux de neurones convolutifs conçus avec plusieurs couches de pooling et de convolution. En 1979, il a développé un réseau de neurones artificiels multi-couches, appelé Neocognitron . Son travail, qui a été fortement influencé par Hubel et Wiesel, a conduit au développement des premiers réseaux de neurones convolutifs, basés sur l'organisation du cortex visuel chez l'animal.Ce sont des variantes de perceptrons multi-couches conçues pour utiliser un minimum de prétraitement.[38]

Fin des années 90 et au début des années 2000. C'est également à cette époque que la deuxième rupture du développement de l'intelligence artificielle a eu lieu (1985-1990), ce qui a également permis d'effectuer des recherches sur les réseaux de neurones et le Deep Learning. En 1995, Dana Cortes et Vladimir Vapnik ont mis au point la machine à vecteurs de support (SVM). La LSTM (mémoire à court et long terme) pour les réseaux de neurones récurrents a été développée en 1997 par Sepp Hochreiter et Juergen Schmidhuber.[38]

La prochaine étape importante de l'évolution du Deep Learning a eu lieu en 1999,lorsque les ordinateurs ont commencé à accélérer le traitement des données et que les unités de traitement graphique (GPU) ont été développées.[38]

Vers l'année 2000, le problème du gradient évanescent est apparu. Il a été découvert que les caractéristiques formées dans les couches inférieures n'étaient pas apprises par les couches supérieures, car aucun signal d'apprentissage n'atteignait ces couches. En 2009, l'atelier NIPS travaillant sur l'apprentissage profond pour la reconnaissance vocale découvre qu'avec un ensemble de données suffisamment important, les réseaux de neurones n'ont pas

besoin d'un apprentissage préalable et que les taux d'erreur chutent considérablement. D'où l'apparition de la base de données ImageNet créée par Fei-Fei Li, professeur d'intelligence artificielle à Stanford. Il s'agit d'une base de données gratuite de plus de 14 millions d'images étiquetées.[38]

En 2011, la vitesse des GPU avait considérablement augmenté, ce qui permettait d'entraîner des réseaux de neurones convolutifs sans l'apprentissage préalable couche par couche. Avec l'augmentation de la vitesse de calcul.

AlexNet, un réseau de neurones convolutif créé par Alex Krizhevsky dont l'architecture a remporté plusieurs concours internationaux en 2011 et 2012, en est un exemple. AlexNet a construit et amélioré LeNet5[38]

4.3. Avantages de l'apprentissage en profondeur:

L'apprentissage en profondeur présente plusieurs avantages par rapport aux méthodes traditionnelles d'apprentissage automatique, dont voici quelques-uns:

- **Apprentissageautomatique des fonctionnalités :**

les algorithmes d'apprentissage en profondeur peuvent automatiquement apprendre les fonctionnalités à partir des données, ce qui signifie qu'ils ne nécessitent pas que les fonctionnalités soient conçues à la main. Ceci est particulièrement utile pour les tâches où les fonctionnalités sont difficiles à définir, telles que la reconnaissance d'images.[39]

- **Traitement de données volumineuses et complexes :**

les algorithmes d'apprentissage en profondeur peuvent gérer des ensembles de données volumineux et complexes qui seraient difficiles à traiter pour les algorithmes d'apprentissage automatique traditionnels. Cela en fait un outil utile pour extraire des informations à partir de données volumineuses.[39]

- **Amélioration des performances :**

il a été démontré que les algorithmes d'apprentissage en profondeur atteignent des performances de pointe sur un large éventail de problèmes, notamment la reconnaissance d'images et de la parole, le traitement du langage naturel et la vision par ordinateur.[39]

- **Gestion des données structurées et non structurées :**

les algorithmes d'apprentissage en profondeur peuvent gérer à la fois des données structurées et non structurées telles que des images, du texte et de l'audio.[39]

- **Gestion des données manquantes :**

les algorithmes d'apprentissage en profondeur peuvent gérer les données manquantes tout en faisant des prédictions, ce qui est utile dans les applications du monde réel où les données sont souvent incomplètes.[39]

- **Gestion des données séquentielles :**

les algorithmes d'apprentissage en profondeur tels que les réseaux de neurones récurrents (RNN) et les réseaux de mémoire longue à court terme (LSTM) sont particulièrement adaptés pour gérer les données séquentielles telles que les séries chronologiques, la parole et le texte. Ces algorithmes ont la capacité de maintenir le contexte et la mémoire au fil du temps, ce qui leur permet de faire des prédictions ou des décisions basées sur des entrées passées.[39]

- **Évolutivité :**

les modèles d'apprentissage en profondeur peuvent être facilement mis à l'échelle pour gérer une quantité croissante de données et peuvent être déployés sur des plates-formes cloud et des appareils périphériques.[39]

- **Généralisation :**

les modèles d'apprentissage en profondeur peuvent bien se généraliser à de nouvelles situations ou de nouveaux contextes, car ils sont capables d'apprendre des représentations abstraites et hiérarchiques des données.[39]

L'apprentissage en profondeur présente plusieurs avantages par rapport aux méthodes d'apprentissage automatique traditionnelles, notamment l'apprentissage automatique des fonctionnalités, la gestion de données volumineuses et complexes, l'amélioration des performances, la gestion des relations non linéaires, la gestion des données structurées et non structurées, la modélisation prédictive, la gestion des données manquantes, la gestion des données séquentielles, l'évolutivité et capacité de généralisation.[39]

4.4. Inconvénients du Deep Learning:

Bien que l'apprentissage en profondeur présente de nombreux avantages, il existe également certains inconvénients à prendre en compte :

• **Coût de calcul élevé:** la formation de modèles d'apprentissage en profondeur nécessite des ressources de calcul importantes, notamment des GPU puissants et de grandes quantités de mémoire. Cela peut être coûteux et prendre du temps.[39]

• **Sur-ajustement:** le sur-ajustement se produit lorsqu'un modèle est trop bien entraîné sur les données d'entraînement et fonctionne mal sur de nouvelles données invisibles. Il s'agit d'un

problème courant dans l'apprentissage en profondeur, en particulier avec les grands réseaux de neurones, et peut être causé par un manque de données, un modèle complexe ou un manque de régularisation.[39]

• **Manque d'interprétabilité:** les modèles d'apprentissage en profondeur, en particulier ceux comportant de nombreuses couches, peuvent être complexes et difficiles à interpréter. Cela peut compliquer la compréhension de la façon dont le modèle fait des prédictions et l'identification d'erreurs ou de biais dans le modèle.[39]

• **Dépendance à la qualité des données:** les algorithmes d'apprentissage en profondeur reposent sur la qualité des données sur lesquelles ils sont formés. Si les données sont bruyantes, incomplètes ou biaisées, les performances du modèle seront affectées négativement.[39]

• **Problèmes de confidentialité et de sécurité des données:** étant donné que les modèles d'apprentissage en profondeur reposent souvent sur de grandes quantités de données, la confidentialité et la sécurité des données suscitent des inquiétudes. L'utilisation abusive des données par des acteurs malveillants peut entraîner de graves conséquences telles que le vol d'identité, la perte financière et l'atteinte à la vie privée.[39]

• **Manque d'expertise dans le domaine:** l'apprentissage en profondeur nécessite une bonne compréhension du domaine et du problème que vous essayez de résoudre. Si l'expertise du domaine fait défaut, il peut être difficile de formuler le problème et de sélectionner l'algorithme approprié.[39]

• **Conséquences imprévues:** les modèles d'apprentissage en profondeur peuvent entraîner des conséquences imprévues, par exemple, un modèle biaisé peut discriminer certains groupes de personnes, ce qui entraîne des préoccupations éthiques.[39]

• **Limité aux données sur lesquelles il a été formé:** les modèles d'apprentissage en profondeur ne peuvent faire des prédictions que sur la base des données sur lesquelles ils ont été formés. Ils peuvent ne pas être en mesure de généraliser à de nouvelles situations ou à de nouveaux contextes qui n'étaient pas représentés dans les données de formation.[39]

• **Modèles de boîte noire:** certains modèles d'apprentissage en profondeur sont considérés comme des modèles de « boîte noire », car il est difficile de comprendre comment le modèle fait des prédictions et d'identifier les facteurs qui influencent les prédictions.[39]

bien que l'apprentissage en profondeur présente de nombreux avantages, il présente également certaines limites, telles que le coût de calcul élevé, le surajustement, le manque d'interprétabilité, la dépendance à la qualité des données, les problèmes de confidentialité et de sécurité des données, le manque d'expertise du domaine, les conséquences imprévues,

limité aux données qu'il est formé modèles on et black-box. Il est important de tenir compte de ces limitations lors de l'application de l'apprentissage en profondeur à un problème.[39]

4.5. Quels sont les différents types de fonctions d'activation utilisées dans le deeplearning?

Dans l'apprentissage en profondeur, plusieurs types différents de fonctions d'activation sont utilisés. Chacun d'eux a sa force et sa faiblesse. Certaines des fonctions d'activation les plus courantes sont les suivantes.

• **Fonction sigmoïde** : Elle mappe toute valeur entre 0 et 1. Elle est principalement utilisée dans les problèmes de classification binaire. où il mappe la sortie de la couche cachée précédente dans la valeur de probabilité.[40]

• **Fonction Softmax :** C'est l'extension de la fonction sigmoïde utilisée pour les problèmes de classification multi-classes dans la couche de sortie du réseau de neurones, où elle mappe la sortie de la couche précédente dans une distribution de probabilité entre les classes, donnant à chaque classe une valeur de probabilité entre 0 et 1 avec la somme des probabilités sur toutes les classes égale à 1. La classe qui a la valeur de probabilité la plus élevée est considérée comme la classe prédite.[40]

• **Fonction ReLU (RectifiedLinear Unit):** C'est une fonction non linéaire qui renvoie la valeur d'entrée pour les entrées positives et 0 pour les entrées négatives. Les réseaux de neurones profonds utilisent fréquemment cette fonction car elle est à la fois simple et efficace.[40]

• **Fonction LeakyReLU:** Elle est similaire à la fonction ReLU, mais elle ajoute une petite pente pour les valeurs d'entrée négatives afin d'éviter les neurones morts.[40]

• **Fonction Tanh (tangente hyperbolique):** il s'agit d'une fonction d'activations non linéaires qui mappe la valeur de l'entrée entre -1 et 1. Elle est similaire à la fonction sigmoïde mais elle fournit des résultats positifs et négatifs. Il est principalement utilisé pour les tâches de régression, où la sortie sera des valeurs continues.[40]

4.6. Comment optimiser un modèle de Deep Learning ?

Un modèle de Deep Learning peut être optimisé en modifiant ses paramètres et hyper paramètres pour augmenter ses performances sur une tâche particulière. Voici quelques méthodes typiques d'optimisation des modèles d'apprentissage en profondeur:

- ✓ Choisir la bonne architecture
- ✓ Réglage du tauxd'apprentissage
- ✓ Régularisation
- ✓ Augmentation des données

✓ Apprentissage par transfert

✓ Réglage des hyper paramètres[40]

5. Réseaux de neurones:

5.1. Historique:

Dès le début des années 40, les chercheurs se sont intéressés au fonctionnement du neurone biologique et à sa modélisation [McCullogh and Pitts, 1943]. Ceci a conduit au modèle du neurone formel encore utilisé aujourd'hui.

En 1949, Hebb [Hebb, 1949] propose d'associer plusieurs neurones pour réaliser l'apprentissage. Ceci a amené quelques années plus tard au perceptron [Rosenblatt, 1958] composé d'une couche de neurones d'entrée et d'une couche de neurones de sortie. Ce réseau très simple s'est avéré incapable d'apprendre des fonctions non linéaires mais a servi de base pour construire le perceptron multicouches.[37]

5.2. Les réseaux de neurones :

Les réseaux de neurones ont été développés comme un modèle mathématique générique afin de modéliser les neurones biologiques. Ils comportent un certain nombre d'éléments de traitement d'information appelés neurones. Chaque neurone a son propre état interne interprété par la fonction d'activation. Il envoie son activation aux autres neurones sous forme de signaux. La connexion entre les neurones est réalisée via des liens orientés et pondérés[35].

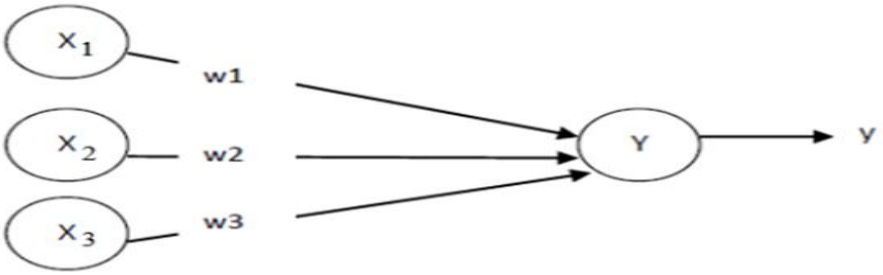


Figure 2.6 : Neurone artificiel avec une seule sortie

Le neurone Y reçoit les entrées de X_1,X_2 et X_3 qui ont comme valeurs de sortie x_1, x_2, x_3. Les poids des liens de connexion de X_1, X_2, X_3 sont . La valeur d'entrée de neurone Y est : y = W_1x_3+W_2x_2+W_3x_3. Le signal de sortie y . est déterminée par la fonction d'activation f(y) . Les réseaux de neurones sont caractérisés par l'architecture (l'organisation des neurones), l'apprentissage (méthode de détermination des poids de connexions), et par leur fonction d'activation.[35]

5.3. Architecture:

Les réseaux de neurones sont souvent classifiés en deux architectures : les réseaux de neurone avec un seul niveau et multi-niveaux. Le nombre de niveaux est calculé sans prendre en considération les unités[35]

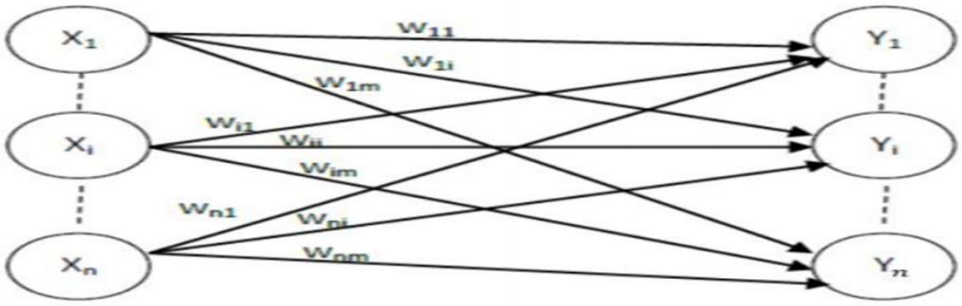


Figure 2.7 : Le réseau de neurones à un seul niveau

Les neurones de la couche d'entrée doivent uniquement passer et distribuer les entrées et ne pas effectuer de calcul. Ainsi, la seule vraie couche de neurones est celle de droite Chacune des entrées X_2 , X_2, $X_3 \ldots X_N$est connectée à chaque neurone de la couche de sortie à travers le poids de lien. Comme chaque valeur des sorties Y_1 , Y_2 , Y_3 est calculée à partir du même ensemble de valeurs d'entrée, chaque sortie est modifiée en fonction des poids de liens.[35]

5.4. Fonctionnement des réseaux de neurones artificiels:

La plupart des algorithmes d'apprentissage profond reposent sur des modèles appelés réseaux de neurones artificiels. Il existe de nombreuses variantes de ces réseaux qui ont été adaptées pour des problématiques spécifiques, et, par simplicité, les éléments communs seront présentés dans cette section. Par la suite, un type de modèle très utilisé pour la reconnaissance d'image sera présenté : les réseaux de neurones convolutifs.[30]

Avant de présenter le fonctionnement d'un réseau de neurones, intéressons-nous à l'anatomie d'un réseau de neurones artificiels.

