République Algérienne Démocratique et Populaire Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saâd Dahlab Blida 1
Faculté des sciences
Département d'informatique

Mémoire de fin d'études Pour l'obtention du diplôme de Master
Option : Ingénierie des Logiciels

Un modèle d'apprentissage en profondeur pour détecter les faux profils.

Réalisé Par : Houria Belkebir Boukherouba Ahlem

Membres de jurys composés de :

Mme. Ouahrani	Président
Mme. Ferdi	Examinateur
Mme. BOUMAHDI Fatima	Promotrice
Mr. REMMIDE Mohamed Abdelkarim	Co-Promoteur

2021/ 2022

Remerciements

Au nom de Dieu, le Très Miséricordieux, le Très Miséricordieux Louange à Dieu, Seigneur des mondes Nous levons la plume pour écrire et exprimer nos remerciements et notre gratitude au Superviseur Boumehdi Fatima et au Superviseur Remmide Mohamed Abdel Karim, qui ont été d'un grand soutien durant cette période, en plus de leur motivation et de leur patience à notre égard. demandons à Dieu le succès, et nous sommes reconnaissants à tous nos aides

Abstract

Social networking sites are popular applications for posting videos and photos, due to their great popularity, despite their development to the fact that they are exposed to several dangers, among these dangers are fake personal accounts. The attacker fakes accounts to spread misinformation such as malware, viruses, or malicious URLs. In this work, we formulate the problem of detecting malicious accounts in communities, a deep learning solution to detect malicious accounts, where we implemented five algorithms, the first one being Convolutional neural network (CNN) ,Temporal Convolutional Networks (TCN), the third algorithm was a mix of (TCN) and (LSTM), and the fourth algorithm was a mix of (TCN + LSTM + GRU) , and the fifth algorithm (TCN + LSTM + Bi GRU) was applied to two datasets, each group containing two parts.

The highest accuracy was reached in the first data set, especially in the 2 CLASS part, was about 90.81%, and that was in the (TCN + LSTM) As for the 4 CLASS part, the highest accuracy was found in the model (TCN + LSTM + Bi GRU), reaching 90.12%. A For the second data set, the highest accuracy was found in the model (TCN + LSTM + GRU) was 97.46%.

Keywords: Deep learning, Neural networks, Social networks, Detection of fake accounts (Sybils).

Résumé

Les sites de réseaux sociaux sont des applications populaires pour publier des vidéos et des photos, en raison de leur grande popularité, malgré leur développement du fait qu'ils sont exposés à plusieurs dangers, parmi ces dangers figurent les faux comptes personnels. L'attaquant falsifie les comptes pour diffuser des informations erronées telles que des logiciels malveillants, des virus ou des URL malveillantes. Dans ce travail, nous formulons le problème de détection des comptes malveillants dans les communautés, une solution d'apprentissage en profondeur pour détecter les comptes malveillants, où nous avons implémenté cinq algorithmes, le premier étant Convolutional neural network (CNN), Temporal Convolutional Networks (TCN), le troisième l'algorithme était un mélange de (TCN) et (LSTM), et le quatrième algorithme était un mélange de (TCN + LSTM + GRU) , et le cinquième algorithme (TCN + LSTM + Bi GRU) , a été appliqué à deux ensemble de données, chaque groupe contenant deux parties.

La précision la plus élevée a été atteinte dans le premier ensemble de données, en particulier dans la partie 2 CLASS, était d'environ 90,81%, et c'était dans le modèle (TCN + LSTM) Quant à la partie 4 CLASS, la précision la plus élevée a été trouvée dans le modèle (TCN + LSTM + Bi GRU), atteignant 90,12 %. A Pour le deuxième ensemble de données, la précision la plus élevée a été trouvée dans le modèle (TCN + LSTM + GRU) était de 97,46 %

Mots clés: L'apprentissage en profondeur, Réseaux de neurones, Réseaux sociaux, Détection de faux comptes (Sybils).

ملخص

مواقع التواصل الاجتماعي هي تطبيقات شهيرة لنشر مقاطع الفيديو والصور ، نظرا لشعبيتها الكبيرة ، على الرغم من تطورها لكنها معرضة لعدة مخاطر ، من بينها الحسابات الشخصية المزيفة. يقوم المهاجم بتزوير الحسابات لنشر معلومات خاطئة مثل البرامج الضارة أو الفيروسات أو عناوين URL الضارة. في هذا العمل ، قمنا بصياغة مشكلة اكتشاف الحسابات الضارة في المجتمعات ، وهو حل تعليمي عميق لاكتشاف الحسابات الضارة ، حيث قمنا بتنفيذ خمس خوارزميات ، أولها الشبكة العصبية الالتفافية (CNN) ، والشبكات التلافيفية المؤقتة (TCN) ، والثالثة كانت الخوارزمية مزيجًا من (TCN) و (LSTM) ، وكانت الخوارزمية الرابعة مزيجًا من (TCN + LSTM + GRU) ، والخوارزمية الخامسة (TCN + LSTM + Bi GRU) ،تم تطبيقه على مجموعتي بيانات ، تحتوي كل مجموعة على جزأين. تم الوصول إلى أعلى دقة في مجموعة البيانات الأولى ، خاصة في الجزء 2 CLASS ، حيث كانت حوالي 90.81٪ ، وكان ذلك في النموذج (TCN + LSTM) أما بالنسبة للجزء 4 CLASS ، فقد تم العثور على أعلى دقة في النموذج (TCN + LSTM + Bi GRU) تصل إلى 90.12٪. A لمجموعة البيانات الثانية ، تم العثور على أعلى دقة في النموذج (TCN + LSTM + GRU) كانت 97.46٪.

الكلمات المفتاحية: التعلم العميق ، الشبكات العصبية ، الشبكات الاجتماعية ، كشف الحسابات المزيفة (Sybils).

Table des matières

Introduction générale 12

1.1 Introduction 14

1.2 Type d'apprentissage 14

1.2.1 Apprentissage supervisé 14

1.2.2 L'apprentissage non supervisé 17

1.2.3 L'apprentissage semi-supervisé 19

1.2.4 L'apprentissage en profondeur 21

- **LSTM** 24
- **Gated Recurrent Unit** 25
- **Bidirectional GRU** 27

1.3 Conclusion 28

Chapitre 2 Détection des faux profils 29

2.1 Introduction 29

2.2 Profil 29

2.3 Faux profils 29

2.4 Les buts de la création d'un faux profil 30

2.5 Travaux connexes 31

2.6 Conclusion 38

Chapitre 3 La solution proposée 40

3.1 Introduction 40

3.2 L'architecture globale de notre solution 40

3.3 Base de données 42

3.4 Prétraitement 45

3.5 Le modèle CNN 47

3.6 Le modèle TCN 50

3.6.1 Convolutions causales 50

3.6.2 Convolutions dilatées 51

3.6.3 Connexions résiduelles 51

3.7 Conclusion 56

Chapitre 4 Test et résultats 57

4.1 Introduction 57

4.2 Choix techniques 57

4.2.1 Environnement du développement 57

4.2.2 Equipment 57

4.3 Le langage de programmation utilisé 58

4.3.1 Les bibliothèques utilisées 58

4.4 Les mesures d'évaluation 60

4.5 Paramètre de compilation 62

4.6 Évaluation des résultats et comparaison 63

4.7 La comparaison avec les travaux connexes 71

4.8 Conclusion 79

Conclusion générale 80

Références 81

Liste des figures

Figure1- 1 : représente comment le SVM sépare les 2 catégories avec le hyperplan.(Chauhan et al., 2019) 16
Figure 1- 2: les deux types de clustering hiérarchique/non hiérarchique.(Oumiloud & Mokeddem, 2014) 18
Figure 1- 3 : Un exemple une fonction discriminante apprise de manière semi supervisée sur des 20
Figure 1- 4 : La illustre la relation entre apprentissage en profondeur, apprentissage automatique, intelligence artificielle. 21
Figure 1- 5 :représente l'architecture de NN (Dongare et al., 2012) 22
Figure 1- 6 : Architecture de LeNet un réseau de neurones convolutifs (Y. Zhang et al.2020) 23
Figure 1- 7 : Modèle conceptuel de CNN(Y. Zhang et al., 2020) 24
Figure 1- 8 : Le modèle Lstm 25
Figure1- 9 : La structure cellulaire d'une GRU 26
Figure 1- 10 :la structure bidirectionnelle du GRU 27
Figure 2 - 11 :Les algorithmes utilisés dans les articles. 38
Figure 3- 12 : Organigramme proposé 40
Figure3 - 13 : schéma global de projet 42
Figure 3 - 14 : changement des valeurs de classe des data de 2 classe et 4 classe. 45
Figure 3- 15: partition de l'ensembe de données 47
Figure 3- 16:Le réseau de neurones convolutifs 48
Figure 3- 17 : Couches et paramètres utilisés dans le réseau de neurones CNN 48
Figure 3 - 18 : architecture de modèle tcn (Lara-Benítez et al., 2020) 50
Figure 3- 19:combinaison entret TCN et LSTM. 54
Figure 3 - 20 :combinaison entre TCN et Lstm et GRU. 55
Figure 3- 21 :combinaison entre TCN et Lstm et Bi GRU. 56
Figure 4 - 22 : schéma des bibliothèques et leur utilité 60
Figure 4 - 23 :l'histogramme du premier ensemble de données 2 classes 64
Figure 4 - 24 : Matrices de confusion de model CNN sur le data de 2 classe 64
Figure 4 - 25 : Matrice de confusion et les métriques d'évaluation du modéle TCN sur le data de 2 classe. 65
Figure 4 - 26 : les métriques d'évaluation de model TCN+LSTM sur le data de 2 classe 65
Figure 4- 27 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM +GRU sur dataset de 2 classes 66
Figure 4- 28 :Matrice de confusion et les métriques d'évaluation du model TCN+LSTM+Bi GRU sur le data de 2 classe. 66
Figure 4 - 29 : l'histogramme du 2eme dataset (4 classes) 68
Figure 4 - 30 : la matrice de confusion les métriques d'évaluation de model CNN sur le data de 4 classe. 68
Figure 4 - 31 : Matrice de confusion et les métriques d'évaluation de model TCN sur le dataset de 4 classes. 69
Figure 4 - 32 : Matrice de confusion et les métriques d'évaluation du model TCN+LSTM sur le data de 4 classe 69
Figure 4- 33 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM+GRU sur le data de 4 classe 70
Figure 4 - 34 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM+Bi GRU sur le data de 4 classe 70
Figure 4 - 35 : les matrices de confusion après le changement d'optimiseur de data de 2 classe 74
Figure 4 - 36 : matrices de confusion après le changement d'optimiseur de data de 4 classe 75
Figure 4 - 37 : Matrice de confusion et les métriques d'évaluation du model CNN sur fake account detection dataset. 76
Figure 4- 38 : Matrice de confusion et les métriques d'évaluation du model TCNsur fake account detection dataset. 76
Figure 4- 39 : Matrice de confusion et les métriques d'évaluation du model TCN+LSTM sur fake account detection dataset 77
Figure 4 -40 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM+GRU sur fake account detection dataset 77
Figure 4- 41 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM+Bi gru sur fake account detection dataset. 78

Liste des tableaux

Tableau 2- 1 :un tableau qui résume les travaux connexes 35
Tableau 3 - 2 :dataset 44
Tableau 3 - 3: La liste des fonctions. 44
Tableau 4- 4 : La matrice de confusion 61
Tableau 4 - 5 : Tableau Les paramètres de travai 65
Tableau 4 - 6 : résultats de tous les modèles 68
Tableau 4- 7 : Les paramètres de travail sur le dataset 4 classes 69
Tableau 4 - 8 : résultats des 5 modèles 72
Tableau 4 - 9 : comparaison entre les modèles utilisées et les travaux connexes sont appliquées sur le data 2 classe. 72
Tableau 4 - 10 : : comparaison entre les modèles utilisées et les travaux connexes sont appliquées sur le data 4 classe 73
Tableau 4 - 11 : tableau représente les résultats après le changement d'optimiseur de dataset de 2 classes. 75
Tableau 4 - 12 : tableau représente les résultats après le changement d'optimiseur de data de 4classe 76
Tableau 4- 13 : Résume les résultats des 5 modèles sur ensemble de données de détection de faux comptes. 79

Liste des équation

Équation 3-1 52

Équation 3-2 53

Équation 3-3 57

Équation 3-4 57

Équation 3-5 57

Équation 3-6 58

Équation 4-1 :Equation de l'accuracy 67

Équation 4-2 : Equation de précision 68

Équation 4-3 : Equation de rappel 68

Équation 4-4: Equation de F1 mesure 68

Tableau des abréviations

Abréviations	Signification
DL	Deep Learning(L'apprentissage en profondeur)
ML	Machine learning(Apprentissage automatique)
RN NN	Réseaux de neurones Neural networks
RS	Réseaux sociaux
DFC	Détection des faux comptes
CNN	Convolutional Neural Network(Réseau de neurones convolutifs)
LSTM	Long short term memory(Mémoire longue à court terme)
TCN	Temporal Convolutional Network(Réseau convolutif temporel)
GRU	Gated recurrent units(Unités récurrentes fermées)
Bi GRU	Bidirectional Gated recurrent units(Unités récurrentes bidirectionnelles fermées)
RNN	Recurrent neural network(Réseau neuronal récurrent)
KNN	k plus proches voisins(k-nearest neighbors)
SVM	Support vector machine

Introduction générale

Les réseaux sociaux en ligne tels que Twitter, Facebook et Instagram sont devenus une plate-forme très nécessaire pour la communication et l'échange d'informations entre les personnes et les utilisateurs du même parcours, des intérêts professionnels et des activités, étant donné la popularité des sites de réseaux sociaux qu'il est une bonne arme et plateforme de cybercriminalité.

La vie en ligne a beaucoup évolué car Instagram a récemment accru sa notoriété parmi les clients de réseautage en ligne. Avec plus d'un milliard de clients dynamiques, Instagram est devenu l'une des destinations de réseautage les plus utilisées sur Internet (Halim et al., 2011). Après qu'Instagram ait développé le cas de vie basé sur le Web, il y a eu une augmentation progressive du nombre d'influenceurs des médias sociaux. Ces influenceurs de la vie en ligne font la promotion de leurs entreprises et de leurs produits via les médias sociaux (Prem Jacob et al., 2018). En raison de l'utilisation généralisée d'Internet.

Problématique

La détection des faux comptes dans les réseaux sociaux protège à la fois le système de réseaux sociaux et les informations personnelles de leurs utilisateurs contre diverses activités illégales, trompeuses et malveillantes.
Les mécanismes de la détection tentent à classer les profils des vrais utilisateurs comme profil authentique réel et les profils malveillants comme faux profil.

Objectif

Les objectifs principaux de notre travail sont :

- Création d'un nouveau modèle d’apprentissage en profondeur.
- Sécuriser les informations personnelles des utilisateurs.
- Effectuer une détection rapide et évolutive des faux profils.

Plan de mémoire

Dans cette note, nous avons expliqué plusieurs domaines afin de trouver des solutions appropriées, comme **le chapitre 1** expliquant l'apprentissage automatique et ses domaines, car il se compose de quatre types, à savoir : Apprentissage supervisé. Apprentissage non supervisé. Apprentissage semi supervisé, d'apprentissage en profondeur. Étant donné que chaque type a son propre ensemble d'algorithmes.

Le chapitre 2 a expliqué un ensemble de techniques qui doivent être utilisées pour détecter les faux calculs, et nous résumons des travaux scientifiques similaires.

Dans **le chapitre 3** nous avons présenté nos solutions proposées, qui sont un ensemble d'algorithmes convolutional neural network (CNN),temporal convolutional network(TCN), temporal convolutional network + long short term memory(TCN+LSTM) ,Réseau convolutif temporel + mémoire à long terme + Gated Recurrent Unit(TCN+LSTM+GRU),Réseau convolutif temporel + mémoire à long terme + Bidirectional Gated Recurrent Unit(TCN+LSTM+Bi GRU)qui visent tous à détecter les faux comptes, car l'un d'eux a ses propres caractéristiques et son mode de fonctionnement qui le distingue des autres.

Et dans **le chapitre 4**, Nous avons présenté les résultats obtenus et les avons comparés avec d'autres travaux et nous avons choisi le meilleur entre eux.

Chapitre1 Apprentissage

1.1 Introduction

Intelligence artificielle (IA) est un domaine scientifique qui cherche à résoudre des problèmes logiques ou algorithmiques, il constitue des dispositifs imitant ou remplaçant l'être humain. (Graesser et al., 2018)

IA est un vaste domaine qui contient plusieurs branches telles que l'apprentissage automatique et l'apprentissage en profondeur alors dans ce chapitre nous allons expliquer les concepts les plus importants de ces branches et leurs algorithmes.

1.2 Type d'apprentissage

L'apprentissage automatique (Karayemiş, 2014) (en anglais Machine Learning ML) est un champ d'étude de l'IA. Il est le processus qui utilise des probabilités statistiques pour donner aux ordinateurs la capacité d'apprendre par eux-mêmes sans programmation explicite.

Il existe également de différents types d'apprentissage automatique (Çelik, 2018):

- ✔ **Apprentissage supervisé.**
- ✔ **Apprentissage non supervisé.**
- ✔ **Apprentissage semi-supervisé.**

1.2.1 Apprentissage supervisé

L'apprentissage supervisé (Theobald, 2017) se concentre sur les modèles d'apprentissage en reliant la relation entre les variables et les résultats connus et en travaillant avec des ensembles de

données étiquetés. L'apprentissage supervisé confère au modèle la possibilité de prédire des valeurs d'étiquette sur des données non étiquetées supplémentaires.

Il existe deux types de sous problèmes en apprentissage supervisé (Brownlee, 2016):

- Classification (où catégorisation) : Les algorithmes de classification sont utilisés lorsque la variable à prédire Y est discrète. Une classification peut avoir des variables d'entrée à valeurs réelles ou discrètes. Il est courant que les modèles de classification prédisent des valeurs continues comme les probabilités d'appartenance à chaque classe de sortie. Une probabilité prédite peut être convertie en une valeur de classe en sélectionnant l'étiquette de la classe qui présente la probabilité la plus élevée.
- Régression : Les algorithmes de régression sont quant à eux utilisés lorsque la variable à prédire Y est continue. Ces variables d'entrée peuvent être des valeurs réelles ou discrètes.

  Cet apprentissage a plusieurs algorithmes :

  - **Support Vector Machine (SVM)**

  SVM support vector machine (Chauhan et al., 2019) est un algorithme d'apprentissage automatique qui permet de résoudre des problèmes de régression, classification ou de détection d'anomalie.

  Le SVM ont été développés dans les années 1990 par les informaticiens russes Vladimir Vapnik et Alexey Chervonenkis , il a été rapidement adopté en raison de sa capacité à travailler avec des données de grandes dimensions et sa grande flexibilité ainsi que leur simplicité d'utilisation, ses garanties théoriques et les bons résultats réalisés en pratique.

  SVM effectue le test de classification en dessinant un hyperplan qui est une ligne qui sépare les 2 classes (catégories) en telles sort que les points d'une catégorie soient d'un côté et que tous les points d'autre catégorie soient d'un autre côté et il puisse avoir plusieurs hyperplans.

  SVM essaie de trouver le meilleur hyperplan qui sépare les catégories qu'il maximise la distance au point d'une ou l'autre catégorie cette distance est appelée la marge et les points qui tombent exactement sur la marge sont appelées les vecteurs de support.

Quand nous avons une nouvelle donnée d'entrée, nous la mettons dans le SVM, comme la figure1-1 le montre si la donnée est sous dessous l'hyperplan elle sera assignée à la classe rouge sinon à la classe bleue.

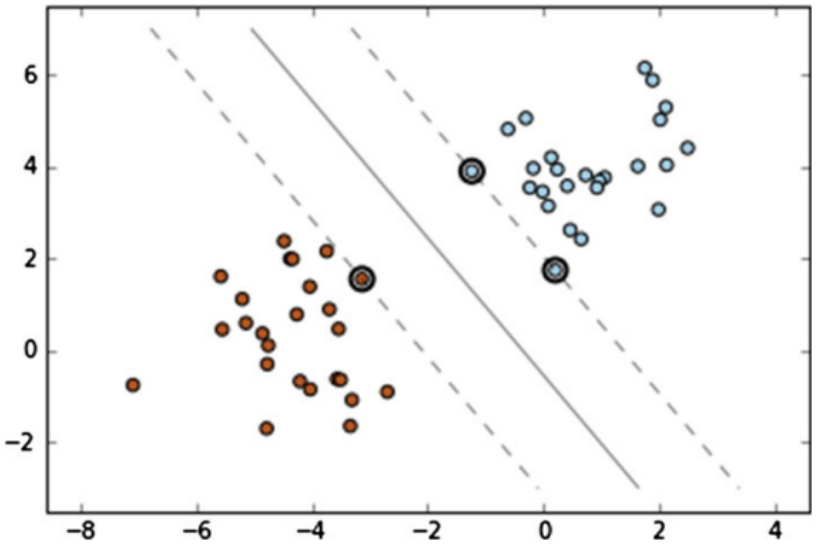


Figure1- 1 : représente comment le SVM sépare les 2 catégories avec le hyperplan.(Chauhan et al., 2019)

- **K plus proches voisins (KNN)**

La méthode des k plus proches voisins (KNN) est une méthode de classification non paramétrique, simple mais efficace dans de nombreux cas (Guo et al, 2003). Pour qu'un enregistrement de données t soit classé, ses k plus proches voisins sont récupérés, ce qui forme un voisinage de t. Le vote majoritaire parmi les enregistrements de données dans le voisinage est généralement utilisé pour décider de la classification de t avec ou sans considération de la pondération basée sur la distance.

Cependant, pour appliquer la méthode KNN, nous devons choisir une valeur appropriée pour k, et le succès de la classification dépend fortement de cette valeur. En un sens, la méthode KNN est biaisée par k, Il existe de nombreuses façons de choisir la valeur de k, mais une méthode simple consiste à exécuter l'algorithme plusieurs fois avec différentes valeurs de k et à choisir celle qui présente les meilleures performances.

- **Naive Bayesian**

Le classifieur naïf bayésien est un type de classification bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses,

appartenant à la famille des classifieurs linéaires. Il suppose que l'existence d'une caractéristique pour une classe est indépendante de l'existence d'autres caractéristiques (H. Zhang, 2004).

- **Avantages**

Méthode facile à comprendre et adaptée aux domaines où chaque classe est représentée par plusieurs prototypes et où les frontières sont irrégulières (ex. Reconnaissance de chiffre manuscrits ...).(Berry, 2020)

- **Inconvénient**

Prédiction lente car il faut revoir tous les exemples à chaque fois et la méthode gourmande en termes de mémoire et sensible aux attributs. (Theobald, 2017).

1.2.2 L'apprentissage non supervisé

L'apprentissage non supervisé traite des données non étiquetées, et le but est d'identifier automatiquement les caractéristiques communes des observations.

Il dépend de plusieurs méthodes, dont les méthodes de réduction des dimensions, qui ont été représentées dans l'analyse de sa composante principale, en plus de l'estimation de la densité de probabilité, qui fait partie de l'apprentissage non supervisé.

Les algorithmes doivent ici analyser et regrouper les données, sans aucune intervention humaine, en découvrant les patterns au sein des masses de données (Captcha, 2022)On distingue aussi les approches de classification non hiérarchiques et les méthodes de classification hiérarchiques comme la figure 1-2 le montre.(Captcha, 2022)

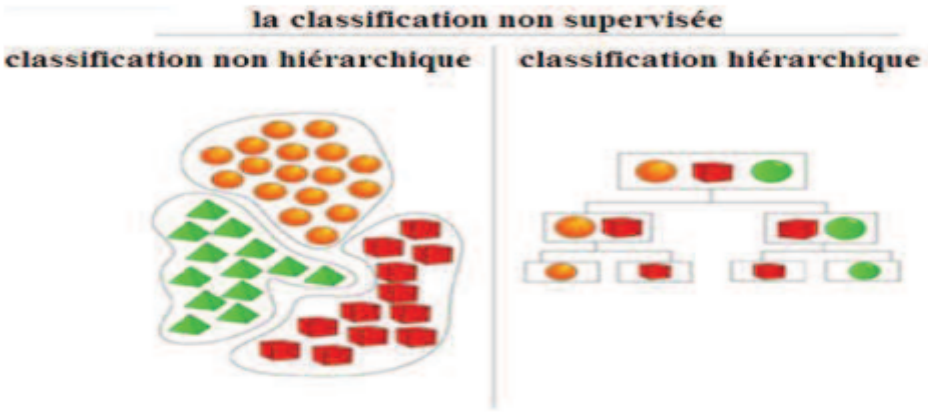


Figure 1- 2: les deux types de clustering hiérarchique/non hiérarchique.(Oumiloud & Mokeddem, 2014)

- **Méthodes non hiérarchiques**

Les méthodes non hiérarchiques sont l'une des méthodes qui dirigent la production on la divisant en un nombre spécifié de couches ou un nombre fixe.

