

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Akli Mohand Oulhadj - Bouira**  
**Faculté des Sciences Exactes**  
**Département d'Informatique**



**Mémoire de fin d'études**

Pour l'obtention du diplôme de Master recherche en Informatique

**Spécialité: Ingénierie des Systèmes d'Information et des Logiciels**

**Par :**

**OMARI Amina ET OMARI Akila**

**Thème**

**Conception d'un modèle d'apprentissage à partir des réseaux sociaux pour la gestion des catastrophes et l'éducation des catastrophes**

*Devant le jury composé de :*

- *BENNOUAR Djamal*
- *ZAIR Bouzidi*
- *BRAHIMI Farida*
- *SAOUD Afaf*

*Président*  
*Encadreur*  
*Examineur1*  
*Examineur2*



## التصريح الشرفي الخاص بالالتزام بقواعد النزاهة العلمية



انا الممضي اسفله،

السيد(ة)..... OMARI Amina .....الصفة: طالب ( ماستر ) / دكتوراه (

الحامل(ة) لبطاقة التعريف الوطنية:..... 119910337007960001 .....والصادرة بتاريخ 07-09-2020

المسجل(ة) بكلية / معهد sciences et sciences appliqués قسم Informatique

تخصص: Ingénierie des systèmes d'information et des logiciels

والمكلف(ة) بإنجاز اعمال بحث(مذكرة، التخرج، مذكرة ماستر، مذكرة ماجستير، اطروحة دكتوراه).

عنوانها: Conception d'un model d'apprentissage a partir des réseaux sociaux pour la gestion des catastrophes et l'éducation des catastrophes

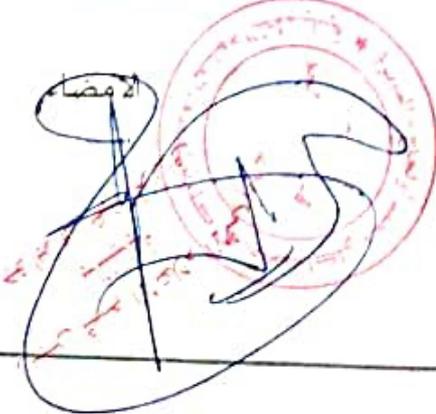
أصرح بشرفي اني ألتزم بمراعاة المعايير العلمية والمنهجية الاخلاقيات المهنية والنزاهة الاكاديمية المطلوبة في انجاز البحث المذكور أعلاه.

توقيع المعني (ة)

التاريخ: 25 - 06 - 2023

هيئة مراقبة السرقة العلمية:

البويرة في: .....



%

27

النسبة:



## التصريح الشرفي الخاص بالالتزام بقواعد النزاهة العلمية للإنجاز بحث

انا الممضي اسفله،

السيد (ة) ..... عماري عقيلة ..... الصفة: طالب ( ماستر / دكتوراه )

الحامل (ة) لبطاقة التعريف الوطنية: ..... 100524602 ..... والصادرة بتاريخ: 2016.04.12

المسجل (ة) بكلية / معهد ..... رياضيات وعلوم الى ..... قسم ..... اعلام الى

تخصص: ..... هندسة الانظمة المعلوماتية والبرمجيات

والمكلف (ة) بإنجاز اعمال بحث (ملاكرة، التخرج، مذكرة ماستر، مذكرة ماجستير، اطروحة دكتوراه).

عنوانها: ..... Conception d'un modèle d'apprentissage à partir des

réseaux sociaux pour la gestion des catastrophes et

l'éducation des catastrophes

أصرح بشرفي اني ألتزم بمراعاة المعايير العلمية والمنهجية الاخلاقيات المهنية والنزاهة الاكاديمية المطلوبة  
في انجاز البحث المذكور أعلاه.

التاريخ: ..... 24.06.2023

توقيع المعني (ة)

الورقة ل: .....	هيئة مراقبة السرقة العلمية:
	النسبة: % 27

# Dédicaces

“

*Les mots ne peuvent jamais exprimer mon amour et ma gratitude envers mon père  
'CHERIF'.*

*Mon amour et mon affection vont à ma chère mère 'ZOHRA' à qui je dois une  
grande dette.*

*À ma précieuse grand-mère 'FATIMA'.*

*À mes trois chers frères : SOFIANE et Sa Femme, KHALED et Sa Femme et ses  
enfants 'ADEM et MARAM', et AISSA.*

*À mes chères sœurs : HOUDA, HOURIA et son mari ALI, leurs enfants : ALAA,  
AHMED et MARIA.*

*À mes meilleurs amis qui ne m'ont jamais laissé tomber.*

*À toute personne qui aura l'occasion de lire ce modeste travail.*

”

**- Amina**

# Dédicaces

“

*A mes chers parents "Amar Omari ,Hadda Abd elmoumene " qui m'ont soutenu à chaque étape dans ma vie .*

*A mes chers sœurs : Aldja , Hanane , Nassrine et leurs enfants : Amina, Inas, Chaima, Abd allah, Rahil, Khalil .*

*A mes chers frères : Abd ellatif, Abd elaziz.*

*A mon cher mari "Haicheur Mahfoud" .*

*Je tiens à exprimer ma reconnaissance envers tous les enseignants qui m'ont accompagné tout au long de mon parcours universitaire.*

*Enfin, je souhaite dédier ce mémoire à moi-même, en reconnaissant les efforts, la persévérance et la détermination que j'ai déployés tout au long de cette expérience d'apprentissage. C'est avec fierté et accomplissement que je franchis cette étape de ma vie académique.*

”

**- Akila Omari**

# Remerciements

Tout d'abord, nous exprimons notre gratitude envers Dieu Tout-Puissant pour avoir instillé en nous le courage et la patience nécessaires pour mener à bien ce travail.

Cette thèse marque la fin de notre parcours universitaire et l'obtention de notre diplôme. Nous tenons à exprimer notre reconnaissance envers toutes les personnes qui nous ont soutenus tout au long de cette période.

Nous souhaitons adresser nos chaleureux remerciements à notre directeur de recherche, avec qui nous avons eu le privilège de collaborer.

Nous sommes également profondément reconnaissants envers tous les membres de notre famille, qui ont été une source constante de soutien et de motivation tout au long de nos études. Ils ont été présents dans les moments difficiles, nous encourageant à exceller et à atteindre nos objectifs académiques.

Enfin, nous exprimons notre sincère gratitude envers toutes les personnes qui ont contribué, de quelque manière que ce soit, à la réalisation de cette thèse, que ce soient nos camarades de classe, nos professeurs ou les membres de la communauté scientifique. Leurs contributions et encouragements ont joué un rôle crucial dans la concrétisation de cette recherche.

Nous vous remercions tous du fond du cœur.

# Résumé

Les médias sociaux, tels que Twitter, jouent un rôle crucial en tant que source d'informations importantes sur les catastrophes. Dans cette étude, nous proposons deux modèles capable de détecter les risques majeurs et les grandes catastrophes en exploitant les tweets et en utilisant des modèles d'apprentissage profond tels que LSTM et BERT.

Avec la participation de millions d'utilisateurs partageant leurs histoires et leurs préoccupations sur Twitter, un flux massif de tweets est généré. Cependant, trouver des informations pertinentes parmi cette abondance de tweets nécessite des efforts considérables.

Les résultats obtenus lors des tests de classification ont montré une précision et un rappel satisfaisants, ce qui constitue une première étape prometteuse dans la détection des catastrophes par l'analyse des tweets. Cette étude offre donc une occasion précieuse pour faciliter une réponse plus rapide à ces situations critiques en analysant le contenu des médias sociaux.

---

**Mots clés :** Apprentissage Profond, Classification, Catastrophe, tweets, BERT, LSTM.

---

# Abstract

Social media platforms, such as Twitter, play a crucial role as important sources of information about disasters. In this study, we propose two models capable of detecting major risks and large-scale disasters by leveraging tweets and using deep learning models such as LSTM and BERT.

With the participation of millions of users sharing their stories and concerns on Twitter, a massive flow of tweets is generated. However, finding relevant information among this abundance of tweets requires significant effort.

The results obtained from the classification tests showed satisfactory precision and recall, which represents a promising first step in disaster detection through tweet analysis. This study offers a valuable opportunity to facilitate a faster response to these critical situations by analyzing the content of social media.

---

**Keywords :** BERT, Deep Learning , LSTM, Classification, Disaster, Tweets.

---

## ملخص

تلعب منصات وسائل التواصل الاجتماعي، مثل تويتر، دورًا حاسمًا كمصادر هامة للمعلومات عن الكوارث. في هذه الدراسة، نقترح نموذجين قادرين على اكتشاف المخاطر الكبرى والكوارث ذات المقياس الكبير من خلال استغلال التغريدات واستخدام نماذج التعلم العميق مثل LSTM وBERT. مع مشاركة الملايين من المستخدمين الذين يشاركون قصصهم واهتماماتهم على تويتر، يتم إنشاء تدفق ضخم من التغريدات. ومع ذلك، فإن العثور على المعلومات ذات الصلة بين هذا الكم الهائل من التغريدات يتطلب جهدًا كبيرًا أظهرت النتائج التي تم الحصول عليها من اختبارات التصنيف دقة واسترجاع مرضية، مما يمثل خطوة واعدة في اكتشاف الكوارث من خلال تحليل التغريدات تقدم هذه الدراسة فرصة قيمة لتسهيل الاستجابة الأسرع لهذه الحالات الحرجة من خلال تحليل محتوى وسائل التواصل الاجتماعي .

---

كلمات مفتاحية : التعلم العميق ، التصنيف، الكوارث، التغريدات.