Dans sa forme la plus simple, le plus petit élément d'un réseau est le neurone. Il prend la forme de plusieurs nœuds entrants qui sont connectés à un nœud de sortie (figure 2.8). Un poids est associé à chaque nœud entrant permettant de calculer une somme pondérée en multipliant chaque poids par la valeur des nœuds entrant. Le résultat de cette somme est transmis à une fonction d'activation qui va calculer la valeur de sortie de ce neurone. La valeur de sortie de ce neurone est ensuite transmise à un autre neurone

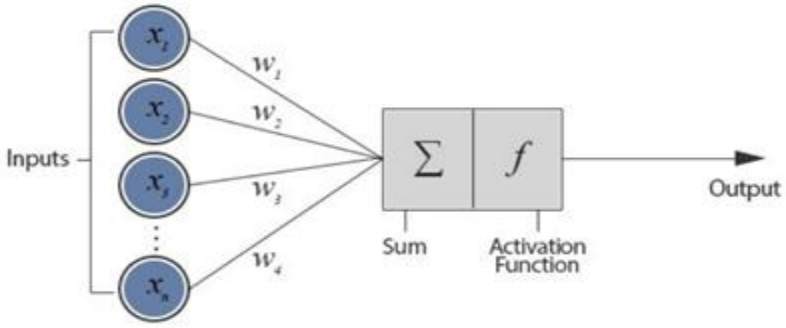


Figure 2.8: Représentation d'un neurone artificielle extraite et adaptée de Data Camp(2018)

La fonction d'activation est un élément essentiel aux réseaux de neurones. Elle introduit une non linéarité dans le modèle et permet d'étendre la dimension de l'espace des hypothèses. Sans cette fonction d'activation, le modèle pourrait seulement apprendre des transformations linéaires ce qui restreindrait trop l'espace des hypothèses.[30]

Les fonctions d'activation les plus courantes sont :

- La fonction ReLU (RectifiedLinear Activation)
- La fonction sigmoïde

5.5. Avantages des réseaux de neurones :

Les réseaux de neurones représentent des avantages, tels que :

1) Les réseaux de neurones sont souples et génériques. Ils peuvent résoudre différents types de problèmes dont le résultat peut être : une classification, analyse de données,etc

2) Ils traitent des problèmes non structurés sur lesquels aucune information n'est disponible à l'avance.

3) Les réseaux neuronaux se comportent bien parce que même dans des domaines très complexes, ils fonctionnent mieux que les arbres de statistique ou de décision.

4) Les réseaux de neurones fonctionnent sur des données incomplètes ou bruitées. Cette lacune d'information peut être complétée par l'ajout d'autres neurones à la couche cachée[35]

5.6.Inconvénients des réseaux de neurones :

Les réseaux de neurones ont aussi des inconvénients, tels que :

1) La lenteur d'apprentissage.

2) La difficulté de choisir des valeurs initiales des poids de connexion ainsi que l'adaptation du pas d'apprentissage.

3) L'apprentissage au détriment de la généralisation.

4) En cas d'erreur dans les résultats de sorties, l'utilisateur n'a aucune information sur le fonctionnement interne.[35]

6. Architectures de réseaux de neurones profonds:

6.1. Réseaux de neurones convolutifs (CNN):

Les réseaux de neurones convolutifs (CNN) ont été introduits par Lecun . La particularité des CNN est l'utilisation de l'opération de convolution dans les premières couches intermédiaires du réseau de neurones. À l'origine, cette opération est utilisée comme fifiltre dans le domaine de l'image ou du son afifin de mettre en évidence des motifs ou réduire un type de bruits. [14]

Un CNN classique est généralement composé de quatre types de couches :

✓ les couches convolutives, qui contiennent plusieurs opérations de convolutions appliquées sur la même entrée,

✓ les couches d'opérations de mise en commun,

✓ les couches d'activations,

✓ les couches toutes connectées.

a. **Les couches convolutives:** À l'origine, l'opération de convolution est utilisée sur des données temporelles (sons) ou spatiales (images) en tant que fifiltre linéaire

b. **Opération de mise en commun:** Les couches convolutives peuvent être suivies d'une opération de mise en commun. Elle a pour but de réduire la dimension des couches de neurones en regroupant les informations présentes sur les neurones proches les uns des autres. Les CNN sont principalement utilisés dans le domaine de l'image, dans lequel ils dépassent les autres méthodes d'apprentissage[41]

6.1.1. Avantages CNN:

1) Très grande précision dans les problèmes de reconnaissance d'image.
2) Détecte automatiquement les caractéristiques importantes sans aucune supervision humaine.
3) Partage du poids.[42]

6.1.2. Désavantages CNN:

1) CNN n'encode pas la position et l'orientation de l'objet.
2) Manque de capacité à être spatialement invariant aux données d'entrée.
3) De nombreuses données d'entraînement sont nécessaires.[42]

6.2. Quelques réseaux convolutifs célèbres:

❖ **LeNet :**LeNet est la première architecture de type CNN proposée pour la classification supervisée. Ce réseau est composé de 7 couches au total : 3 couches de convolution , 2 couches de Avg-pooling et 2 couches entièrement connectées .L'apprentissage à base de rétro propagation sur la base d'apprentissage MNIST a montré l'efficacité de l'algorithme

d'apprentissage profond LeNet par rapport aux algorithmes d'apprentissage automatique classiques SVM et KNN. La base d'apprentissage MNIST a été conçue pour la classification des chiffres manuscrits et contient 60 000 instances pour l'apprentissage et 10 000 instances pour le test.[43]

❖ **AlexNet:** En 2012, le réseau AlexNet a été proposé dans la compétition ImageNet de reconnaissance visuelle à grande échelle (ILS VRC). Cette compétition utilise un sous ensemble de la base d'apprentissage ImageNet composé de 1000 catégories, 1.2 million instances d'apprentissage, 50 000 instances de validation, et 150 000 instances de test .Dans cette compétition, le réseau AlexNet a atteint le meilleur taux d'erreur par rapport aux autres méthodes d'apprentissage automatique classiques. Ce résultat était un point important dans l'historique des réseaux d'apprentissage profond, car l'intérêt s'est attiré vers les méthodes DL plus que les méthodes ML classiques en vision par ordinateur. Le réseau AlexNet a une architecture similaire et plus profonde par rapport à LeNet. Ce réseau est composé de 5 couches de convolution et 3 couches entièrement connectées[43]

❖ **ZFnet :** Malgré la performance de AlexNet dans ILSVRC, n'ont pas justifié le choix des hyper paramètres (taille des filtres, nombre de couches) et comment les ajuster pour améliorer la performance du CNN. En plus, le comportement du réseau et son fonctionnement interne restent ambigus d'un point de vue scientifique .Affin de comprendre le comportement des CNN et améliorer leur per formance, ont proposé une nouvelle méthode de visualisation qui permet de déchiffrer le contenu des couches intermédiaires. Le but de cette visualisation et d'étudier le comportement de Alex Net et de proposer la version optimisée ZFNet .La technique de Visual sation est basée sur le réseau déconvolutionnel multicouche (Deconvnet) . Ce réseau permet de projeter les cartes des caractérise tiques internes aux entrées affin de visualiser leur contenu.[43]

❖ **GoogLeNet :** .Proposé en 2014 par C. Szegedy et al. le réseau de neurones appelé GoogleNet gagne le challenge ILSVRC (challenge de classification d'images sur la base de données ImageNet) cette même année en proposant de multiples améliorations à l'architecture du CNN. La plus notable est l'utilisation de modules à branches appelés module d'Inception. L'idée de base est de multiplier les filtres avec des tailles différentes. le module contient 4 branches, 3 couches de convolutions avec des filtres de taille 5 × 5, 3 × 3 et 1 × 1, et une opération de mise en commun avec des fenêtres de taille 3×3. Les opérations de convolution 1×1 consistent à mettre en relation uniquement les différents canaux sur une même position de l'image .[41]

❖ **ResNet:**ResNet est un réseau de neurones convolutif basé sur des blocs résiduels. Le but principal de cette architecture est de résoudre le problème de dégradation de gradient. Ce problème apparait dans les réseaux très profonds, où la précision commence à être saturée ensuite dégrade rapidement à cause de la diminution des valeurs des gradients. Affin de résoudre ce problème, les blocs résiduels ont été introduits.[43]

❖ **VGG Net :** VGGNet est un réseau de neurones convolutif qui est basé sur les mêmes principes du réseau AlexNet. Le but de cette version est de proposer des configurations profondes (16 à 19 couches) en se basant sur la technique de stabilisation structurelle. Cette technique permet de contrôler le nombre des paramètres dans les réseaux profonds affin de diminuer les risques de sur-apprentissage.[43]

6.3. Réseaux de neurones récurrents (RNN):

Les réseaux de neurones récurrents (RNN) ont été développés affin de garder un contexte temporel pour chaque événement en entrée. Ils ont été particulièrement utilisés pour de l'analyse de séries temporelles, de données audio, ou de textes dans lesquelles le contexte est important affin d'analyser chaque nouvelle entrée. L'idée consiste à garder des informations au cours du temps à l'intérieur des couches de neurones affin de donner un contexte aux données analysées.[14]

6.3.1. Avantages RNN :

1) Un RNN se souvient de chaque information au fil du temps. Il est utile dans la prédiction de séries chronologiques uniquement en raison de la fonctionnalité permettant de mémoriser également les entrées précédentes. C'est ce qu'on appelle la mémoire à long terme.

2) Les réseaux de neurones récurrents sont même utilisés avec des couches convolutives pour étendre le voisinage effectif des pixels[15].

6.3.2. Désavantages RNN:

1) Problèmes de disparition et d'explosion de dégradé.

2) La formation d'un RNN est une tâche très difficile.

3) Il ne peut pas traiter de très longues séquences s'il utilise tanh ou relu comme fonction d'activation.[15]

6.4. Cellules LSTM:

Les cellules Long Short-Term Memory (en français les réseaux récurrents à mémoire courte et long terme) ont été introduites par Hochreiter et al. en1997. Le but étant de faire face aux problèmes de disparition du gradient lors que l'élément courant et son contexte sont trop éloignés dans le temps Les cellules LSTM ont été particulièrement utilisées à l'intérieur de réseaux de neurones profonds[41]

6.5. Différence entre CNN ET RNN :

	CNN	RNN
Type de données	Données d'image	Données de séquence
Partage de paramètres	Oui	Oui
Entrée de longueur fixe	Oui	Non
Connexions récurrentes	Non	Oui
Dégradé de disparition et d'explosion	Oui	Oui
Relation spatiale	Oui	Non
Performance	CNN est considéré comme plus puissant que RNN.	RNN inclut moins de compatibilité de fonctionnalités par rapport à CNN.
Application	Reconnaissance faciale, numérisation de texte et traitement du langage naturel	Conversions texte-parole
Principaux avantages	Haute précision dans les problèmes de reconnaissance d'image, partage de poids.	Se souvient de chaque information, prédiction de séries chronologiques
Désavantages	Grandes données d'entraînement nécessaires, n'encodez pas la position et l'orientation de l'objet.	Gradient qui disparaît, qui explose.

Tableaux 2.2 : Différentes entre CNN et RNN

6.6. Perceptron Multicouche:

Le Perceptron Multicouche (MLP de l'anglais MultiLayer Perceptron) est un réseau de neurones composé d'au moins 3 couches :une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. La couche d'entrée est passive : elle n'applique pas d'opération mathématiques et ne fait que transférer l'entrée à la première couche cachée. couche précédente, lui applique un produit matriciel, ajoute un biais et applique une opération non-linéaire qui, dans la majorité des cas, est une tangente hyperbolique, une sigmoïde ou une unité linéaire rectifiée (ReLU de l'anglais RectifiedLinear Unit (Malik et Perona,

1990)).De façon plus formelle, la sortie de la $k^{ème}$ couche (note X^k)d'un MLP se calcule de la façon suivante :

$X^k = \sigma(b^k + w^k * X^{K-1})$ [44]

Avec b^k et w^k le biais et la matrice associés à la $k^{ème}$ couche cachée dont les coefficients font partie des paramètres du MLP. σ dénote une fonction non linéaire. Un MLP répète ce type d'opération autant de fois qu'il y a de couches cachées jusqu'à la couche de sortie .La sortie de chaque couche d'un MLP (et plus généralement d'un réseau de neurones à plusieurs couches) peut être vue comme un vecteur de caractéristiques représentant les données en entrée. Plus la couche est profonde dans l'architecture, plus le vecteur de caractéristiques qu'il renvoie est difficilement interprétables parun humain. Ce sont ces différents niveaux de représentations qui permettent aux réseaux de neurones de modéliser les données en entrée de façon utile pour la tâche considérée .La principale différence entre l'apprentissage profond à base de réseaux de neurones et les méthodes traditionnelles d'apprentissage automatique est que ces dernières modélisent les données à l'aide de vecteurs de caractéristiques dont les règles de construction sont définies par des experts du domaine de la tâche à résoudre tandis que les réseaux de neurones apprennent automatiquement des vecteurs de caractéristiques adaptés à la tâche considérée.[44]

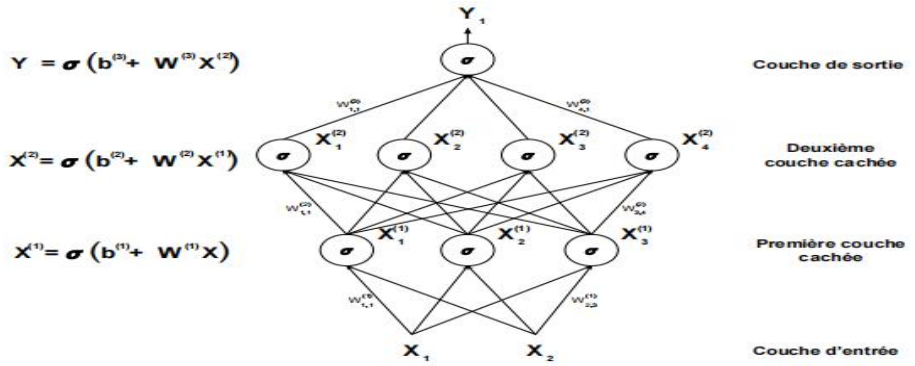


Figure 2.9 : Architecture d'un MLP de R2 dans R à deux couches cachées de tailles h1 " 3 et h2 " 4

7.hybride basé sur l'apprentissage profondsystèmes de recommandation :

Les méthodes traditionnelles basées sur CF utilisent la matrice utilisateur-élémentqui encode les préférences individuelles des utilisateurs pour les élémentspour apprendre à faire des recommandations. Dans les applications réelles,la matrice de notation est généralement très clairsemée, ce qui entraîne uneméthodes de dégrader considérablement les performances des recommandations. Dans ce cas, certaines méthodes CF améliorées utilisent lequantitécroissante d'informations secondaires pour traiter les donnéesproblème de

parcimonie ainsi que le problème de démarrage à froid. Cependant,les facteurs latents appris peuvent ne pas être efficaces en raison de lanature clairsemée de la matrice utilisateur-élément et des informations annexes.L'idée de base de la méthode de recommandation hybride basée sur l'apprentissage profond est de combiner la recommandation basée sur le contenuméthodes avec des méthodes de recommandation de filtrage collaboratif, intègrent l'apprentissage des fonctionnalités de l'utilisateur ou de l'élément et leprocessus de recommandation dans un cadre unifié.[45]

Nous proposons une hybridation du FC et FBC basé sur les réseaux de neurones.Les couches de notre modèle sont décrites comme suit :

Couche d'entrée (Input Layer). Nous considérons en entrée les identifiants des utilisateurs et des items, en plus des informations et caractéristiques recueillies relatives à chacun d'entre eux. Ces informations seront par la suite ajoutées aux facteurs appris par le modèle (i.e. seront concaténées à l'Embedding).