Où **n** personnes sont regroupées en **k** catégories pour que ces individus soient de la même catégorie et se ressemblent le plus possible et les classes sont bien distinctes Parmi ces méthodes on trouve :

a) k-means :

Où il construit K catégories à partir d'un groupe de n individus similaires tout en réduisant la quantité.(Sinaga & Yang, 2020)

Le pseudo algorithme général de ces méthodes :

Donnée : k le nombre maximum de classe désiré.

Début

(1) Choisir k individus au hasard (comme centre des classes initiales)

(2) Affecter chaque individu au centre le plus proche

(3) Recalculer le centre de chacune de ces classes

(4) Répéter l'étape (2) et (3) jusqu'à stabilité des centres

(5) Editer la partition obtenue

Sortie : classe

Fin

Afin de calculer la variance des groupes, les K-means sont tirés plusieurs fois avec différentes valeurs de k, qui est considéré comme le nombre de groupes, car la valeur idéale ne peut pas être connue à l'avance, c'est-à -dire la somme des distances entre chaque centre de masse et les notes incluses sont dans le même groupe. Le but est de déterminer le nombre optimal de groupes pour que la valeur de k minimise la distance intra-classe.

❖ Avantages

-L’avantage de ces algorithmes est avant tout leur grande simplicité. ·

- Tend à réduire l’erreur quadratique.(Captcha, 2022)

❖ Inconvénients

- Nécessité de spécifier le nombre de clusters k.

- Complexité de chaque itération.(Sinaga & Yang, 2020)

1.2.3 L’apprentissage semi-supervisé

L'apprentissage semi-supervisé se situe quelque part entre les deux. Il résout les problèmes de classification, ce qui signifie que vous aurez finalement besoin d'un algorithme d'apprentissage supervisé pour la tâche. Mais en même temps, vous souhaitez entraîner votre modèle sans étiqueter chaque exemple d'entraînement, pour lequel vous obtiendrez l'aide de techniques d'apprentissage automatique non supervisées. Consiste plutôt à fournir une information qualificative (des étiquettes de classification) ou une information quantitative (valeur réelle à des fins de régression) pour seulement un sous-ensemble des points d'entraînement soumis à l’algorithme d’apprentissage. L’algorithme peut quand même extraire de l’information sur la nature dti ou des phénomène(s) observé(s) à partir des vecteurs qui sont fournis sans information de qualification ou de quantification.

Dans la Figure 1-3 on peut voir une surface de décision apprise à des fins de classification étant donné certains points étiquetés comme appartenant à la première ou à la deuxième classe et certains point dont l’étiquette n’est pas révélée.**(*Université de Montréal Apprentissage Semi-Supervisé Par Réduction de Dimensionnalité Non Linéaire*, 2004)**

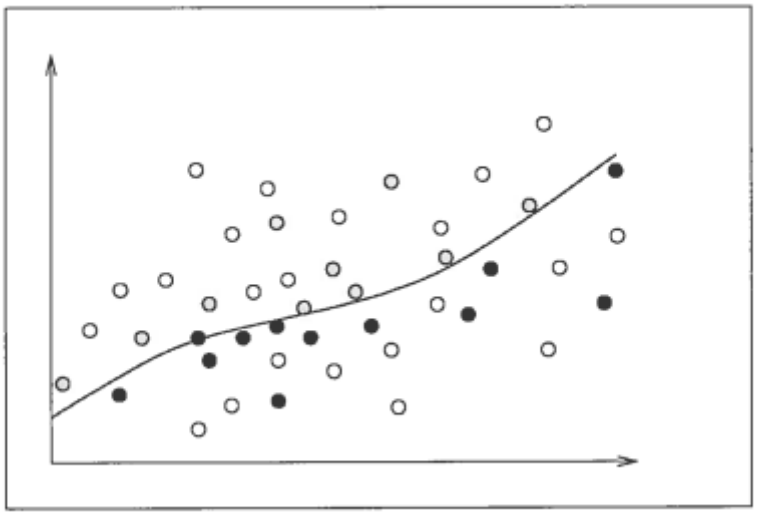


Figure 1- 3 : Un exemple une fonction discriminante apprise de manière semi supervisée sur des points exprimés dans un espace à 2 dimensions (*Université de Montréal Apprentissage Semi-Supervisé Par Réduction de Dimensionnalité Non Linéaire*, 2004)

- **Les algorithmes d'apprentissage semi-supervisée**

✔ L'ALGORITHME COP-KMEANS : la méthode K-means modifiée qui prend en considération des contraintes de comparaison entre paires d'objets est appelée COP-KMEANS.

✔ L'ALGORITHME SKMS : L'algorithme Kernel Mean Shift Clustering semi-supervisé, qui utilise des contraintes paires.

✔ L'ALGORITHME SKLR : regroupement semi supervisé par des contraintes de distances relatives.

✔ L'algorithme EM pour les t-distributions.

✔ L'algorithme FNC (Fuzzy Noise Clustering) (Saint-Jean& Classification, 2007).

❖ **Avantages**

L'apprentissage semi-supervisé permet à l'algorithme d'apprendre à partir d'une petite quantité de documents texte étiquetés tout en classant une grande quantité de documents texte non étiquetés dans les données d'apprentissage. (Brownlee, 2016).

1.2.4 L'apprentissage en profondeur

L'apprentissage en profondeur est une approche de l'IA comme la figure 1-4 le montre. Plus précisément, il s'agit d'un type d'apprentissage automatique, une technique qui permet aux systèmes informatiques de s'améliorer avec l'expérience et les données. L'apprentissage en profondeur atteint une grande puissance et une grande flexibilité en apprenant à représenter le monde comme une hiérarchie imbriquée de concepts, chaque concept étant défini par rapport à des concepts plus simples et des représentations plus abstraites calculées en termes de concepts moins abstraits. (Courville, 2016)

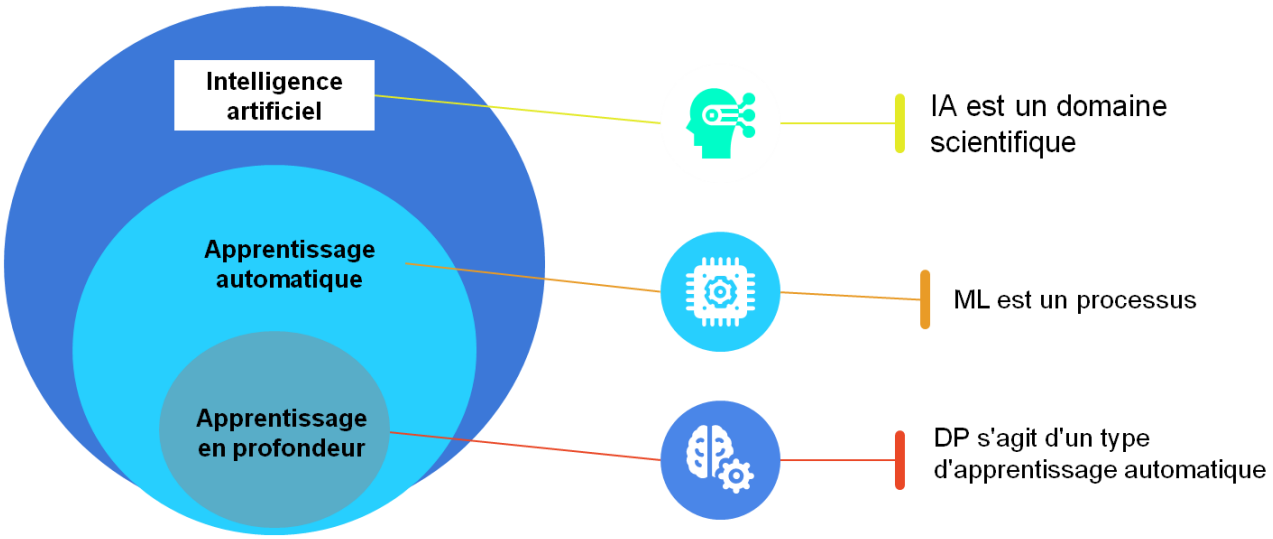


Figure 1- 4 : La illustre la relation entre apprentissage en profondeur, apprentissage automatique, intelligence artificielle.

- **Les algorithmes d'apprentissage en profondeur**

  - **Les réseaux de neurones**

Les réseaux de neurones (Dongare et al., 2012) forment la base d'apprentissage en profondeur (sous domaine d'apprentissage automatique), ils sont inspirés de la méthode de travail du cerveau humain qui est totalement différente de celle d'un ordinateur.

Les réseaux de neurones prennent en compte les données et s'entraînent à reconnaître les modèles de ces données puis prédire les outputs pour un nouvel ensemble de données similaires.

NN sont constitués de couches de neurones et ce dernier est les unités de traitement central du réseau.

Comme la figure 1-5 montre, il y a input layer qui reçoit l'entrée et il y a la dernière couche prédit la sortie finale et entre ces 2 couches il existe les couches cachées (hidden layer) qui effectuent la plupart des calculs requis par le réseau.

Chaque donnée est alimentée en entrée de chaque neurone de la première couche, les neurones d'une couche sont connectés aux neurones de la couche suivante à travers des canaux et chacun de ces canaux est attribué avec une valeur numérique appelée poids.

Alors les entrés sont multipliées par les poids correspondants et leur somme est envoyée en entrée aux neurones de couche cachée et chacun de ces neurones est associées à une valeur numérique appelée le biais, qui est ensuite ajouter à la somme d'entrées et après la valeur sera transmise à une fonction de seuil c'est la fonction d'activation et le résultat de cette fonction détermine si le neurone particulier sera activé ou pas , les neurones activés transmet les données aux neurones de la couche suivante sur les canaux donc le résultat de la couche de sortie finale est utilisé comme solution au problème.

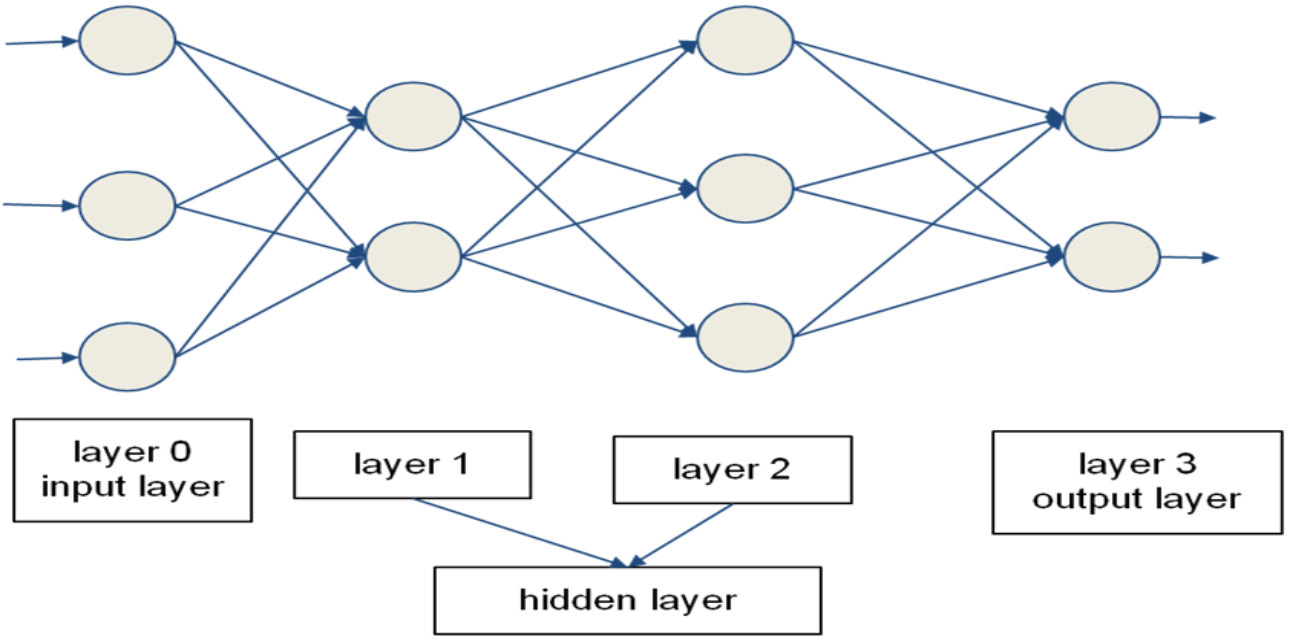


Figure 1- 5 :représente l'architecture de NN (Dongare et al., 2012)

- CNN

Parmi les différentes architectures d'apprentissage en profondeur, un type spécial de réseau neuronal multicouche pour les données spatiales sont Convolutional Neural Network (ou CNN ou ConvNet.). L'architecture de CNN s'inspire de la perception visuelle des êtres vivants. Bien qu'il soit devenu populaire après les performances record d'AlexNet(Y. Zhang et al., 2020) en 2012, il a en fait été lancé en 1980. Après 2012, le CNN a pris le rythme pour prendre en charge différents domaines de la vision par ordinateur, du traitement du langage naturel et bien d'autres.

Ce réseau est le premier modèle théorique de CNN. En 1990, LeCun et al. Il a développé le cadre CNN moderne appelé LeNet-5 (LeCun et al., 1998) pour reconnaître les nombres manuscrits.

L'entraînement avec l'algorithme de rétropropagation LeNet-5 (Rumelhart et al., 2013) a aidé à reconnaître les modèles visuels de Images brutes directement sans utiliser de géométrie de fonction distincte , la figure 1-6 montre son architecture.

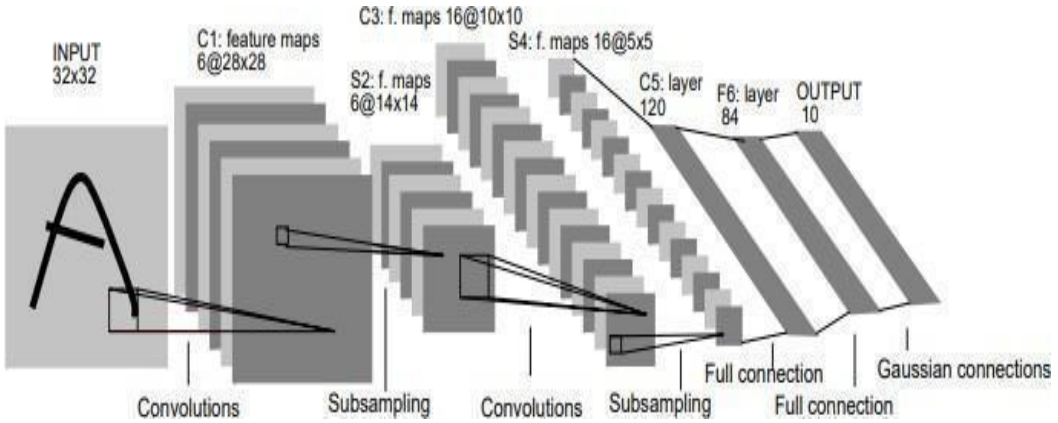


Figure 1- 6 : Architecture de LeNet un réseau de neurones convolutifs (Y. Zhang et al.2020)

o CNN Architectures

Un réseau neuronal convolutif traditionnel est composé d'un ou de plusieurs blocs de couches de convolution et de mise en commun, suivis d'une ou plusieurs couches entièrement connectées (FC) et d'une sortie couche. La couche synaptique est la chambre d'angle du réseau neuronal synaptique CNN. Cette couche vise à apprendre les représentations de caractéristiques de l'entrée. La couche convolutive est composée de plusieurs noyaux ou filtres de convolution qui sont utilisés pour calculer différentes cartes de caractéristiques.

Chaque unité de la carte des caractéristiques est connectée à un champ récepteur dans la couche précédente. La nouvelle carte des caractéristiques est produite en convoluant l'entrée avec les noyaux et en appliquant une fonction d'activation non linéaire par élément sur le résultat convolé. La propriété de partage des paramètres de la couche convolutive réduit la complexité du modèle. La couche de mise en commun ou de sous-échantillonnage prend une petite région de la sortie convolutive comme entrée et la sous-échantillonne pour produire une seule sortie. Ils sont différents techniques de sous-échantillonnage comme par exemple regroupement maximal, regroupement minimal, regroupement moyen, etc.(Y. Zhang et al., 2020)

- **Concepts de réseaux de neurones convolutifs**

Le réseau neuronal convolutif (CNN), également appelé ConvNet, est un type de réseau neuronal artificiel (ANN), comme la figure 1-7 montre qu'il possède une architecture d'alimentation profonde et une capacité de généralisation étonnante comme Comparé à d'autres réseaux avec des

couches FC, il peut apprendre les caractéristiques d'objets très abstraits Surtout les données spatiales et peuvent être identifiés plus efficacement. Un modèle CNN profond se compose d'un ensemble limité de couches de traitement qui peuvent apprendre différentes fonctionnalités. (Y. Zhang et al., 2020)

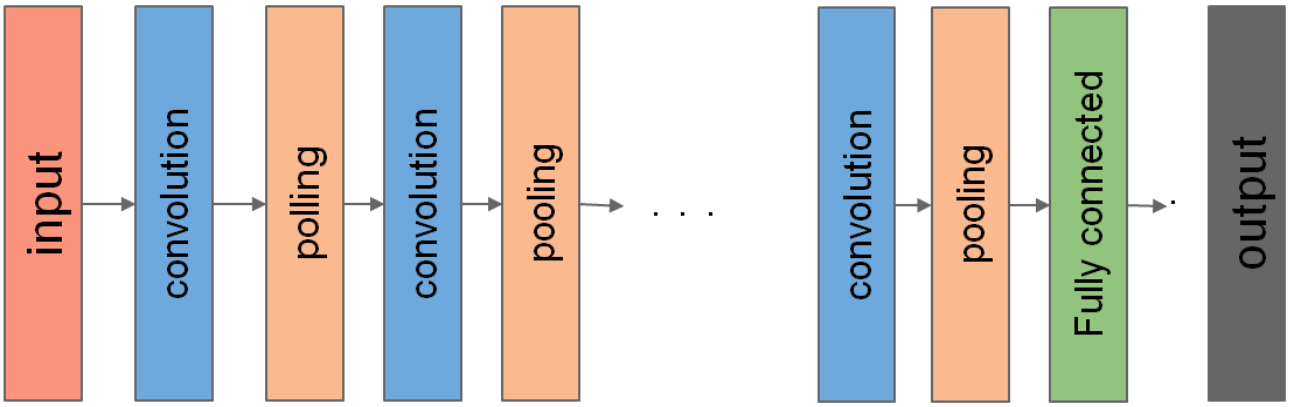


Figure 1- 7 : Modèle conceptuel de CNN(Y. Zhang et al., 2020)

- **Un réseau neuronal récurrent (RNN)**

RNN ou Récurrent Neural Network (Subasi, 2020) est un type particulier de réseau neuronal artificiel, il est connu comme un modèle de séquence.il est utilisé dans le domaine du traitement du langage naturel ainsi que dans d'autres domaines tels que la traduction de la parole en texte, la surveillance de l'activité vidéo, etc.

Le réseau de RNN possédé des connexions de rétroaction veut dire il utilise des boucles de rétroaction pour modéliser les effets des premières parties de la séquence sur les dernières parties de la séquence car il permet aux informations de persister, effet souvent assimilé à la mémoire.

- **LSTM**

Les réseaux de longue mémoire à court terme (LSTM) sont un type spécial des réseaux neuronaux récurrents et ces unités sont utilisées comme unités de construction pour les couches d'un RNN (un réseau de LSTM) et cette unité se compose d'une cellule mémorise les valeurs et trois portes d'entrée et de sortie et la dernière la porte d'oubli régulent le flux d'informations entrant et sortant de la cellule.

Il est un modèle utilisé pour l'analyse de données séquentielles (prédiction de données de séries chronologiques).

Il est bien adapté pour apprendre des expériences importantes qui ont des retards très longs entre les deux et il permet aux RNN de souvenirs leurs intrants sur une longue période de temps parce que les LSTM contiennent leurs informations dans une mémoire, ce qui ressemble beaucoup à la mémoire d'un ordinateur parce que le LSTM peut lire, écrire et supprimer des informations de sa mémoire et la figure 1-8 représente l'architecture de modèle LSTM. (Staudemeyer & Morris, 2019).

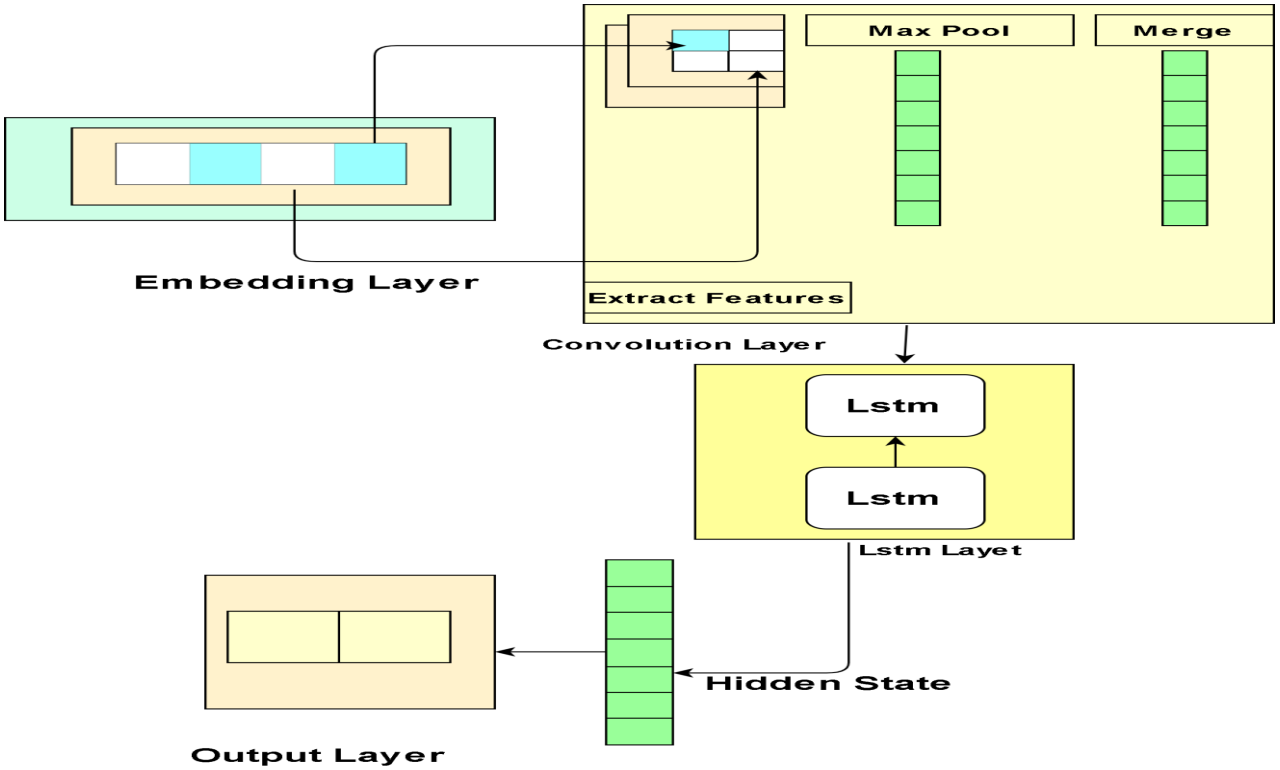


Figure 1- 8 : Le modèle Lstm

- **Gated Recurrent Unit**

Gated Recurrent Unit (GRU) a été introduite par (Cho et al., 2014)car elle est inspirée de l'unité lstm, mais elle se caractérise par sa structure interne plus simple, plus facile à calculer et à mettre en œuvre, et plus facile à former, et moins d'opérations arithmétiques sont nécessaire pour mettre à niveau les états internes. Le GRU est un type spécial de réseau neuronal récurrent amélioré basé sur LSTM car le GRU est très similaire à celui interne de LSTM, car le GRU combine le port entrant et le port oublié, tandis que dans LSTM, il est considéré comme un seul port moderne.

Le port de mise à jour contrôle la quantité d'informations d'état du moment précédent conservées dans l'état actuel. Le port de réinitialisation détermine si ces deux états doivent être combinés. (Cho et al., 2014) la figure1-9 montre la structure interne du GRU.