---

# Table des matières

Dédicaces . . . . .	I
Dédicaces . . . . .	II
Remerciements . . . . .	III
Résumé . . . . .	IV
Abstract . . . . .	V
VI . . . . .	ملخص
Introduction générale . . . . .	1
<b>1 Généralités et concepts de base . . . . .</b>	<b>3</b>
1.1 Introduction . . . . .	4
1.2 Catastrophe . . . . .	4
1.2.1 Définition . . . . .	4
1.3 La Gestion des catastrophes . . . . .	5
1.4 Le Cycle de gestion des catastrophes . . . . .	5
1.5 Réseaux sociaux . . . . .	6
1.6 Twitter . . . . .	7
1.7 Education . . . . .	7
1.8 Conclusion . . . . .	7
<b>2 Apprentissage profond . . . . .</b>	<b>8</b>
2.1 Introduction . . . . .	9
2.2 Définition . . . . .	10
2.2.1 Traitement du langage naturel(NLP) . . . . .	10
2.2.2 Apprentissage Automatique . . . . .	10
2.2.3 Apprentissage profond . . . . .	10
2.3 Réseaux de neurones artificiels . . . . .	10
2.3.1 Définition . . . . .	10
2.3.2 Fonctions d'activation . . . . .	11
2.4 Quelques méthodes d'apprentissage profond . . . . .	13
2.4.1 Réseaux neuronaux récurrents (RNN) . . . . .	13
2.4.2 Mémoire longue à court terme (LSTM) . . . . .	14
2.4.3 Réseaux de Neurones Transformers . . . . .	16
2.4.4 Représentation de l'encodeur bidirectionnel des transformateurs BERT . . . . .	17
2.4.5 Architecture de BERT . . . . .	18
2.5 Evaluation des performances . . . . .	20
2.5.1 La Matrice de Confusion . . . . .	20
2.5.2 Critère dévaluation . . . . .	21
2.6 Problème d'underfitting et d'overfitting . . . . .	21
2.6.1 Principe de l'overfitting . . . . .	21
2.6.2 Principe de l'underfitting . . . . .	21
2.6.3 quelques approches pour réduire l'overfitting : . . . . .	22

2.7	Conclusion . . . . .	23
<b>3</b>	<b>Etat de l'art . . . . .</b>	<b>24</b>
3.1	Introduction . . . . .	25
3.2	Ouvrages connexes . . . . .	25
3.3	Analyse et Comparaison . . . . .	26
3.4	Conclusion . . . . .	29
<b>4</b>	<b>Modélisation et Implémentation . . . . .</b>	<b>30</b>
4.1	Introduction . . . . .	31
4.2	Environnement de développement . . . . .	31
4.2.1	Matériel . . . . .	31
4.2.2	Les logiciels . . . . .	31
4.3	Description du dataset . . . . .	34
4.4	Modèle LSTM . . . . .	35
4.4.1	Le Prétraitement . . . . .	35
4.4.2	Le traitement . . . . .	36
4.4.3	Évaluation du modèle . . . . .	38
4.5	Modèle BERT . . . . .	39
4.5.1	Analyse des ensembles de données . . . . .	39
4.5.2	Fractionner l'ensemble de données . . . . .	39
4.5.3	Préprocesseur et Encodeur BERT . . . . .	39
4.5.4	Création du modèle . . . . .	39
4.5.5	Entraînement et Evaluation le modèle . . . . .	40
4.6	Résultats du modèle LSTM . . . . .	40
4.7	Résultats du modèle BERT . . . . .	43
4.8	Comparaison et analyse des performances . . . . .	44
4.9	Conclusion . . . . .	45
	<b>Conclusion et perspectives . . . . .</b>	<b>47</b>
	<b>Bibliographie . . . . .</b>	<b>49</b>

# Table des figures

1.1	Cycle de gestion des catastrophes. . . . .	6
2.1	AI vs ML vs DL [11] . . . . .	9
2.2	Machine Learning et Deep Learning [12]. . . . .	9
2.3	Architecture du réseau neuronal artificiel[15]. . . . .	11
2.4	Une structure en couches de neurones Learning[16]. . . . .	11
2.5	Graphe de fonction sigmoïde [17]. . . . .	12
2.6	Graphe de fonction ReLU [17]. . . . .	12
2.7	Graphe de fonction Softmax [17]. . . . .	13
2.8	Graphe de fonction tanh [18]. . . . .	13
2.9	La représentation de l'exécution des réseaux de neurones récurrents.[19]. . . . .	14
2.10	Cellule LSTM[20]. . . . .	15
2.11	La porte d'oubli[20]. . . . .	15
2.12	La porte d'entrée[20]. . . . .	15
2.13	La porte d'entrée[20]. . . . .	16
2.14	la porte de sortie[20]. . . . .	16
2.15	Différentes variantes de BERT . . . . .	18
2.16	Architecture de BERT[29]. . . . .	18
2.17	Input Emedding [30]. . . . .	19
2.18	Auto-attention multi-têtes[22]. . . . .	20
2.19	La Matrice de Confusion [31] . . . . .	20
2.20	Overfitting / Underfitting [31] . . . . .	22
2.21	Dropout : une méthode simple pour prévenir l'overfitting des réseaux neuronaux. [36] . . . . .	22
4.1	Google Colab logo . . . . .	31
4.2	Python logo. . . . .	32
4.3	TensorFlow. . . . .	32
4.4	Keras. . . . .	33
4.5	Numpy. . . . .	33
4.6	Pandas. . . . .	33
4.7	Scikit-learn. . . . .	34
4.8	nombre de tweets positifs et négatifs . . . . .	34
4.9	Les cinq premières lignes de dataset . . . . .	34
4.10	Résumé du modèle LSTM . . . . .	38
4.11	Structure du modèle BERT . . . . .	40
4.12	Variation de la perte(loss) en fonction du nombre d'époques . . . . .	41
4.13	Accuracy en fonction du nombre d'époques . . . . .	41
4.14	Variation de la précision en fonction du nombre d'époques . . . . .	41
4.15	Evolution du rappel(recall) en fonction du nombre d'époques . . . . .	42
4.16	visualisation graphique de la matrice de confusion . . . . .	42
4.17	Receiver Operating Characteristic (ROC) . . . . .	42
4.18	Variation de la perte(loss) en fonction du nombre d'époques . . . . .	43
4.19	Accuracy en fonction du nombre d'époques . . . . .	43
4.20	Variation de la précision en fonction du nombre d'époques . . . . .	43
4.21	Evolution du rappel(recall) en fonction du nombre d'époques . . . . .	43
4.22	Visualisation graphique de la matrice de confusion . . . . .	44

4.23 Receiver Operating Characteristic (ROC) . . . . . 44

# Liste des tableaux

1.1	Derniers Evénements Catastrophiques . . . . .	5
2.1	Différentes variantes de BERT . . . . .	17
3.2	Étude comparative des travaux connexes . . . . .	27
4.1	Matériel . . . . .	31
4.2	Les Résultats . . . . .	44

# Liste des sigles et acronymes

<b>IA</b>	<i>Intelligence Artificiel</i>
<b>ML</b>	<i>Machine learning</i>
<b>DL</b>	<i>Deep Learning</i>
<b>NLP</b>	<i>Natural language Processing</i>
<b>Tanh</b>	<i>Tangente Hyperbolique</i>
<b>ReLU</b>	<i>Rectified Linear Unit</i>
<b>RNN</b>	<i>Recurrent neural network</i>
<b>LSTM</b>	<i>Long short-term memory</i>
<b>BLSTM</b>	<i>Bidirectional Long short-term memory</i>
<b>BERT</b>	<i>Bidirectional Encoder Representations from Transformers</i>
<b>CNN</b>	<i>Convolutional Neural Network</i>
<b>SVM</b>	<i>Support vector machine</i>
<b>GB</b>	<i>Gradient Boosting</i>
<b>NLTK</b>	<i>Natural Language Toolkit</i>
<b>NumPy</b>	<i>Numerical Python</i>
<b>ROC</b>	<i>Receiver Operating Characteristic</i>
<b>AUC-ROC</b>	<i>Area Under the Receiver Operating Characteristic Curve</i>
<b>TPR</b>	<i>True Positive Rate</i>
<b>FPR</b>	<i>Folse Positive Rate</i>

# Introduction générale

L'utilisation des médias sociaux dans la gestion des catastrophes constitue un outil puissant et efficace pour faciliter la communication, collecter des informations et coordonner les secours. Les médias sociaux sont devenus une plateforme populaire où les individus partagent des informations en temps réel, expriment leurs préoccupations et se mobilisent pour apporter leur aide en cas de crise.

En outre, les médias sociaux facilitent également la collecte d'informations provenant de diverses sources. Les personnes sur le terrain peuvent partager des photos, des vidéos et des témoignages, permettant aux gestionnaires de catastrophes de mieux comprendre la situation et de prendre des décisions éclairées. Les médias sociaux peuvent également être utilisés comme plate-forme pour recueillir des demandes d'assistance, des signalements de personnes disparues et d'autres informations importantes pour les opérations de secours.

L'intégration de l'apprentissage en profondeur dans la gestion des catastrophes basée sur les médias sociaux présente certains défis. En analysant les messages et les tweets liés aux catastrophes naturelles, des modèles et des informations précieux peuvent être extraits pour prédire les catastrophes potentielles. Les techniques d'apprentissage en profondeur nous permettent d'analyser les mégadonnées provenant de ces plateformes et de découvrir des modèles et des relations qu'il serait difficile de découvrir manuellement. Des modèles tels que les réseaux de neurones profonds et les techniques de traitement du langage naturel peuvent être utilisés pour classer et analyser les tweets et les messages liés aux catastrophes.

Grâce à ces analyses, les mots clés, les phrases et les déclencheurs partagés dans les tweets liés aux catastrophes naturelles peuvent être identifiés, ce qui contribue à la surveillance et à la prévision des catastrophes potentielles. Par exemple, ces techniques peuvent être utilisées pour identifier le modèle linguistique des tweets qui font référence à des tremblements de terre, des inondations ou des ouragans, ce qui aide à la prédiction précoce et à la réponse rapide.

En général, l'utilisation des plateformes de médias sociaux pour prévoir les catastrophes naturelles et l'apprentissage en profondeur améliorent l'efficacité de la réponse et de la préparation aux catastrophes, ce qui contribue à protéger les vies et les biens et à réduire l'impact négatif de ces catastrophes.

## Problématique et objectifs

Aujourd'hui, la gestion des catastrophes représente un défi majeur face à la multiplication d'événements imprévus et dévastateurs. Malgré l'abondance de données textuelles en temps réel sur les réseaux sociaux, il demeure complexe d'analyser efficacement ces informations pour en extraire des éléments pertinents liés aux catastrophes. Notre objectif dans ce projet est de mettre en place un modèle d'apprentissage profond qui utilise le traitement des données textuelles pour améliorer considérablement la capacité à exploiter les données des réseaux sociaux dans la gestion des catastrophes. Cette approche permettra de faciliter une réponse plus rapide, mieux informée et plus efficace face à ces événements critiques.