Couches d'inclusion (Embedding Layer). L'embedding est la représentation de variables discrètes en un vecteur de nombres continus. En termes de réseaux de neurones, il s'agit de vecteurs, de représentation de variables discrètes, appris. Nous prenons comme exemple le Word Embedding qui s'est avéré très efficace dans l'interprétation du langage naturel. Les mots sont préalablement codés en One-Hot Encoding.

Le vecteur sera donc très long et surtout très creux. Le but étant de concevoir un algorithme capable de former des vecteurs moins creux et qui posséderaient une relation logique entre eux.[46]

7.1 rnn dans les systèmes de recommandation :

Les RNN sont idéaux pour modéliser des données contenant des séquences.

Ils sont différents des réseaux de neurones traditionnels car RNN a des boucles et des mémoires qui enregistrent les calculs. Des variantes comme LSTM et GRU sont introduites comme solution au problème des gradients de fuite. Le concept sous-jacent des systèmes de recommandation utilisant les RNN est qu'ils peuvent modéliser l'effet de l'activité séquentielle passée des utilisateurs sur les actions récentes de l'utilisateur, après quoi proposer des choses et prévoir le comportement des utilisateurs . Les RNN peuvent modéliserdonnées séquentielles sans notion de délais entre événements successifs, en tenant compte du temps variable intervalles. Néanmoins, les décalages temporels peuvent être très utiles connaissances explicitement pour les tâches de recommandation. Par exemple, Si deux biens de consommation sont éloignés, le premier bien peut ne pas être un prédicteur équitable, car les préférences des clients ont évidemment évolué au fil du temps. Les RNN peuvent être

développés pour décrire l'utilisateur actions plus précisément et identifier les interconnexions entre actions de consommation que celles du recommandeur existant des systèmes incapables de refléter la dynamique temporelle dans intérêts des utilisateurs.[47]

7.2cnn dans les systèmes de recommandation :

Le réseau de neurones convolutifs est utilisé dans systèmes de recommandation en raison de ses performances dans la capture caractéristiques locales et globales issues du visuel et du textuel les sources de données. Ce modèle est utile pour résoudre les problème de classification et recommandation de balises en se basant sur l'extraction de caractéristiques visuelles à partir d'images correctifs ou en sélectionnant des mots informatifs informations textuelles

7.3mlp dans les systèmes de recommandation :

Notre architecture profonde est de type MLP. C'est l'architecture la plus adaptée aux problèmes de classification avec caractéristiques des données déjà définies en entrée. Notre MLP est composé de plusieurs couches avec un nombre variable de neurones et fonctionnellement regroupés et entièrement connectés. Cela consiste en une couche d'entrée, couches cachées et une couche de sortie . Cette architecture est obtenue après plusieurs variantes du paramètres réseau (nombre de couches cachées, nombre de neurones par couche, la fonction d'activation, etc.)[48].Cette partie se charge de l'apprentissage de la fonction d'interaction :

$$y_{ui} = \begin{cases} 1 & s'il\ y\ a\ une\ interaction\ entre\ l'utilisateur\ u\ et\ l'item\ i \\ 0 & sinon \end{cases}$$

La description textuelle des différentes couches du modèle HybMLP est donnée comme suit (voir figure 2.) :[46]

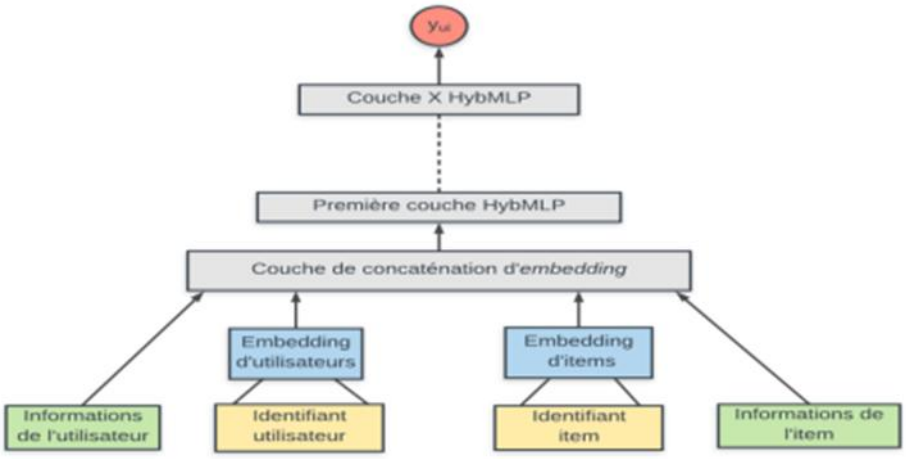


Figure2.10: Architecture du modèle HybMLP

— Embedding d'utilisateurs HybMLP : Vecteurs de facteurs latents des utilisateurs.

— Embedding d'items HybMLP : Vecteurs de facteurs latents des items.

— Couche de concaténation : dans cette couche s'effectue la concaténation des embeddings avec les caractéristiques non-apprises données en entrée. Cette couche constitue la première couche cachée du modèle HybMLP. Etant donné, que la concaténation ne suffit pas pour apprendre l'interaction entre les caractéristiques des utilisateurs et des items, ces données seront passées par d'autres couches cachées.

— Autre(s) couche(s) cachée(s) (HybMLPHidden Layer(s)) : Ces couches sont ajoutées pour permettre l'apprentissage des interactions comme expliqué précédemment. La fonction d'activation utilisée dans chaque couche est ReLU (RectifiedLinear Unit : $ReLU(x) = \max(0, x)$) car elle permet de réduire les risques de sur-apprentissage et la saturation des neurones.

Pour décider du nombre de nœuds optimal de chaque couche nous en fixons un pour la dernière que nous appellerons PF (Facteurs Prédictifs), la couche précédente aura le double de ce nombre de nœuds et ainsi de suite jusqu'à la première.[46]

8. Conclusion :

Dans ce chapitre nous avons introduit et explique l'apprentissage profond et expliqué le fonctionnement des algorithmes d'apprentissage automatique les plus populaires qui peuvent être utilisé pour faire de la Classification.

CHAPITRE 04 :
APPROCHE PROPOSÉ

1. Introduction :

Afin de prendre en compte les différents choix des utilisateurs, nous introduisons une nouvelle approche de recommandation hybride qui combine le filtrage collaboratif (CF) avec une approche basée sur le contenu (CBF). Cette approche hybride permet d'améliorer la pertinence et la précision des recommandations en exploitant à la fois les informations provenant des utilisateurs (filtrage collaboratif) et les caractéristiques des éléments recommandés (basé sur le contenu).

Dans ce chapitre, nous abordons deux aspects importants. Tout d'abord, nous présentons la création de la base de données de service web (BDDSW). Cette base de données constitue un élément essentiel de notre étude. Ensuite, nous proposons notre modèle d'architecture qui se décompose en quatre étapes : prétraitement, traitement, modèle de recommandation. Ces étapes sont construites en utilisant trois types de modèles d'apprentissage profond : RNN (Réseau de Neurones Récurrents), CNN (Réseaux de Neurones Convolutifs) et MLP (Perceptron Multi-Couches).

2. Les démarches de création de la base de données

2.1 Diagramme des classes

On a créé notre base de données par le biais d'un diagramme de classe proposée (figure 1.1).

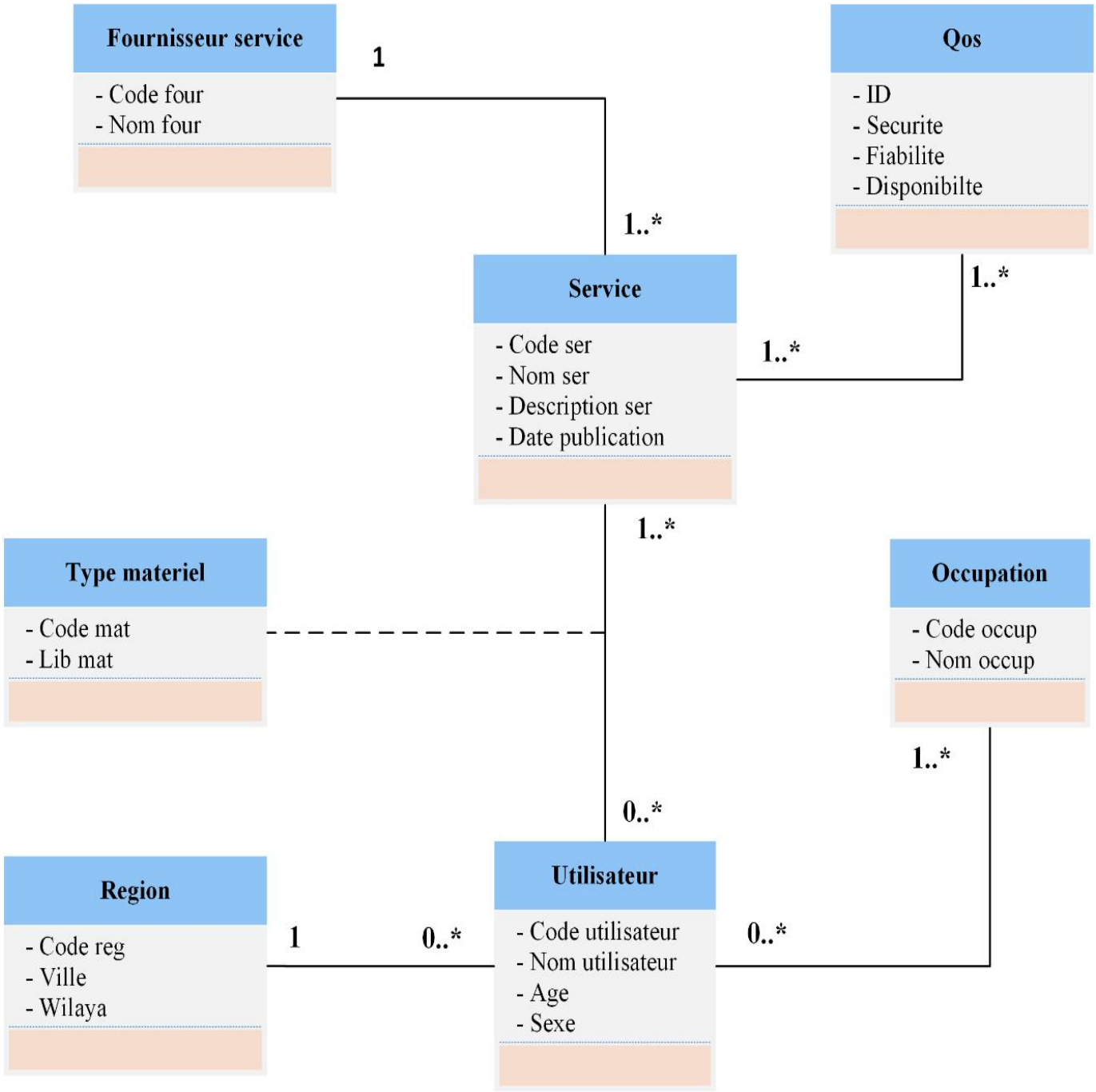


Figure 3.1 : Diagramme de classe proposée

2.2 Algorithme random dataset :

- Pour notre approche, nous avons créé une base de données simulée en utilisant une méthode de génération aléatoire.

Algorithme random dataset

Entrée : les tables sous forme de fichiers CSV
Sortie : BDDSW (Base de données simulée)
Début :
-Lire les fichiers CSV des tables et extraire les colonnes utilisées pour chaque table.
-Initialiser un fichier CSV vide pour stocker les données simulées.
-Définir le nombre d'observations souhaitées.
-Pour chaque observation :
a. Pour chaque colonne :
i. Générer une valeur aléatoire en utilisant une distribution de probabilité appropriée.
ii. Ajouter cette valeur au fichier CSV.
Retourner le fichier CSV généré.
FIN

Figure 3.2 : Algorithme random dataset

- Pour les fichiers d'entrée, nous allons créer une base de données à partir de diagrammes de classe à l'aide de phpMyAdmin. La base de données comprendra les fichiers suivants :

✓ Fichier "Utilisateur" : Ce fichier contiendra les informations sur les utilisateurs. Il comprendra les attributs suivants **Figure 3.3** :

- "codeu" : Identifiant de l'utilisateur.
- "tranche_age" : Tranche d'âge de l'utilisateur.
- "nomu" : Nom de l'utilisateur.
- "sexe" : Genre de l'utilisateur.

#	Nom	Type
1	codeu	int(10)
2	trancheage	int(5)
3	nomu	varchar(10)
4	sexe	char(3)

Figure 3.3: Table utilisateur

✓ Fichier "Service" : Ce fichier contiendra les informations sur les services web. Il comprendra les attributs suivants **Figure 3.4** :

- "codes" : Code du service.
- "noms" : Nom du service.
- "descrs" : Description du service.
- "date_pubs" : Date de publication du service.
- "langs" : Langue du service.

#	Nom	Type
1	codes	int(5)
2	noms	varchar(25)
3	descrs	text
4	date_pubs	date
5	langs	char(5)

- ✓ Fichier "Type Matériel" : Ce fichier contiendra les informations sur les types de matériel utilisés par les utilisateurs. Il comprendra les attributs suivants **Figure 3.5** :
  - "codetm" : Identifiant du matériel.
  - "libtm" : Type de matériel

#	Nom	Type
1	codetm	varchar(5)
2	libtm	text

Figure 3.5: Table type matériel

- ✓ Fichier "Région" : Ce fichier contiendra des informations sur les localisations géographiques. Il permettra de stocker des données sur les régions **Figure 3.6 .**

#	Nom	Type
1	coder	varchar(5)
2	ville	varchar(10)
3	wilaya	int(3)

Figure 3.6: Table région

- ✓ Fichier "QOS" : Ce fichier contiendra les informations sur quality of service. Il comprendra les attributs suivants **Figure 3.7** :
  - Codeq : identifient quality of service
  - Secq : security du service.
  - Fiaq : friability du service.
  - Disq : disponibilité du service.