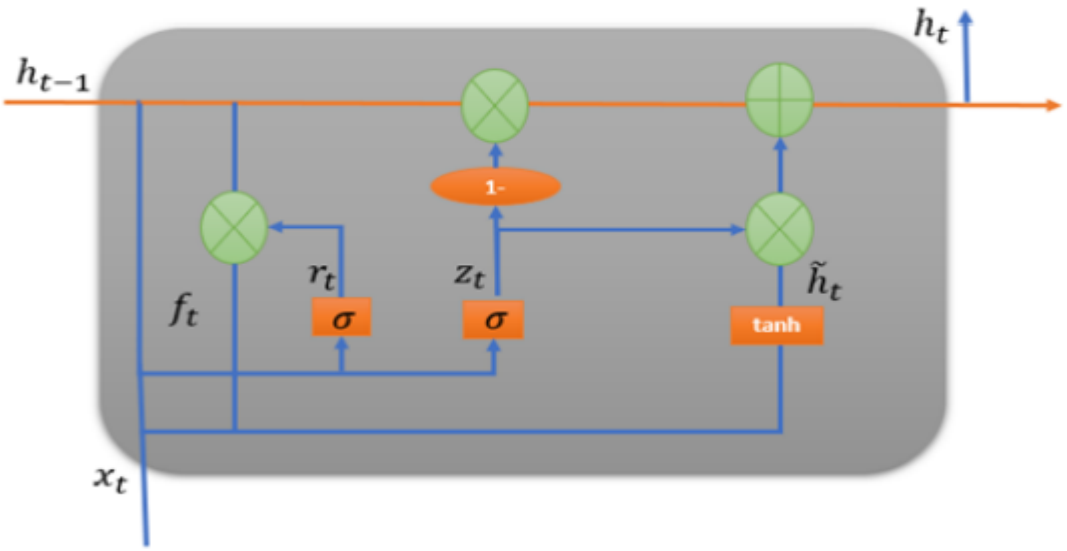


Figure1- 9 : La structure cellulaire d'une GRU

- zt = σ(xtWz + ht−1U z + bz) (7)

  Équation 3-3

- rt = σ(xtWr + ht−1U r + br)(8)

  Équation 3-4

- ˜ ht = tan(rt × ht−1U + xtW + b) (9)

  Équation 3-5

- ht = (1 − zt) × ˜ht + zt × ht−1

  Équation 3-6

où est

- ✔ Wr, Wz et W désignent les matrices de poids des vecteurs d'entrée
- ✔ Uz, Ur et U sont les matrices de pondération pour le pas de temps précédent
- ✔ Représente br, bz et b et un espace où il désigne la fonction logistique sigmoïde
- ✔ rt signifie porte de réinitialisation

✔ Zt fait référence au portail de mise à jour

✔ ht est une classe cachée(Lynn et al., 2019)

Le GRU dispose d'un port de mise à jour et d'un port d'affectation similaire au port d'oubli et d'insertion du module LSTM. Le travail du port de mise à jour est de mettre à jour la quantité d'ancienne mémoire qui doit être conservée, tandis que le port de carte a pour tâche de définir comment la nouvelle entrée est combinée avec l'ancienne mémoire.

Le GRU peut différer en ce sens qu'il affiche l'intégralité du contenu de sa mémoire en utilisant uniquement l'intégration. L'hyper Information est importante pour les modèles de réseau GRU et représente le nombre d'unités cachées dans les couches récursives, la valeur d'abandon et la valeur du taux d'apprentissage.

Là où ces modèles peuvent affecter les performances des modèles neuronaux LSTM et GRU, comme le montrent des études (Reimers & Gurevych, 2017)(Merity et al., 2018)où meilleur est le processus d'apprentissage, meilleurs sont les résultats.

- **Bidirectional GRU**

Un GRU bidirectionnel est une forme spéciale d'un RNN bidirectionnel qui divise un GRU régulier en deux parties, une liée vers l'avant avec des données héritées et l'autre avec une liaison vers l'avant, permettant à la fois aux données héritées d'entrée et aux données futures d'être utilisées simultanément.(Ju et al., 2019)

Cette structure permet d'améliorer efficacement les performances de classification du GRU normal. Elle donne également au GRU bidirectionnel plus de variabilité et de puissance. La figure 1-10 représente la structure bidirectionnelle du GRU.

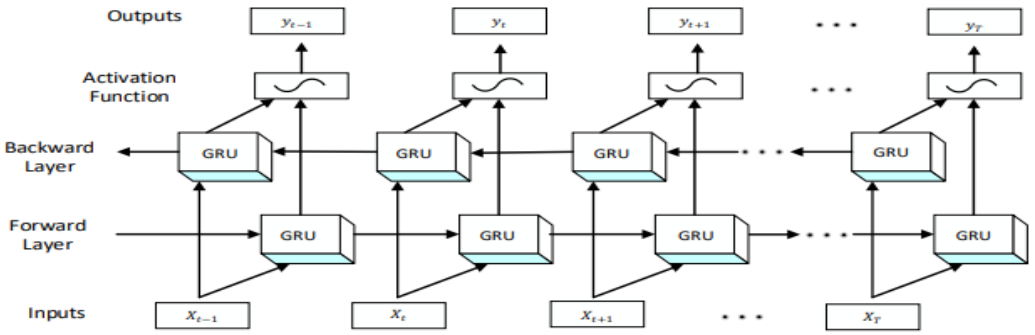


Figure 1- 10 :la structure bidirectionnelle du GRU

Les GRU bidirectionnelle ont leurs propres états car il n'y a pas de connexion directe entre eux.

Comme le cas direct se propage au temps 1-> T, et que le cas inverse se propage à partir du temps T -> 1, et que le cas direct est inconnu au moment t = 0, et le cas inverse est inconnu au moment t = T, donc les deux besoin d'un réglage manuel.

Le processus de formation GRU bidirectionnel est le suivant :

- On calcule d'abord le cas direct le long de la direction 1-> T, et on calcule le cas inverse le long de T -> 1 et enfin on calcule le résultat
- Deuxièmement, nous calculons le gradient de la couleur jaune le long de T->1 et le gradient inverse le long de 1->T.
- Troisièmement, mettez à jour les paramètres du modèle en fonction des valeurs de gradient calculées ci-dessus.(Ju et al., 2019)

1.3 Conclusion

Dans ce premier chapitre, nous avons présenté de manière globale les grands axes du machine learning, quelques définitions, leurs types (Apprentissage supervisé. Apprentissage non supervisé. Apprentissage semi supervisé, d'apprentissage en profondeur), certains des algorithmes utilisés et leurs avantages et inconvénients.

Dans le chapitre suivant, nous expliquerons la signification du faux profil, certaines des techniques qui sont utilisées pour distinguer ces profils, et nous étudierons certains travaux connexes.

Chapitre 2 Détection des faux profils

2.1 Introduction

La vie sociale en ligne est devenue très exigeante pour les gens à mesure que leur vie y est de plus en plus connectée, mais il existe un danger pour ces réseaux car la quantité massive de données personnelles attire des attaquants frauduleux pour voler ces données. Et de fausses nouvelles. De plus, les réseaux sociaux ont fait du monde un petit village, notamment Twitter, Facebook, etc. À ce stade, les chercheurs ont entrepris d'enquêter et de rechercher des techniques efficaces pour détecter les activités anormales et les calculs fantômes. Afin de protéger les comptes du piratage et d'identifier les faux comptes, plusieurs algorithmes ont été utilisés pour résoudre ce problème et éliminer la menace.

2.2 Profil

Les profils peuvent être considérés comme les briques de base du RS. Les profils contiennent généralement des informations démographiques basic sur l'utilisateur tels que son nom, son sexe, sa ville natale et son emplacement actuel…etc. Parallèlement à ces informations personnelles considérées essentielles pour chaque profil, la plupart des réseaux sociaux encouragent également les utilisateurs à écrire une courte biographie sur eux-mêmes et de partager leurs goûts et leurs intérêts. (Madjarov, 2005)

2.3 Faux profils

Un faux profil représente l'identité (profil) de personnes qui prétendent être quelqu'un qu'elles ne sont pas. Selon Facebook, si l'on utilise un compte Facebook autre que son compte principal, c'est faux. Les faux comptes sont généralement créés pour effectuer diverses activités illégales, trompeuses, malveillantes ou discriminatoires sur le réseau, constituant une menace pour le réseau ainsi que pour ses utilisateurs. Le motif derrière les faux profils varie généralement selon le type de réseau pour lequel ils ont été créés. Un nombre croissant de pirates informatiques créent de fausses identités sur des réseaux.

Comme Facebook et Twitter pour accéder aux informations personnelles des utilisateurs, pour soutenir une marque ou une personne en particulier, pour diffamer un utilisateur, etc. Les adversaires peuvent cibler des sites professionnels comme LinkedIn dans le but de suivre l'activité des membres ou gagner la confiance des professionnels. Les attaquants ciblent souvent les sites de rencontres pour profiter des personnes qui recherchent des compagnons à des fins monétaires, des cadeaux ou des informations personnelles (Kadam & Patidar, 2020).

2.4 Les buts de la création d'un faux profil

Exploiter les informations recueillies afin de conduire plusieurs types d'actions, des attaques informatiques : usurpation d'identité, envoi de spams, phishing/spearphishing ou courriel accompagné d'une pièce jointe piégée avec un logiciel malveillant (virus, cheval de Troie…) et en plus pour propager de la désinformation (des rumeurs)(Ghosh & Doshi, 2021).

Ces initiatives sont envisagées par des personnes malveillantes qui cherchent à communiquer avec des salariés travaillant dans des entreprises internationales ciblées, et c'est la première étape de la démarche, notamment depuis l'étranger.

Les salariés ciblés reçoivent des demandes de mise en contact émanant de ces faux profils. Dès l'établissement de la relation sur les réseaux sociaux professionnels, les auteurs de la démarche sont en mesure de cartographier les cercles de relations professionnelles et obtenir de nombreuses informations à des fins utilisables telles que des appels téléphoniques, des adresses e-mail et des sites de communication

Après avoir gagné la confiance de leur nouveau contact, les faussaires pourront exploiter les informations recueillies afin de conduire plusieurs types d'actions, notamment des attaques informatiques (Kadam & Patidar, 2020) :

- usurpation d'identité
- envoi de spams
- Le phishing ou le courrier électronique peut être accompagné d'une pièce jointe malveillante telle qu'un virus ou un chevaux de Troie
- Espionner une personne (un petite amie, un patron, des enfants, un ennemi, …),
- Augmenter le nombre de fans d'une Page Facebook.(Ranjana et al., 2021)
- Spammer des « amis » en toute impunité
- Monter tous types d'escroqueries (très souvent du chantage).
- Porter atteinte à la réputation de personnes ou d'entreprises connues

- **Comment les reconnaître ?**

Il existe des techniques pour repérer ces profils : (M. Meligy et al., 2017)

Technique n°1 : la photo de profil. " Photo de profil Il ne s'agit pas de rechercher pour savoir si la personne qui possède le fichier semble être réelle ou non, mais aussi de savoir si l'image qui a été utilisée en effectuant une recherche inversée dans Google. "

Technique n°2 : l'URL. " Adresse URL On peut observer que parfois les URL qui considèrent les adresses des profils ne correspondent pas au nom et prénom du propriétaire du profil, et cela indique que le compte a peut-être été piraté."

2.5 Travaux connexes

Plusieurs chercheurs ont déjà proposé des solutions pour détecter les faux profils :

La popularité des réseaux sociaux a entraîné différentes problèmes à cause de ces faux comptes qui violent les règles ,c'est pour ça y' a une étude (Erşahin et al., 2017) proposé une méthode de classification pour détecter ces faux comptes sur twitter , en utilisant une technique de discrétisation supervisé EMD (Entropy Minimization discretization) sur les critères numériques, pour discrétiser les données avec la longueur minimal de description (MDL) ,comme critères d'arrêt et la deuxièmes technique naïves bayes pour l'analyse des résultat ,avec cette dernière ils ont augmenté la précession par rapport au autre recherches de 85.5% à 90.41% , en prétraitant seulement en utilisant EMD sur les caractères sélectionnés ,mais cette étude est appliqué seulement sur twitter non pas sur d'autres plateforme.

L'étude faite par Nikita Kadam et Harish Patidar (Kadam & Patidar, 2020) est motivée par l'étude de la plateforme de médias sociaux, et de ses défis en matière de sécurité et de confidentialité, ils sont utilisés la technique d'exploration.

Le LinkedIn est le plus grand réseau social pour établir des relations professionnels mais les faux profils ont un effet négatif sur la fiabilité du réseau, ils sont difficile à identifier c'est pour sa ShalindaAdikaisi et Koushik Dutta (Adikari & Dutta, 2014) ont proposé une combinaison des techniques basée sur NN SVM (noyaux radieux et polynomiaux), et WA avec et sans sélection de caractéristique PCA pour chaque cas. Les objectifs de cette combinaison est de faire différencier entre profils légitimes des faux et en plus ils ont atteint une grande précision 87% avec un taux de vrais négatifs 94%.

D'autre part, le groupe a été séparé en sous-groupes spécifiques d'emplois. Le réseau de neurones NN et le support vectoriel svm ont été utilisés, les caractéristiques de profil ont été utilisées avec PCA, MIB a été utilisé pour leur étude, qui se compose de trois ensembles de données collectées sur Twitter. Trois algorithmes de classification de sélection-formation ont été utilisés pour former la sélection des trois groupes de données et, à la fin, ils ont comparé l'exactitude. Un nouvel algorithme appelé SVM-NN a été développé, dans lequel les valeurs des décisions du modèle d'entraînement et l'utilisation du modèle SVM ont été utilisées. Chaque modèle NN a été formé en utilisant. Ils ont également utilisé un algorithme hybride entre NN et SVM . La machine a été choisie comme algorithme principal car sa précision est meilleure que celle de SVM et celle de NN, qui a une précision d'environ 98%.

D'autre part, ils ont utilisé un algorithme de réseau de neurones profonds et créé un CNN dynamique pour former un modèle d'apprentissage afin de classer un faux profil. L'étude (Wanda & Jie, 2020) se concentre sur la diversité des différentes approches qui consistent en un algorithme d'apprentissage automatique traditionnel et statistique basé sur des règles sur l'apprentissage et la construction d'un nouveau modèle pour détecter un faux profil. Ils ont été collectés et extraits pour créer un ensemble de données contenant l'identité. Les noms complets ont été divisés en deux groupes, un groupe d'entraînement et un groupe de test. Après avoir entré l'ensemble de données dans le modèle Deep Profile, il a montré une grande précision par rapport au modèle SVM et au modèle Naïve Bayes. Le modèle Deep Profile est présenté avec le modèle cnn, qui a une précision de 95,7%.

Cette architecture (Khaled et al., 2019) utilise une stratégie d'agrégation telle qu'une machine vectorielle, une forêt aléatoire et des réseaux de neurones profonds pour fournir des profils d'alias ou de canaux en temps réel aux NMO dans le remplissage d'attributs efficace en temps réel processus en plus de la modification des données, Cela peut inclure le lissage et le regroupement, permettant ainsi la sélection d'informations précieuses pour le cadre, l'algorithme Random Forest, une utilisation multi-méthode qui effectue des tâches de classification et de régression contenant approximativement la même hyperinformation, a été utilisé pour détecter les faux comptes sur les réseaux sociaux situés en ligne, où le réseau contient de nombreuses informations telles que le sexe, les amis, les commentaires, l'éducation, etc. Il a une précision de 96%.

Le modèle a été formé (Ghosh & Doshi, 2021) afin que les comptes puissent être classés comme faux comptes ou comptes réels, et que les faux comptes puissent être identifiés à l'aide de différentes méthodologies au moyen d'un réseau de neurones et d'une machine pour prendre en charge les vecteurs. Les algorithmes de réseau de neurones et de Support Vector Machin ont été utilisés. Les données ont été collectées à partir du réseau Facebook qui comprend le nombre d'abonnés, le nombre de statuts, le nombre d'amis, l'icône, la langue, ainsi que le sexe. Les données ont été mises en forme et alimentées avec de nouvelles données pour obtenir les résultats. Précision de la forme. L'ensemble de données utilisé contient 2818 lignes et 15 colonnes, ainsi que des propriétés. De plus, avant d'utiliser ces données, elles ont été purifiées et les algorithmes ont été entraînés dessus en tant que solution. 98% de précision est obtenue car il a déjà été détecté.

Utiliser des réseaux de neurones artificiels pour identifier la contrefaçon apparence L'apprentissage automatique, en particulier un réseau de neurones artificiels, a été utilisé pour déterminer si la demande d'ami sur Facebook est authentique ou fausse. Ils ont utilisé un ensemble de données obtenu à partir de Facebook. Ce groupe contient plusieurs informations représentées par l'âge, le sexe, le nom et le nombre d'amis. Informations sur les numéros de compte à utiliser, De plus, ils ont utilisé un groupe de bibliothèques représentées dans Pandas, NumPy, MATLAB, theano, scikit-learn, Keras, TensorFlow.Au final, la précision de l'utilisation était de 98%.

Le monde est devenu un petit village grâce au réseau social, mais ces réseaux souffrent d'usurpation d'identité en ligne et de faux comptes, un modèle d'appareil Vector Support et un réseau de neurones profond ont été utilisés pour identifier ces comptes, et un ensemble de données obtenu de Facebook a été utilisé qui contient 2818 lignes et 15 colonnes,. Ce groupe a été nettoyé en supprimant les valeurs en double et les valeurs nulles, afin d'utiliser Vector Support machine, La précision d'apprentissage était de 98%.

Détection de faux profil sur Facebook Dans la culture de l'incident actuel, les réseaux sociaux basés sur le Web sont une fonction essentielle dans la vie de la société, l'utilisation des médias sociaux est devenu pour blesser des individus, les intimider et répandre de fausses nouvelles,a été utilisé Random Forest ,Un ensemble de données disponibles sur Internet a été utilisé pour Facebook, , La précision de la prédiction était d'environ 0,96.

Il y a un développement notable dans les sciences appliquées ces jours-ci... Les téléphones portables se transforment en téléphones intelligents. La technologie est associée aux réseaux

sociaux sur Internet, qui sont apparus sous la forme de Département en présence de chaque individu pour se faire de nouveaux amis et garder des amis, leurs passe-temps sont plus faciles . Mais cette amplification des réseaux sur Internet pose de nombreux problèmes comme la contrefaçon de leur profil. Afin de détecter ces faux fichiers, ils ont utilisé un ensemble de données composé de 1 337 faux clients et 1 448 vrais clients, après avoir divisé l'ensemble de données en un ensemble d'entraînement et de test, Où ils ont appliqué des réseaux de neurones artificiels et la précision du spin est d'environ 98%.

Les communautés de développeurs en ligne comme GitHub offrent des services comme En tant que contrôle de version distribué et gestion des tâches, permettant l'extension d'un nombre Un grand nombre de développeurs pour collaborer en ligne. Cependant , l'ouverture des communautés les rend vulnérables à divers types d'attaques malveillantes, car les attaquants peuvent facilement rejoindre et interagir avec des utilisateurs légitimes . Afin de détecter les comptes malveillants, GitSec est proposé qui est basé sur l'apprentissage en profondeur et qui à son tour consiste à deux réseaux.Vérifiez LSTM (Long Term Memory) et Network PLSTM (Phased LSTM), Un jeu de données GitHub a été utilisé, les données publiques pour chaque utilisateur GitHub se composent d'une partie descriptive et d'une partie dynamique . La partie descriptive indique principalement des informations sur le profil de l'utilisateur et un ensemble de mesures statistiques de ses activités. La partie dynamique couvre les enregistrements exacts des activités, L'ensemble de données contient 59 857 utilisateurs GitHub. Capacité d'intensité de précision 0,940.

Le tableau 2-1 ci-dessous illustre une comparaison entre les travaux connexes

Tableau 2- 1 :un tableau qui résume les travaux connexes

Articles	**Techniques et approches Utilisée**	**Incon/Avant**	**Data set**	**Resultat**	**Reseaux sociaux**
Ersahin 2017	Technique de discrétisation supervisé (nommée Entropy Minimization Discretization EMD L'algorithme naïves bayes.	Ya de perte plus elle est appliquée juste sur plateformes elle n'est pas appliquer sur les autres plateformes.	-501 faux comptes. -499réels comptes.	Précision de naïves bayes Avant la discrétisation :86.1%. Après la discrétisation : 90.9%.	Twitter

Nitika kadam et harish patidar mars 2020	Méthode D'estimation (évaluation les attributs des profiles). Techniques basées sur l'analyse de contenu.	Les techniques sont utiles pour la prise de décision, La reconnaissance des formes et les prédictions. Estimer et classer le niveau de risque d'un profil.	82 profils Twitter sont collectés et analysés.	AUC=0.86%	Twitter
Wanda & Jie, 2020	CNN DNN	- Une grande précision et une perte plus faible. -Combinaison entre plusieurs techniques pour faire différencier entre profils.	1500 authentiques profils données réelles et 1500 faux profils	AUC de CNN =0.94% AUC de DNN =0.9547 %	Les réseaux Sociaux
Shalinda Adikari Kaushik Dutta 2006	NN,SVM (noyaux radiaux et polynomiaux)		sur la base des preuves existantes en ligne et pour identifier ces faux profils, , Ils ont réussi à collecter 34 faux profils sur Linkedin	Précision 87% avec un taux de vraies négatives de 94% (COMPARAISON cette méthode apporte une amélioration d'environ 14% de précision)	Linkedin

Khaled et al, 2019	Un nouvel algorithme appelé SVM-NN a été développé Quatre parties principales : prétraitement des données, réduction des fonctionnalités Classification des données et comparaison précise pour améliorer la précision de la classification	Les inconvénients sont que la théorie ne couvre réellement que la détermination des paramètres pour une valeur donnée des paramètres de régularisation et du noyau et le choix du noyau	L'ensemble de données MIB a été utilisé et il a été divisé en deux groupes Le groupe Réalistes contient 1481 comptes réels Le groupe de faux participants contient 3 351 faux comptes	AUC= 0.983%	Twitter
(Khaled et al., 2019)	L'algorithme de forêt aléatoire	Il effectue des tâches de classification et de régression, est facile à lire et a une grande précision	Les réseaux sociaux sur Internet ont été utilisés	La précision du test était 0.89%	Facebook

(Ghosh & Doshi, 2021)	SVM , Deep Neural Network comme méthode de classification	L'avantage est que ces algorithmes peuvent traiter de grands ensembles de données sans réduire la précision	L'ensemble de données utilisé contient 2 818 lignes et 15 colonnes nombre de VARIABLE S 9 nombre d'observ ations 2818 Cellules manquantes 1749 lignes en double26	La précision était de 98%	Réseaux Sociaux
(Ranjana et al., 2021)	Random Forest	//	Un ensemble de données Facebook été utilisé	AUC=0.96	Facebook
Gong et al , 2019	GitSec est proposé qui est basé sur l'apprentissage en profondeur et qui à son tour consiste à deux réseaux. Vérifiez LSTM (Long Term Memory) et Network PLSTM (Phased LSTM)	GitSec distingue les comptes malveillants des comptes légitimes en fonction des profils de compte ainsi que des caractéristiques d'activité dynamiques	Un jeu de données GitHub a été utilisé, les données publiques pour chaque utilisateur GitHub se composent d'une partie descriptive et d'une partie dynamique.	AUC= 0.940	GitHub
(Shama* et al., 2019)	Le modèle utilisé est celui des réseaux de neurones artificiels(ANN)	//	Un ensemble de données composé de 1337 faux clients et 1481 vrais clients a été utilisé	AUC=98%	Facebook

Le schéma 2-11 suit présente les algorithmes utilisés dans la littérature

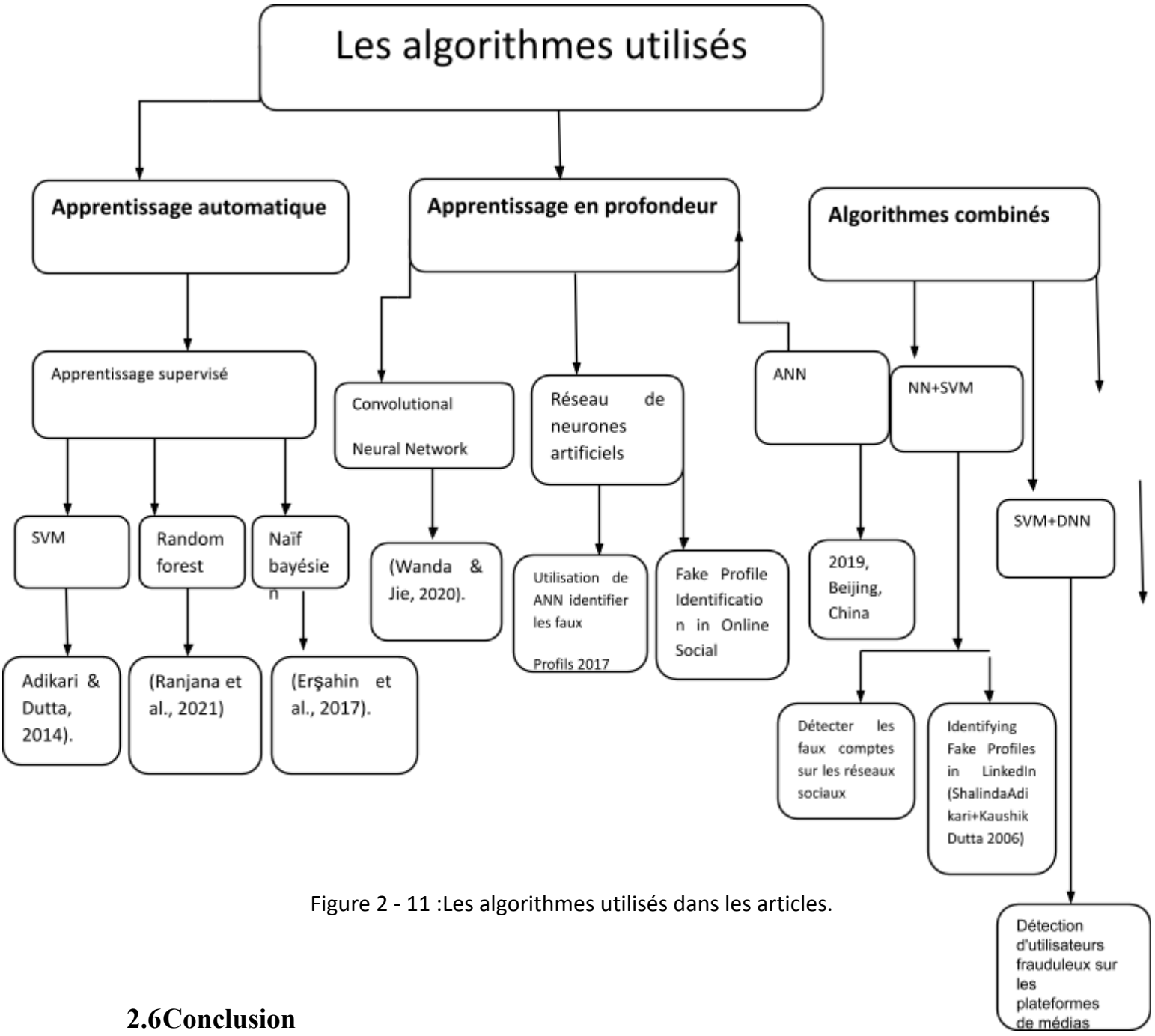


Figure 2 - 11 :Les algorithmes utilisés dans les articles.