## Organisation du mémoire

Ce mémoire est organisé comme suit :

chapitre "1" : **"Généralités et concepts de base"** nous allons présenter introduction générale sur

la gestion des catastrophes et présenter les concepts de base nécessaires à la compréhension des enjeux liés à l'analyse des données textuelles sur les réseaux sociaux

chapitre "2" : "**Apprentissage Profond**" Nous présentons de manière générale l'apprentissage profond. Ses principes et quelques méthodes intéressantes d'apprentissage Profond pour notre travail.

Chapitre "3" : "**Etat de l'art**" nous allons aborder d'une manière abrégée les travaux antérieurs qui se portent sur les multiples techniques de ML et DL avec divers datasets.

Chapitre "4" : "**Modélisation et Implémentation**" nous détaillons la construction et l'application de notre modèle, en expliquant les choix faits à chaque étape et en fournissant une évaluation approfondie des résultats obtenus. Cela nous permet de comprendre l'efficacité et la pertinence de notre modèle dans le contexte de notre étude.

# Chapitre 1

## Généralités et concepts de base

### 1.1 Introduction

nous entamons notre premier chapitre par présenter les fondements essentiels liés aux généralités et les principes de base en matière de gestion des catastrophes. Nous commencerons par explorer l'origine du concept de risque "majeur", qui a été introduit pour la première fois en France en 1981 par Patrick LAGADEC dans son ouvrage réputé "La civilisation du risque : Catastrophes technologiques et responsabilité sociale". Depuis lors, ce concept a suscité d'importants débats politiques, administratifs et scientifiques, mettant en lumière les risques technologiques et naturels auxquels nous sommes confrontés.

Nous examinerons également la distinction entre les notions de risque et de catastrophe. Alors que la catastrophe est définie comme un événement soudain et réel causant des dommages dépassant un seuil acceptable par la société, le risque se caractérise par l'incertitude et la probabilité de survenance d'un événement indésirable. Nous mettrons en évidence l'interdépendance entre les risques et les catastrophes, soulignant l'importance de comprendre les aléas et la vulnérabilité dans la gestion de ces situations critiques.

De plus, nous explorerons l'importance croissante des réseaux sociaux, en mettant en évidence la plateforme Twitter, en tant que sources d'informations précieuses sur les catastrophes. Les millions d'utilisateurs qui partagent leurs histoires et préoccupations génèrent un volume considérable de tweets, constituant une mine d'informations potentiellement utiles. Toutefois, il est essentiel de pouvoir extraire les informations pertinentes de cette masse de données, ce qui requiert des efforts significatifs.

Enfin, nous soulignerons l'objectif de notre étude qui vise à exploiter les données textuelles des médias sociaux en utilisant des modèles d'apprentissage profond pour détecter les risques majeurs et les catastrophes à partir des tweets. Les résultats de nos tests de classification offrent une première étape prometteuse pour améliorer la détection des catastrophes naturelles grâce à l'analyse des tweets.

En somme, ce chapitre jettera les bases nécessaires pour comprendre les concepts clés de la gestion des catastrophes et soulignera l'importance des médias sociaux dans cette discipline.

### 1.2 Catastrophe

#### 1.2.1 Définition

Les catastrophes telles que les tremblements de terre, les inondations, les incendies et les tsunamis entraînent des souffrances humaines désastreuses, des destructions de biens et d'autres effets néfastes.

En plus des catastrophes existantes, plusieurs catastrophes anthropiques ont surgi au cours des deux dernières décennies, principalement en raison de la mondialisation, des réseaux interconnectés et d'une croissance technologique importante. La contrefaçon de produits, les risques biologiques. [1, 2].

No	Événement Catastrophique	Période	Dommages
1	Tremblement de terre Haïti	Oct 2007	230 000 décès
2	Feu de forêt Californie	Jan 2010	203 décès, 6 152.9 Km2 de terres ravagées
3	Inondations Thaïlande	juillet 2011	815 décès
4	Tsunami tremblement de terre Japon	Avril 2011	15 896 décès, 6 157 blessées, 2 537 disparus
5	L'ouragan Sandy USA	Oct 2012	220 décès
6	Typhoo Haiyan Philippines	Nov 2013	26 626 blessées
7	Inondation de l'elbe Allemagne	Juin 2013	25 décès
8	Attentat dans le métro Russie	Avril 2017	15 décès, 50 blessées
9	Attentat suicide Angleterre	Mai 2017	22 décès, 116 blessées
10	Trois explosions Indonésie	Mai 2018	9 décès, 40 blessées
11	Japon Inondations Japon	Juin 2018	235 décès, 13 disparus
12	Tremblement de terre en Indonésie	Sep 2018	2 000 décès, 1.5 million blessées
13	Séisme Incendie Haïti	Oct 2018	18 décès, 548 blessées
14	Attaque terroriste Strasbourg	Dec 2018	5 décès, 10 blessées
15	Épidémie d'Ebola au Kivu Congo	Août 2018- Juin 2020	14 739 450 malades, 1 162 guéris, 2299 décès
16	Pandémie de coronavirus COVID-19	21jan-23 Juil 2020	14 739 450 malades, 8 332 461 guéris, 610 776 décès

TAB. 1.1 : Derniers Evénements Catastrophiques

### 1.3 La Gestion des catastrophes

Les étapes suivantes sont incluses dans le cycle de récupération après une catastrophe : avertissement, planification, action, prévention, atténuation et restauration. En matière de gestion de catastrophes, il existe au moins six éléments clés [2] : prévention, atténuation, planification, réponse (secours), restauration et reconstruction. Le processus de gestion des urgences, cependant, est défini en quatre phases, à savoir : atténuation (avant la catastrophe)[3], préparation (avant la catastrophe)[4], réponse (pendant la catastrophe)[3] et récupération (après la catastrophe)

### 1.4 Le Cycle de gestion des catastrophes

C'est un processus continu qui comprend plusieurs phases distinctes, visant à prévenir, atténuer, préparer, répondre et récupérer les communautés touchées par une catastrophe. Voici un aperçu du cycle de gestion des catastrophes :

**Prévention** : éviter ou à réduire les risques de catastrophe. Elle implique l'identification des menaces potentielles, l'évaluation des vulnérabilités, la mise en place de mesures de prévention telles que la réglementation, la sensibilisation et l'éducation des populations, ainsi que la mise en œuvre de projets d'atténuation des risques.

**Atténuation (avant la catastrophe)** : minimiser les impacts potentiels d'une catastrophe. Elle implique la mise en place de mesures structurelles et non structurelles pour réduire les risques, telles que la construction de structures résistantes aux catastrophes, la gestion durable des ressources naturelles et la mise en place de systèmes d'alerte précoce.

**Préparation (avant la catastrophe)** : se préparer à faire face à une catastrophe imminente. Elle implique la planification d'urgence, la formation et l'exercice des équipes de réponse aux catastrophes, la mise en place de plans d'évacuation, la préparation des abris d'urgence et la constitution de stocks de fournitures d'urgence.

**Réponse** : Cette phase survient pendant et immédiatement après une catastrophe. Elle implique la mobilisation des ressources d'urgence, la coordination des secours, la gestion des abris d'urgence, la fourniture de soins médicaux, la recherche et le sauvetage, ainsi que la communication et l'information aux populations affectées.

**Récupération (après la catastrophe)** : restaurer et reconstruire les communautés touchées par une catastrophe. Elle implique la réhabilitation des infrastructures endommagées, la réparation des services essentiels, la réinstallation des populations déplacées, la reconstruction des moyens de subsistance, ainsi que la promotion de la résilience à long terme.

Il est important de noter que ces phases du cycle de gestion des catastrophes sont interdépendantes et nécessitent une approche holistique pour assurer une gestion efficace des risques et des catastrophes. La prévention et la préparation sont des éléments clés pour minimiser les pertes potentielles, tandis que la réponse et la récupération sont essentielles pour faire face aux conséquences immédiates et à long terme d'une catastrophe[5].

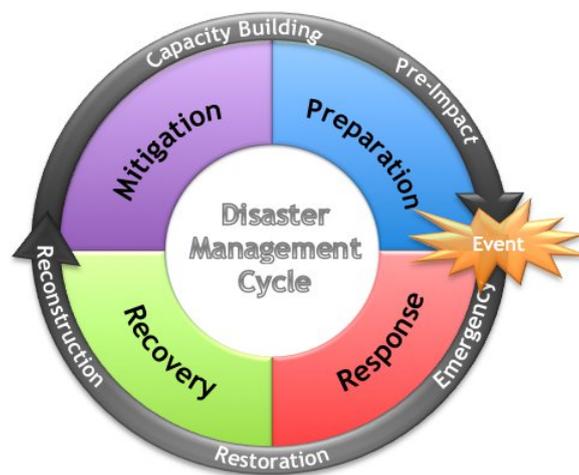


FIG. 1.1 : Cycle de gestion des catastrophes.

[1]

## 1.5 Réseaux sociaux

Les réseaux sociaux jouent un rôle très important dans la gestion des catastrophes. Ils servent à collecter et classer les contenus (informations et images) publiés sur les plateformes de réseaux sociaux pour aider les organisations humanitaires à prendre conscience de la situation et à lancer des opérations de secours. Cependant, comprendre ces données volumineuses et à haute vitesse est une tâche difficile qui nécessite l'utilisation des techniques de crowdsourcing, de crowdtasking et de gestion collaborative des catastrophes.

Avec l'augmentation des sites et des applications des réseaux sociaux, ceux-ci ont attiré l'attention des chercheurs et des praticiens. D'une part, les praticiens développent des compétences pour mieux utiliser cette technologie de communication en ligne d'autre part, les chercheurs développent des modèles pour examiner divers aspects de l'utilisation des réseaux sociaux dans différents environnements et pour la gestion des catastrophes. Il est important de noter que les médias sociaux ne sont pas considérés comme un substitut aux médias traditionnels tels que la télévision, la radio et les journaux, mais leur importance est indéniable en tant que moyen de communication supplémentaire [4], surtout lorsque le partage d'informations est essentiel - comme dans la gestion des catastrophes.