#	Nom	Type
1	codeq	varchar(5)
2	secq	int(3)
3	fiaq	int(3)
4	disq	int(3)

Figure 3.7 : Table QOS

- Ces fichiers constitueront la base de données utilisée pour stocker les informations relatives aux utilisateurs, aux services web, aux types de matériel et aux régions.

3. L'architecture proposé :

Dans l'architecture de notre modèle, nous détaillons chaque partie en mettant en évidence son rôle spécifique. Tout d'abord, nous effectuons le prétraitement des données. Pour cela, nous utilisons la fonction Label encoder de la bibliothèque sklearn afin de convertir une variable catégorielle en variables factices/indicatrices. Cette étape est essentielle pour préparer les données en vue de la recommandation.

La deuxième étape de notre modèle combine à la fois le filtrage collaboratif (CF) et l'approche filtrage basée sur le contenu (CBF). Nous utilisons des couches dense et des couches d'embedding pour capturer les informations pertinentes dans les deux approches. Cela nous permet d'exploiter entre utilisateurs (CF) ainsi que les caractéristiques des éléments recommandés (CBF).

Ensuite, nous avons développé un modèle hybride pour la recommandation en utilisant des modèles de deeplearning tels que le RNN, le CNN et le MLP. Ces modèles sont adaptés pour traiter les données séquentielles et textuelles, ce qui nous permet d'obtenir des recommandations plus précises et personnalisées.

Enfin, nous effectuons la recommandation des N meilleurs services pour chaque utilisateur. Cette étape consiste à sélectionner les services les plus pertinents en fonction des préférences et des caractéristiques de chaque utilisateur, telles que capturées par notre modèle.

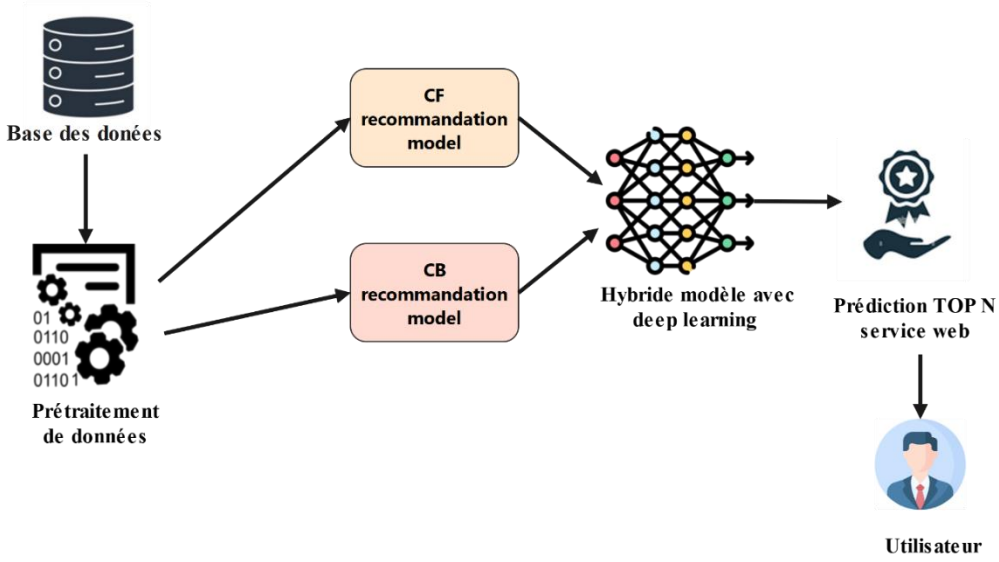


Figure3.8 : Architecture de l'approche.

3.1 Prétraitement:

Ces étapes visent à améliorer et préparer les données pour l'analyse en effectuant plusieurs actions. Tout d'abord, les attributs inutiles sont supprimés pour simplifier le jeu de données et se concentrer sur les attributs pertinents. Ensuite, les attributs catégoriels, tels que 'materiel' et 'region', sont convertis en données catégorielles en utilisant le type de données 'category'. Cette conversion permet de représenter ces attributs de manière plus appropriée pour les analyses ultérieures.

De plus, les colonnes 'secq' et 'fiaq' sont converties en labels binaires (0 ou 1) à l'aide de la fonction numpywhere(). Toutes les valeurs supérieures ou égales à 5 sont transformées en 1,

tandis que les valeurs inférieures à 5 sont transformées en 0. Cette conversion binaire permet de simplifier la prédiction de ces attributs dans les modèles d'apprentissage automatique.

En outre, des encodeurs de libellé (LabelEncoder) sont créés pour les caractéristiques catégorielles afin de les convertir en valeurs numériques. Ces encodeurs attribuent un identifiant unique à chaque catégorie présente dans les attributs catégoriels, ce qui facilite l'utilisation de ces données dans les modèles d'apprentissage automatique qui requièrent des valeurs numériques.

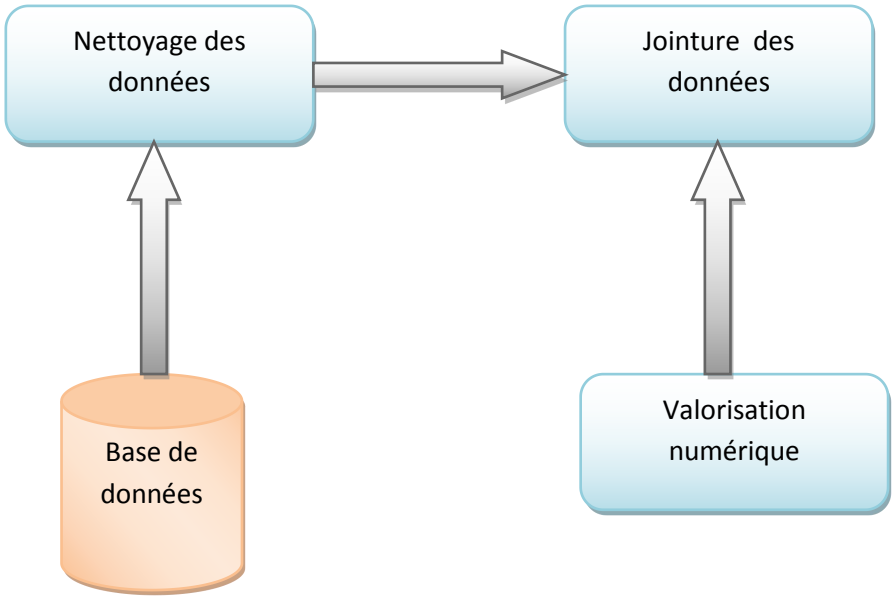


Figure3.9 : Architecture de prétraitement

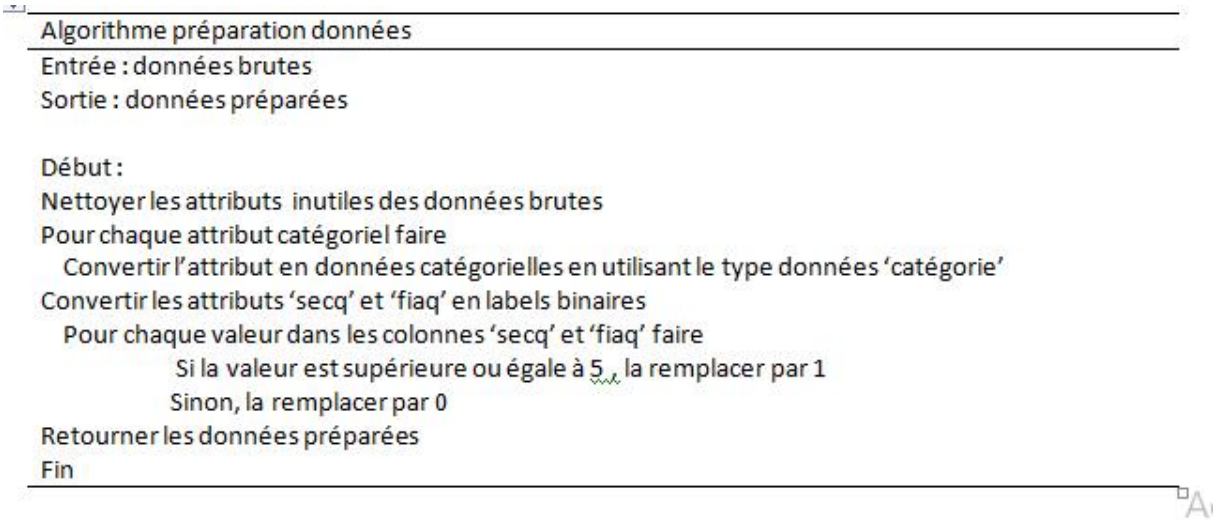


Figure3.10 : Algorithme préparation données

3.2 CF et CB model :

Dans le cadre de la construction des modèles CF et CBF, nous avons utilisé une approche de représentation des données basée sur les couches Embedding. Nous avons appliqué une couche Embedding pour les attributs "user" et "service". Cette couche Embedding permet de transformer les attributs catégoriques en vecteurs denses de valeurs réelles, ce qui facilite l'apprentissage par le modèle.

Après l'application des couches Embedding, nous avons utilisé la couche Flatten pour aplatir les sorties des couches Embedding en un seul vecteur continu. Cette étape est importante pour préparer les données avant de les alimenter dans les couches suivantes du modèle.

Pour le modèle CF, nous avons créé une couche Dense avec 16 nœuds, suivie d'une couche de dropout avec un taux de 0.2 pour réduire le surapprentissage. Enfin, nous avons ajouté une couche de sortie avec un seul nœud. La fonction d'activation ReLU a été utilisée dans la couche Dense pour introduire de la non-linéarité dans le modèle, tandis que la fonction d'activation sigmoid a été utilisée dans la couche de sortie pour produire une prédiction binaire.

Pour le modèle CBF, nous avons créé une couche Dense avec 16 nœuds, qui capture des caractéristiques complexes basées sur le contenu des attributs du service. Ensuite, nous avons ajouté une couche de sortie avec un seul nœud. Les mêmes fonctions d'activation que celles du modèle CF ont été utilisées dans les couches Dense et de sortie

```
Algorithme Création Model CF CBF
Entrée : user embedding, service embedding, materiel embedding, region embedding
Sortie : CF model, CB model

Début
    # Création du modèle Collaborative filtrage (CF)
    -Ajouter les couches  embedding
    -Ajouter la couche d'entrée pour CF
    -Ajouter la couche d'abandon (Dropout)
   -Implémenter les couches cachées pour CF
    - Ajouter la couche de sortie pour CF
   # Création du modèle filtrage  base sur le contenu  (CBF)
   - Ajouter les couches embedding
   -Ajouter la couche d'entrée pour CBF
   - Ajouter la couche d'abandon (Dropout)
   -Implémenter les couches cachées pour CBF
   -Ajouter la couche de sortie pour CBF
  Retourner CF model, CBF model

Fin
```

Figure 3.11 : Algorithme création model CF et CBF

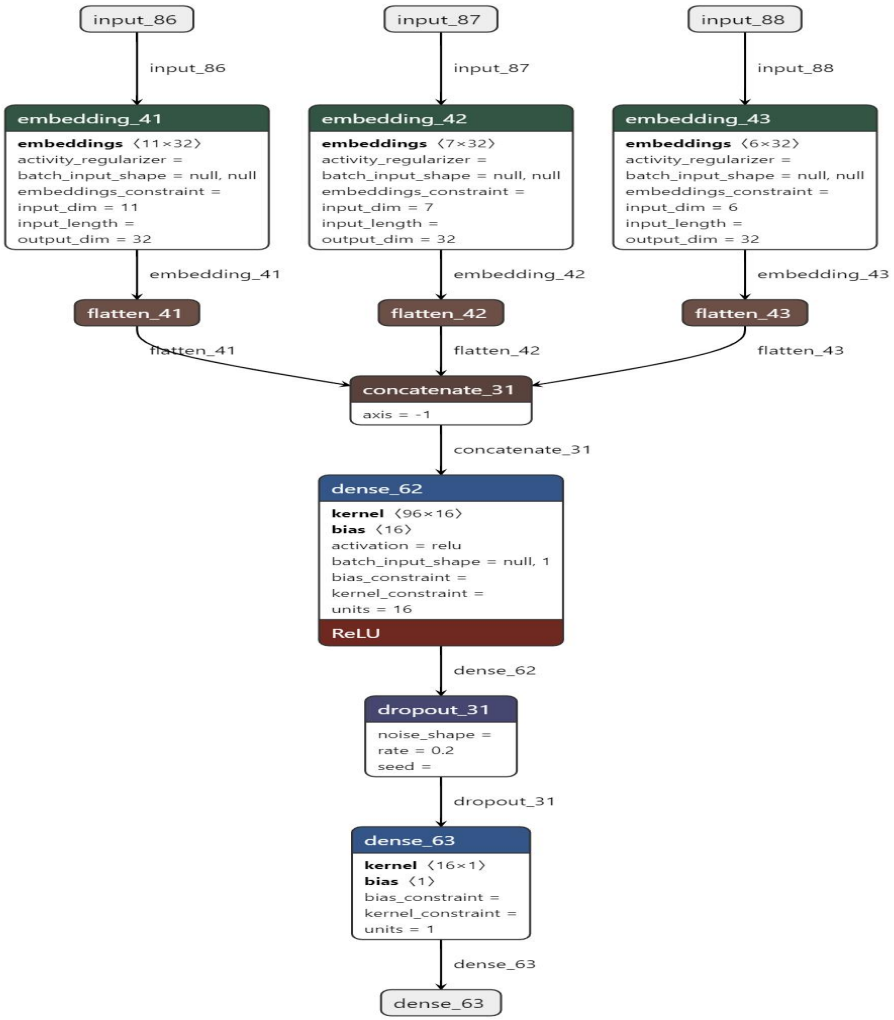


Figure3.12: Architecture CBF model

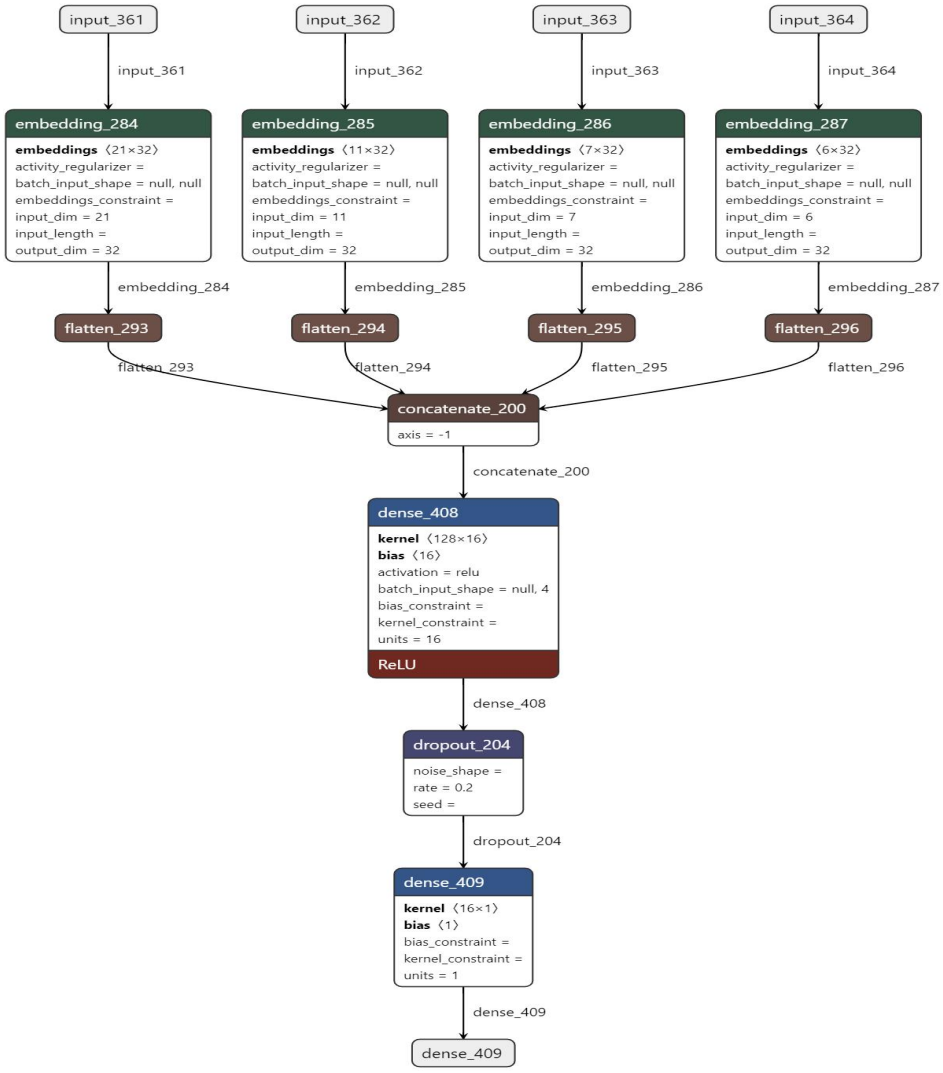


Figure3.13: architecture CF model

3.3 Hybridemodèle:

Pour hybride recommandation on a fait trois modèle pour comparées entre eux.chaque modèle nous suivons la valeur accuracy pour chaque epochs.