2.6Conclusion

Dans ce chapitre, nous avons expliqué le concept de faux profils, profil, et le but de la contrefaçon de profils. De plus, nous avons résumé les travaux scientifiques similaires …..

Dans le chapitre suivant, nous appliquerons Les modèles suivants sont suggérés sur les données collectées sur les réseaux sociaux afin que nous puissions détecter les faux profils et nous vous expliquerons le travail de chaque modèle.

Chapitre 3 La solution proposée

3.1 Introduction

Lors de ce chapitre, on mettra en avant les processus suivis pour réaliser notre objectif la détection des faux profils. D'abord on commence par la collecte des données après qu'elles passent par le prétraitement commence le processus de la partition des données en données d'entrainement et le reste pour le test et par la suite on va évaluer trois modèles avec des architectures différente pour atteindre les résultats

3.2 L'architecture globale de notre solution

L'organigramme 3-12 et le schéma 3-13 montrent la méthodologie de notre projet et les étapes les plus importantes qu'il a traversées. Étant donné que nous avons utilisé deux modèles de l'ensemble de données, où chaque ensemble contient deux parties. En outre, chaque partie a apporté des modifications, telles que la modification des valeurs. Le groupe a également été divisé pour les tests Après cela, nous avons testé un ensemble de modèles. De plus, nous avons organisé les résultats obtenus, les avons comparés et avons choisi le modèle qui donne le meilleur résultat.

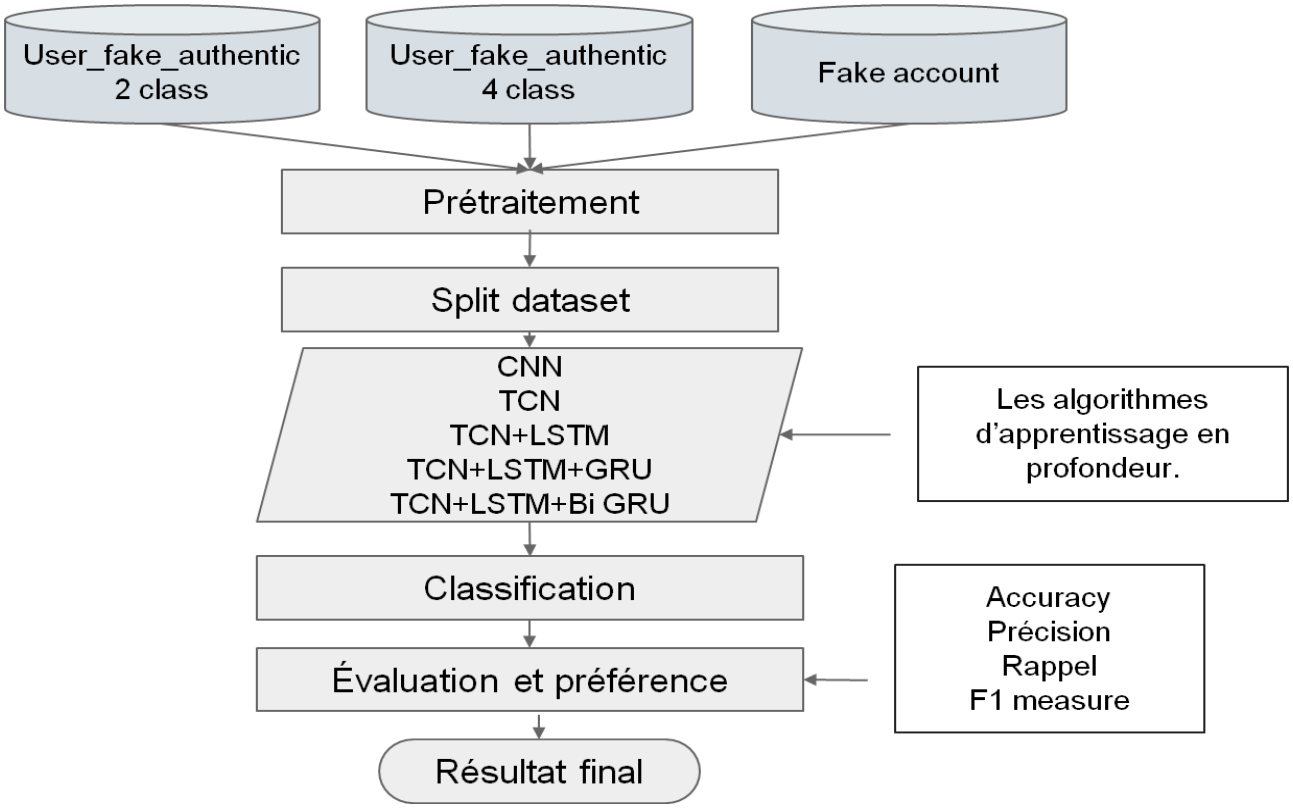
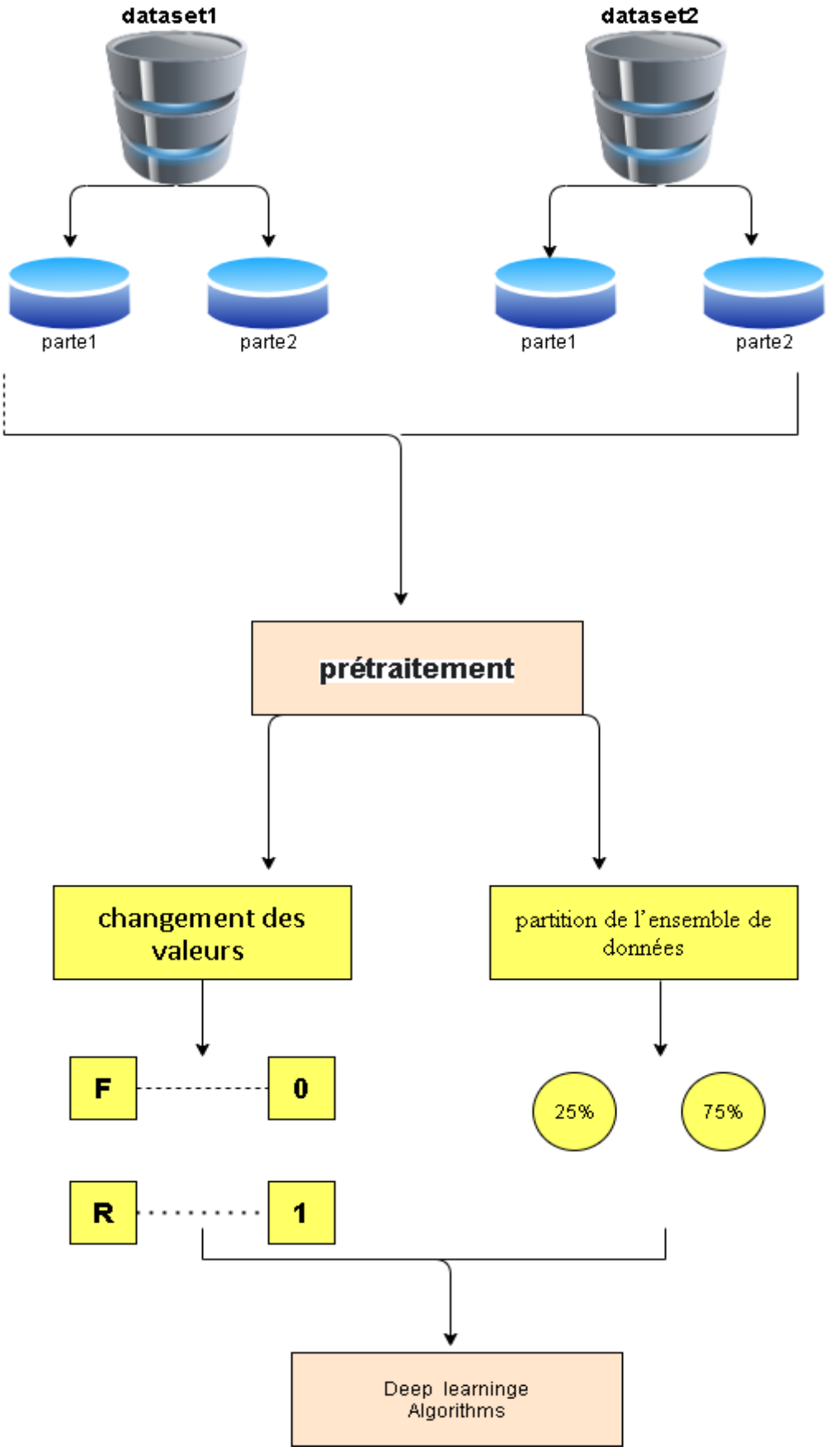


Figure 3- 12 : Organigramme proposé


dataset1
dataset2
parte1
parte2
parte1
parte2
prétraitement
changement des valeurs
partition de l'ensemble de données
F
0
R
1
25%
75%
Deep learninge Algorithms

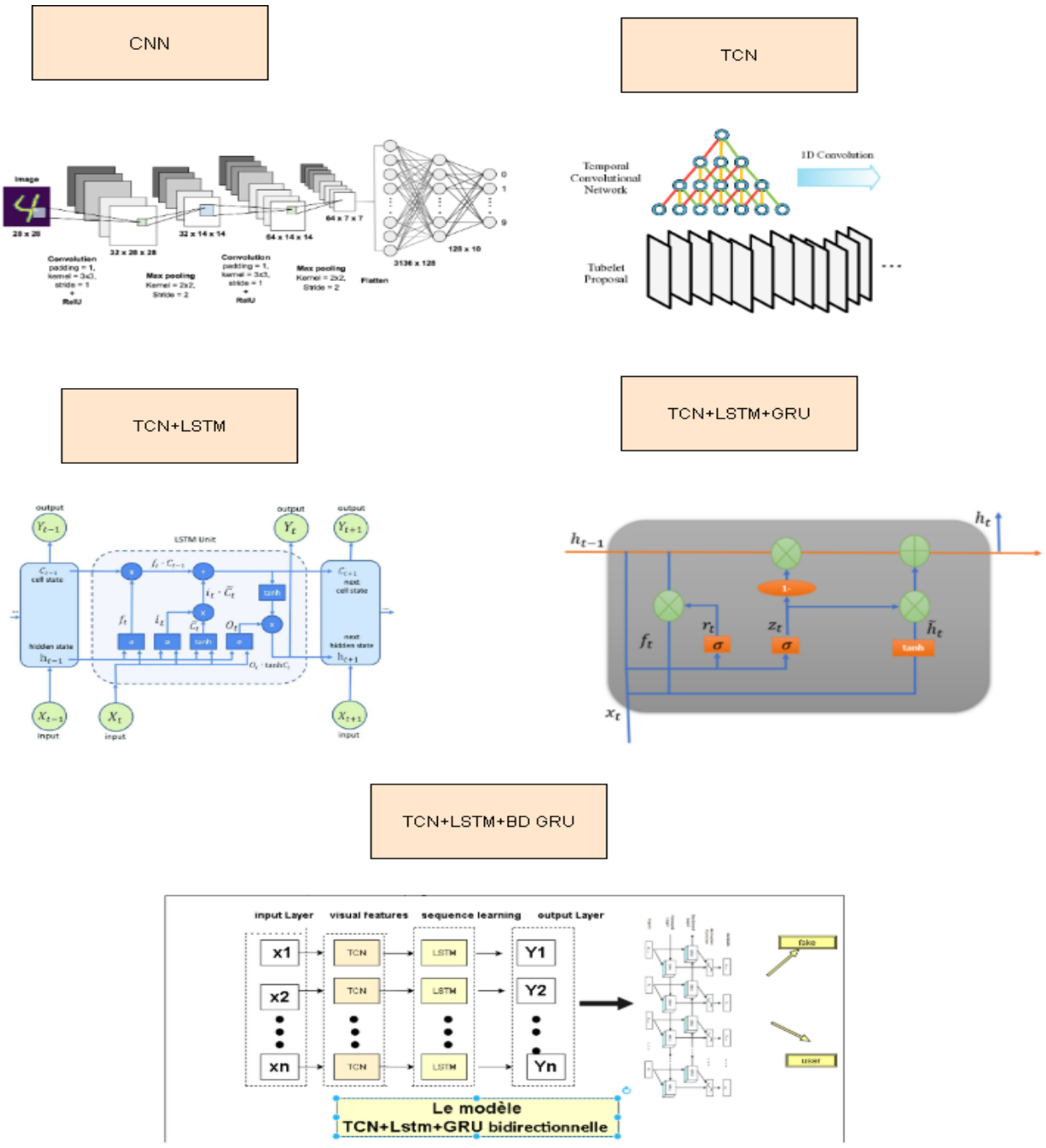


Figure3 - 13 : schéma global de projet

3.3 Base de données

Le premier ensemble de données (Purba et al., 2020) que nous avons utilisé , il contient 2 datasets :

- 2-class User classes: r (real/authentic user), f (fake user / bought followers).

- 4-class User classes: r (authentic/real user), a (active fake user), i (inactive fake user), s (spammer fake user).

Tableau 3 - 2 :dataset

Id	Classification	Number of fake users	Authentic users	Total users	Media (fake)	Media (auth)
1	2-classes	32,869 fake Users	32,460 users	65,329	376,357	460,923
2	4-classes	12,054 a, 10,549 i, 10,263 s	10,441	43,307	376,357	141,371

La liste des fonctions utilisateur utilisées pour le modèle de classificateur est expliquée dans le Tableau 3-2. Il existe cinq sources différentes de fonctions, à savoir les métadonnées, les informations sur les médias, l'engagement, la balise média et la similarité des médias :

Tableau 3 - 3: La liste des fonctions.

	Catégorie	Nom du var	Nom de la fonctionnalité	La description
1	Métadonnées	Pos	Nombre de messages	Nombre total de messages que l'utilisateur a déjà publiés.
2	(M)	Flg	Nombre de suivis	Nombre d'abonnés
3		Flr	Nombre d'abonnés	Nombre d'abonnement
4		Bl	Longueur de la biographie	Longueur (nombre de caractères) de la biographie de l'utilisateur
5		Pic	Disponibilité des photos	Valeur 0 si l'utilisateur n'a pas de photo de profil, ou 1 s'il a
6		Lin	Disponibilité du lien	Valeur 0 si l'utilisateur n'a pas d'URL externe, ou 1 si
7	Infos médias	Cl	Longueur moyenne des sous-titres	Le nombre moyen de caractères des sous-titres dans les médias 8

8	(MI)	Cz	Légende zéro	Pourcentage (0,0 à 1,0) de sous-titres qui ont une longueur presque nulle (<=3)
9		Ni	Pourcentage sans image	Pourcentage (0,0 à 1,0) de médias sans image. Il existe trois types de médias sur une publication Instagram, à savoir l'image, la vidéo, le carrousel
10	Engagement (E)	Erl	Taux d'engagement (J'aime)	Le taux d'engagement (ER) est généralement défini comme (num likes) diviser par (num media) diviser par (num followers)
11		Erc	Taux d'engagement (Comm.)	Semblable à ER comme, mais c'est pour com
12	Balises média	Lr	Pourcentage de balise de localisation	Pourcentage (0,0 à 1,0) de messages tagués avec l'emplacement
13	(MT)	Hc	Nombre moyen de hashtags	Nombre moyen de hashtags utilisés dans une publication
14	Similitude des médias	Pr	Mots clés promotionnels	Utilisation moyenne des mots-clés promotionnels dans le hashtag
15	(MS)	Fo	Mots-clés abonnés	Utilisation moyenne des mots clés de chasseur de followers dans le hashtag, c'est-à-dire {follow, like, folback, folback, f4f}
16		Cz	Similitude cosinus	Similitude cosinus moyenne entre toutes les paires de deux messages qu'un utilisateur a
17		Pi	Intervalle de publication	Intervalle moyen entre les publications (en heures)

Et nous avons utilisé un 2eme ensemble de données (Akyon & Esat Kalfaoglu, 2019)qui est une fusion de deux ensembles de données:

- Fake account detection dataset qui n’est équilibré pas, il contient 1002 comptes réels et 201 faux comptes avec les features suivantes :

Nombre total de médias du compte et d'abonnés du compte.

Suivi du décompte du compte.

Nombre de chiffres présents dans le nom d'utilisateur du compte.

Si le compte est privé ou non (fonctionnalité binaire).

- Automated account detection dataset qui est équilibré, il contient 700 comptes réels et 700 Compte automatisé.

Nombre total de médias des comptes et d'abonnés du compte.

Suite au décompte du compte.

Si le compte a au moins une bobine de surbrillance, ou non (fonctionnalité binaire).

Indique si le compte a une URL externe dans le profil ou non (fonctionnalité binaire).

Nombre de photos sur lesquelles l'utilisateur est tagué par quelqu'un d'autre.

Nombre moyen de hashtag média récent.

3.4 Prétraitement

Après avoir étudié l'ensemble de données, où nous y avons apporté quelques modifications, nous avons changé deux valeurs dans le champ Où nous avons changé une valeur **r** de 1 et une valeur **f** de 0, la figure 3-14 montrent ce changement.

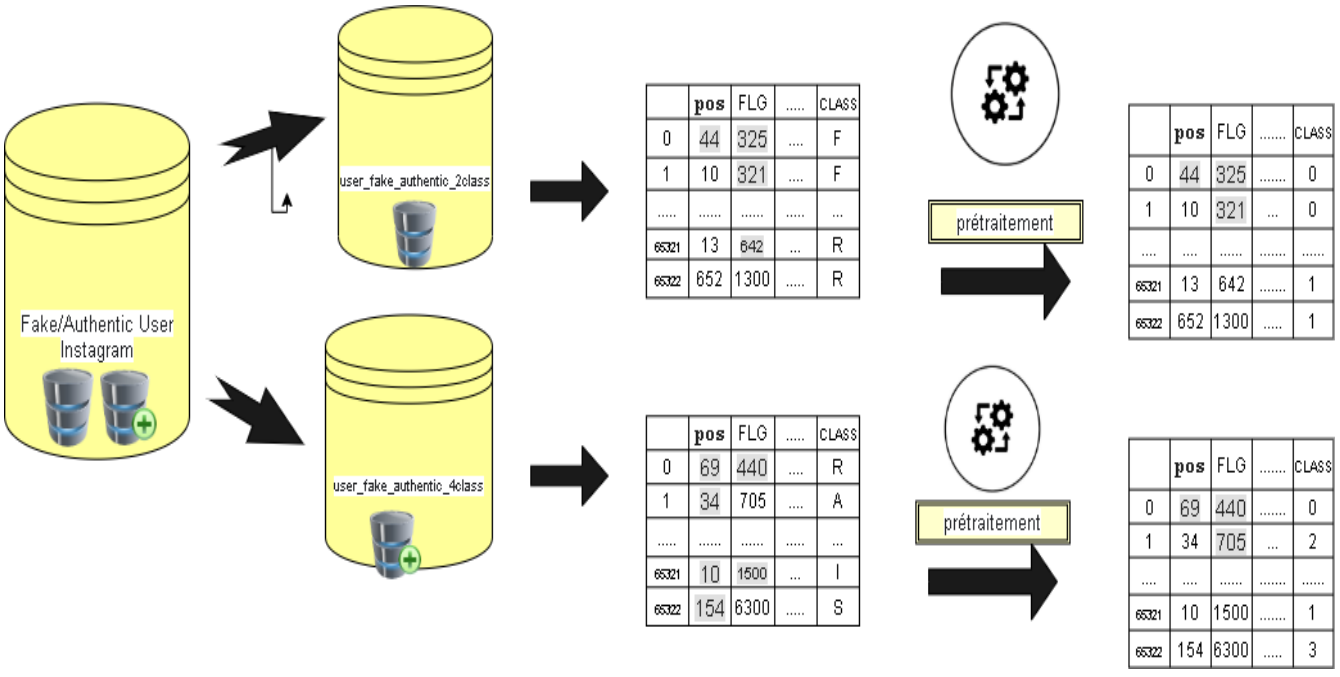


Figure 3 - 14 : changement des valeurs de classe des data de 2 classe et 4 classe.

Pseudo algorithme prétraitement de dataset de 2 classes:

```
Début
ENTRÉE : dataset<-profiles
SORTIE : newdataset
Pour chaque profil
Si c==F alors remplacer F  par 1
Sinon remplacer R par 0
Finsi;
Fin pour;
Retourner newdataset ;
Fin.
```

Pseudo algorithme prétraitement de dataset de 4 classes:

```
Début
ENTRÉE : dataset<-profiles
SORTIE : newdataset
Pour chaque profil
 Si c==R alors remplacer R  par 0
Sinon
    Si c==I alors remplacer I par 1
     Sinon
       Si c==S alors remplacer S  par 2
        Sinon
          Si c==A alors remplacer A  par 3
          Finsi
       Finsi
      Finsi
   Finsi
Fin pour;
Retourner newdataset
Fin.
```

En termes de division, nous avons divisé les deux ensembles de données (2 classes et 4 classes) en deux groupes un groupe de training qui vaut 75% et un groupe test qui vaut 25% et le dernier ensemble de données (dataset fake user detection) est divisé en 70% training et 30% test, La figure 3-15 illustre cela.