### 1.6 Twitter

Twitter est défini par Mollett, Moran et Dunleavy (2011) comme : "Un type de microblogging gratuit qui permet aux utilisateurs d'envoyer et de recevoir de courts messages publics appelés tweets.

Twitter est défini par Mollett, Moran et Dunleavy (2011, p. 1) comme suit : "Une forme de microblogging gratuit qui permet aux utilisateurs d'envoyer et de recevoir de courts messages publics appelés tweets. Les tweets sont limités à 140 caractères maximum et peuvent inclure des liens vers des blogs, des pages web, des images, des vidéos et tout autre contenu en ligne". Comme plus de 200 millions de tweets sont envoyés chaque jour dans le monde entier (Marketing Gum, 2011), il est impossible de les suivre sans un outil d'organisation. C'est pourquoi les hashtags (#) et les symboles "@" sont utilisés. Les tweets envoyés avec le symbole "@" sont généralement adressés à une personne, tandis que le hashtag "#" est utilisé pour indiquer un sujet d'intérêt, par exemple "#TwitterEdTalk" ou "#teaching". D'autres moyens de suivre les sujets et les personnes qui intéressent l'utilisateur sont les programmes logiciels et/ou les applications sur les appareils mobiles[6].

### 1.7 Education

L'éducation est importante et représente une priorité, car si les êtres humains ne prennent pas conscience des risques de catastrophe, n'acquièrent pas les connaissances nécessaires et ne développent pas les comportements, attitudes et niveaux d'implication appropriés, ils ne seront pas en mesure de les prévenir. Il est important que l'éducation facilite et contribue à la création d'une culture de prévention et que la population et les communautés prennent des mesures pour se préparer à une catastrophe.

L'éducation en matière de gestion de catastrophes joue un rôle clé, tout d'abord en aidant les citoyens à prendre conscience des dangers des catastrophes, puis en les encourageant à développer des compétences en matière de gestion de catastrophes. Cet environnement est conçu pour soutenir une formation d'initiation à la gestion des catastrophes pour les stagiaires et les futurs gestionnaires de catastrophes. Les membres de la communauté partagent généralement leurs connaissances et s'entraident en matière de préparation aux catastrophes. Des exercices d'évacuation ont souvent lieu dans les écoles, les entreprises et les communautés [4], [7], [8], [9], [10].

### 1.8 Conclusion

enfin, nous avons souligné l'importance cruciale de comprendre les notions fondamentales et les généralités relatives à la gestion des catastrophes. Nous avons défini ce qu'est une catastrophe et examiné le cycle de gestion associé. De plus, nous avons souligné le rôle des réseaux sociaux, en particulier Twitter, dans la collecte et l'analyse des données textuelles liées aux catastrophes. Enfin, nous avons abordé l'aspect de l'éducation dans la gestion des catastrophes, soulignant son importance pour une réponse plus efficace et mieux informée. Ces concepts et notions jetteront les bases nécessaires pour explorer plus en profondeur l'utilisation de l'apprentissage profond dans ce domaine dans les chapitres suivants.

## Chapitre 2

# Apprentissage profond

## 2.1 Introduction

L'Intelligence Artificielle (IA) consiste à mettre en œuvre des systèmes informatiques pouvant imiter l'intelligence humaine et même de prendre des décisions. Le terme Apprentissage automatique ou Machine Learning (ML) date du milieu du 20<sup>ème</sup> siècle. En 1950, le mathématicien britannique Alan Turing a proposé la première machine capable d'apprendre et de simuler l'intelligence réelle. Depuis, plus d'un demi-siècle, les techniques d'IA sont devenues pluridisciplinaires et connues un développement considérablement dans le but de créer des algorithmes capables d'apprendre et de s'améliorer de manière autonome. Ce qui fait l'appariation de l'apprentissage profond qui est un algorithme de haut niveau d'abstraction qui permet de modéliser des grands ensembles de données.

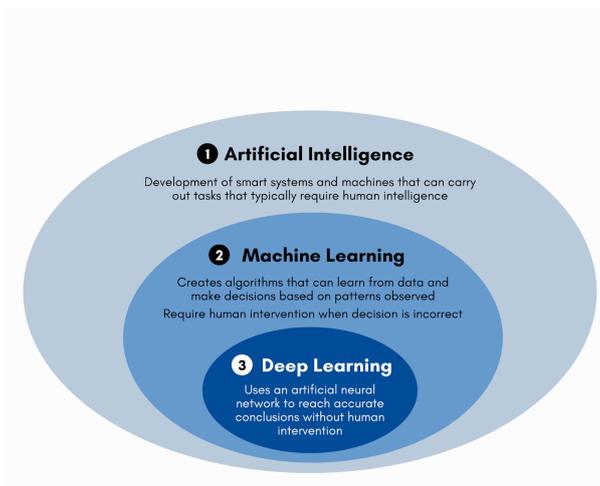


FIG. 2.1 : AI vs ML vs DL [11]

Les réseaux d'apprentissage profond (Deep Learning) dérive de l'apprentissage automatique qui été largement utilisé dans différents domaines quelque soit leur degrés de complexité grâce à son avantage d'extraction automatique des caractéristiques la chose pour laquelle il a prouvé son efficacité par rapport aux méthodes d'apprentissage classique.

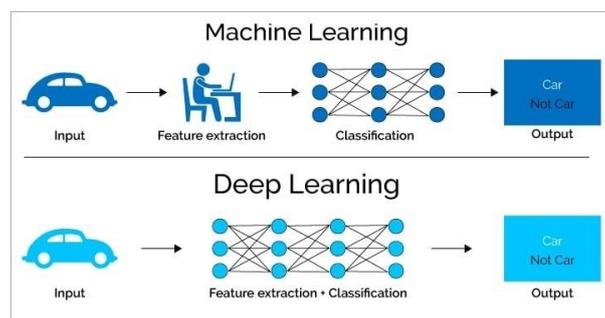


FIG. 2.2 : Machine Learning et Deep Learning [12].

Dans ce chapitre, nous allons explorer les principes fondamentaux de l'apprentissage profond, en mettant l'accent sur deux modèles spécifiques. Nous examinerons leur structure, leur mécanisme de traitement du langage naturel et leur capacité de classification de texte. Nous discuterons également de l'utilisation des réseaux de neurones, des fonctions d'activation et des méthodes d'évaluation des performances des modèles. En outre, nous aborderons les problèmes courants rencontrés lors du traitement des données et les approches pour les résoudre.

## 2.2 Définition

### 2.2.1 Traitement du langage naturel(NLP)

Le Traitement du Langage Naturel (NLP) est un domaine qui se concentre principalement sur l'analyse, la manipulation et la génération de langage naturel par des machines. Il joue un rôle crucial en faisant le lien entre l'informatique et la linguistique, en permettant aux machines d'interagir directement avec les êtres humains.

### 2.2.2 Apprentissage Automatique

machine learning, est une branche de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances dans des tâches spécifiques. Les algorithmes d'apprentissage automatique utilisent des méthodes statistiques et des modèles mathématiques pour extraire des informations des données, sans programmation explicite. Les données d'entraînement sont utilisées pour enseigner au modèle les motifs nécessaires à la prise de décisions ou aux prédictions. L'apprentissage automatique est appliqué dans de nombreux domaines tels que la reconnaissance vocale, la vision par ordinateur et le traitement du langage naturel. Il permet aux machines d'acquérir des connaissances à partir des données et de prendre des décisions autonomes, ouvrant ainsi de vastes possibilités dans différents domaines.

### 2.2.3 Apprentissage profond

Actuellement les plus populaires dans la recherche et les applications du traitement du langage naturel (NLP) sont les modèles d'apprentissage profond. Ils généralisent encore mieux que les approches classiques d'apprentissage machine. Ils n'ont pas besoin de fonctionnalités créées manuellement ni d'ingénierie des fonctionnalités, car ils fonctionnent automatiquement comme des extracteurs de fonctionnalités, ce qui permet une formation de modèle de bout en bout. Les capacités d'apprentissage des modèles d'apprentissage profond sont plus puissantes que celles des modèles d'apprentissage automatique superficiels/classiques, ce qui leur a permis d'avoir les meilleurs résultats dans diverses tâches complexes de NLP.

## 2.3 Réseaux de neurones artificiels

### 2.3.1 Définition

Un réseau neuronal artificiel est un sous-domaine du cœur des algorithmes d'apprentissage profond (Deep Learning). La conception des réseaux neuronaux s'inspire du fonctionnement des réseaux neuronaux biologiques, ainsi plusieurs neurones (ou nœuds) sont interconnectés pour programmer les ordinateurs afin qu'ils puissent prendre des décisions, résoudre des problèmes de classification ou de reconnaissance. Les réseaux neuronaux sont plus performants lorsque les données d'entrée sont massives et variées, car l'algorithme apprend de l'expérience[14].

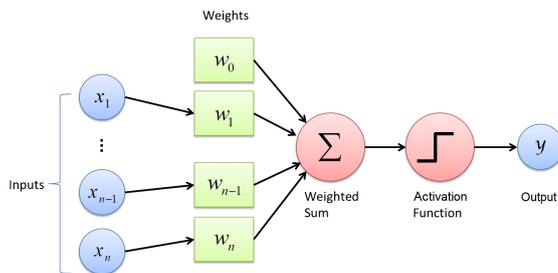


FIG. 2.3 : Architecture du réseau neuronal artificiel[15].

Un réseau neuronal artificiel se compose de trois composants importants : la couche input qui accepte tous les types de données, la couche cachée où les entrées sont examinées et où tous les calculs sont effectués. Il est à noter qu'en fonction de l'algorithme d'apprentissage, il est possible d'avoir une propagation de l'information à reculons (Backpropagation), chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante. Enfin, le dernier composant est la couche output renvoie les résultats de la prédiction en utilisant les sorties de la couche cachée.

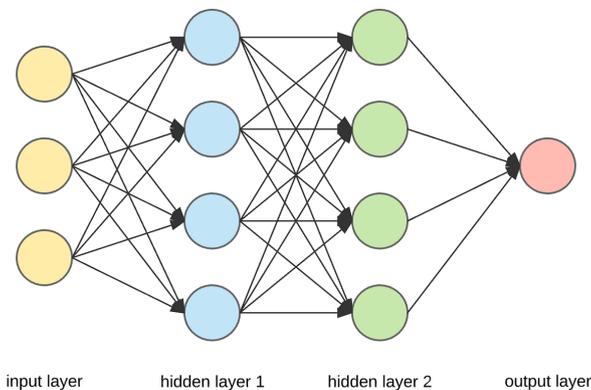


FIG. 2.4 : Une structure en couches de neurones Learning[16].