3.3.1 Hybride avec RNN:

Le modèle est basé sur une architecture de réseau de neurones récurrent(RNN) utilisant la couche SimpleRNN. Il se compose de deux couches SimpleRNN avec 16 nœuds chacune, une couche cachée avec 16 nœuds et une couche de sortie avec 2 nœuds.

La première couche SimpleRNN avec 16 nœuds effectue des calculs récurrents sur les données d'entrée.

La deuxième couche SimpleRNN avec 16 nœuds est similaire à la première, mais elle agit sur les sorties de la première couche SimpleRNN. La couche cachée avec 16 nœuds sert de pont entre les couches SimpleRNN et la couche de sortie.

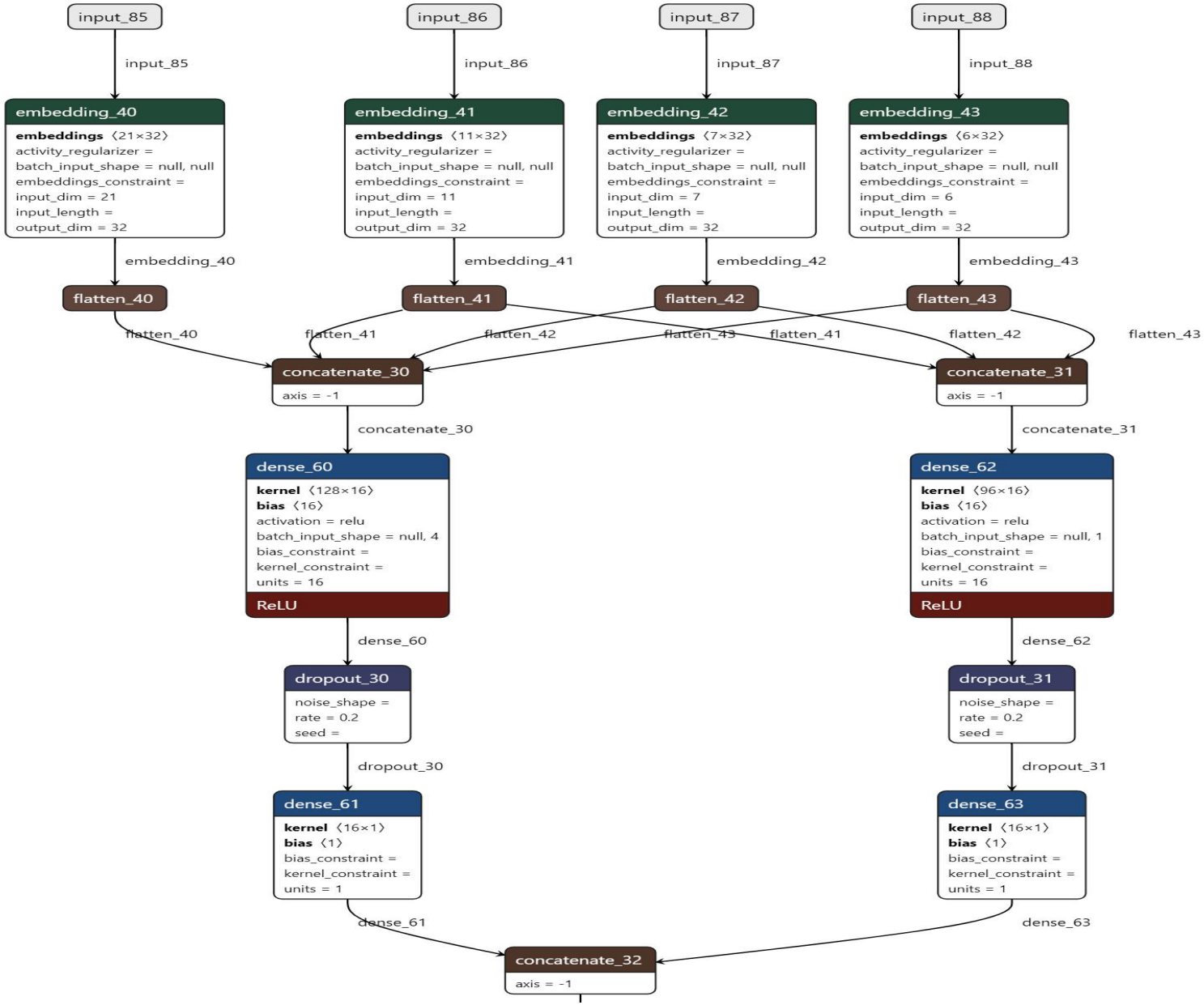
La couche de sortie avec 2 nœuds est la dernière couche du modèle. Elle applique une fonction d'activation sigmoid pour produire les prédictions finales.

La fonction de perte utilisée est la loss binaire croisée (binary cross-entropy), qui est couramment utilisée pour les problèmes de classification binaire. L'optimiseur utilisé est Adam, un algorithme d'optimisation populaire pour l'apprentissage en profondeur. La métrique d'évaluation utilisée est l'accuracy, qui mesure la précision des prédictions du modèle.

Le modèle est entraîné en utilisant la fonction `fit()` avec un batch_size de 128 et un nombre total d'époch de 100.

```
Algorithme création hybride RNN model
Entrée : concaténation sortie CF et CB model
Sortie : courbe d'accuracy.
Début :
      -implémenter les deux couche Simple RNN ;
      -ajouter la couche cachée ;
      -ajouter la couche de sortie ;
      -implémenter le modèle ;
      -compiler modèle avec optimisation Adam, Loss  fonction
binary_ crossentropy
      -fit modèle avec X train et Y train
      -plot modèle
FIN
```

Figure3.14 : Algorithme création hybride RNN


input_85
input_86
input_87
input_88
embedding_40
embeddings ⟨21×32⟩
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 21
input_length =
output_dim = 32
embedding_41
embeddings ⟨11×32⟩
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 11
input_length =
output_dim = 32
embedding_42
embeddings ⟨7×32⟩
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 7
input_length =
output_dim = 32
embedding_43
embeddings ⟨6×32⟩
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 6
input_length =
output_dim = 32
flatten_40
flatten_41
flatten_42
flatten_43
concatenate_30
axis = -1
concatenate_31
axis = -1
dense_60
kernel ⟨128×16⟩
bias ⟨16⟩
activation = relu
batch_input_shape = null, 4
bias_constraint =
kernel_constraint =
units = 16
ReLU
dense_62
kernel ⟨96×16⟩
bias ⟨16⟩
activation = relu
batch_input_shape = null, 1
bias_constraint =
kernel_constraint =
units = 16
ReLU
dropout_30
noise_shape =
rate = 0.2
seed =
dropout_31
noise_shape =
rate = 0.2
seed =
dense_61
kernel ⟨16×1⟩
bias ⟨1⟩
bias_constraint =
kernel_constraint =
units = 1
dense_63
kernel ⟨16×1⟩
bias ⟨1⟩
bias_constraint =
kernel_constraint =
units = 1
concatenate_32
axis = -1

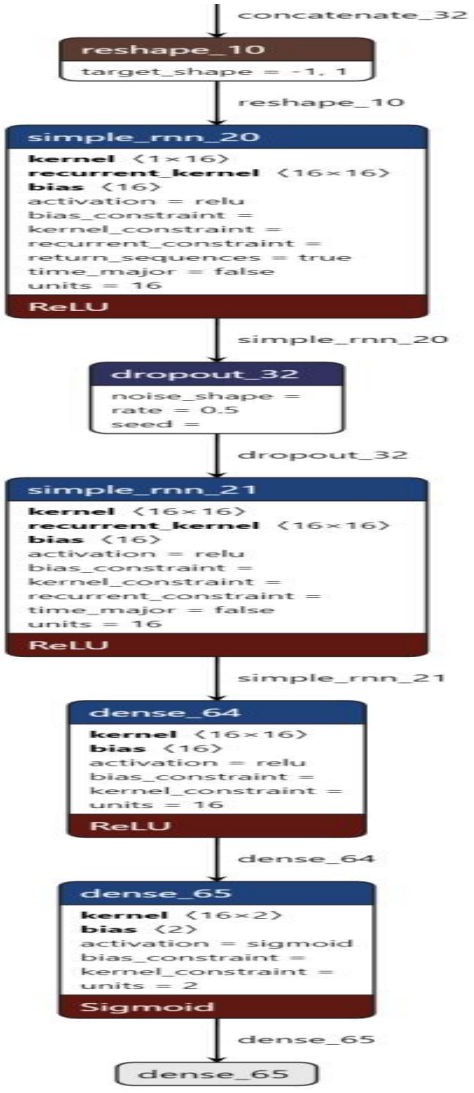


Figure3.15: Architecture SimpleRNN model

3.3.2 hybride avec CNN:

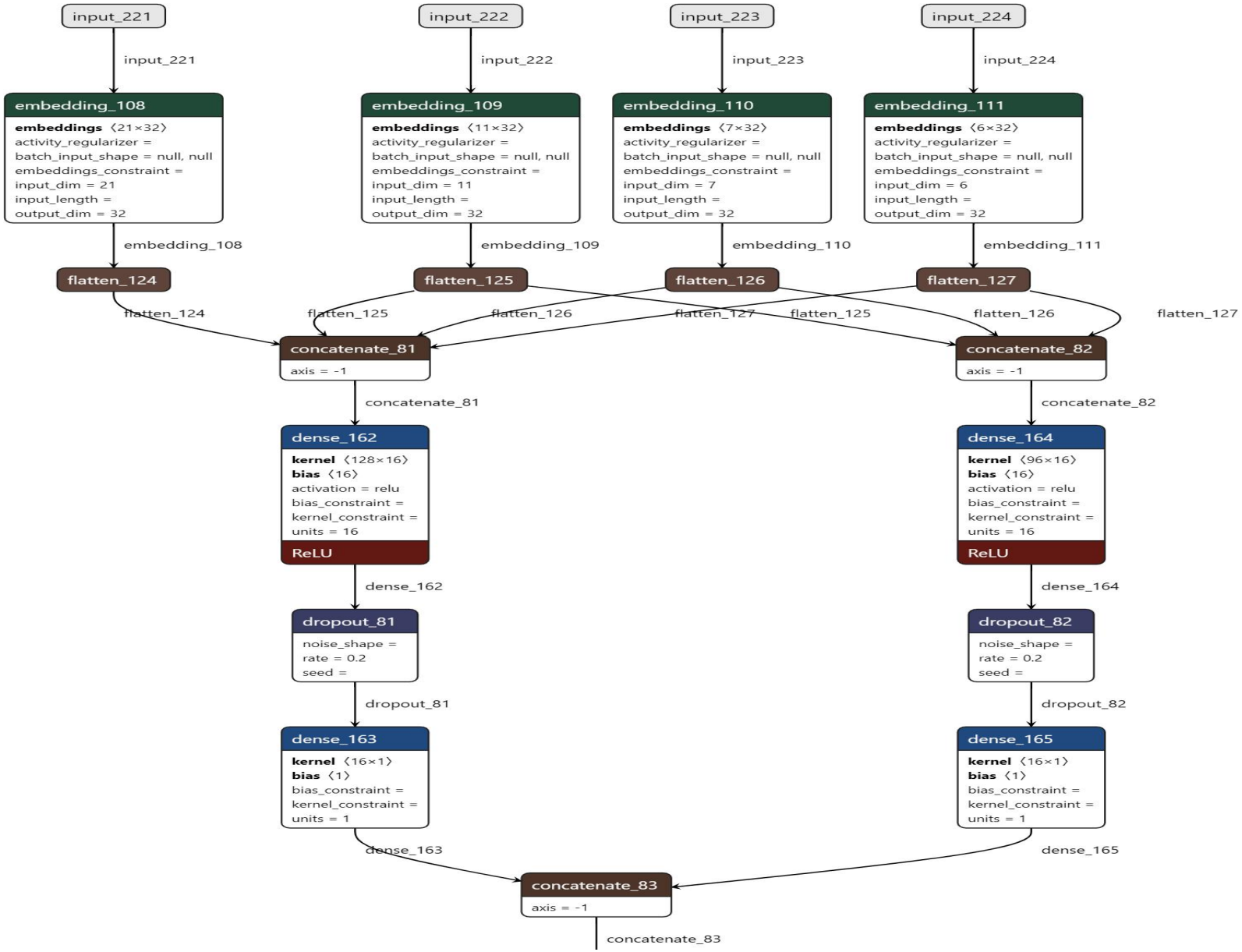
Le modèle est basé sur une architecture de réseau de neurones convolutifs (Conv1D) pour effectuer des prédictions. Il se compose de deux couches Conv1D avec 16 nœuds chacune, une couche cachée avec 16 nœuds et une couche de sortie avec 2 nœuds.

La première couche Conv1D avec 16 nœuds applique une opération de convolution sur les données d'entrée, suivie d'une fonction d'activation Relu.