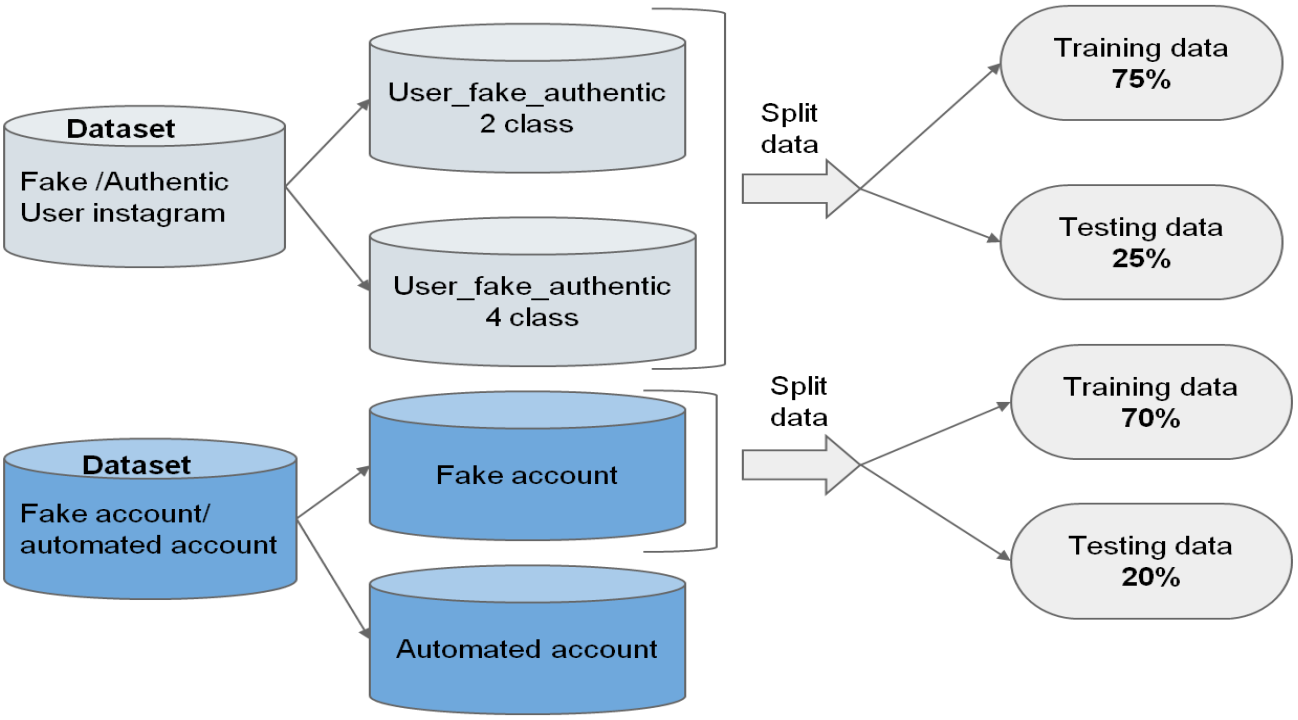


Figure 3- 15: partition de l'ensembe de données

Pseudo algorithme de partition de l'ensemble de données :

Début

ENTRÉE : dataset (Fake/Authentic User Instagram).

SORTIE: training data, test data

D: dataset;
Function splitting_data (D): training data, test data
Retourner (training data, test data)
Fin.

3.5 Le modèle CNN

Les informations sont appelées noyau ou filtre. Le noyau se déplace avec une foulée spécifique et chaque fois qu'une opération de multiplication matricielle est effectuée entre le noyau et une partie du vecteur d'entrée qui se superpose au noyau. Cela réduit non seulement la puissance de calcul nécessaire pour traiter les données, mais est également utile pour extraire les caractéristiques clés qui sont positionnées invariant et contribuer à une formation efficace du modèle.

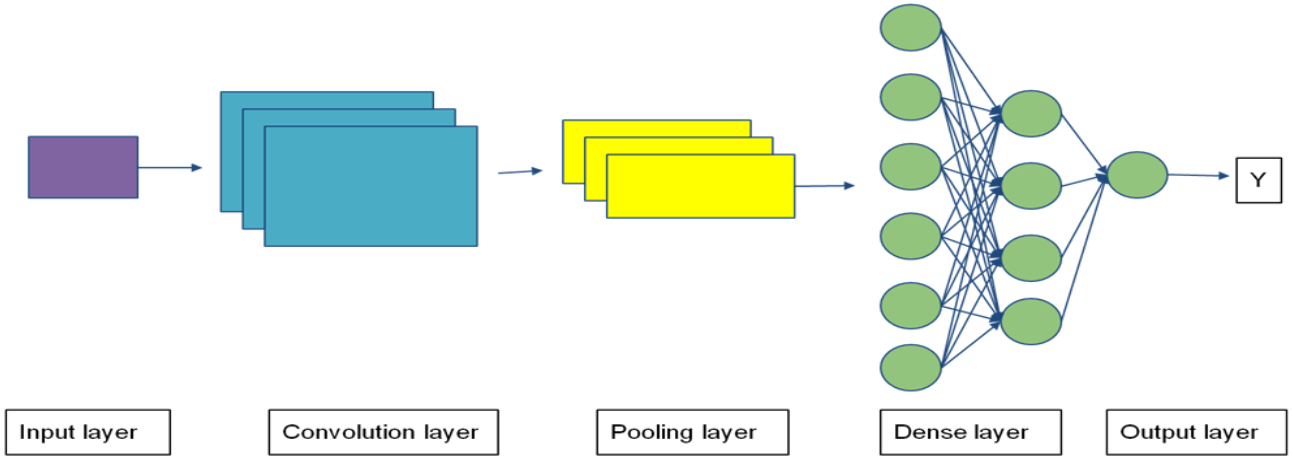


Figure 3- 16:Le réseau de neurones convolutifs

Dans cette étude, des couches convolutives, des couches de regroupement, des couches d'abandon et des couches denses ont été utilisées(convolutional layers, pooling layers, drop-out layers, and dense layers). Le réseau neuronal utilisait principalement des couches convolutionnelles pour la formation. L'architecture de CNN pour classer le texte .

Les paramètres du réseau sont repris par les couches entièrement connectées. Les caractéristiques d'entrée sont extraites par la couche de convolution et la matrice de convolution résultante est obtenue à partir du regroupement.

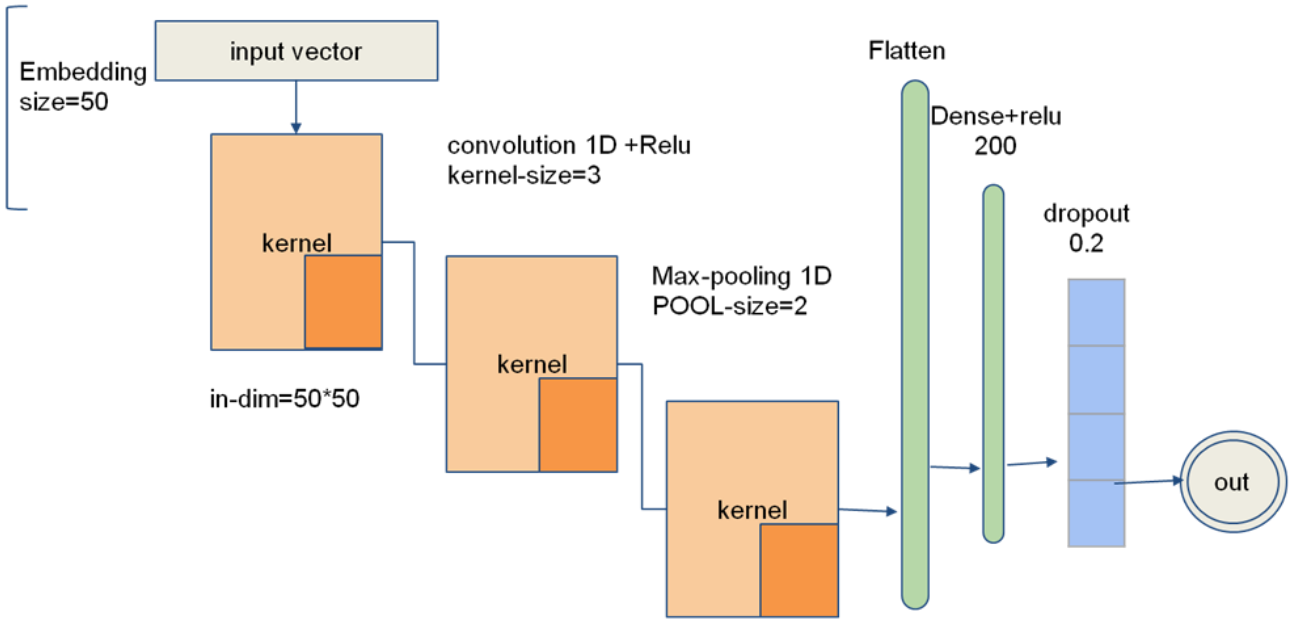


Figure 3- 17 : Couches et paramètres utilisés dans le réseau de neurones CNN

Le problème de sur-ajustement est résolu par la couche d'abandon et, par conséquent, lors de la formation, Nous avons présenté l'application de CNN + Maxpool pour améliorer les performances de classification. Pour la couche de convolution, nous avons utilisé une taille de noyau 3 × 3 avec une foulée de 1 suivie d'une couche Max-pooling avec une taille de noyau 2 × 2. Enfin, les caractéristiques alambiquées sont introduites dans une couche dense. Étant donné que les paramètres du réseau sont proches de ~ 1 M, nous avons utilisé une couche "abandon" avec une "probabilité de conservation" de 0,2 pour éviter de surajuster les données d'apprentissage.

- **Couche convolutive (Convolutional Layer)**

La couche convolutive contient plusieurs noyaux de convolution. Chaque noyau de convolution effectue des opérations de convolution sur la zone partielle des données d'entrée, puis ajouté un terme de biais et obtient les données de sortie après la fonction d'activation. L'opération de convolution peut aider à extraire différentes caractéristiques de l'image d'entrée telles que les bords ou les lignes, etc.,

- **Couche de regroupement (pooling Layer)**

La couche de regroupement est généralement définie après la couche convolutive. La couche de regroupement peut conserver les caractéristiques de sortie de la couche convolutive tout en réduisant la taille des données de sortie, réduisant ainsi le calcul global du CNN. La méthode la plus couramment utilisée de la couche de mise en commun est la mise en commun maximale, qui utilise la plus grande valeur dans les données de zone comme sortie représentative. pseudo CODE de modèle CNN

ENTRÉE : données (Fake/Authentic User Instagram).

SORTIE : Faux(F)/Utilisateur(U).

Pour chaque classe dans data {

*Conv1D 3*3*

*MaxPool1D 2*2*

Dropout (0.2)

Dense(200)

Dense + sigmoïde

SORTIE : Faux(F)/Utilisateur(U).

Fin pour

Évaluer la précision, le rappel, le score F et la matrice de confusion

3.6 Le modèle TCN

Le réseau convolutif temporel fait partie des réseaux de neurones artificiels qui ont connu des domaines scientifiques très avancés, où l'on assiste à plusieurs nouvelles applications de ce réseau. Notez que le réseau convolutif temporel est considéré comme un modèle de réseau de neurones car il utilise des convolutions et des expansions transversales afin de l'adapter à un grand ensemble de données avec ses champs temporels et larges, et la structure du réseau convolutif temporel a été proposée par (Bai et al., 2018).

3.6.1 Convolutions causales

TCN a deux avantages :

1- Convolutions causales, ce qui signifie que la sortie se rapporte uniquement aux entrées actuelles et historiques et non aux entrées futures.

2- L'architecture peut prendre une série temporelle de n'importe quelle longueur et la tracer avec les données de sortie de la même longueur que le fait rnn, La première couche de TCN est un réseau convolutif unidimensionnel complet r où chaque couche doit avoir la même taille que la couche d'entrée. Où les canaux de taille nulle sont ajoutés pour que les canapés suivants soient de la même taille que les canapés précédents. Pour la deuxième caractéristique, TCN joint les entrées de temps actuelles et précédentes afin d'obtenir un volume d'historique efficace et long. (Lara-Benítez et al., 2020) la figure3-18 suivant représente l'architecture du modèle TCN.

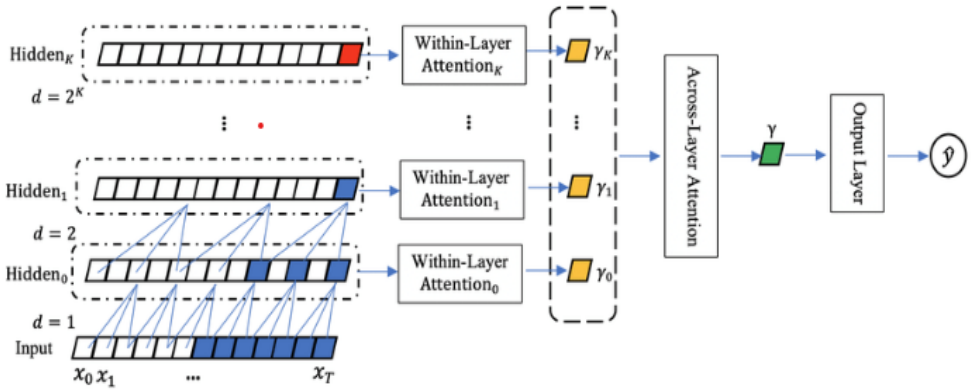


Figure 3 - 18 : architecture de modèle tcn (Lara-Benítez et al., 2020)

3.6.2 Convolutions dilatées

Il convient d'utiliser des Convolutions dilatées qui permettent d'obtenir un champ qui reçoit une taille exponentielle, afin d'appliquer la convolution causale à des séries temporelles avec une longue histoire (Ju et al., 2019).Car la convolution dilatée permet d'augmenter la taille des séries temporelles révélées par le réseau, en ajoutant des poids au noyau de convolution. L'opération de convolution dilatées F sur les éléments de série temporelle unidimensionnels x et le filtre f est défini comme suit

$$\mathbf{F(s) = (x * f)(s) = Xk - 1i = 0f(i)xs - di}$$

Équation 3-1

où k est la taille du filtre, d est le facteur d'expansion et s - d est la direction du passé.

Par conséquent, la dilatation est égale à l'utilisation du volume des lignes fixes entre deux prises de filtre adjacentes.Lorsque d = 1, la torsion étirée est réduite à une chaîne uniforme.

TCN a deux façons d'étendre le champ récepteur en augmentant le facteur d'expansion ou en sélectionnant des tailles de filtre plus grandes K. (Ju et al., 2019)

3.6.3 Connexions résiduelles

Afin de résoudre le problème de la détérioration des performances d'un réseau neuronal convolutif lorsque le nombre de ses couches est trop profond, le réseau restant est utilisé, car le chevauchement de la convolution causale et de la convolution en expansion rend le nombre de couches du réseau neuronal progressivement plus profond. (Lara-Benítez et al., 2020)

Les connexions restantes sont insérées dans la couche de sortie de TCN, la couche de sortie de TCN, l'entrée X et la sortie du réseau convolutif sont combinées comme suit :

$$\mathbf{o = Activation(x + F(x))}$$

Équation 3-2

- où **F(x)** est le produit de la couche convolutionnelle
- L'activation est la fonction de simplification

Les connexions résiduelles permettent effectivement à la couche d'apprendre à modifier et à définir l'identité plutôt que de basculer entièrement.

TCN contient deux convolutions qui étendent la cause et l'indifférence linéaire, de sorte que la fonction d'unité linéaire corrigée (relu) est utilisée comme fonction de simplification.

Une couche de restauration est également ajoutée après chaque enveloppement prolongé pour éviter une déformation excessive.

Le réseau TCN se caractérise par son architecture unique qui le rend adapté aux tâches de séquencement, qui sont représentées dans :

1) **Parallélisme** : les convolutions peuvent être calculées en parallèle car le même filtre est utilisé dans chaque couche. Ainsi, pendant l'apprentissage, une longue séquence d'entrée peut être traitée dans son ensemble plutôt que séquentiellement. (Lara-Benítez et al., 2020)

2) **Taille flexible du champ récepteur** : La taille du champ récepteur peut être ajustée de plusieurs manières, par exemple, en tassant plus largement les couches convolutives en utilisant des facteurs d'expansion ou en augmentant la taille filtrée, permettant ainsi à TCN de mieux contrôler le taille de la mémoire du modèle. (Lara-Benítez et al., 2020)

3) **Faible besoin en mémoire pour la formation** : dans TCN, les filtres sont partagés sur une seule couche avec un chemin de rétropropation uniquement vers la profondeur du réseau, contrairement à LSTM et GRU, ils peuvent utiliser beaucoup de mémoire pour stocker les résultats partiels de leurs multiples portes de cellules. (Lara-Benítez et al., 2020) c'est le pseudo CODE de modèle TCN

ENTRÉE : données (Fake/Authentic User Instagram).

SORTIE : Faux(F)/Utilisateur(U).

Début
time_steps=17 , input_dim=17
Pour chaque profil
Appliquer TCN
tcn_layer = TCN(input_shape=(time_steps, input_dim))
Dense (1, activation='sigmoid ')
Fin pour
Pour chaque époque
Evaluer accuracy, Evaluer loss, Evaluer validation accuracy, Evaluer validation loss;
Fin pour
Evaluer précision, rappel, F1 mesure, Matrice de confusion ;
Retourner validation accuracy, validation loss, précision, rappel, F1 mesure, Matrice de confusion
Fin

Combinaison entre LSTM et TCN

Nous avons proposé une approche combinant TCN et LSTM (un type de RNN), pour des performances de classification plus élevées afin de prédire les faux utilisateurs.

Le TCN utilise des techniques telles que plusieurs couches de convolutions dilatées et de remplissage de séquences d'entrée afin de gérer différentes longueurs de séquence et de détecter les dépendances entre des éléments qui ne sont pas côte à côte.

Au début nous avons appliqué directement le TCN sur l'ensemble de données (les profils) après nous avons appliqué sur les outputs de TCN l'algorithme LSTM. Grâce à cette combinaison la perte était minimisée pour une modélisation et une prédiction efficace des données. le pseudo ci dessous et la figure 3-19 représente la Combinaison entre LSTM et TCN:

```
ENTRÉE : données (Fake/Authentic User Instagram).
SORTIE : Faux(F)/Utilisateur(U).
Début
        time_steps=17, input_dim=17
        Pour chaque profil
    Appliquer TCN
tcn_layer = TCN(input_shape=(time_steps, input_dim))
    Appliquer LSTM
lstm_layer = LSTM(input_shape=(time_steps, input_dim))
    Dense (1, activation='sigmoid ')
        Fin pour
        Pour chaque epoch
    Evaluer accuracy, Evaluer loss, Evaluer validation accuracy,
Evaluer validation loss;
        Fin pour
    Evaluer précision, rappel, F1 mesure, Matrice de confusion ;
Output : faux profil, authentique profil
Fin
```

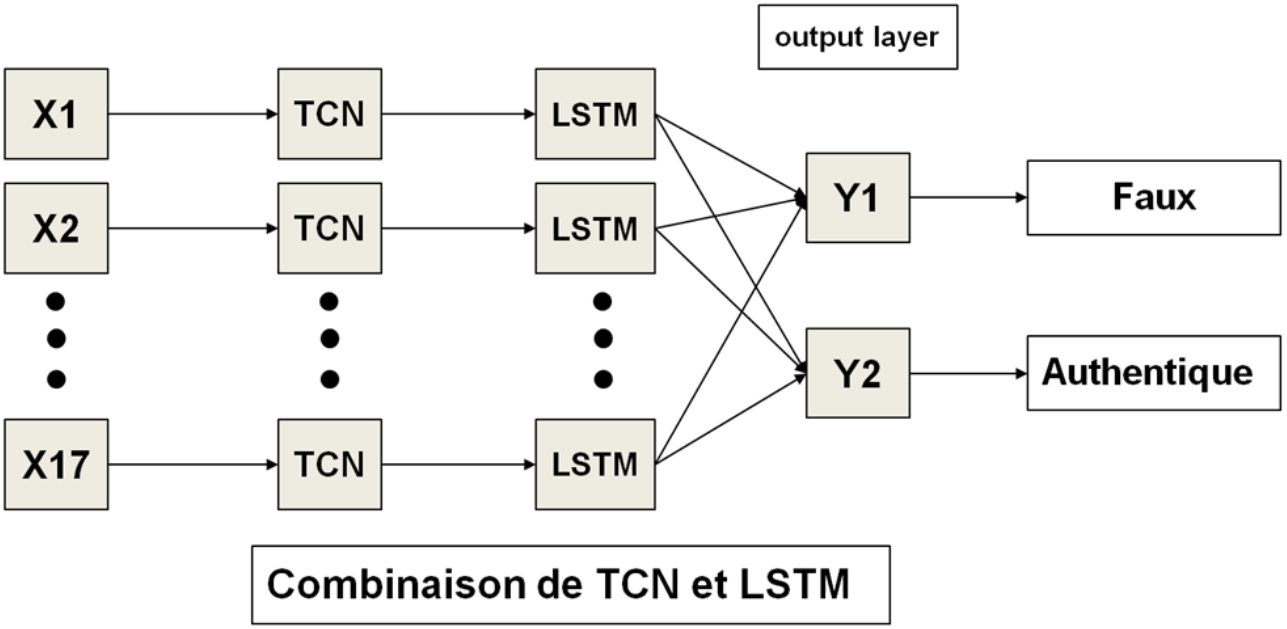


Figure 3- 19:combinaison entret TCN et LSTM.

Nous avons appliqué les 3 modèles TCN+LSTM+GRU comme le montre le pseudo algorithme ci dessous et la figure 3-20 (la combinaison des 3 modèles entre LSTM et TCN st GRU) :

ENTRÉE : données (Fake/Authentic User Instagram).
SORTIE : Faux(F)/Utilisateur(U).
Début
time_steps=17 , input_dim=17
Pour chaque profil
Appliquer TCN
tcn_layer = TCN(input_shape=(time_steps, input_dim))
Appliquer LSTM
lstm_layer = LSTM(input_shape=(time_steps, input_dim))
Appliquer GRU
gru_layer = GRU(input_shape=(time_steps, input_dim))
Dense (1, activation='sigmoid ')
Fin pour
Pour chaque epoch
Evaluer accuracy, Evaluer loss, Evaluer validation accuracy, Evaluer validation loss;
Fin pour
Evaluer précision, rappel, F1 mesure, Matrice de confusion ;
retourner Faux(F)/Utilisateur(U).
Fin

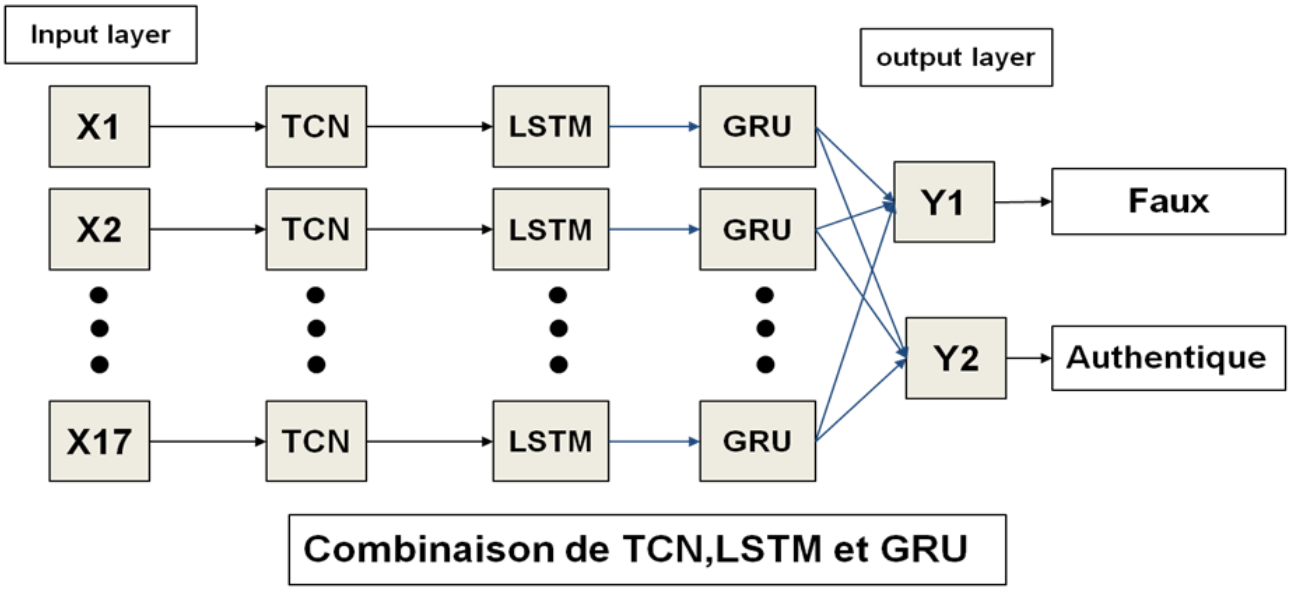


Figure 3 - 20 :combinaison entre TCN et Lstm et GRU.