### 2.3.2 Fonctions d'activation

Une fonction d'activation est une fonction mathématique appliquée aux sorties des neurones dans un réseau de neurones artificiels. Elle introduit une non-linéarité dans le modèle et permet de modéliser des relations complexes entre les entrées et les sorties. Les fonctions d'activation ont un rôle majeur dans l'apprentissage des réseaux de neurones en permettant la propagation du signal à travers le réseau et en déterminant l'état d'activation d'un neurone. Différentes fonctions d'activation, telles que la sigmoïde, ReLU, tanh, softmax, etc., sont utilisées en fonction du problème à résoudre et des propriétés souhaitées du modèle. Chaque fonction d'activation a ses propres caractéristiques et influences sur la représentation et l'apprentissage des données dans le réseau de neurones.

Nous allons mentionner quelques types de fonction d'activation :

**Fonction Sigmoïde** est une fonction mathématique utilisée dans les réseaux de neurones artificiels et d'autres modèles d'apprentissage automatique. Elle transforme une valeur d'entrée en une valeur de sortie comprise entre 0 et 1.

$$g(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

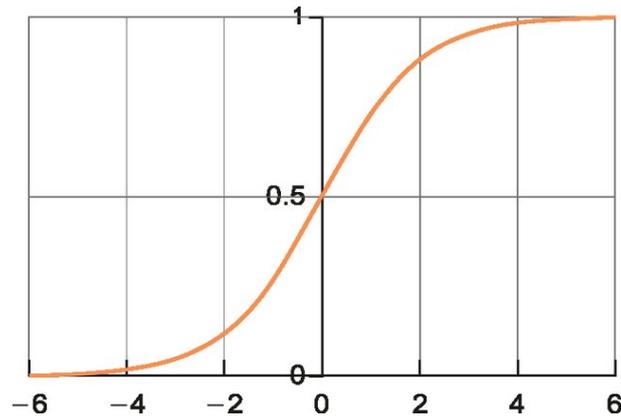


FIG. 2.5 : Graphe de fonction sigmoïde [17].

**Fonction ReLU** est une fonction non linéaire qui retourne la valeur d'entrée si elle est positive, sinon elle renvoie 0. Ainsi, lorsque la valeur d'entrée est positive, la fonction ReLU conserve cette valeur, tandis que pour les valeurs négatives, elle les annule. Cette propriété permet à la fonction ReLU d'introduire de la non-linéarité dans le modèle.

$$g(x) = \max(0, x) \tag{2.2}$$

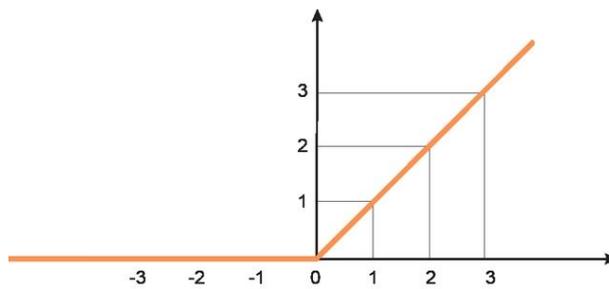


FIG. 2.6 : Graphe de fonction ReLU [17].

**Fonction Softmax** La fonction softmax applique la fonction exponentielle à chaque élément du vecteur d'entrée, puis normalise les résultats en les divisant par la somme de toutes les exponentielles. Cela garantit que les valeurs de sortie se situent entre 0 et 1, et que leur somme est égale à 1, ce qui les interprète comme des probabilités.

$$g(x) = \frac{e^{x_i}}{\sum_{i=1}^K e^{x_i}} \tag{2.3}$$

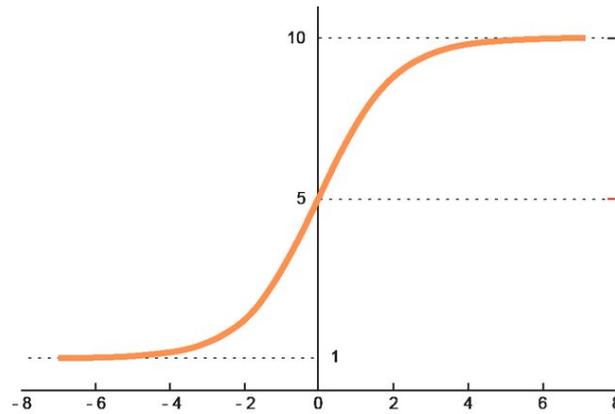


FIG. 2.7 : Graphe de fonction Softmax [17].

**Fonction Tanh** est une fonction mathématique permet transformer une valeur d'entrée en une valeur de sortie dans l'intervalle  $[-1, 1]$ . Pour les valeurs d'entrée positives, la sortie se rapproche de 1, pour les valeurs d'entrée négatives, la sortie se rapproche de -1, et pour une valeur d'entrée de zéro, la sortie est égale à zéro.

$$g(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.4)$$

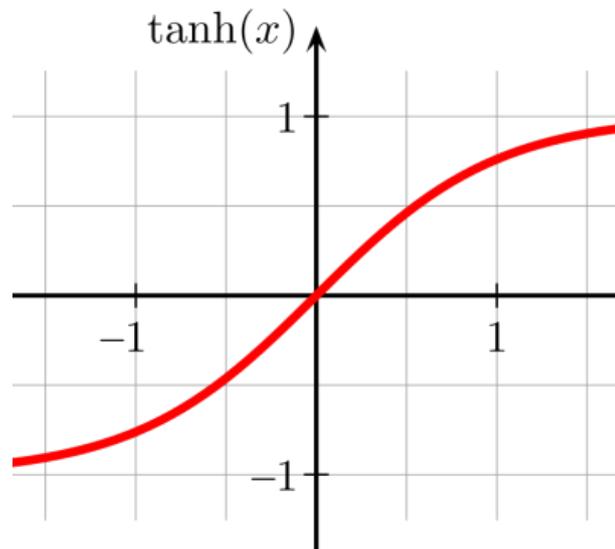


FIG. 2.8 : Graphe de fonction tanh [18].

## 2.4 Quelques méthodes d'apprentissage profond

### 2.4.1 Réseaux neuronaux récurrents (RNN)

Les gens ne pensent pas à nouveau à chaque seconde. Lorsque vous lisez un livre, vous comprenez chaque mot en fonction de votre compréhension du mot précédent. On oublie tout et on recommence à réfléchir. Nos pensées sont persistantes. Un inconvénient majeur réside dans l'incapacité des réseaux de

neurones traditionnels à réaliser cette tâche. Supposons que vous souhaitiez catégoriser les événements qui se produisent au cours de chaque phase d'un film. On ne sait pas comment les réseaux de neurones conventionnels utilisent l'inférence sur les événements antérieurs du film pour informer les événements ultérieurs. Les RNN traitent ce problème. Ce sont des réseaux avec des boucles qui peuvent conserver des informations

Les RNN sont appelés boucles car ils effectuent la même tâche pour chaque élément d'une séquence et la sortie dépend des calculs précédents.

Les RNN utilisent les sorties précédentes comme entrées supplémentaires, ce qui les rend bien adaptées au traitement des données. Par exemple, si la séquence que nous traitons est une phrase de trois termes, le réseau est déployé dans un réseau de neurones à trois couches (une couche par mot). Le schéma suivant illustre cette idée :

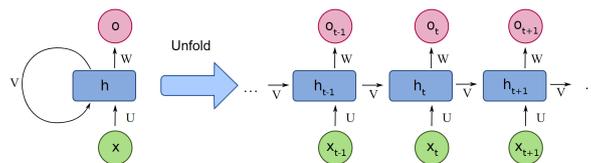


FIG. 2.9 : La représentation de l'exécution des réseaux de neurones récurrents.[19].

avec :

$x$  : L'entrée. Il peut être un mot dans une phrase ou un autre type de données séquentielle

$o$  : La sortie. Par exemple, ce que le réseau pense le prochain mot sur une phrase devrait être donné les mots précédents

$h$  : Le bloc principal du RNN. Il contient les poids et les fonctions d'activation du réseau

$v$  : Représente la communication d'un pas temporel à l'autre.

### 2.4.2 Mémoire longue à court terme (LSTM)

Les RNN traditionnels sont limités dans l'apprentissage et rencontrent souvent un problème pendant la phase d'apprentissage à cause de la longueur de la séquence temporelle .

la fonction d'activation  $\tanh$  utilisée dans le RNN prend trop de valeur proche de zéro lors des opérations de dérivés pendant la descente du gradient. De plus, les RNNs « classiques » ne sont capables de mémoriser que le passé dit proche, et Les LSTM permettent de passer outre ce problème – par l'utilisation de la fonction sigmoïde – et disposent d'une mémoire interne qui se modifie en permanence de manière dynamique en fonction des séquences des données d'entrée sur de très longues périodes de temps, Ils ont été introduits par Hochreiter Schmidhuber (1997) et sont devenus des technologies importantes pour le traitement des données temporelles. [20]

#### Le fonctionnement de la mémoire longue à court terme

Une cellule LSTM est une variante d'une couche récurrente qui facilite la transmission des signaux provenant de couches profondes antérieures vers la cellule en cours, elle est composée d'une porte d'oubli (Forget Gate), d'une porte d'entrée (Input Gate) et d'une porte de sortie (Output Gate) et gère une mémoire dynamique (notée  $C$ )

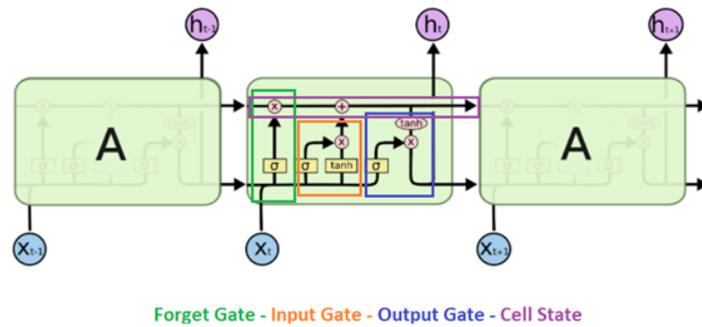


FIG. 2.10 : Cellule LSTM[20].