La deuxième couche Conv1D avec 16 nœuds est similaire à la première, mais elle agit sur les caractéristiques extraites par la première couche Conv1D. Cela permet au modèle de capturer des motifs plus complexes et de les combiner pour des prédictions plus précises.

```
Algorithme création hybride CNN model
Entrée : concaténation sortie CF et CBF model
Sortie : courbe d'accuracy.
Début :
    -implémenter les deux couche conv1D ;
    -ajouter la couche cachée ;
    - ajouter la couche de sortie ;
    -implémenter le modèle ;
    -compiler modèle avec optimizer adam, loss fonction
'binary_crossentropy' ;
     -Fit modèle avec X train et Y train
     -plot modèle
FIN
```

Figure3.16 :Algorithme hybride CNN model


input_221
input_222
input_223
input_224
embedding_108
embeddings 〈21×32〉
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 21
input_length =
output_dim = 32
embedding_109
embeddings 〈11×32〉
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 11
input_length =
output_dim = 32
embedding_110
embeddings 〈7×32〉
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 7
input_length =
output_dim = 32
embedding_111
embeddings 〈6×32〉
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 6
input_length =
output_dim = 32
flatten_124
flatten_125
flatten_126
flatten_127
concatenate_81
axis = -1
concatenate_82
axis = -1
dense_162
kernel 〈128×16〉
bias 〈16〉
activation = relu
bias_constraint =
kernel_constraint =
units = 16
ReLU
dense_164
kernel 〈96×16〉
bias 〈16〉
activation = relu
bias_constraint =
kernel_constraint =
units = 16
ReLU
dropout_81
noise_shape =
rate = 0.2
seed =
dropout_82
noise_shape =
rate = 0.2
seed =
dense_163
kernel 〈16×1〉
bias 〈1〉
bias_constraint =
kernel_constraint =
units = 1
dense_165
kernel 〈16×1〉
bias 〈1〉
bias_constraint =
kernel_constraint =
units = 1
concatenate_83
axis = -1

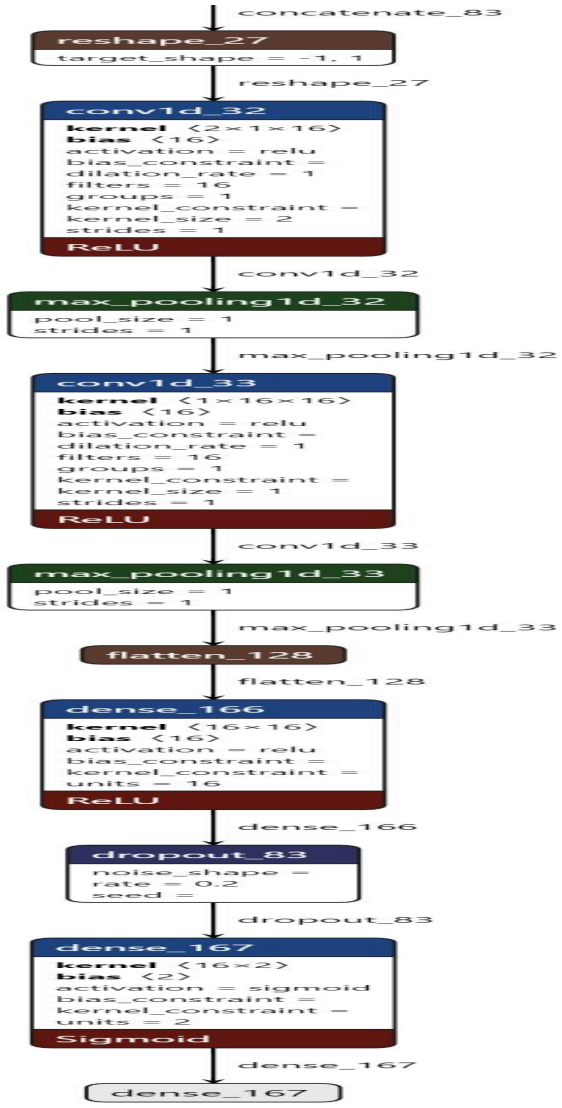


Figure3.17 : Architecture CNN model

3.3.3 Hybride avec MLP :

Dans le modèle hybride MLP (Multi-Layer Perceptron), nous avons développé une architecture en plusieurs étapes. Tout d'abord, une couche d'entrée avec 32 nœuds a été ajoutée pour recevoir les données d'entrée et les transmettre à la couche suivante.

Afin d'améliorer la généralisation du modèle et prévenir le surapprentissage, nous avons inclus une couche Dropout avec un taux de 0.2. Pendant l'entraînement, cette couche désactive aléatoirement certains nœuds, ce qui réduit la dépendance excessive entre eux et favorise une meilleure généralisation.

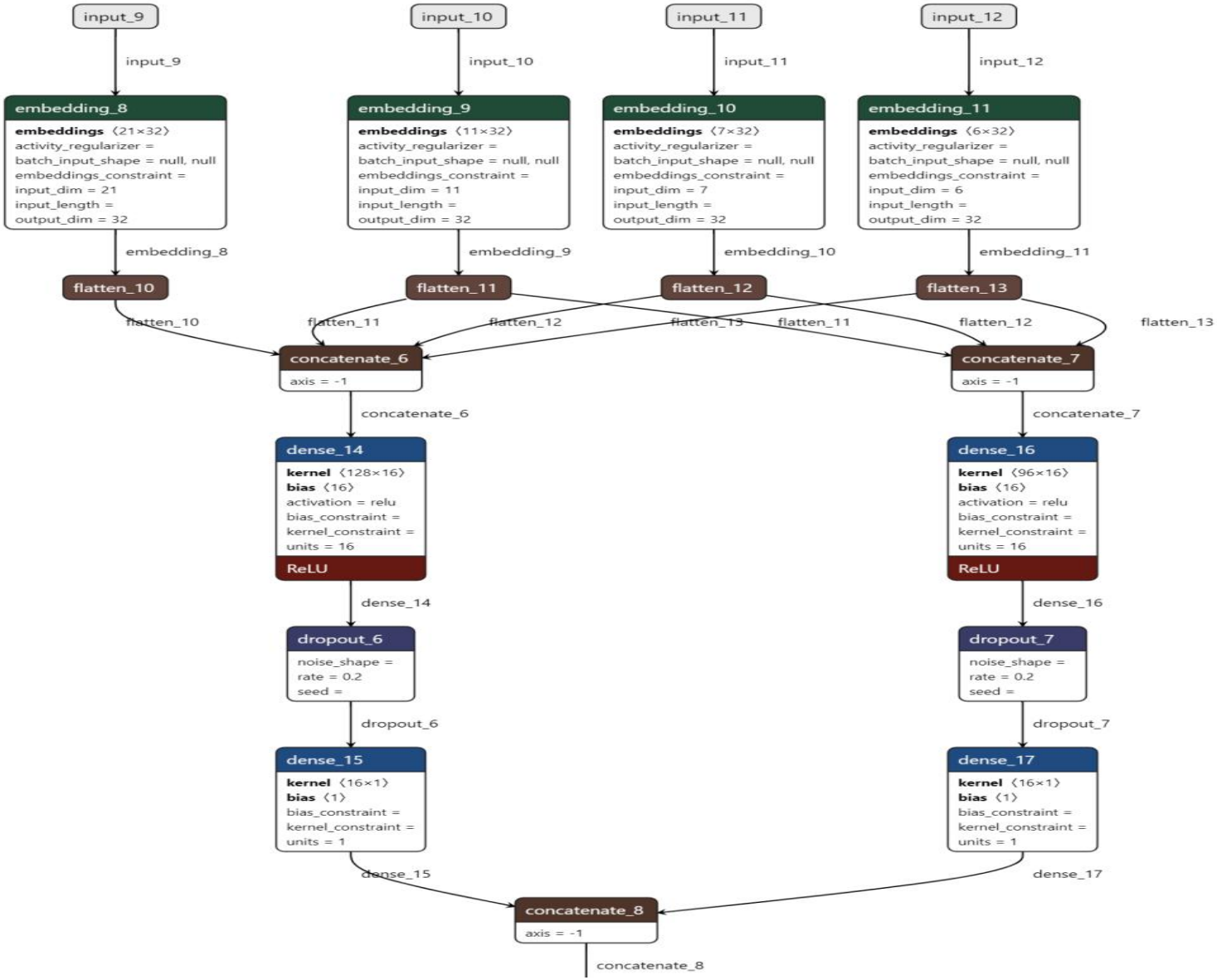
Ensuite, nous avons intégré une couche cachée comprenant 32 nœuds. Cette couche effectue des calculs et des transformations non linéaires sur les données provenant de la couche précédente. Les nœuds de cette couche contribuent à la représentation intermédiaire des caractéristiques des données.

Enfin, une couche de sortie avec 2 nœuds a été ajoutée. Cette couche applique une fonction d'activation appropriée, telle que la fonction sigmoid pour un problème de classification binaire, afin de générer les prédictions finales du modèle.

```
Algorithme hybride MLP model
Entrée : concaténation sortie CF et CBF model
Sortie : courbe d'accuracy.
Début :
    -ajouter la couche d'entrée ;
    -implémenter la couche cachée ;
    -ajouter la couche de sortie ;
    -implémenter le modèle
    -compiler modèle avec optimizer adam, loss fonction
'binary_crossentropy' ;
    -fit modèle avec  X train et  Y train
    -plot modèle

FIN
```

Figure3.18: Algorithmehybride MLP model


input_9
input_10
input_11
input_12
embedding_8
embeddings ⟨21×32⟩
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 21
input_length =
output_dim = 32
embedding_9
embeddings ⟨11×32⟩
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 11
input_length =
output_dim = 32
embedding_10
embeddings ⟨7×32⟩
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 7
input_length =
output_dim = 32
embedding_11
embeddings ⟨6×32⟩
activity_regularizer =
batch_input_shape = null, null
embeddings_constraint =
input_dim = 6
input_length =
output_dim = 32
flatten_10
flatten_11
flatten_12
flatten_13
concatenate_6
axis = -1
concatenate_7
axis = -1
dense_14
kernel ⟨128×16⟩
bias ⟨16⟩
activation = relu
bias_constraint =
kernel_constraint =
units = 16
ReLU
dense_16
kernel ⟨96×16⟩
bias ⟨16⟩
activation = relu
bias_constraint =
kernel_constraint =
units = 16
ReLU
dropout_6
noise_shape =
rate = 0.2
seed =
dropout_7
noise_shape =
rate = 0.2
seed =
dense_15
kernel ⟨16×1⟩
bias ⟨1⟩
bias_constraint =
kernel_constraint =
units = 1
dense_17
kernel ⟨16×1⟩
bias ⟨1⟩
bias_constraint =
kernel_constraint =
units = 1
concatenate_8
axis = -1

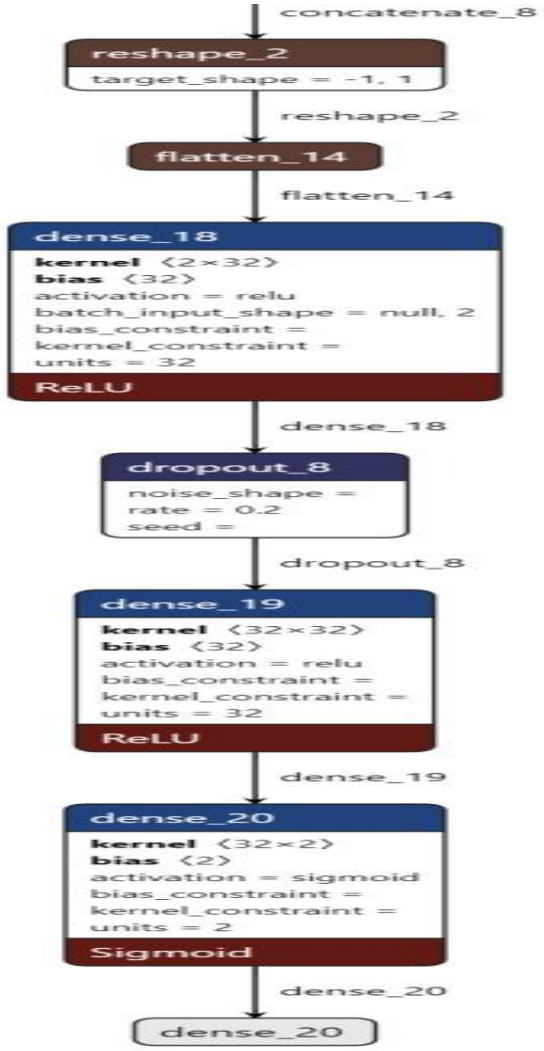


Figure3.19 : Architecture MLP model

4. Algorithme get top N service :

Algorithme get top N service :

Début :
Entrée :user_id, materiel, region ,model
Sortie :top 10 service name .

1. Créer une liste de tous les identifiants de service possibles à partir des données existantes:
   - Utiliser np.arange pour générer une séquence d'entiers de 1 jusqu'au nombre total d'identifiants de service unique dans les données.
2. Initialiser les variables pour les entrées utilisateur, les entrées de service, les entrées de matériel et les entrées de région:
   - num_combinations représente le nombre de combinaisons pour les recommandations (10 dans ce cas).
   - user_input est un tableau de dimensions (num_combinations, 1) contenant l'identifiant de l'utilisateur répété num_combinations fois.
   - service_input est un sous-ensemble des identifiants de service, représentant les num_combinations premiers identifiants de service. Il est remodelé en un tableau de dimensions (num_combinations, 1).
   - materiel_encoded est un tableau de dimensions (num_combinations, 1) contenant l'encodage de la variable materiel à l'aide d'un LabelEncoder. La valeur est répétée num_combinations fois.
   - region_encoded est un tableau de dimensions (num_combinations, 1) contenant l'encodage de la variable region à l'aide d'un LabelEncoder. La valeur est répétée num_combinations fois.
3. Imprimer les longueurs des tableaux user_input, service_input, materiel_encoded et region_encoded (optionnel).
4. Obtenir les prédictions du modèle pour les 10 meilleures combinaisons:
   - Utiliser la méthode model.predict en passant les tableaux user_input, service_input, materiel_encoded et region_encoded en tant qu'entrées.
5. Créer une liste de tuples contenant les noms de service, les scores de fiabilité et les scores de sécurité prédits pour chaque combinaison:
   - Parcourir les identifiants de service et les prédictions correspondantes.
   - Pour chaque identifiant de service, trouver le nom de service correspondant dans les données.
   - Créer un tuple contenant le nom de service, le score de fiabilité et le score de sécurité prédits.
   - Ajouter ce tuple à la liste prediction_scores.
6. Trier les prédictions en fonction du score de fiabilité dans l'ordre décroissant:
   - Utiliser la fonction sorted en passant prediction_scores comme argument et spécifier key=lambda x: x[1] pour trier en fonction du deuxième élément du tuple (score de fiabilité).
7. Sélectionner les 10 meilleures recommandations (services avec les scores de fiabilité les plus élevés):

Figure3.20:Algorithme get topN service

5. Conclusion:

Dans ce chapitre, nous avons présenté les différentes étapes de construction d'une architecture hybride de recommandation. Nous avons élaboré des modèles de filtrage collaboratif (CF) et de filtrage basé sur le contenu (CBF), ainsi qu'un modèle hybride qui combine différentes approches de deeplearning. Pour commencer, nous avons développé le modèle CF et CBF en utilisant des techniques avancées telles que les embeddings et les couches de neurones. Ce modèle intègre des informations sur le contenu des éléments afin de fournir des recommandations personnalisées aux utilisateurs.