Nous avons remplacé le modèle GRU avec le médele Bi GRU comme le montre le pseudo algorithme ci dessous (la combinaison des 3 modèles entre LSTM et TCN et Bi GRU) et la figure 3-21 représente la combinaison des 3 modèles:

ENTRÉE : données (Fake/Authentic User Instagram).
SORTIE : Faux(F)/Utilisateur(U).
time_steps=17 , input_dim=17
Début
Pour chaque profil
Appliquer TCN
tcn_layer = TCN(input_shape=(time_steps, input_dim))
Appliquer LSTM
lstm_layer = LSTM(input_shape=(time_steps, input_dim))
Appliquer Bidirectional GRU
Bi_gru_layer = Bidirectional GRU(input_shape=(time_steps, input_dim))
Dense (1, activation='sigmoid ')
Fin pour
Pour chaque epoch
Evaluer accuracy, Evaluer loss, Evaluer validation accuracy, Evaluer validation loss;
Fin pour
Evaluer précision, rappel, F1 mesure, Matrice de confusion ;
retourner Faux(F)/Utilisateur(U).
Fin

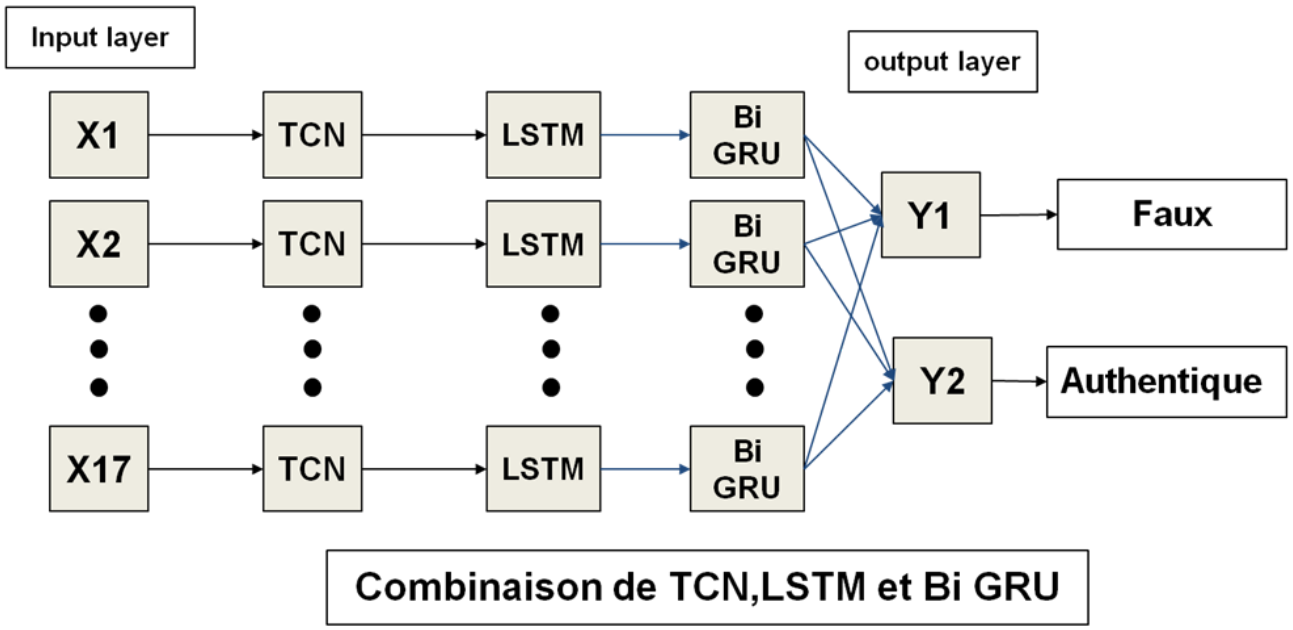


Figure 3- 21 :combinaison entre TCN et Lstm et Bi GRU.

La dernière étape de notre travail est une étape indispensable dans la validation des caractéristiques qui mènent à donner un meilleur algorithme.

3.7 Conclusion

Dans ce chapitre, nous avons proposé un ensemble de modèles afin de détecter les faux comptes. Nous avons appliqué ces modèles à deux ensembles de données, et nous avons également expliqué certaines des modifications que nous avons apportées à l'ensemble de données utilisé.

Dans ce qui suit, nous présenterons les tests et les résultats obtenus de chaque modèle. Nous verrons les résultats obtenus et les comparerons entre eux, afin de valider la solution que nous avons proposée.

Chapitre 4 Test et résultats

4.1 Introduction

Dans ce chapitre, nous présenterons les outils de développement adoptés et le langage de programmation utilisé ainsi que différentes librairies utilisées, parmi elles keras et tensorflow qui sont les deux importantes bibliothèques déployées pour la réalisation de nos modèles et les mesures d'évaluation utilisées.

Par la suite, nous discutons et comparons les résultats obtenus de nos modèles afin de sélectionner le meilleur modèle et pour atteindre notre objectif qui est la détection des faux profils.

4.2 Choix techniques

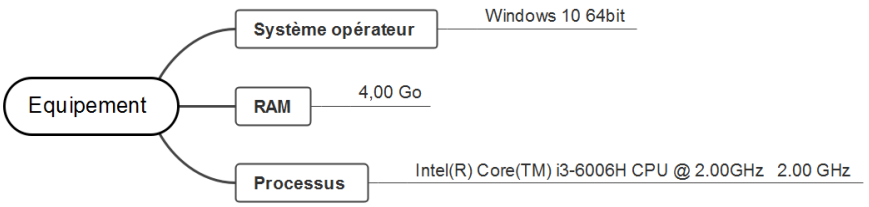
Dans cette partie, nous présenterons le matériel sur lequel nous avons développé notre modèle, les différents outils utilisés.

4.2.1 Environnement du développement

Google Collaboratory ou Google Colab en bref. Google Colab (Gunawan et al., 2020) est un environnement de bloc-notes Jupiter gratuit qui ne nécessite aucune configuration et s'exécute entièrement dans le cloud. Avec Google Colab, vous pouvez écrire et exécuter du deep learning and Machine learning codes, enregistrer et partager vos analyses et accéder à de puissantes ressources informatiques, le tout gratuitement depuis votre navigateur.

4.2.2 Equipment

Durant les différentes étapes d'implémentations, d'entraînements et de tests de nos modèles, nous avons exploité les ressources matérielles de notre station personnelle qui possède les spécifications suivantes :



4.3 Le langage de programmation utilisé

Python (Rossum & Boer, 1991)est un langage de programmation puissant et il est idéal pour les scripts et le développement rapide d'applications car il a une syntaxe élégante, un typage dynamique et interprétabilité.

Il s'agit aussi d'une structure de données de haut niveau efficace et d'une approche efficace de la programmation orientée objet.

4.3.1 Les bibliothèques utilisées

Dans notre travail, nous avons utilisées les bibliothèques suivantes :

Pandas

Pandas (Vo.T.H & Czygan, 2015) est un package Python qui prend en charge des structures de données riches et des fonctions d'analyse de données,

Il se concentre sur l'amélioration des bibliothèques de données de Python et c'est un outil idéal pour les systèmes qui ont besoin de structures de données complexes ou de fonctions de séries chronologiques hautes performances telles que les applications d'analyse de données financières.

TensorFlow

La bibliothèque TensorFlow (Dillon et al., 2017)est un outil open source a été développé par l'équipe Google et lancé en 2015 pour aider les développeurs à concevoir, créer et entraîner les modèles et pour le calcul numérique qui rend l'apprentissage automatique et le développement de réseaux de neurones plus rapides et plus faciles.

Tensorflow est une des frameworks les plus utilisés pour le deep learning.

Numpy

NumPy (Travis, 2007) est l'abréviation de «Numerical Python» et c'est un ensemble fondamental pour le calcul scientifique en Python. NumPy fournit à Python une vaste bibliothèque mathématique capable d'effectuer des calculs numériques de manière efficace et efficiente afin de pouvoir travailler avec des tableaux multidimensionnels et des structures de données matricielles, très courante dans les domaines de la science des données et de l'apprentissage automatique.

Scikit-learn

Le scikit-learn (Buitinck et al., 2013) est une bibliothèque open source d'apprentissage automatique utilisant le langage de programmation Python. Il prend en charge divers modèles d'apprentissage automatique, tels que les algorithmes de classification, de régression et de clustering, interopérables avec les bibliothèques numériques et scientifiques Python NumPy et SciPy.

Matplotlib

Matplotlib (Tosi, 2009)est un package Python pour la visualisation de données. Il permet de créer facilement divers tracés, y compris des tracés linéaires, dispersés, à barres, en boîte et radiaux, avec une grande flexibilité pour un style raffiné et des annotations personnalisées. Le module polyvalent BSUJTU permet aux développeurs de définir pratiquement n'importe quel type de visualisation. Pour une utilisation régulière, Matplotlib propose une interface orientée objet simpliste, le module QZQMPU, pour un traçage facile. Autre la génération de graphiques statiques, Matplotlib prend également en charge une interface interactive qui non seulement aide à créer une grande variété de tracés, mais est également très utile pour créer des applications Web.

Keras

Keras (Ketkar, 2017) est une bibliothèque qui fournit des blocs de construction très puissants et abstraits pour construire des réseaux d'apprentissage en profondeur. Les blocs de construction fournis par Keras sont construits à l'aide de Theano (couvert précédemment) ainsi que de TensorFlow (qui est une alternative à Theano pour la construction de graphiques de calcul, la dérivation automatique de gradients, etc.). Keras prend en charge les calculs CPU et GPU et est un excellent outil pour prototyper rapidement des idées.

La figure 4-22 suivant résume les bibliothèques avec leur utilité :

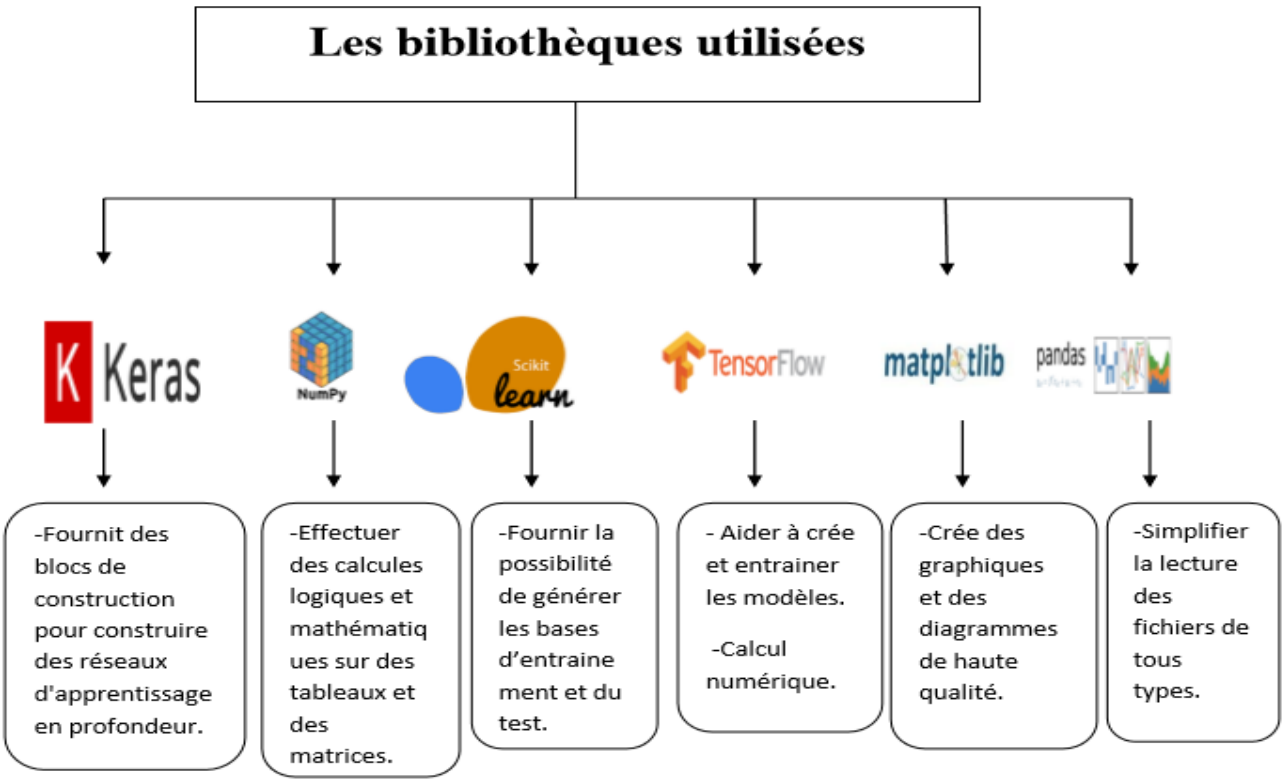


Figure 4 - 22 : schéma des bibliothèques et leur utilité

4.4 Les mesures d'évaluation

Pour calculer les performances sur les algorithmes implémentés, nous avons utilisé des métriques de recherche d'informations extraites de la matrice de confusion.

La matrice de confusion (Townsend, 1971)est un outil de mesure de performance des modèles de classification à 2 classes ou plus. Elle permet également de visualiser des paramètres tels que l'accuracy, F1-score, précision et le rappel.

Tableau 4- 4 : La matrice de confusion

		Classe prédite	
		Authentique profil (Classe r, 0)	Faux profil (Classe f, 1)
Classe réelle	Authentique profil (classe r, 0)	VP	FN
	Faux profil (Classe f, 1)	FP	VN

Tel que :

- VP Vrai Positif : les cas dans lesquels le modèle a prédit que c'est un profil authentique et la sortie réelle étaient également vrai un profil authentique.
- VN Vrai Négatif : les cas dans lesquels le modèle a prédit que c'est un faux profil et la sortie réelle étaient également un faux profil.
- FP Faux Positif : les cas dans lesquels le modèle a prédit que c'est un profil authentique et la sortie réelle étaient un faux profil.
- FN Faux Négatif : les cas dans lesquels le modèle a prédit que c'est faux profil et la sortie réelle étaient un profil authentique.

Accuracy (Kovatchev et al., 2004) est une mesure qui permet d'évaluer les modèles de classification et L'équation 1-4 représente la formule de calcul de cette mesure :

$$ACCURACY = \frac{Vrai\ Positif \pm Vrai\ Négatif}{Vrai\ Positif \pm Vrai\ Négatif \pm Faux\ Positif \pm Faux\ Négatif}$$

Équation4-1 : Equation de l'accuracy

Précision (Laloë, 1993) désigne la proportion des prédictions correctes effectuées par le modèle et elle est la mesure utilisée pour déterminer quel modèle est le meilleur pour détecter les faux profils. L'équation 4-2 représente la formule de calcul de cette mesure :

$$Précision = \frac{Vrai\ Positif}{Vrai\ Positif \pm Faux\ positif}$$

Équation 4-2 :Equation de précision

Rappel (Recall) (Lehnert et al., 1979) est calculé comme le rapport du nombre d'échantillons positifs correctement classés comme positifs au nombre total d'échantillons positifs. L'équation 4-3 représente la formule de calcul Le rappel:

$$Rappel = \frac{Vrai\ Positif}{Vrai\ Positif\ \pm Faux\ Négatif}$$

Équation 4-3 : Équation de rappel

F1 mesure Moyenne harmonique de la précision et du rappel (Sasaki, 2007). On calcule F1 mesure avec la formule de L'équation 4-4 :

$$F1\ mesure\ =\ 2\ *\ \frac{précision*rappel}{précision\pm rappel}$$

Équation 4-4 :Equation de F1 mesure

4.5 Paramètre de compilation

Fonction perte (loss function)

Perte est une fonction (Bottou, 2012)utilisée pour évaluer dans quelles mesure l'algorithme modélise l'ensemble de données donc si la valeur obtenue de cette fonction est plus élevée alors les prédictions sont totalement erronées sinon elles sont assez bonnes.

Il existe plusieurs fonctions de perte :

Binary cross entropy (Rosasco et al., 2004) est utilisée pour mesurer l'erreur dans les tâches de classification binaire c'est-à-dire lorsqu'il n'y a que deux classes d'étiquettes (exemple : oui ou non, 1 ou 0 et comme notre cas faux ou authentique profil).

Categorical cross entropy (Koidl et al, 2013) est utilisé dans le modèle de classification multi classe où il y a plusieurs classes d'étiquettes (comme notre cas de 4 classes).

Fonction d'activation

Une fonction d'activation (Sharma et al, 2020) décide si un neurone doit être activé ou non et elle dérive la sortie d'un ensemble de valeurs d'entrée transmises à un nœud (ou une couche).

La fonction d'activation sigmoïde (Ito, 1991)est l'une des fonctions d'activation utilisées dans l'apprentissage en profondeur qu'elle permet de contrôler la sortie d'un réseau de neurones.

Elle est placée comme dernière couche d'un modèle d'apprentissage automatique et elle peut être utilisée pour convertir la sortie du modèle en un score de probabilité (les probabilités sont toujours comprises entre 0 et 1).

Softmax (Peng et al., 2017) elle est une fonction d'activation qui est utilisée pour la classification multi-classe donc si nous avons plusieurs neurones de sortie, et chacun représente une

classe avec les valeurs de ces neurones en entrée nous pouvons obtenir les probabilités de chaque classe et la somme des probabilités de softmax est toujours égale à 1.

Les optimiseurs

Les optimiseurs ont des algorithmes chargés de réduire les pertes et d'améliorer la précision en modifiant les attributs du réseau de neurones tels que les poids et le taux d'apprentissage. Il existe plusieurs types d'optimiseur et dans notre cas nous avons travaillé avec les quatre optimiseurs suivants :

ADAM (Kingma & Ba, 2015) est l'abréviation d'Adaptive moment estimation, est l'algorithme d'optimisation le plus populaire dans le domaine de l'apprentissage en profondeur. Il calcule les taux d'apprentissage adaptatif pour chaque paramètre.

Nadam: (Xiao et al, 2019) est l'abréviation Nesterov-accelerated Adaptive Moment Estimation est une extension de l'algorithme d'Adam qui intègre l'impulsion de Nesterov et peut avoir pour résultat une meilleure performance de l'algorithme d'optimisation.

Sgd (Bottou et Léon ,1998) (Stochastic gradient descent) c'est une méthode d'optimisation qui fait effectuer des calcules sur quelques données qui sont sélectionné d'une manière aléatoire au lieu de prendre tous l'ensemble de données pour chaque itération.

RMSprop (Kumar Reddy et al, 2019) : est l'abréviation de Root Mean Square propagation. C'est l'une des méthodes d'optimisation en apprentissage automatique et en apprentissage profond, qui est utilisée pour augmenter la précision des modèles.

4.6 Évaluation des résultats et comparaison

Pour atteindre notre objectif nous avons passé par 4 expériences.

- ✔ Expérience 1 : Appliquer 5 modèles sur dataset de 2 classes.
- ✔ Expérience 2 : Appliquer 5 modèles sur dataset de 4 classes.
- ✔ Expérience 3 : l'impact de changement d'optimiseur sur les 2 datasets
- ✔ Expérience 4 : Appliquer 5 modèles sur fake account detection dataset.

Expérience 1 : Appliquer 5 modèles sur dataset de 2 classes.

Dans cette expérience nous avons appliqué 5 modèles sur la première dataset 2 classes (Faux et authentique utilisateur) qui est équilibrée comme le montre l'histogramme la figure 4-23 ci- dessous et dans cette expériences nous avons fixé des paramètres comme le montre le tableau

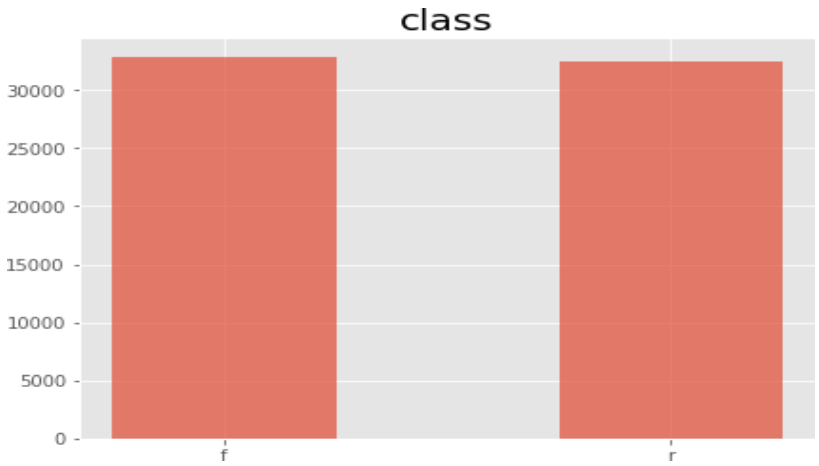
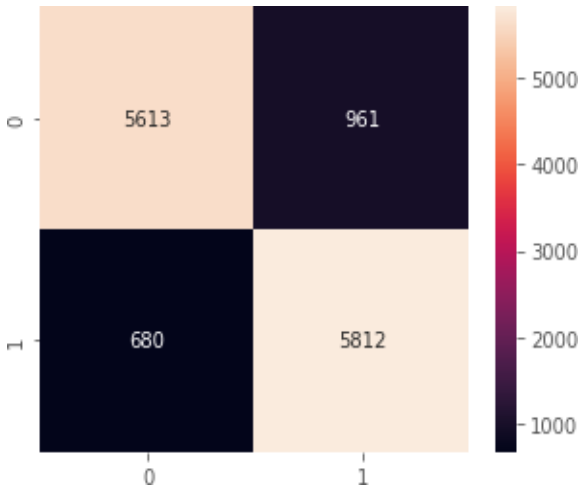


Figure 4 - 23 :l'histogramme du premier ensemble de données 2 classes

Tableau 4 - 5 : Tableau Les paramètres de travai

Paramètres	Loss	Optimiseur	Activation
Valeur	Binary crossentropy	ADAM	Sigmoid

Donc nous avons commencé par appliquer le modèle CNN sur l'ensemble de données alors nous avons obtenu comme résultats accuracy 0.87 et 0.29 la perte. La figure 4-24 représente les métriques d'évaluation précision et rappel et f1 mesure et représente la matrice de confusion

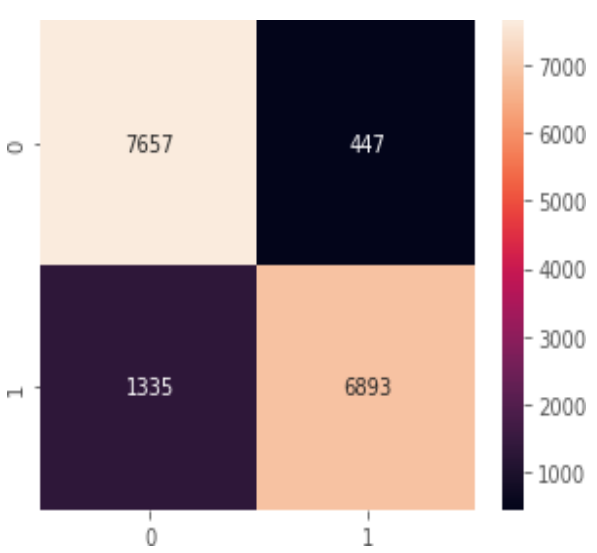


	precision	recall	f1-score	support
0	0.89	0.85	0.87	6574
1	0.86	0.90	0.88	6492
accuracy			0.87	13066
macro avg	0.88	0.87	0.87	13066
weighted avg	0.88	0.87	0.87	13066

103/103 [==============================] - 1s 10ms/step

Figure 4 - 24 : Matrices de confusion de model CNN sur le data de 2 classe

Après nous avons appliqué le modèle TCN alors nous avons obtenus accuracy 0.8909 et 0.2489 la perte. La figure 4-25 représente les métriques d'évaluation, et représente la matrice de confusion et le tableau représente les paramètres de compilation utilisée.



	precision	recall	f1-score	support
authentic	0.85	0.94	0.90	8104
fake	0.94	0.84	0.89	8228
accuracy			0.89	16332
macro avg	0.90	0.89	0.89	16332
weighted avg	0.90	0.89	0.89	16332

Figure 4 - 25 : Matrice de confusion et les métriques d'évaluation du modéle TCN sur le data de 2 classe.