- **La porte d'oubli** : Cette porte détermine les informations à conserver ou à supprimer

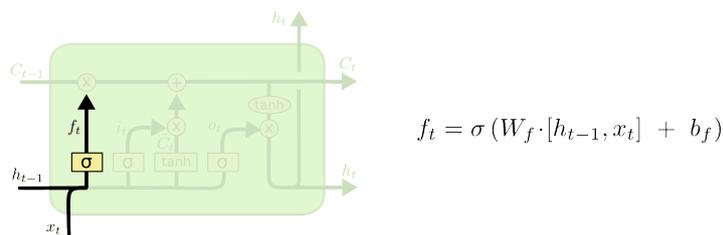


FIG. 2.11 : La porte d'oubli[20].

- **La porte d'entrée** : est d'extraire des informations à partir des données actuelles :

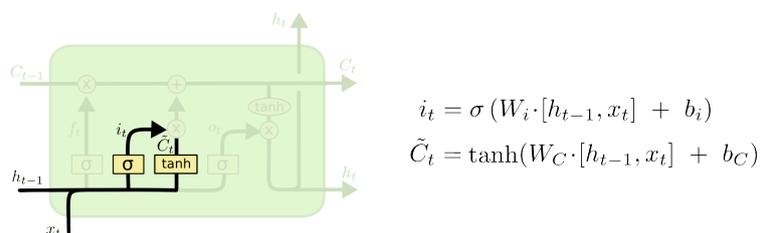


FIG. 2.12 : La porte d'entrée[20].

— Sigmoid renverra un vecteur avec des coordonnées proches de zéro, ce qui signifie que les coordonnées à la position équivalente dans le vecteur de jointure n'ont pas d'importance. Sinon, les coordonnées proches de 1 seront considérées comme "importantes".

— La fonction tanh normalisera simplement les valeurs (les comprimera) entre -1 et 1 afin d'éviter les problèmes de surcharge informatique lors des calculs.

— Ainsi, le produit des deux éléments permettra de conserver uniquement les informations importantes, tandis que les autres seront pratiquement remplacés par 0. Tel que :

$C_t$  : Une valeur candidate.

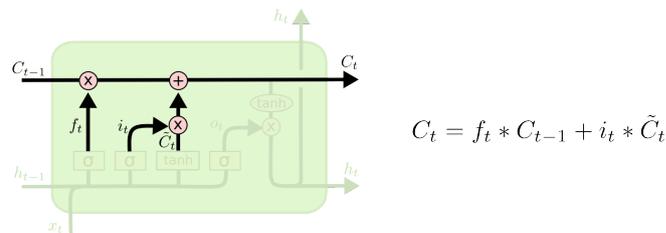


FIG. 2.13 : La porte d'entrée[20].

L'état d'une cellule peut être facilement calculé à partir de la porte d'oubli et de la porte d'entrée : Tout d'abord, multipliez les coordonnées par les coordonnées de la sortie de l'oubli et l'ancien état de la cellule. Cela permet d'oublier certaines informations de l'état précédent qui ne sont pas utilisées pour de nouvelles prédictions. Ajoutez ensuite le tout (coordonnée à coordonnée) à la sortie de la porte d'entrée. Cela permet à LSTM de stocker tout ce qu'il juge pertinent (états cachés d'entrée et précédents)

Tel que :

$C_t$  : État interne.

- **La porte de sortie** : doit décider quel doit être le prochain état caché, il contient des informations sur les entrées précédentes dans le réseau et il est utilisé pour prédire

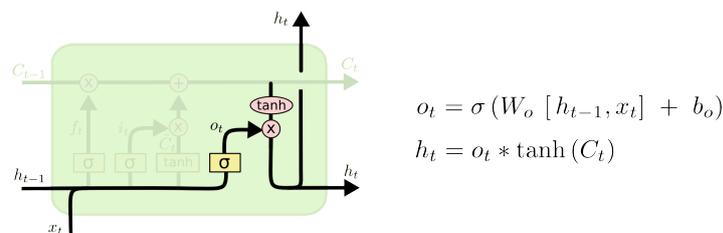


FIG. 2.14 : la porte de sortie[20].

Pour ce faire, le nouvel état de la cellule calculé juste avant est normalisé entre -1 et 1 grâce à tanh. Le vecteur concaténant l'entrée courante et l'état caché précédent est alors transformé en un sigmoïde dont le but est de décider quelle information conserver (proche de 0 signifie oublier, proche de 1 signifie conserver coordonnées de l'état de la cellule)

Tel que :

$h_t$  : La sortie.

### 2.4.3 Réseaux de Neurons Transformers

Un transformateur est un modèle d'apprentissage en profondeur . Il se distingue par son adoption de l'auto-attention , pondérant différemment la signification de chaque partie des données d'entrée (qui inclut la sortie réursive). Il est principalement utilisé dans les domaines du NLP et de la vision par ordinateur (CV)[21].

Comme les réseaux de neurones récurrents (RNN), les transformateurs sont conçus pour traiter des données d'entrée séquentielles, telles que le langage naturel, avec des applications pour des tâches telles que la traduction et la synthèse de texte . Cependant, contrairement aux RNN, les transformateurs traitent l'intégralité de l'entrée en une seule fois. Le mécanisme d'attention fournit un contexte pour toute position dans la séquence d'entrée. Par exemple, si les données d'entrée sont une phrase en langage naturel, le transformateur n'a pas à traiter un mot à la fois. Cela permet une plus grande parallélisation que les RNN et réduit donc les temps de formation[22].

Les transformateurs ont été introduits en 2017 par une équipe de Google Brain [1] et deviennent de plus en plus le modèle de choix pour les problèmes de NLP[23], remplaçant les modèles RNN tels que la mémoire longue à court terme (LSTM). Par rapport aux modèles RNN, les transformateurs se prêtent

mieux à la parallélisation, ce qui permet de s'entraîner sur des ensembles de données plus volumineux. Cela a conduit au développement de systèmes préformés tels que BERT (Représentations d'encodeurs bidirectionnels de transformateurs) et le GPT original (transformateur préformé génératif), qui ont été formés avec de grands ensembles de données linguistiques, tels que Wikipedia Corpus et Common Crawl, et peut être ajusté pour des tâches spécifiques[24].

### Domaines d'application de Réseaux de Neurons Transformers

- Séquence de texte [22]
- Patches d'image [25]
- Graphes [26]
- Schéma de base de données
- Discours
- Séries temporelles
- Intégration mathématique lorsqu'elle est formulée comme un problème de traduction

#### 2.4.4 Représentation de l'encodeur bidirectionnel des transformateurs BERT

C'est un modèle de traitement du langage naturel (NLP) introduit par Google en 2018. BERT est un type de modèle d'apprentissage profond basé sur l'architecture Transformer, qui a révolutionné diverses tâches NLP[27].

Il a été entraîné sur un large corpus de texte, y compris Wikipédia, pour capturer la structure de la langue naturelle de manière plus précise que les modèles précédents.

L'innovation technique clé de BERT consiste à appliquer l'entraînement bidirectionnel du Transformer. Contrairement aux modèles unidirectionnels, qui lisent le texte saisi de manière séquentielle (de gauche à droite ou de droite à gauche), l'encodeur du Transformer dans le modèle de BERT lit toute la séquence de mots en une fois. Cette caractéristique permet au modèle d'apprendre le contexte d'un mot en fonction de tout son environnement (gauche et droite du mot).

Il existe différentes variantes de BERT en fonction de la taille du modèle et des données d'entraînement utilisées selon le tableau suivant :

Model	Couches	Taille Cachée	Têtes d'auto-attention	feed-forward	Paramètres Total
BERT-base	12	768	12	3072	110 millions
BERT-large	24	1024	16	4096	350 millions

TAB. 2.1 : Différentes variantes de BERT

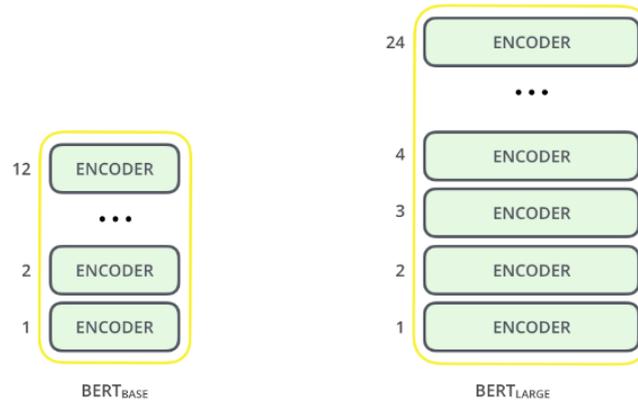


FIG. 2.15 : Différentes variantes de BERT

### 2.4.5 Architecture de BERT

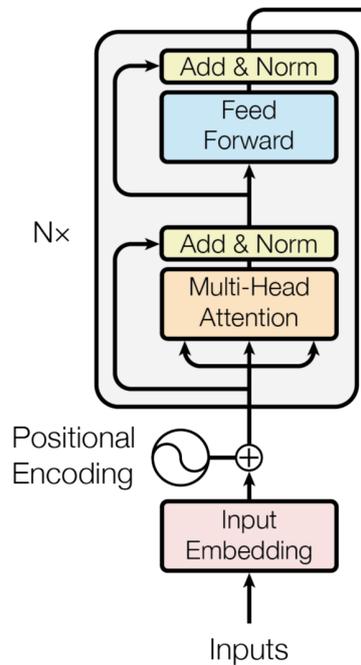


FIG. 2.16 : Architecture de BERT[29].

- **Input Embedding** : L'encodeur BERT attend une séquence de jetons. La figure ci-dessous montre comment les jetons sont traités et convertis. [CLS] est un jeton spécial inséré au début de la première phrase. [SEP] est inséré à la fin de chaque phrase. Nous avons créé des intégrations de segments en ajoutant un segment 'A' ou 'B' pour distinguer les phrases. Nous ajoutons également la position de chaque jeton dans la séquence pour obtenir des intégrations de position.