Enfin, nous avons créé un modèle hybride en combinant les prédictions des modèles CF et CBF. Cette approche nous permet de capitaliser sur les forces de chaque approche et d'améliorer la qualité globale des recommandations proposées.

Dans le prochain chapitre, nous allons discuter des résultats de prédiction obtenus en utilisant cette architecture hybride. Nous évaluerons les performances du modèle en utilisant des métriques telles que la précision, afin de mesurer l'efficacité de notre système de recommandation..

CHAPITRE04 :
EXPÉRIMENTATIONET EVALUATION

1. Introduction:

Dans nos expérimentations, nous avons recueilli 6000 utilisations de différents services effectuées par plusieurs utilisateurs dans la base de données BDDSW. L'objectif de cette collecte était d'évaluer l'approche de SR (Système de Recommandation) hybride. Ce chapitre est structuré de la manière suivante : tout d'abord, nous présentons le fichier CSV de BDDSW, qui contient les données recueillies. Ensuite, nous décrivons les outils et langages que nous avons utilisés pour mettre en œuvre les modèles de recommandation. Enfin, nous discutons des résultats obtenus à partir de nos expérimentations. Au travers de cette structure, nous cherchons à fournir une vue d'ensemble complète de notre approche et de ses performances.

2. Base de donnée :

Afin de pouvoir mettre en pratique notre démarche, il est essentiel de disposer d'un jeu de données adéquat. À cette fin, nous avons créé une base de données appelée BDDSW. Cette base de données contient les informations relatives à 6000 utilisations de services. Le format utilisé pour stocker ces données est un fichier CSV. Chaque entrée de ce fichier est composée des champs suivants : user_id (identifiant de l'utilisateur), service_id (identifiant du service), service_name (nom du service), secq (sécurité du service), fiaq (fiabilité du service), matériel (matériel utilisé) et région (localisation géographique).(**Tableau4.1**)

Champ	descriptions
User_id	L'identificationd'utilisateur
Service_id	L'identificationdeservice
Service_name	Nomdeservice
Secq	Securitydeservice(valeur1a10)
Fiaq	Fiabilitédeservice(valeur1à 10)
Matériel	Typematérielquiutilisateurutiliser
Région	Localisationd'utilisateur

Tableau 4.1:Description des champsBDDSW

3. Langage de programmation :

Lors de la réalisation de notre système, nous avons choisi d'utiliser le langage de programmation Python en raison de ses nombreux avantages. Python [49]est un langage de haut niveau qui prend en charge la programmation orientée objet, impérative séquentielle et fonctionnelle. Sa syntaxe simplifiée et concise permet aux programmeurs de gagner du temps et de développer plus rapidement.

Python offre une vaste gamme de modules et de bibliothèques qui facilitent le développement de projets complexes. Parmi les bibliothèques utilisées pour notre système, nous avons inclus numpy[50], pandas[51], sklearn[52], tensorflow[53]et keras[54].

4. Métriquesd'évaluation:

Pour évaluer notre système de recommandation, nous avons utilisé trois modèles différents: un modèle utilisant SimpleRNN, un autre utilisant CONV1D(conventionnel neural network avec 1 dimension) et un troisième modèle utilisant MLP (Multi-Layer Perceptron). L'objectif était de réaliser une hybridation entre le filtrage collaboratif (CF) et le filtrage basé

sur le contenu (CBF) pour chaque modèle, puis de comparer leurs performances.

Parmi les différents protocoles d'évaluation disponibles, nous avons choisi d'utiliser le protocole Hold-out. Ce protocole consiste à diviser le jeu de données en deux parties distinctes : l'ensemble d'entraînement (training set) et l'ensemble de test (test set). Le pourcentage de division est fixé à 80%, ce qui signifie que 80% des données sont utilisées pour l'entraînement et les 20% restants sont réservés au test.

L'objectif de l'ensemble d'entraînement est de permettre au modèle d'apprendre à partir de ces données. Il s'agit de fournir à l'algorithme des exemples d'entrée (attributs ou features en anglais) et les réponses correspondantes (label ou output). Cette relation entre l'entrée et la sortie est essentiellement une relation de question/réponse. Pendant la phase d'entraînement, le modèle analyse les données, ajuste les poids et les valeurs ajoutées, et cherche à établir une corrélation entre les entrées et les sorties.

Dans notre approche, nous utilisons précision (accuracy), et test_accuracy qui nous permet d'évaluer la précision de notre système de recommandation hybride et d'évaluer son efficacité lors de la prédiction de nouvelles données. Ces mesures nous aident à prendre des décisions éclairées sur les améliorations potentielles et à garantir que notre système fournit des recommandations pertinentes et précises aux utilisateurs.

Ces métriques d'évaluation sont définies dans les équations (1)-(2), où chaque mesure nécessite d'estimer le nombre de vrais positifs (TP), de faux positifs (FP), de vrais négatifs (TN) et de faux négatifs (FN).[55]

$$Precision = \frac{TP}{TP + FP} \qquad (1)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad (2)$$

La formule de la fonction loss Binary Cross-entropy (entropie croisée binaire) est la suivante :

$$Loss = -\left(y_{true} * \log\left(y_{pred}\right) + (1 - \mathrm{y_{true}}) * \log\left(1 - \mathrm{y_{pred}}\right)\right)$$

- y_true représente la valeur vraie de l'étiquette de la classe (0 ou 1) pour l'exemple donné.
- y_pred représente la valeur prédite par le modèle pour cet exemple.

4. Résultats et diagnostic:

4.1 modèle SimpleRNN :

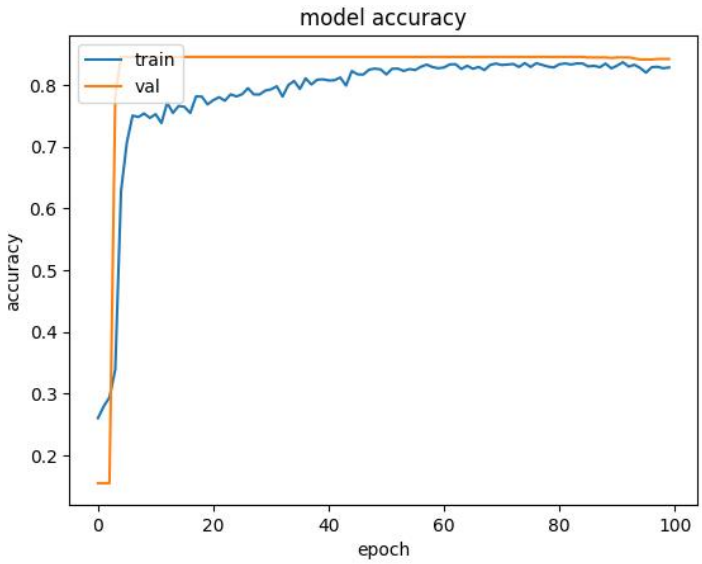


Figure4.1 : Graph accuracy simplernn

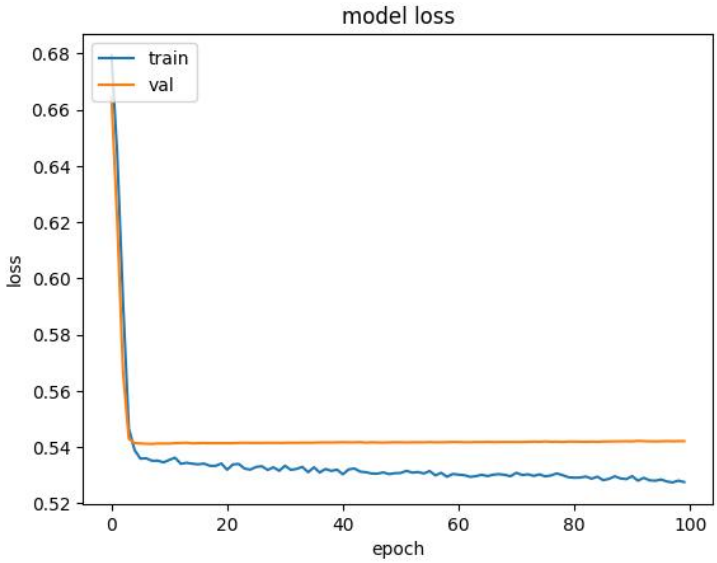


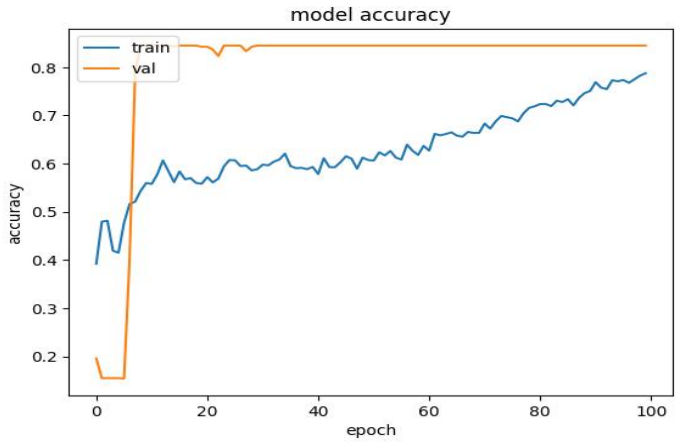
Figure4.2 : Graph loss simplernn

Diagnostic:

L'exactitude initiale du modèle est très faible, environ 26,02%, et reste constamment basse au cours des premières époques. Cependant, à partir de l'époque 5, l'exactitude du modèle sur l'ensemble de validation commence à augmenter considérablement, atteignant environ 84,5% et se maintient à ce niveau jusqu'à l'époque 86, avec des valeurs oscillant entre 84,42% et 84,16%. Après la quatrième époque, l'exactitude s'améliore de manière significative, atteignant environ 62,77% et continuant à augmenter jusqu'à atteindre une valeur de 83,74%.

D'autre part, la perte (loss) diminue progressivement au fur et à mesure des époques, indiquant que le modèle s'améliore et s'ajuste aux données d'entraînement.

4.2 modèle CONV1D:



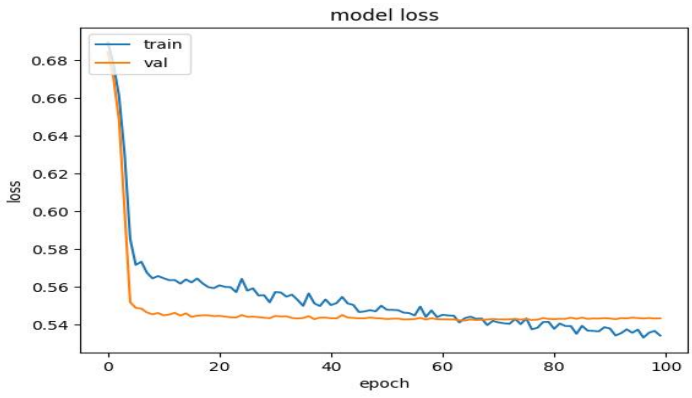
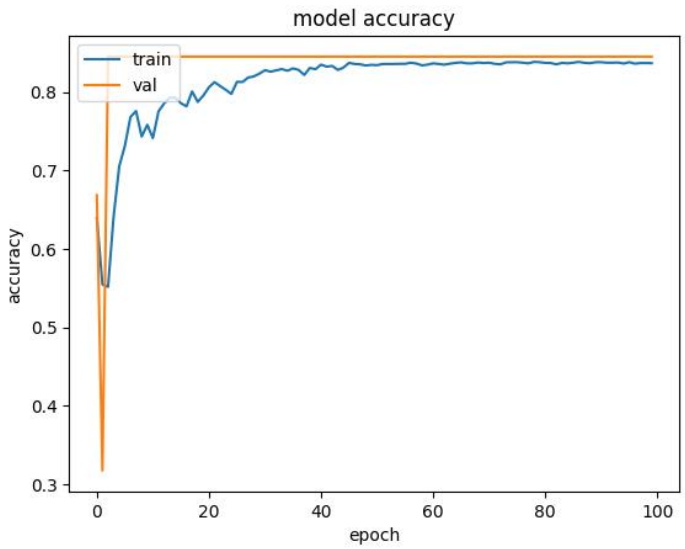


Figure4.3: Graph accuracyCNN1D **Figure4.4:**Graph lossCNN1D

Diagnostic:

D'après l'entraînement, la précision finale obtenue sur les données d'entraînement est d'environ 83,65%, et la précision finale obtenue sur les données de validation est d'environ 84,50 %. Les valeurs de perte diminuent également au fil des époques, ce qui indique que le modèle a appris à faire de meilleures prédictions. Ces résultats suggèrent que le modèle a réussi à capturer une partie des motifs présents dans les données et à généraliser ses prédictions sur de nouvelles données.

4.3 modèle MLP:



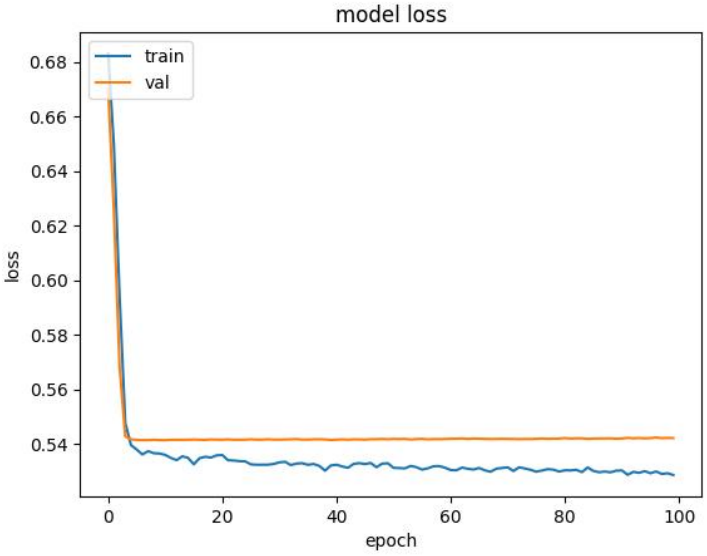


Figure4.5 : Graph accuracy MLP **Figure4.6:**Graph Loss MLP

Diagnostic:

D'après l'entraînement, la précision finale obtenue sur les données d'entraînement est d'environ 83,67%, et la précision finale obtenue sur les données de validation est d'environ 84,50 %. Les valeurs de perte diminuent jusqu'à du valeur 52,87%.

4.4 Evaluation des modèles utilisés:

L'évaluation des métriques utilise principalement l'exactitude (accuracy). Pour effectuer cette évaluation, nous utilisons la fonction "evaluate" en utilisant les données de test pour calculer l'exactitude du modèle sur l'ensemble de test et les données d'entraînement pour

calculer l'exactitude du modèle sur l'ensemble d'entraînement. Voici le tableau présentant l'évaluation de l'exactitude pour chaque modèle :

Modèle / Métrique	SimpleRNN%	CNN%	MLP%
Testaccuracy	84,17	**84.50**	**84.50**
Trainaccuracy	83,74	83,65	83.67

Tableau 4.2: Résultats de précision (accuracy)

Diagnostic:

Les trois modèles (SimpleRNN, Conv1D, MLP) ont des performances similaires en termes d'accuracy sur l'ensemble de test, avec environ 84,50%. Les différences entre les modèles sont minimes. Les modèles présentent également des performances similaires sur l'ensemble d'entraînement, avec des précisions entre 83,65% et 83,74%. Une évaluation plus détaillée est nécessaire pour une comparaison approfondie.