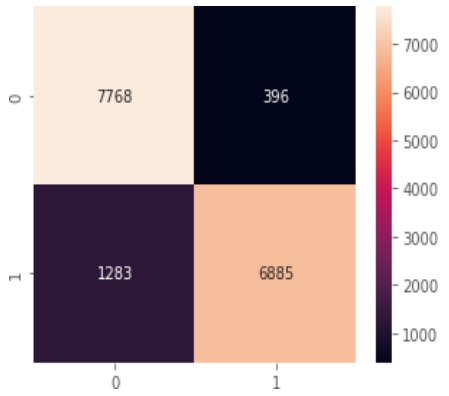
Après nous avons fait une combinaison du modèle précédent TCN avec le modèle LSTM alors nous avons obtenu accuracy 0.9081 et 0.2127 la perte. La figure 4-26 représente la précision, rappel et f1 mesure, et le tableau représente les paramètres de compilation utilisée.

	precision	recall	f1-score	support
0	0.86	0.94	0.90	8099
1	0.94	0.85	0.89	8233
accuracy			0.90	16332
macro avg	0.90	0.90	0.89	16332
weighted avg	0.90	0.90	0.89	16332

```
128/128 [==============================] - 1s 12ms/step
```

Figure 4 - 26 : les métriques d'évaluation de model TCN+LSTM sur le data de 2 classe

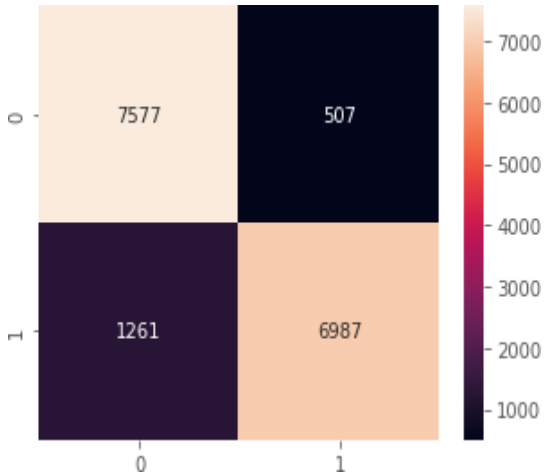
Après nous avons fait une autre combinaison du TCN, LSTM avec le modèle gru alors nous avons obtenus accuracy 0.8972 et 0.2410 la perte. La figure 4-27 représente la précision, rappel et f1 mesure et représente la matrice de confusion.



	precision	recall	f1-score	support
authentic	0.86	0.95	0.90	8164
fake	0.95	0.84	0.89	8168
accuracy			0.90	16332
macro avg	0.90	0.90	0.90	16332
weighted avg	0.90	0.90	0.90	16332

Figure 4- 27 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM +GRU sur dataset de 2 classes

Après nous avons fait une combinaison du TCN et LSTM+Bi gru alors nous avons obtenus accuracy 0.8917 et 0.2493 la perte. La figure 4-28 représente la précision, rappel et f1 mesure et représente la matrice de confusion.



	precision	recall	f1-score	support
authentic	0.85	0.94	0.90	8104
fake	0.94	0.84	0.89	8228
accuracy			0.89	16332
macro avg	0.90	0.89	0.89	16332
weighted avg	0.90	0.89	0.89	16332

Figure 4- 28 :Matrice de confusion et les métriques d'évaluation du model TCN+LSTM+Bi GRU sur le data de 2 classe.

Discussion 1:

Le tableau 4-6 est un récapitulatif des résultats de tous les modèles.

Tableau 4 - 6 : résultats de tous les modèles

Optimiseur	Algorithme	Accuracy	Précision	Rappel	F1 score
ADAM	CNN	87.44%	88%	87%	87%
	TCN	89.08%	**90%**	89%	89%
	TCN+ LSTM	**90.81%**	**90%**	**90%**	89%
	TCN+LSTM+ GRU	89.72%	**90%**	**90%**	**90%**
	TCN+ LSTM+ Bidirectional GRU	89.17%	**90%**	89%	89%

Nous observons très nettement que nous avons obtenus le meilleur résultat d'accuracy avec le modèle où nous avons fait une combinaison du modèle TCN et LSTM car il prédit correctement à la fois les individus positifs et négatifs et même les autres modèles nous avons obtenus des résultats assez similaires pour les mesures de performances précision(90%) , rappel (varie entre 89% et 90%) et f1 mesure (se varie entre 89% et 90%) sauf le modèle de CNN le résultat d'accuracy est très peu et même les autres performances précision, rappel, f1 score sont très faible.

Expérience 2 : Appliquer 5 modèles sur dataset de 4 classes**.**

Dans cette expérience nous avons appliqué 5 modèles sur la deuxième dataset de 4 classes (authentique, active faux utilisateur, inactive faux utilisateur, spammeur) qui n'est pas équilibrée comme le montre l'histogramme dans la figure 4-29 ci-dessous et dans cette expériences nous avons fixé des paramètres comme le montre le tableau 4-7

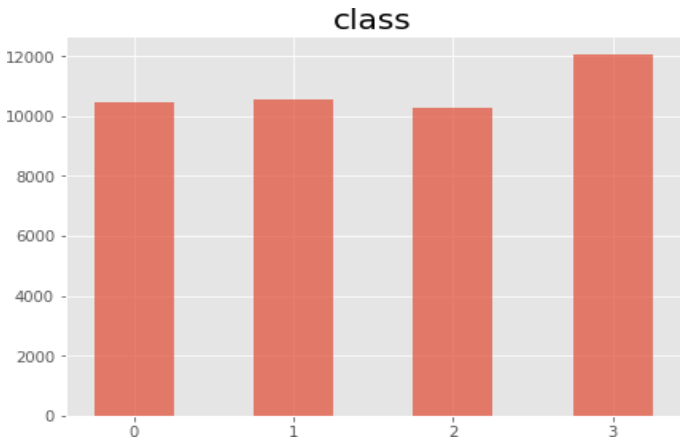
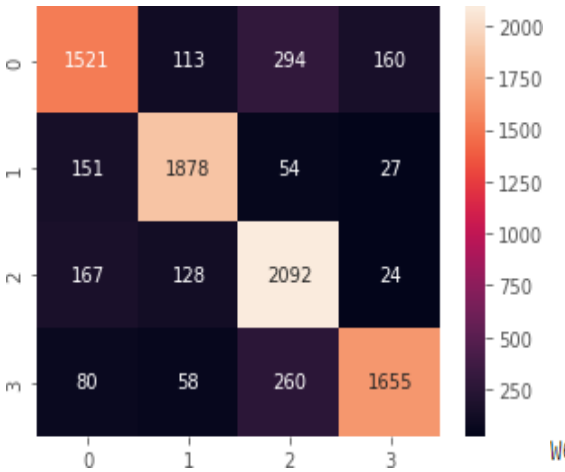


Figure 4 - 29 : l'histogramme du 2eme dataset (4 classes)

Tableau 4- 7 : Les paramètres de travail sur le dataset 4 classes

Paramètres	Loss	Optimiseur	Activation
Valeur	Catégorial crossentropy	ADAM	Softmax

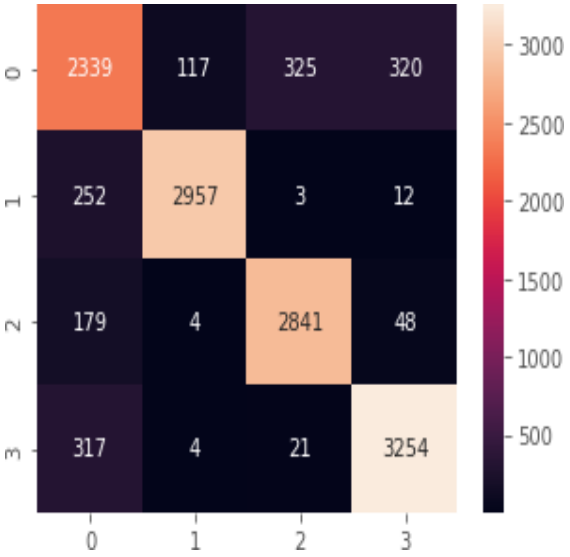
Donc nous avons commencé par appliquer le modèle CNN sur l'ensemble de données alors nous avons obtenus comme résultats accuracy 0.8250 et 0.5571 la perte. La figure 4-30 représente la précision et rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
0	0.79	0.73	0.76	2088
1	0.86	0.89	0.88	2110
2	0.77	0.87	0.82	2411
3	0.89	0.81	0.84	2053
accuracy			0.82	8662
macro avg	0.83	0.82	0.82	8662
weighted avg	0.83	0.82	0.82	8662

Figure 4 - 30 : la matrice de confusion les métriques d'évaluation de model CNN sur le data de 4 classe.

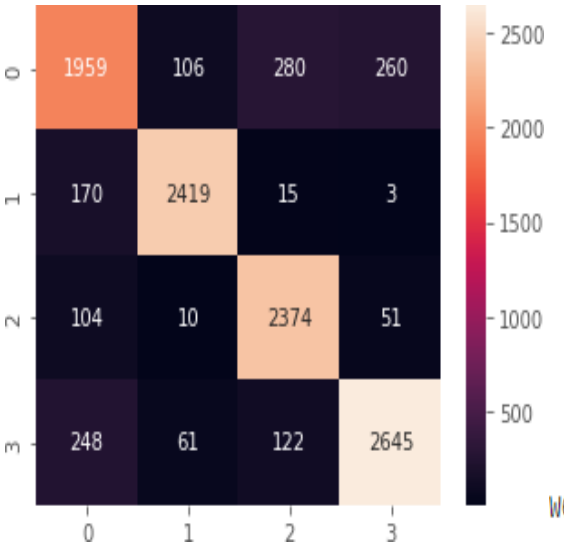
Après nous avons appliqué le modèle TCN sur dataset alors nous avons obtenus accuracy 0.8767 et 0.3149 la perte. La figure 4-31 représente la matrice de confusion, les métriques d'évaluation la précision, rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
0	0.76	0.75	0.76	3101
1	0.96	0.92	0.94	3224
2	0.89	0.92	0.91	3072
3	0.90	0.90	0.90	3596
accuracy			0.88	12993
macro avg	0.88	0.88	0.88	12993
weighted avg	0.88	0.88	0.88	12993

Figure 4 - 31 : Matrice de confusion et les métriques d'évaluation de model TCN sur le dataset de 4 classes.

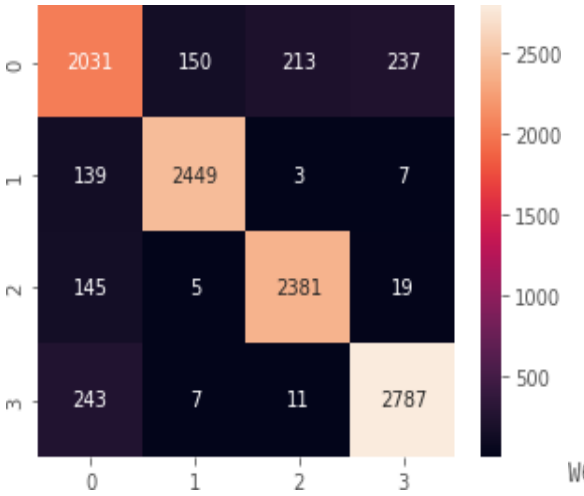
Après nous avons fait une combinaison du TCN et LSTM alors nous avons obtenu accuracy 0.8679 et 0.3149 la perte. La figure 4-32 représente la précision, rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
0	0.79	0.75	0.77	2605
1	0.93	0.93	0.93	2607
2	0.85	0.94	0.89	2539
3	0.89	0.86	0.88	3076
accuracy			0.87	10827
macro avg	0.87	0.87	0.87	10827
weighted avg	0.87	0.87	0.87	10827

Figure 4 - 32 : Matrice de confusion et les métriques d'évaluation du model TCN+LSTM sur le data de 4 classe

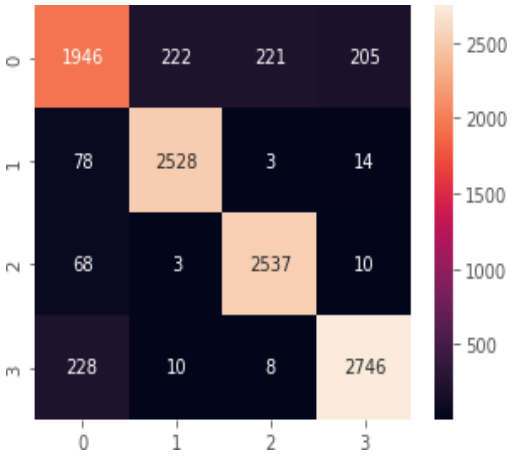
Après nous avons fait une combinaison du TCN et LSTM et GRU alors nous avons obtenus accuracy 0.8911 et 0.3815 la perte. La figure 4-33 représente la précision, rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
0	0.79	0.77	0.78	2631
1	0.94	0.94	0.94	2598
2	0.91	0.93	0.92	2550
3	0.91	0.91	0.91	3048
accuracy			0.89	10827
macro avg	0.89	0.89	0.89	10827
weighted avg	0.89	0.89	0.89	10827

Figure 4- 33 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM+GRU sur le data de 4 classe

Après nous avons fait une combinaison du TCN et LSTM et Bi gru alors nous avons obtenus accuracy 0.9012 et 0.2312 la perte. La figure 4-34 représente la précision, rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
0	0.84	0.75	0.79	2594
1	0.91	0.96	0.94	2623
2	0.92	0.97	0.94	2618
3	0.92	0.92	0.92	2992
accuracy			0.90	10827
macro avg	0.90	0.90	0.90	10827
weighted avg	0.90	0.90	0.90	10827

Figure 4 - 34 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM+Bi GRU sur le data de 4 classe

Discussion 2: Le tableau 4-8 est un récapitulatif des résultats des 5 modèles sur le même ensemble de données.

Tableau 4 - 8 : résultats des 5 modèles

Optimiseur	Algorithme	Accuracy	Précision	Rappel	F1 score
ADAM	CNN	82.50%	83%	83%	83%
	TCN	87.67%	88%	88%	88%
	TCN+ LSTM	86.79%	87%	87%	87%
	TCN+LSTM+ GRU	89.11%	89%	89%	89%
	TCN+ LSTM+ Bidirectional GRU	**90.12%**	**90%**	**90%**	**90%**

Comme le tableau suivant indique que nous avons obtenus les meilleurs résultats avec le modèle où la combinaison du 3 modèles le TCN, LSTM et le bidirectionnel GRU, même le modèle TCN+LSTM+ GRU nous avons obtenus des résultats très élevées pour toutes mesures de performances précision (90%), rappel (89%) et f1 mesure (89%) et les 2 modèles TCN, TCN+LSTM les résultats sont bien mais le modèle CNN est très faible.

4.7 La comparaison avec les travaux connexes

Le tableau suivant résume la comparaison entre nos modèles utilisées et les travaux connexes précédents (Purba et al., 2020) qui sont appliquées sur le même ensemble de données (2 classes):

Comme le montre le tableau 4-9, le modèle TCN + LSTM surpasse les autres algorithmes dans la classification de 2 classes.

Tableau 4 - 9 : comparaison entre les modèles utilisées et les travaux connexes sont appliquées sur le data 2 classe.

	Algorithmes	Accuracy	Précision	Rappel	F1 Mesure
Les modèles de l'article (Purba et al., 2020)	Random Forest	90.09%	**90.7%**	**90.1%**	**90.1%**
	Multilayer Perceptron	81.73%	81.8%	81.7%	81.7%
	Logistic Regression	80.94%	81%	80.9%	80.9%
	Naive Bayes	73.12%	75.9%	73.1%	72.4%
	J48 Decision Tree	88.34%	88.6%	88.3%	88.3%
Nos modèles	CNN	87.44%	88%	87%	87%
	TCN	89.08%	**90%**	89%	89%
	TCN+LSTM	**90.81%**	**90%**	**90%**	89%
	TCN+LSTM+GRU	89.72%	**90%**	**90%**	**90%**
	TCN+ LSTM+ Bidirectional GRU	89.17%	**90%**	89%	89%

Le tableau 4-10 suivant résume la comparaison entre les modèles utilisées et les travaux connexes précédents qui sont appliquées sur le même ensemble de données (4 classes).

Tableau 4 - 10 : : comparaison entre les modèles utilisées et les travaux connexes sont appliquées sur le data 4 classe

	Algorithmes	Accuracy	Précision	Rappel	F1_mesure
Les modèlesde l'article (Purba et al., 2020)	Random Forest	**91.76%**	**91.7%**	**91.%**	**91.7%**
	Multilayer Perceptron	73.75%	73.8%	73.7%	73.5%
	Logistic Regression	68.54%	68.1%	68.5%	68.1%
	Naïve Bayes	54.22%	60.6%	54.2%	49.3%
	J48 Decision Tree	88.28%	88.2%	88.3%	88.2%
Nos modèles	CNN	82.50%	83%	82%	82%
	TCN	87.67%	88%	88%	88%
	TCN +LSTM	86.79%	87%	87%	87%
	TCN +LSTM+GRU	89.11%	89%	89%	89%
	TCN+ LSTM+ Bidirectional GRU	90.12%	90%	90%	90%

Comparaison

Nous avons comparé qualitativement les travaux reliés présentés dans le Tableau 4-6 et le tableau 4-7.

La comparaison se fait selon un ensemble de critères : classification, base de données utilisée, les métriques de performance.

Dans le tableau 4-6, Nous comparons notre modèle proposé TCN et LSTM avec le modèle Random Forest de l'article (Purba et al., 2020) les 2 modèles sont appliqués sur le même l'ensemble de données classification 2 classes nous observons que notre modèle TCN LSTM donne un meilleur résultat d'accuracy (90.81% très élevée) par rapport à Random Forest , la précision et le rappel des 2 modèles sont sémantiques mais la valeur de f1 mesure est un peu élevé du modèle Random Forest.

Et nous remarquons que même nos résultats d'accuracy, précision, rappel et f1 mesure des autres modèles CNN, TCN, TCN+LSTM+GRU et TCN+LSTM+GRU sont très élevés par rapport à leurs modèles Naive Bayes Logistic Regression Multilayer Perceptron .

Mais le tableau 4-8 a montré que le modèle TCN LSTM ne fonctionne pas bien sur le deuxième ensemble de données classification de 4 classes mais quand nous avons combiné ce

modèle avec le modèle Bidirectionnel GRU nous avons obtenus des résultats élevés mais c'est peu par rapport au leur modèle Random Forest par ce que nous avons utilisé des modèles de deep learning.

En générale Les résultats de nos modèles sont plus élevés par rapport à leurs modèles :

Si nous prenons notre faible modèle CNN nous remarquons qu'il est mieux que Naive Bayes Logistic Regression Multilayer Perceptron et le modèle.

Et même le modèle Tcn +Lstm+GRU est mieux que tous leurs modèles sauf random forest.

Expérience 3 : l'impact de changement d'optimiseur sur les 2 datasets.

Dans cette expérience nous avons appliqué un changement d'optimiseur des modèles qui ont l'accuracy les plus élevées et sur les 2 datasets.

Dataset 2 classes : la figure 4-35 ci-dessous représente les matrices de confusion après le changement d'optimiseur et le tableau 4-11 représente les résultats.

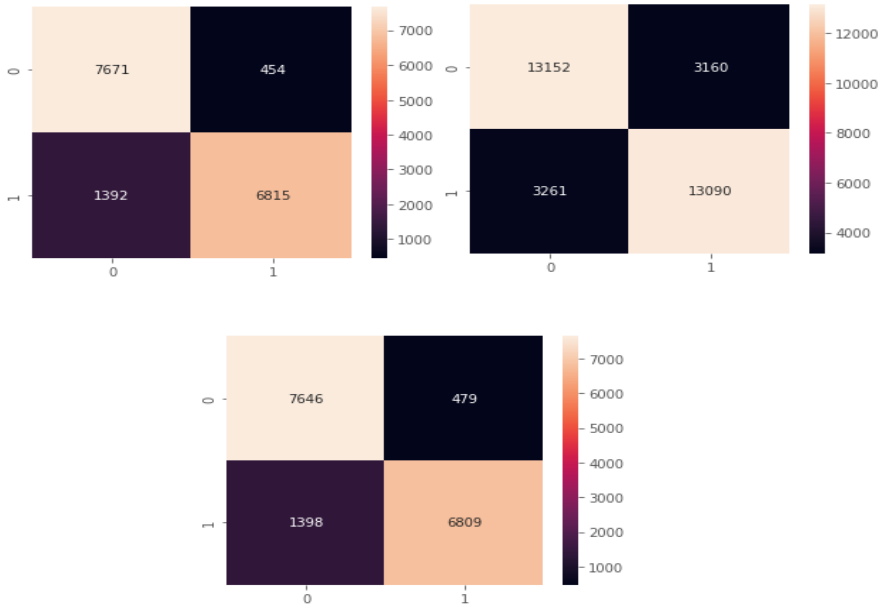


Figure 4 - 35 : les matrices de confusion après le changement d'optimiseur de data de 2 classe

Optimiseur	Accuracy	Précision	Rappel	F1 score
ADAM	**90.81%**	**90%**	**90%**	**89%**
NADAM	88.51%	89%	89%	88%
SGD	85.77%	80%	80%	80%
RSMPROP	88.70%	89%	89%	**89%**

Tableau 4 - 11 : tableau représente les résultats après le changement d'optimiseur de dataset de 2 classes.

Dataset 4 classes les figures 4-36 ci-dessous représentent les matrices de confusion après le changement d'optimiseur et le tableau 4-12 représente les résultats.

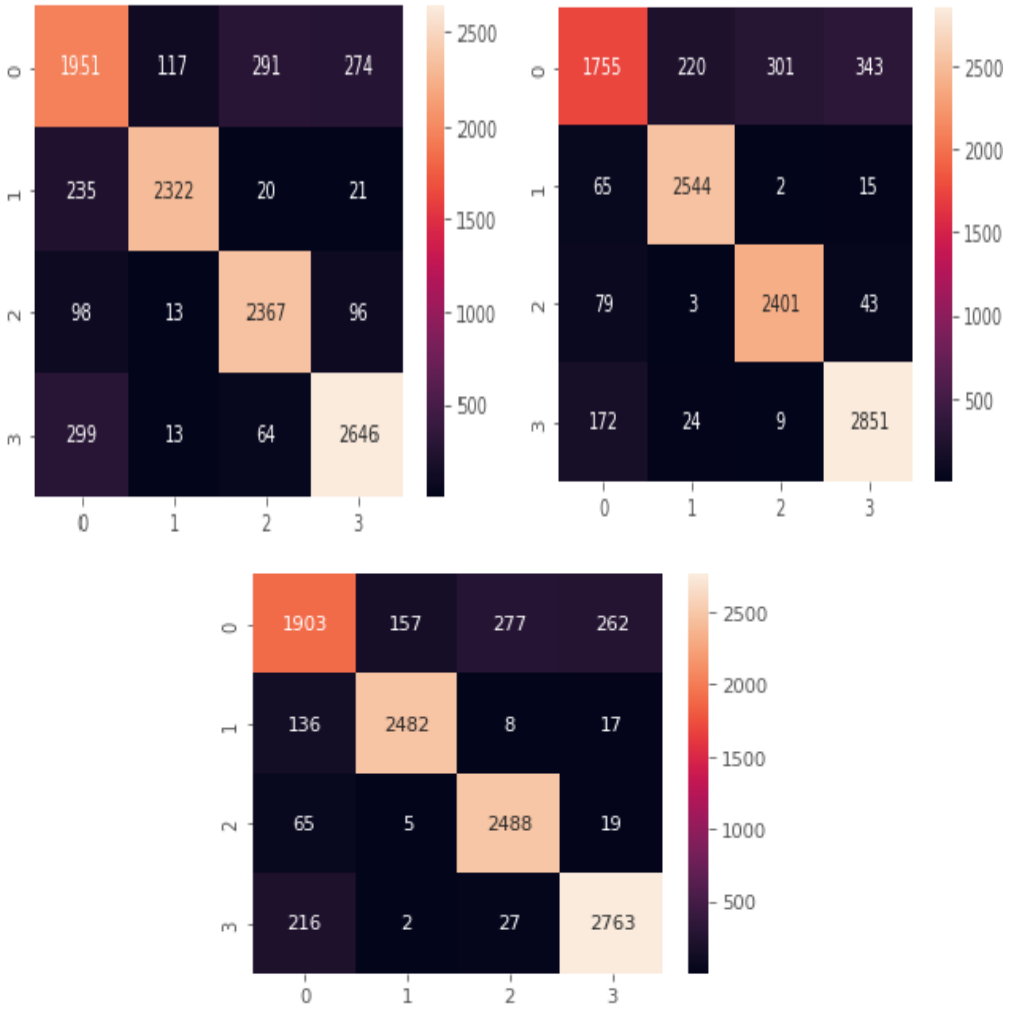


Figure 4 - 36 : matrices de confusion après le changement d'optimiseur de data de 4 classe

Tableau 4 - 12 : tableau représente les résultats après le changement d'optimiseur de data de 4 classes

Optimiseur	Accuracy	Précision	Rappel	F1 score
ADAM	**90.12%**	**90%**	**90%**	**90%**
NADAM	88.21%	88%	88%	88%
SGD	85.77%	86%	86%	86%
RSMPROP	89.00%	89%	89%	89%

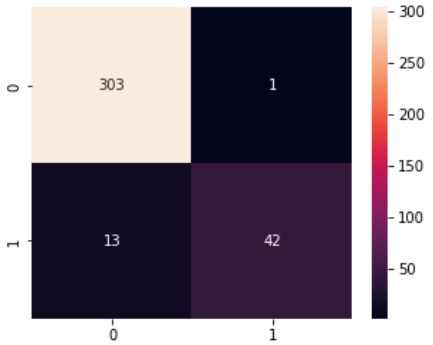
Dans tous les cas différents modèles et même déférentes dataset nous avons remarqué que les résultats d'optimiseur ADAM est très élevés et il est très rapide par rapport aux autres optimiseur et l'optimiseur SGD ces résultats sont les plus faibles.