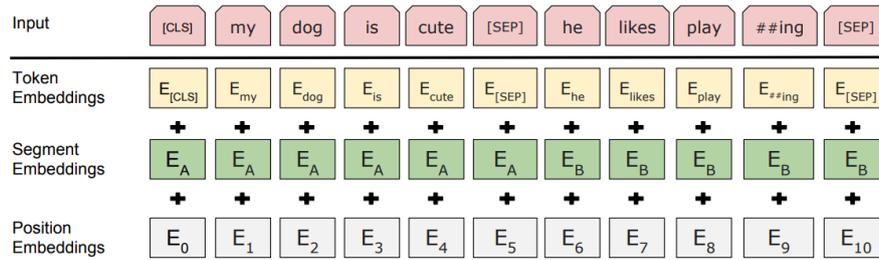


FIG. 2.17 : Input Emedding [30].

La somme des trois intégrations ci-dessus est l'entrée finale de l'encodeur BERT.

- **Encodeur BERT** : BERT utilise une architecture de transformer basée sur l'auto-attention, qui est composée de plusieurs couches de blocs de transformer. Chaque bloc de transformer a deux sous-couches principales :

**Auto-attention multi-têtes** : Cette sous-couche permet à BERT de capturer les dépendances à longue distance entre les mots dans une phrase.

Elle calcule les poids d'attention pour chaque paire de mots dans la phrase, en attribuant des poids aux mots qui sont plus importants pour la tâche de modélisation du langage masquée.

Auto-attention multi-têtes utilise la fonction de l'échelle-point-produit-attention sur une liste donnée de requêtes (Q), clés (K) et valeurs (V). la fonction d'attention à l'échelle du produit peut être formulée comme suit : [22]

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.5)$$

Avec :

$d_k$  : la dimension de Q et K

la fonction effectue une attention à l'échelle multiple-point-produit en parallèle sur plusieurs têtes - chacune ayant des représentations linéaires différentes de Q, K et V. Les résultats de l'attention dans chaque tête sont alors concaténés et linéairement projetés. Ceci peut être formalisé comme :

$$h_i = Attention(QW_i^q, KW_i^k, VW_i^v) \quad (2.6)$$

$$MH(Q, K, V) = Concat(h_0, \dots, h_j)W_o \quad (2.7)$$

avec :

$i = 0, \dots, j$

$j$  : le nombre de têtes.

L'utilisation de la même séquence pour Q, K et V permet l'auto-attention sur l'entrée donnée.

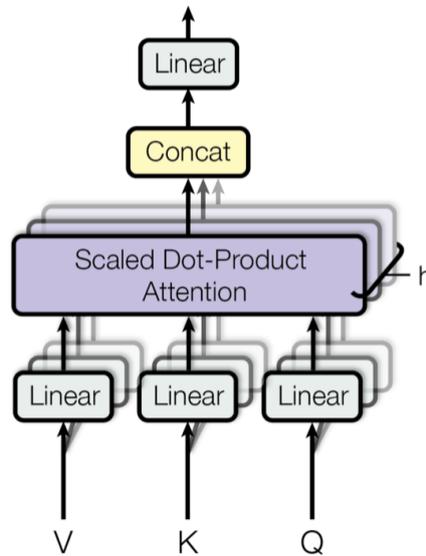


FIG. 2.18 : Auto-attention multi-têtes[22].

**Feed forward :** En plus des sous-couches d’attention, chacune des couches de notre encodeur contient un réseau de transmission connecté, qui est appliqué à chaque position séparément et à l’identique.

Cette couche se compose de deux transformations linéaires avec une activation ReLU entre elles

$$FF(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2.8)$$

Bien que les transformations linéaires soient identiques pour différentes positions, elles utilisent des paramètres différents de couche en couche. Une autre façon de décrire cela est comme deux convolutions avec une taille de noyau de 1. La dimensionnalité de l’entrée et de la sortie est  $d_{model} = 512$ , et la couche interne a une dimensionnalité  $d_{ff} = 2048$ .

## 2.5 Evaluation des performances

### 2.5.1 La Matrice de Confusion

Une matrice de confusion résume les prédictions qui ont été fournies après l’étude d’un problème de classification dans un processus d’apprentissage automatique. Le résultat de ces prédictions peut être correctes ou incorrectes et seront rangés par catégorie de classe. Il s’agit d’un tableau présentant 4 combinaisons différentes de valeurs prédites et réelles[31].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

FIG. 2.19 : La Matrice de Confusion [31]

- **TP (vrais positifs)** : Le cas où la prédiction est positive et la valeur réelle est positive.
- **TN (vrais négatifs)** : Le cas où la prédiction est négative et la valeur réelle est négative.
- **FP (faux positifs)** : Le cas où la prédiction est positive, mais la valeur réelle est négative.
- **FN (faux négatifs)** : lorsque la prédiction est négative, mais que la valeur réelle est positive.

### 2.5.2 Critère d'évaluation

L'évaluation des performances est une étape nécessaire pour vérifier la qualité du modèle, pour assurer la fiabilité des résultats de prédiction du modèle.

#### Exactitude (Accuracy) :

L'accuracy (ou exactitude en français) est une mesure de performance couramment utilisée pour évaluer la précision d'un modèle de classification. Elle représente la proportion de prédictions correctes par rapport à l'ensemble des prédictions effectuées par le modèle.

$$\text{Exactitude} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.9)$$

#### Rappel (Recall) :

Le terme "rappel" est utilisé pour désigner la mesure de performance. Le rappel est une mesure qui évalue la capacité d'un modèle ou d'un système à retrouver tous les éléments pertinents d'un ensemble de données.

$$\text{Rappel} = \frac{TP}{TP + FN} \quad (2.10)$$

#### Précision (Precision) :

La précision est une mesure de performance couramment utilisée pour évaluer la justesse d'un modèle de classification. Elle représente la proportion de prédictions correctes parmi les prédictions positives effectuées par le modèle.

$$\text{Précision} = \frac{TP}{TP + FP} \quad (2.11)$$

## 2.6 Problème d'underfitting et d'overfitting

### 2.6.1 Principe de l'overfitting

L'overfitting est un problème qui est souvent croisé en Machine Learning. Il se manifeste lorsqu'un modèle essaye de s'ajuster excessivement aux données d'entraînement, ce phénomène est appelé (Bruit).

- Ce qui se passe en réalité, c'est que l'apprentissage se focalise sur des données qui

n'ont aucun impact direct avec les données d'entraînement. Ce qu'on traduit par bruit, ce sont les données d'entraînement qui sont présentes sous forme de points et qui présentent des variations et fluctuations de façon aléatoires. L'entraînement sur ce type de données rendent les modèles instables [32].

### 2.6.2 Principe de l'underfitting

underfitting est un scénario en science des données où un modèle de données est incapable de capturer de manière précise la relation entre les variables d'entrée et de sortie, ce qui entraîne un taux d'erreur

élevé à la fois sur l'ensemble d'entraînement et sur les données non vues. Cela se produit lorsque le modèle est trop simple, ce qui peut être le résultat d'un modèle ayant besoin de plus de temps d'entraînement, de plus de caractéristiques d'entrée ou d'une régularisation moins. Tout comme l'overfitting, lorsqu'un modèle est sous-ajusté, il ne peut pas établir la tendance dominante dans les données, ce qui entraîne des erreurs d'entraînement et de mauvaises performances du modèle. Si un modèle ne peut pas généraliser correctement sur de nouvelles données, il ne peut pas être utilisé pour des tâches de classification ou de prédiction[33].

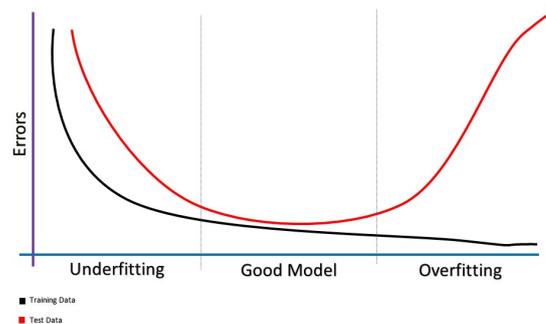


FIG. 2.20 : Overfitting / Underfitting [31]

### 2.6.3 quelques approches pour réduire l'overfitting :

#### Early-stopping :

est une technique très intuitive. Elle consiste simplement à arrêter l'entraînement avant que le modèle ne commence à overfit. Cette technique est surtout utilisée en deep learning pour l'entraînement de réseaux.[34]

#### Utiliser dropout pour les réseaux neuronaux :

est une technique qui peut être appliquée à la sortie de plusieurs couches réseau. Il supprimera de manière aléatoire et périodique certains neurones (ainsi que leurs connexions d'entrée et de sortie) du réseau. [35].

La figure ci-dessous illustre dropout après une couche entièrement connectée :

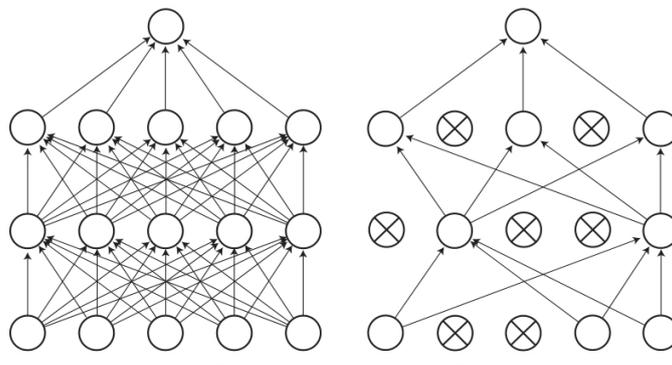


FIG. 2.21 : Dropout : une méthode simple pour prévenir l'overfitting des réseaux neuronaux. [36]

### 2.7 Conclusion

La recherche dans le domaine des réseaux de neurones est en constante évolution pour améliorer les performances des résultats, ce qui en fait un domaine très dynamique et en plein essor.

L'apprentissage profond (DL) a réussi à se positionner comme une technologie révolutionnaire dans de nombreux domaines. Il a transformé la traduction automatique, les moteurs de recherche, les assistants informatiques et de nombreuses applications de notre vie quotidienne, qui sont maintenant alimentés par des techniques d'apprentissage profond.

Les modèles d'apprentissage profond s'efforcent de reproduire de manière précise les modes de traitement de l'information et de communication observés dans le système nerveux biologique.