5. Get_top_recommandations :

Pour les trois modèles de deeplearning, nous avons utilisé la fonction predict pour obtenir les scores de fiabilité et de sécurité des recommandations. Cela nous permet d'obtenir les prédictions pour les combinaisons d'entrée fournies par l'utilisateur. voici le résultats d'un exemple user_id='user3' et materiel= 'hp' et region='ouzra'(figure 4.7)(figure 4.6)(figure 4.8)

```
recommendation avec SimpleRNN
Service Name: ser08
Score de fiabilité : 0.82436895 Score de sécurité : 0.81563205
Service Name: ser01
Score de fiabilité : 0.7869941 Score de sécurité : 0.7776204
Service Name: ser07
Score de fiabilité : 0.78163314 Score de sécurité : 0.7713629
Service Name: ser09
Score de fiabilité : 0.77391183 Score de sécurité : 0.76069784
Service Name: ser10
Score de fiabilité : 0.7643406 Score de sécurité : 0.7493286
Service Name: ser06
Score de fiabilité : 0.75135493 Score de sécurité : 0.7340685
Service Name: ser02
Score de fiabilité : 0.73861045 Score de sécurité : 0.7138131
Service Name: ser05
Score de fiabilité : 0.73230726 Score de sécurité : 0.69852984
Service Name: ser04
Score de fiabilité : 0.7164513 Score de sécurité : 0.6864506
Service Name: ser03
Score de fiabilité : 0.70920414 Score de sécurité : 0.6713114
```

figure 4.7 : Recommandations Simple RNN

```
recommendation avec CONV1D
Service Name: ser04
Score de fiabilité : 0.78791976 Score de sécurité : 0.76137704
Service Name: ser10
Score de fiabilité : 0.7867338 Score de sécurité : 0.75850946
Service Name: ser02
Score de fiabilité : 0.769065 Score de sécurité : 0.74327004
Service Name: ser07
Score de fiabilité : 0.74482524 Score de sécurité : 0.7227379
Service Name: ser03
Score de fiabilité : 0.7427748 Score de sécurité : 0.71955216
Service Name: ser01
Score de fiabilité : 0.74121845 Score de sécurité : 0.71852857
Service Name: ser08
Score de fiabilité : 0.7362965 Score de sécurité : 0.7148641
Service Name: ser06
Score de fiabilité : 0.73531127 Score de sécurité : 0.71489924
Service Name: ser09
Score de fiabilité : 0.7312495 Score de sécurité : 0.7107588
Service Name: ser05
Score de fiabilité : 0.7293156 Score de sécurité : 0.70920986
```

figure 4.8: Recommandation CONV1D

```
recommendation avec MLP
Service Name: ser02
Score de fiabilité : 0.805386 Score de sécurité : 0.7816705
Service Name: ser08
Score de fiabilité : 0.80404806 Score de sécurité : 0.7540266
Service Name: ser10
Score de fiabilité : 0.7983709 Score de sécurité : 0.76680845
Service Name: ser09
Score de fiabilité : 0.792068 Score de sécurité : 0.73494744
Service Name: ser01
Score de fiabilité : 0.79042804 Score de sécurité : 0.7509375
Service Name: ser07
Score de fiabilité : 0.78420156 Score de sécurité : 0.74316007
Service Name: ser03
Score de fiabilité : 0.78338385 Score de sécurité : 0.7372164
Service Name: ser04
Score de fiabilité : 0.7794078 Score de sécurité : 0.76718736
Service Name: ser06
Score de fiabilité : 0.76244795 Score de sécurité : 0.7097619
Service Name: ser05
Score de fiabilité : 0.7449752 Score de sécurité : 0.7348741
```

Figure 4.9: Recommandations MLP

Diagnostic:

- Il est intéressant de noter que les services recommandés diffèrent légèrement d'un modèle à l'autre. Cela peut être dû aux différentes architectures et techniques d'apprentissage utilisées par chaque modèle.
- Les services "ser02", "ser08" et "ser10" sont recommandés avec des scores de fiabilité élevés par deux des trois modèles (MLP et CONV1D). Cela suggère que ces services pourraient être des choix solides pour les utilisateurs.
- Les services "ser04" et "ser07" sont également recommandés par deux des trois modèles, bien que leur ordre de priorité varie légèrement.
- Il est important de prendre en compte les scores de sécurité ainsi que les scores de fiabilité lors de la prise de décision sur les services recommandés. Certains services peuvent avoir des scores de fiabilité élevés mais des scores de sécurité plus faibles, ce qui peut affecter leur pertinence pour certains utilisateurs.

Comparaison:

- Lors de l'utilisation du modèle SimpleRNN, les services recommandés avec les scores de fiabilité les plus élevés sont "ser08", "ser01" et "ser07". En revanche, avec le modèle CONV1D, les services "ser04", "ser10" et "ser02" obtiennent les scores de fiabilité les plus élevés. Enfin, avec le modèle MLP, les services "ser02", "ser08" et "ser10" sont les plus recommandés en termes de fiabilité.

6. Conclusion:

Malgré certaines variations dans les recommandations entre les modèles, il semble exister une cohérence dans les services qui reçoivent des scores de fiabilité élevés. Parmi ces services, "ser02", "ser08" et "ser10" se démarquent comme des choix fiables et sécurisés, d'après les différents modèles. Néanmoins, il est conseillé de prendre en compte les scores de sécurité en plus des scores de fiabilité pour prendre une décision éclairée. Il est important d'examiner attentivement ces indicateurs afin de s'assurer de choisir un service qui répond pleinement à vos besoins en matière de fiabilité et de sécurité.

CONCLUSUION GENERAL

Conclusion général :

En conclusion, ce mémoire a présenté une proposition novatrice de système de recommandation utilisant le deeplearning pour la sélection de services web. L'objectif était d'améliorer la pertinence et la précision des recommandations fournies aux utilisateurs face à la multitude de choix disponibles.

Nous avons élaboré une architecture basée sur des réseaux de neurones Convolutifs (CNN) et des réseaux de neurones récurrents (RNN) et perceptron multicouche (MLP de l'anglais Multi Layer Perceptron) pour modéliser les caractéristiques des utilisateurs et des services web. Ces architectures ont démontré leur capacité à capturer des informations complexes et à apprendre des patterns significatifs dans les données.

Nous avons évalué notre système de recommandation sur une base de données conçue et construite de notre part, a démontré des performances prometteuses. Les résultats ont montré une amélioration significative par rapport aux approches traditionnelles de recommandation, ce qui confirme l'efficacité de l'utilisation du deeplearning dans ce contexte.

Cependant, il convient de noter que notre proposition comporte également certaines limites. Par exemple, la disponibilité de données d'utilisation des services web peut être limitée, ce qui peut affecter les performances du modèle. De plus, l'interprétabilité des recommandations générées par le modèle de deeplearning peut poser des défis.

Pour les perspectives futures, il serait intéressant d'explorer davantage les techniques de transfert de connaissances et d'apprentissage par renforcement pour améliorer encore les performances du système de recommandation. De plus, l'incorporation de données contextuelles supplémentaires, telles que la localisation géographique ou les informations démographiques des utilisateurs, pourrait permettre des recommandations encore plus personnalisées.

En conclusion, notre proposition de système de recommandation basé sur le deeplearning pour la sélection de services web ouvre de nouvelles perspectives dans le domaine de la recommandation personnalisée. En exploitant les capacités du deeplearning, nous avons pu fournir des recommandations pertinentes et adaptées aux besoins spécifiques des utilisateurs, améliorant ainsi leur expérience dans la recherche et l'utilisation de services web.

RÉFÉRENCES

Références

[1] E. Rich, "User modeling via stereotypes," *Cogn. Sci.*, vol. 3, no. 4, pp. 329–354, 1979.

[2] P. Brusilovsky and M. T. Maybury, "From adaptive hypermedia to the adaptive web," *Commun. ACM*, vol. 45, no. 5, pp. 30–33, 2002.

[3] P. Resnick, "An open architecture for collaborative filtering of netnews," in *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina*, 1994.

[4] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-adapt. Interact.*, vol. 12, pp. 331–370, 2002.

[5] L. Terveen and W. Hill, "Beyond recommender systems: Helping people help each other," *HCI New Millenn.*, vol. 1, no. 2001, pp. 487–509, 2001.

[6] I. Benouaret, "Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels To cite this version : HAL Id : tel-01767997 Thèse présentée pour l ' obtention du grade de Docteur de l ' UTC," 2018.

[7] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[8] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Appl. Sci.*, vol. 10, no. 21, p. 7748, 2020.

[9] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, 2014.

[10] E. Saymard, "Mémoire De Fin D ' Études Mémoire De Fin D ' Études," p. 2013, 2010.

[11] M. KARAOUZENZ, "Système de recommandation des services web sémantiques," pp. 2014–2015, 2015, [Online]. Available: http://dspace.univ-tlemcen.dz/handle/112/8210

[12] N. Wang, "Towards a competency recommender system from collaborative traces HAL Id : tel-01728209 Thèse présentée pour l ' obtention du grade de Docteur de l ' UTC," 2018.

[13] J. Gaillard, "Systèmes de recommendation : adaptation Dynamique et Argumentation To cite this version : HAL Id : tel-01168476 Recommender Systems : Dynamic Adaptation and Argumentation," 2015.

[14] N. A. MESROUA and D. SEGHIRI, "Application et exploration de l'approche de découpage des systèmes de recommandations pré-contextuelle." 08-09-2019.

[15] E. HACENE and S. TEKIA, "Développement d'un système de recommandation dans le domaine médical."

[16] A. Hadil and N.-E.-Y. Mandi, "Système de recommandation à base de communautés." Université Ibn Khaldoun-Tiaret-, 2022.

[17] A. GUERBA, N. BENMEHIDI, and F. SADOUKI, "Système de Recommandation Sociale." 2017.

[18] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "Mobile recommender systems in tourism," *J. Netw. Comput. Appl.*, vol. 39, no. 1, pp. 319–333, 2014, doi: 10.1016/j.jnca.2013.04.006.

[19] B. Safia, "Mémoire de Fin d ' études Master Un système de recommandation des ressources en se basant sur les profils des apprenants," 2022.

[20] R. De Croon, L. Van Houdt, N. N. Htun, G. Štiglic, V. Vanden Abeele, and K. Verbert, "Health recommender systems: Systematic review," *J. Med. Internet Res.*, vol. 23, no. 6, 2021, doi: 10.2196/18035.

[21] W. W. W. Consortium, "Web Services Architecture, W3C Working Group Note 11 February 2004," *http//www. w3. org/TR/2004/NOTE-ws-arch-20040211/*, 2004.

[22] D. Hazar, "Informatique Une Approche Effective Pour La Sélection Des Services Web Composites Remerciements," 2019.

[23] M. BOUROUIS, "La sélection de services Web À base de l'algorithme d'abeilles." 2012.

[24] "No Title." http://www.ibm.com/developerworks/library/ws-bpel/

[25] F. ELOUCHDI and M. E. M. KHITRI, "Une approche de sélection des services Web composites à base de critères de vie Privée." 28-06-2022.

[26] N. BOUKAIS, "Réalisation d'un service web selon l'architecture REST." 15-01-2018, 2017.

[27] N. LAACHEMI, C. BENAISSA, and K. BOUDJEBBOUR, "Prediction of web services access using Association Rule Mining." 2021.

[28] F. Chollet, *L'apprentissage profond avec Python* . 2020.

[29] C.-A. Azencott, "Introduction au Machine Learning," --, no., p., 2022.

[30] P. Cornioley, "Intégration d'un module d'apprentissage profond dans l'architecture logicielle d'un SIG Web," *Mémoire présenté pour l'obtention du grade Maître en Sci. Univ. Sherbrooke*, 2018.

[31] Https://www.euro-langues.org/avantages-et-inconvenients-du-langage-dapprentissage-automatique-apprendre-langue/, "No Title."

[32] V. Mathivet, *Artificielle L ' Intelligence*.

[33] G. Saint-Cirgue, "Apprendre le Machine Learning en une semaine," *Mach. Learn. Londres*, 2019.

[34] R. Mifdal, "Application des techniques d'apprentissage automatique pour la prédiction de la tendance des titres financiers." École de technologie supérieure, 2019.

[35] A. Labiad, "Sélection des mots clés basée sur la classification et l'extraction des règles d'association." Université du Québec à Trois-Rivières, 2017.

[36] Https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/, "No Title."

[37] M. Millan, “L’apprentissage profond pour l’évaluation et le retour d’information lors de l’apprentissage de gestes.” Sorbonne université, 2020.

[38] K. Dahmane, “Analyse d’images par méthode de Deep Learning appliquée au contexte routier en conditions météorologiques dégradées.” Université Clermont Auvergne [2017-2020], 2020.

[39] Https://www.geeksforgeeks.org/advantages-and-disadvantages-of-deep-learning/, “No Title.”

[40] Https://www.geeksforgeeks.org/deep-learning-interview-questions/, “No Title.”

[41] C. Hardy, “Contribution au développement de l’apprentissage profond dans les systèmes distribués.” Rennes 1, 2019.

[42] Https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/, “No Title,” *https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/*.

[43] N. Dif, “L’apprentissage profond pour le traitement des images.” Djillali Liabes University, 2020.

[44] J. Frej, “Connaissances a priori pour la Recherche d’Information textuelle basée sur l’apprentissage profond.” Université Grenoble Alpes, 2021.

[45] R. Mu, “A survey of recommender systems based on deep learning,” *Ieee Access*, vol. 6, pp. 69009–69022, 2018.

[46] L. Berkani, I. Kerboua, and S. Zeghoud, “Recommandation Hybride basée sur l’Apprentissage Profond.,” in *EDA*, 2020, pp. 69–76.

[47] A. Rath and S. R. Sahu, “Recurrent neural networks for recommender systems,” *Comput. Intell. Mach. Learn.*, vol. 1, no. 1, pp. 31–36, 2020.

[48] B. Dellal-Hedjazi and Z. Alimazighi, “Hybrid filtering and semantic sentiment analysis by deep learning for recommendation systems,” *Int. J. Inf. Sci. Technol.*, vol. 6, no. 3, pp. 16–28, 2022.

[49] “No Title.” https://www.python.org/

[50] “No Title.” https://numpy.org/

[51] “No Title.” https://pandas.pydata.org/

[52] “No Title.” https://scikit-learn.org/stable/

[53] “No Title.” https://www.tensorflow.org/?hl=fr

[54] “No Title.” https://www.tensorflow.org/guide/keras?hl=fr

[55] M. Jiang, J. Liu, L. Zhang, and C. Liu, “An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms,” *Phys. A Stat. Mech. its Appl.*, vol. 541, p. 122272, 2020.