Expérience 4 : Appliquer 5 modèles sur fake account détection dataset.

Dans cette expérience nous avons appliqué les 5 modèles précédents sur une nouvelle dataset dans cette expériences nous avons fixé des paramètres comme le montre le tableau

Paramètres	Loss	Optimiseur	Activation	Epochs	Train-Test Split
Valeur	Binary cross entropy	ADAM	Sigmoid	100	%70 − %30

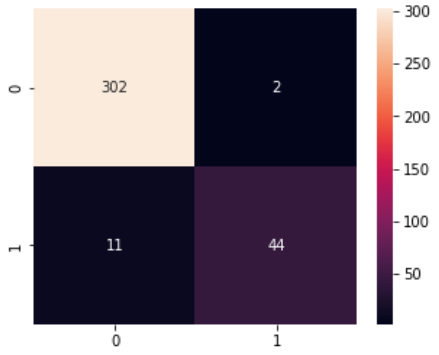
Nous avons commencé par appliquer le modèle CNN sur l'ensemble de données alors nous avons obtenus comme résultats accuracy 0.9610 et 0.1089 la perte. La figure 4-37 représente la précision et rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
real	0.96	1.00	0.98	304
fake	0.98	0.76	0.86	55
accuracy			0.96	359
macro avg	0.97	0.88	0.92	359
weighted avg	0.96	0.96	0.96	359

Figure 4 - 37 : Matrice de confusion et les métriques d'évaluation du model CNN sur fake account detection dataset.

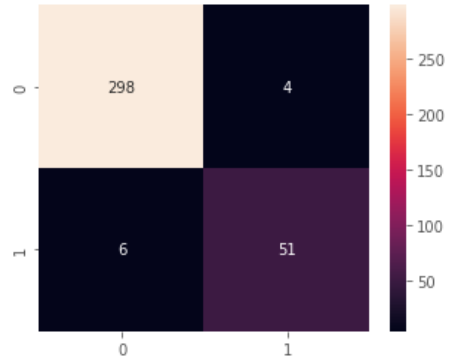
Ensuite nous avons appliqué le modèle TCN alors nous avons obtenus comme résultats accuracy 0.9638 et 0.1853 la perte. La figure 4-38 représente la précision et rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
real	0.96	0.99	0.98	304
fake	0.96	0.80	0.87	55
accuracy			0.96	359
macro avg	0.96	0.90	0.93	359
weighted avg	0.96	0.96	0.96	359

Figure 4- 38 : Matrice de confusion et les métriques d'évaluation du model TCNsur fake account detection dataset.

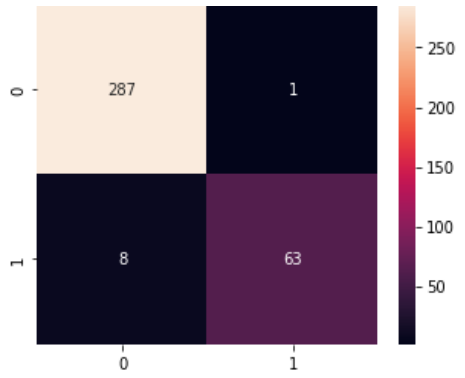
Après nous avons fait une combinaison du TCN et LSTM alors nous avons obtenu accuracy 0.9721 et 0.0998% la perte . La figure4-39 représente la précision, rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
real	0.98	0.99	0.98	302
fake	0.93	0.89	0.91	57
accuracy			0.97	359
macro avg	0.95	0.94	0.95	359
weighted avg	0.97	0.97	0.97	359

Figure 4- 39 : Matrice de confusion et les métriques d'évaluation du model TCN+LSTM sur fake account detection dataset

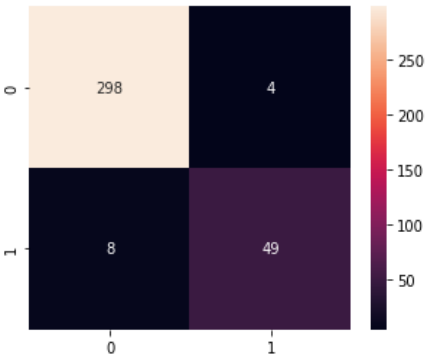
Nous avons fait une combinaison du modèle précédent TCN et LSTM avec GRU alors nous avons obtenu accuracy 0.9749 et 0.1248 la perte. La figure 4-40 représente la précision, rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
real	0.97	1.00	0.98	288
fake	0.98	0.89	0.93	71
accuracy			0.97	359
macro avg	0.98	0.94	0.96	359
weighted avg	0.98	0.97	0.97	359

Figure 4 -40 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM+GRU sur fake account detection dataset

Après nous avons fait une combinaison du TCN et LSTM et Bi gru alors nous avons obtenus accuracy 0.9666 et 0.1339 la perte. La figure 4-41 représente la précision, rappel et f1 mesure et la matrice de confusion.



	precision	recall	f1-score	support
real	0.97	0.99	0.98	302
fake	0.92	0.86	0.89	57
accuracy			0.97	359
macro avg	0.95	0.92	0.94	359
weighted avg	0.97	0.97	0.97	359

Figure 4- 41 : Matrice de confusion et les métriques d'évaluation de model TCN+LSTM+Bi gru sur fake account detection dataset.

Discussion 3

Tableau 4- 13 : Résume les résultats des 5 modèles sur ensemble de données de détection de faux comptes**.**

Algorithme	Accuracy	Précision	Rappel	F1 score
CNN	96.10%	96%	96%	96%
TCN	96.94%	97%	**97%**	**97%**
TCN+ LSTM	97.21%	97%	**97%**	**97%**
TCN+LSTM+ GRU	**97.49%**	**98%**	**97%**	**97%**
TCN+ LSTM+ Bidirectional GRU	96.38%	96%	96%	96%

D'après le tableau 4-13 , on observe que le modèle le performant est la combinaison de TCN, LSTM et GRU car avec ce modèle nous avons obtenus le meilleur résultat d'accuracy 97.49% et avec une précision de 98%, rappel 97% et f1 mesure 97% en comparant avec les autres modèles.
Mais même les autres modèles sont très efficaces et leurs résultats sont très élevés.

Nous comparons nos résultats avec l'article :
Nous remarquons que la précision des 3 modèles (TCN, TCN+LSTM, TCN+LSTM+GRU) est 97% est élevée par rapport à leur précision (96%).

4.8 Conclusion

Dans ce dernier chapitre nous avons présenté l'environnement de travail, les bibliothèques et les paramètres d'évaluations utilisés.

Ensuite, nous avons exposé les résultats obtenus des modèles utilisés et comparé entre les performances de ces modèles avec d'autres modèles des travaux connexes précédents.

Après avoir pu déterminer le bon modèle de chaque ensemble donné nous avons appliqué des changements sur leurs paramètres.

Notre travail prouve son intérêt en donnant de bons résultats et il a prouvé sa capacité de détecter les faux utilisateurs.

Conclusion générale

Les réseaux sociaux via Internet sont une plate-forme de communication nécessaire dans tous les domaines, mais vu la popularité des sites de réseaux sociaux, ils s'exposent à de grands dangers, en ouvrant de faux comptes pour se rapprocher des gens et les menacer, ou pour voler des informations utiles à des entreprises internationales Major, tout cela est considéré comme un cybercrime.

Afin de détecter ces faux comptes, nous avons proposé un ensemble de modèles basés sur l'approche du deep learning. Il est bien connu que pour détecter les faux comptes, notre modèle doit être performant en temps réel avec un faible taux d'erreur.

Dans notre travail, nous avons utilisé deux ensembles de données différents, où chaque groupe contient deux parties. Pour le premier ensemble de données A, il se compose de deux parties, à savoir, la partie 2 CLASS contient 32 869 faux fichiers et 32 460 fichiers réels.

Tandis que la 4 CLASS se compose de 10 441 utilisateurs réels et 12 054 faux utilisateurs actifs et 10 549 faux utilisateurs inactifs et 10 263 faux spammeurs.

Le deuxième ensemble de données B se compose d'une partie de faux compte qui contient 1002 comptes réels 201 faux comptes et d'une partie de compte automatisé qui contient 700 comptes réels et 700 comptes automatisés.

Cinq modèles ont été appliqués à chaque partie des deux ensembles de données représenté dans CNN ,TCN ,TCN+LSTM ,TCN+LSTM+GRU, TCN+LSTM+Bi GRU.

l'ensemble de données 2 class , la précision la plus élevée a été trouvée dans le modèle TCN + LSTM, atteignant 90,81 %. Quant à l'ensemble de données CLASS 4, la précision la plus élevée trouvée dans le modèle TCN + LSTM + Bi GRU était supérieure à celle des autres modèles, atteignant 90,12 %, En ce qui concerne l'ensemble de données sur les faux comptes, la précision la plus élevée a été trouvée dans le modèle TCN + LSTM + GRU, qui s'élevait à environ 97,49 %.

Cependant, malgré ces résultats obtenus, il reste encore un besoin d'améliorations supplémentaires, telles que la modification des modèles en ce qui concerne le temps de réponse, en plus du fait que des modèles plus puissants peuvent être émis afin de compenser le mélange entre modèle

Références

Adikari, S., & Dutta, K. (2014). Identifying fake profiles in linkedin. Proceedings - Pacific Asia Conference on Information Systems, PACIS 2014, 1–30.

Atif, J. (2016). Data Mining / ML e.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. http://arxiv.org/abs/1803.01271

Bottou, L. (2012). Stochastic Gradient Learning. Tutorial2.

Bottou, Léon (1998). "Online Algorithms and Stochastic Approximations". Online Learning and Neural Networks. Cambridge University Press. ISBN 978-0-521-65263-6.

Brownlee, J. (2016). Master Machine Learning Algorithms: Discover how they work and implement them from scratch. MAchine Learning Mastery, 1–163. http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. 1–15. http://arxiv.org/abs/1309.0238

Captcha, M. L. (2022). Introduction à l ' apprentissage automatique.

Chauhan, V. K., Dahiya, K., & Sharma, A. (2019). Problem formulations and solvers in linear SVM: a review. Artificial Intelligence Review, 52(2), 803–855. https://doi.org/10.1007/s10462-018-9614-6

Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation, 103–111. https://doi.org/10.3115/v1/w14-4012

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., & Saurous, R. A. (2017). TensorFlow Distributions.http://arxiv.org/abs/1711.10604

Dr.A, .Usha Ruby. (2020). Binary cross entropy with deep learning technique for Image classification. International Journal of Advanced Trends in Computer Science and Engineering, 9(4), 5393–5397. https://doi.org/10.30534/ijatcse/2020/175942020

Erşahin, B., Aktaş, Ö., Kilmç, D., &Akyol, C. (2017). Twitter fake account detection. 2nd International Conference on Computer Science and Engineering, UBMK 2017, 388–392. https://doi.org/10.1109/UBMK.2017.8093420

F. Sultana, A. Sufian, and P. Dutta. Advancements in image classification using convolutional neural network. In 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pages 122–129,.

Flash DGSI : Les dangers des faux profils sur les réseaux sociaux professionnels Chan-Thaï LAMInformation-09/02/2016

Fundamental Concepts of Convolutional Neural Network Anirudha Ghosh1 , Abu Sufian1,∗ , Farhana Sultana1 , Amlan Chakrabarti2 , Debashis De 2020.

Géron, A. (2019). Hands-on Machine Learning whith Scikit-Learing, Keras and Tensorfow. In O'Reilly Media, Inc.

Ghosh, K., & Doshi, J. (2021). Fraudulent Users Detection on Social Media Platforms Using Machine Learning Techniques. 25(5), 4170–4174.

Graesser, A. C., Hu, X., & Sottilare, R. (2018). Intelligent tutoring systems. In International Handbook of the Learning Sciences. https://doi.org/10.4324/9781315617572

Halim, Z., Gul, M. M., Ul Hassan, N., Baig, R., Ur Rehman, S., & Naz, F. (2011). Malicious users' circle detection in social network based on spatio-temporal co-occurrence. Proceedings - International Conference on Computer Networks and Information Technology, 35–39. https://doi.org/10.1109/ICCNIT.2011.6020904

Hassan, A., Amin, M. R., Azad, A. K. Al, & Mohammed, N. (2017). Sentiment analysis on bangla and romanized bangla text using deep recurrent models. IWCI 2016 - 2016 International Workshop on Computational Intelligence, 51–56. https://doi.org/10.1109/IWCI.2016.7860338

Ho, Y., & Wookey, S. (2020). The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling.

IEEE Access, 8, 4806–4813. https://doi.org/10.1109/ACCESS.2019.2962617

Ito, Y. (1991). Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory. Neural Networks, 4(3), 385–394. https://doi.org/10.1016/0893-6080(91)90075-G

Jarl, G. (2003). Accuracy and precision of a technique to assess residual limb volume with a measuring-tape. June, 1–62.

Ju, Y., Zhang, M., & Zhu, H. (2019). Study on a New Deep Bidirectional GRU Network for Electrocardiogram Signals Classification. 90(Iccia), 355–359. https://doi.org/10.2991/iccia-19.2019.54

K. R. Srinath. (2017). Python -The Fastest Growing Programming Language. International Research Journal of Engineering and Technology, 4(12), 354–357.www.irjet.net

Kadam, N., &Patidar, H. (2020). Social Media Fake Profile Detection Technique Based on Attribute Estimation and Content Analysis Method. International Journal of Recent Technology and Engineering, 8(6), 4534–4539. https://doi.org/10.35940/ijrte.f8417.038620

Ketkar, N. (2017). Deep Learning with Python. In Deep Learning with Python. https://doi.org/10.1007/978-1-4842-2766-4

Khaled et al., 2019)10.1109@BigData.2018.8621913.pdf. (n.d.).

Khedkar, S., Gandhi, P., Shinde, G., & Subramanian, V. (2020). Deep Learning and Explainable AI in Healthcare Using EHR. https://doi.org/10.1007/978-3-030-33966-1_7
Kikuta, K., & Takahashi, A. (2019). On the categorical entropy and the topological entropy. International Mathematics Research Notices, 2019(2), 457–469. https://doi.org/10.1093/imrn/rnx131
Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–15.
Kishor, L., & Goyal, D. (2013). Comparative Analysis of Various Scheduling Algorithms. 2(4), 1488–1491.
Koidl, K. (2013). Loss functions in classification tasks. School of Computer Science and Statistic Trinity College, Dublin.
Kovatchev, B. P., Gonder-Frederick, L. A., Cox, D. J., & Clarke, W. L. (2004). Evaluating the Accuracy of Continuous Glucose-Monitoring Sensors. Diabetes Care, 27(8), 1922–1928. https://doi.org/10.2337/diacare.27.8.1922
Kumar Reddy, R. V., Srinivasa Rao, B., & Raju, K. P. (2019). Handwritten Hindi Digits Recognition Using Convolutional Neural Network with RMSprop Optimization. Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018, Iciccs, 45–51. https://doi.org/10.1109/ICCONS.2018.8662969
Laloë, D. (1993). Precision and information in linear models of genetic evaluation. Genetics Selection Evolution, 25(6), 557. https://doi.org/10.1186/1297-9686-25-6-557
Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M., & Riquelme, J. C. (2020). Temporal convolutional networks applied to energy-related time series forecasting. Applied Sciences (Switzerland), 10(7), 1–17. https://doi.org/10.3390/app10072322
Le Scan : comment ne pas vous faire avoir par des faux profils Facebook.
LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2323.https://doi.org/10.1109/5.726791
Lehnert, W. G. (1979). Text Processing Effects and Recall Memory. YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE.
Lynn, H. M., Pan, S. B., & Kim, P. (2019). A Deep Bidirectional GRU Network Model for Biometric Electrocardiogram Classification Based on Recurrent Neural Networks. IEEE Access, 7, 145395–145405. https://doi.org/10.1109/ACCESS.2019.2939947
Madjarov, I. (2005). Des services Web pour le e-Learning. Revue Électronique Sur La Recherche En TIC (ETI-2005), March, 1–15.
Mandhala, V. N., Sujatha, V., & Devi, B. R. (2015). Scene classification using support vector machines. Proceedings of 2014 IEEE International Conference on Advanced Communication, Control and Computing Technologies, ICACCCT 2014, 63(3), 1807–1810. https://doi.org/10.1109/ICACCCT.2014.7019421
Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—part 1: Literature review. Remote Sensing, 13(13). https://doi.org/10.3390/rs13132450
Merity, S., Keskar, N. S., & Socher, R. (2018). An Analysis of Neural Language
Modeling at Multiple Scales. http://arxiv.org/abs/1803.08240
Nagarajan, G., Minu, R. I., & Jayanthila Devi, A. (2020). Optimal Nonparametric Bayesian Model-Based Multimodal BoVW Creation Using Multilayer pLSA. Circuits, Systems, and Signal Processing, 39(2), 1123–1132. https://doi.org/10.1007/s00034-019-01307-7
Option, I., Encadr, B. M., Djamel, M., & Promotion, E. (2014). Résumé. 1–137. Boursin, L. (2011). Le média humain Sommaire.
Peng, H., Li, J., Song, Y., & Liu, Y. (2017). Incrementally learning the hierarchical softmax function for neural language models. 31st AAAI Conference on Artificial Intelligence, AAAI 2017, 3267–3273. https://doi.org/10.1609/aaai.v31i1.10994
Powers, D. M. W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 37–63. http://arxiv.org/abs/2010.16061
Prem Jacob, T., Pravin, A., & Asha, P. (2018). Arduino object follower with augmented reality. International Journal of Engineering and Technology(UAE), 7(3.27 Special Issue 27), 108–110. https://doi.org/10.14419/ijet.v7i3.27.17665
Purba, K. R., Asirvatham, D., & Murugesan, R. K. (2020). Classification of instagram fake users using supervised machine learning algorithms. International Journal of Electrical and Computer Engineering, 10(3), 2763–2772. https://doi.org/10.11591/ijece.v10i3.pp2763-2772
Ranjana, S., Sathian, R., &Kamalesh, M. D. (2021). Fake Profile Detection in Facebook. In Lecture Notes in Electrical Engineering (Vol. 691). Springer Singapore. https://doi.org/10.1007/978-981-15-7511-2_74
Reimers, N., & Gurevych, I. (2017). Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. http://arxiv.org/abs/1707.06799
Rosasco, L., De Vito, E., Caponnetto, A., Piana, M., & Verri, A. (2004). Are Loss Functions All the Same? Neural Computation, 16(5), 1063–1076. https://doi.org/10.1162/089976604773135104
Rossum, G. van, & Boer, J. de. (1991). Interactively Testing Remote Servers Using the Python Programming Language. In CWI Quarterly (pp. 283–303).
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (2013). Learning Internal Representations by Error Propagation. Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence, V, 399–421. https://doi.org/10.1016/B978-1-4832-1446- 7.50035-2

Sahoo, S. R., & Gupta, B. B. (2021). Real-time detection of fake account in twitter using machine-learning approach. In Advances in Intelligent Systems and Computing (Vol. 1086). Springer Singapore. https://doi.org/10.1007/978-981-15-1275-9_13

Saint-jean, C., & Classification, C. S. (2007). Classification paramétrique robuste partiellement supervisée en reconnaissance des formes To cite this version : HAL Id : tel-00145895 par Classification param ´ etrique robuste partiellement supervis ´ ee en reconnaissance des formes.

Sasaki, Y. (2007). The truth of the F-measure. Teach Tutor Mater, 1–5. http://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf

Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. International Journal of Engineering Applied Sciences and Technology, 04(12), 310–316. https://doi.org/10.33564/ijeast.2020.v04i12.054

Sinaga, K. P., & Yang, M. (2020). Unsupervised K-Means Clustering Algorithm. 8. https://doi.org/10.1109/ACCESS.2020.2988796

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks. 1–42. http://arxiv.org/abs/1909.09586

Townsend, J. T. (1971). Erratum to: Theoretical analysis of an alphabetic confusion matrix. Perception & Psychophysics, 10(4), 256. https://doi.org/10.3758/BF03212817

Travis, O. (2007). Numpybook. 378.

Tripathy, A., Agrawal, A., & Rath, S. K. (2015). Classification of Sentimental Reviews Using Machine Learning Techniques. Procedia Computer Science, 57, 821–829. https://doi.org/10.1016/j.procs.2015.07.523

Use_of_Artificial_Neural_Networks_to_Identify_Fake_Profiles .pdf

Vo.T.H, P., & Czygan, M. (2015). Getting started with Python Data Analysis. http://bibliotecavirtual.uis.edu.co:2110/ehost/detail/detail?vid=0&sid=de3c0a36-bb34-4675-930d-f5029cc2738e%40pdc-v-sessmgr06&bdata=Jmxhbmc9ZXMmc2l0ZT1laG9zdC1saXZl#AN=1091507&db=nlebk

Vujović, Ž. (2021). Classification Model Evaluation Metrics. International Journal of Advanced Computer Science and Applications, 12(6), 599–606. https://doi.org/10.14569/IJACSA.2021.0120670

Wanto, A., Windarto, A. P., Hartama, D., & Parlina, I. (2017). Use of Binary Sigmoid Function And Linear Identity In Artificial Neural Networks For Forecasting Population Density. IJISTECH (International Journal Of Information System & Technology), 1(1), 43. https://doi.org/10.30645/ijistech.v1i1.6

Wiering, M. A., & van Hasselt, H. (2008). Ensemble algorithms in reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 38(4), 930–936. https://doi.org/10.1109/TSMCB.2008.920231

Xiao, B., Liu, Y., & Xiao, B. (2019). Accurate state-of-charge estimation approach for lithium-ion batteries by gated recurrent unit with ensemble optimizer. IEEE Access, 7, 54192–54202. https://doi.org/10.1109/ACCESS.2019.2913078

Xu, C., Lu, C., Liang, X., Gao, J., Zheng, W., Wang, T., & Yan, S. (2016). Multi-loss Regularized Deep Neural Network.IEEE Transactions on Circuits and Systems for Video Technology, 26(12), 2273–2283. https://doi.org/10.1109/TCSVT.2015.2477937

Yeh, C. W., & Chen, J. Der. (2011). Role of ligand conformation in the structural diversity of divalent complexes containing phosphinic amide ligand. Inorganic Chemistry Communications, 14(8), 1212–1216. https://doi.org/10.1016/j.inoche.2011.04.023

Zhang, H. (2004). The optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004, 2, 562–567.

Zhang, Y., Gao, J., & Zhou, H. (2020). Breeds Classification with Deep Convolutional Neural Network. PervasiveHealth: Pervasive Computing Technologies for Healthcare, 145–151. https://doi.org/10.1145/3383972.3383975

Shama*, S., Akram, S. W., Nandini, K. S., Anjali, P. B., & Manaswi, K. D. (2019). Fake Profile Identification in Online Social Networks. International Journal of Recent Technology and Engineering (IJRTE), 8(4), 11190–11194. https://doi.org/10.35940/ijrte.d9933.118419

Ranjana, S., Sathian, R., & Kamalesh, M. D. (2021). Fake Profile Detection in Facebook. In Lecture Notes in Electrical Engineering (Vol. 691). Springer Singapore. https://doi.org/10.1007/978-981-15-7511-2_74

Gong, Q., Zhang, J., Chen, Y., Li, Q., Xiao, Y., Wang, X., & Hui, P. (2019). Detecting malicious accounts in online developer communities using deep learning. International Conference on Information and Knowledge Management, Proceedings, 1, 1251–1260. https://doi.org/10.1145/3357384.3357971

Hajdu, G., Minoso, Y., Lopez, R., Acosta, M., & Elleithy, A. (2019). Use of Artificial Neural Networks to Identify Fake Profiles. 2019 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2019. https://doi.org/10.1109/LISAT.2019.8817330