Ce chapitre a présenté les architectures les plus populaires et les plus utilisées des réseaux de neurones, ainsi que de nouvelles architectures qui semblent prometteuses dans différents domaines d'application de la technologie contemporaine.

## Chapitre 3

### Etat de l'art

### 3.1 Introduction

Ce chapitre passe en revue les travaux connexes publiés sur ce sujet, en nous concentrant sur les approches fondées sur l'apprentissage profond. Nous allons également analyser et comparer les différentes approches pour comprendre leur efficacité dans la gestion des catastrophes naturelles

### 3.2 Ouvrages connexes

**Saloni Jain et al**[37] ont proposé un cadre permettant de prévoir la trajectoire d'une catastrophe en extrayant les données relatives aux catastrophes en temps réel provenant de Twitter. Le cadre implique la collecte de données, le prétraitement, la géolocalisation des tweets, le filtrage de données et l'extrapolation de la courbe des catastrophes pour la prédiction des zones vulnérables. Le cadre est validé par l'analyse des événements passés et le potentiel d'être développé pour devenir un système complet de prévision et d'alerte des populations en cas de catastrophes.

**Jyoti Prakash Singh et al**[38] ont proposé un système de classification des tweets pour identifier les victimes, l'objectif de cette étude est d'étudier l'utilisation des publications sur Twitter lors des catastrophes liées aux inondations et de proposer un algorithme d'apprentissage automatique permettant d'identifier les victimes demandant de l'aide. Le système proposé classe les tweets en priorité élevée ou faible et prédit l'emplacement des tweets hautement prioritaires sans informations de localisation à l'aide du modèle de Markov. Le système vise à aider les victimes lors des catastrophes sur la base de leurs tweets et peut être étendu pour être utilisé dans d'autres situations de catastrophes naturelles ou causées par l'homme. Les auteurs ont utilisé l'API Twitter pour capturer des tweets en direct liés aux inondations dans les états du sud et de l'est de l'Inde. La collecte de données a été effectuée à l'aide de l'API StreamingTwitter à l'aide de la bibliothèque Tweepy Python. Les tweets ont été collectés en novembre-décembre 2015 pour les inondations à Chennai (sud de l'Inde), et en juillet-août 2016 pour le Bihar. Au total, 32 400 tweets ont été collectés avec les mots-clés « inondation », « eau », « Baarh ». Trois algorithmes de classification populaires ont été appliqués, à savoir la machine vectorielle de support (SVM), l'amplification du gradient et la forêt aléatoire. Les résultats du SVM étaient médiocres par rapport aux deux autres algorithmes et n'ont donc pas été publiés. Le système a une précision de classification de 81 % et une précision de prédiction de localisation de 87 %.

**Yasin Kabir et al**[39] ont proposé un modèle et la planification des secours peut aider à la gestion des catastrophes en fournissant des informations précises et à jour permettant une réponse rapide, facile et rentable afin de réduire les pertes potentielles en vies humaines et en biens. L'article propose un modèle d'apprentissage profond combinant la mémoire à long terme bidirectionnelle (BLSTM) basée sur l'attention et le réseau neuronal convolutionnel (CNN) pour classer les tweets, ainsi qu'un algorithme de planification hybride multitâche adaptatif prenant en compte les contraintes de ressources pour effectuer une opération de planification de sauvetage efficace. Le système peut identifier les zones ou les individus ayant besoin d'une aide immédiate et prioriser les opérations de secours en conséquence

**Abhinav Kumar et al** [40] Cette étude compare différents algorithmes d'apprentissage automatique pour classer les tweets liés aux catastrophes en six catégories différentes. Les modèles sont testés avec quatre catastrophes différentes, telles qu'un ouragan, un tremblement de terre, une inondation et un feu de forêt, afin de vérifier l'efficacité des modèles. Les modèles de réseaux neuronaux profonds se révèlent très performants pour classer les tweets liés aux catastrophes, même pour des ensembles de données déséquilibrés. La plage du score F1 varie de 0,61 à 0,88 pour les modèles basés sur des réseaux neuronaux profonds, alors qu'elle varie de 0,16 à 0,80 pour les classificateurs d'apprentissage automatique classiques.

**Congcong Wang et al** [42] ont proposé une approche d'apprentissage multitâche basée sur des transformers pour la classification et la hiérarchisation des tweets en cas de crise, ce qui constitue un problème crucial en matière d'intervention d'urgence. L'approche utilise des codeurs de transformateur pré-entraînés pour coder le texte d'entrée, puis applique un apprentissage multitâche pour optimiser conjointement les tâches de classification et de priorisation. Le document présente également une approche d'ensemble simple qui s'appuie sur plusieurs apprenants multitâches pour classer les tweets de test liés à la crise. L'efficacité de cette approche est évaluée à l'aide de divers indicateurs en soumettant des essais au système TREC Incident Streams (IS). Les auteurs font état d'une précision de 0,764 pour leur approche,

ce qui est supérieur à la précision de référence de 0,732% et les valeurs de rappel vont de 0,63% à 0,87%. La limite de l'approche suggérée est qu'elle ne prend pas en compte d'autres aspects de la réponse aux situations d'urgence, comme l'allocation des ressources ou la prise de décisions.

### 3.3 Analyse et Comparaison

Le tableau ci-dessous présente une étude comparative des approches proposées

- **la date**
- **Auteur**
- **Données**
- **Résultats**
- **Techniques utilisées**
- **Avantages**
- **Inconvénients**

TAB. 3.2 : Étude comparative des travaux connexes

date	Auteur	Données	Résultats	Techniques utilisées	Avantages	Inconvénients
2015	Saloni Jai	4 000 tweets	Former une courbe et l'extrapoler pour trouver les zones vulnérables avec un niveau de confiance de 95 %	-la géolocalisation -le clustering DBSCAN	-la possibilité de développer pour en faire un système complet de prévision et d'alerte des populations en cas de catastrophes - peuvent être envoyés aux chaînes d'information ou diffusés pour une action proactive	-nécessite d'importantes ressources de calcul pour traiter de grandes quantités de données en temps réel
2017	Jyoti Prakash Singh, Yogesh Dwivedi, Nripendra Rana, Abhinav Kumar, Kawaljeet Kaur Kapoor	32 400 tweets ont été collectés avec les mots-clés « inondation », « eau », « Baarh ». Utilisez l'API Twitter pour capturer des tweets en direct liés aux inondations dans le sud et l'est des États indiens	precision RF=0.81 GB=0.79 Rappel RF=0.81 GB=0.71 f1-score RF=0.81 GB=0.75	-Random Forest -Gradient Boosting(GB)	-adaptable à d'autres catastrophes naturelles	-Si un Tweet contient une langue non prise en charge ou contient du sarcasme, le système peut ne pas être en mesure de classer le Tweet
2019	Yasin Kabir, Sanjay Kumar Madria	dataset des ouragans Harvey et Irma, CrisisNLP, CrisisLex	l'article ne fournit pas de résultats numériques spécifiques pour les performances de l'algorithme proposé	-BLSTM -CNN	-model plus performant . -réduit les pertes potentielles en fournissant des informations précises pour une réponse rapide, facile et rentable lors des catastrophes	-il peut ne pas bien se généraliser à d'autres ensembles de données de tweet des catastrophes

2019	Abhinav Kumar, Jyoti Prakash Singh, Sunil Saumya	tweets liés aux catastrophes sur Twitter lors d'urgences	les modèles de réseaux neuronaux profonds sont nettement plus performants f1-score : 0,61 à 0,88 que les outils classiques de classification par apprentissage automatique f1-score : 0,16 à 0,80	-DNN -Algorithme de ML	- améliore la précision de la classification et renforce l'efficacité des opérations de secours en cas de catastrophe	-concentre uniquement sur les tweets publiés en anglais
2021	congcong wang, Paul Nulty, David Lillis	dataset Text Retrieval Conference (TREC) Incident Streams (IS)	LR+STL CF1-H=0.0898 CF1-A=0.1527 Cacc=0.9113 BERT_base+STL CF1-H=0.1084 CF1-A=0.1801 Cacc=0.8960 BERT_base+MTL CF1-H=0.1382 CF1-A=0.1638 Cacc=0.8937	-Multi-Task Learning -BERT_base -Logistic Regression (LR) -Single Task learning (STL)	-permet d'obtenir des performances compétitives	-nécessite une grande quantité de données -basé sur des modèles de transformateurs pré-entraînés, qui peuvent être coûteux en termes de calcul et nécessite des ressources importantes

## 3.4 Conclusion

Pour conclure ce chapitre dans lequel nous avons réalisé un état de l'art à propos de la prédiction des catastrophes à partir des réseaux sociaux. Les travaux connexes que nous avons cités représentent une étude comparative qui nous permet de mieux poser les bases de notre recherche. nous avons décrit chaque approche en détails dans un tableau en résumant brièvement chaque travail.

## Chapitre 4

# Modélisation et Implémentation

### 4.1 Introduction

Dans le chapitre précédent, nous avons cité les différentes méthodes appliquées avec succès pour l'analyse des données textuelles des réseaux sociaux dans le contexte de la gestion des catastrophes. La qualité, la performance et la précision du modèle de Prédiction dépendent fortement de jeux de données utilisés. Après beaucoup de recherches nous avons proposés des modèles et nous les avons testés sur 2 méthodes de deep learning(LSTM, BERT). Nous avons introduit dans ce chapitre les modèles dont nous avons atteint nos meilleurs résultat pour chacune des méthodes cités. Les performances des approches de classification proposées ont été évalué en prenant en compte les différentes mesures d'évaluation des algorithmes de l'apprentissage profond, l'exactitude, la précision, le rappel, Toutes les méthodes ont été implémentées et évalué avec une Classification binaire (catastrophe, non catastrophe ).

### 4.2 Environnement de développement

#### 4.2.1 Matériel

Nous avons utilisé Google Cloud pour implémenter notre modèle et effectuer toutes nos expérimentations.

CPU	Disk Space	RAM
Intel(R) Core(TM) i5-8365U	107.7 GB	12.7 GB

TAB. 4.1 : Matériel

#### 4.2.2 Les logiciels

Il existe de nombreux frameworks open source disponibles, et la plupart d'entre eux prennent en charge le langage Python.

##### Google Colaboratory

également connu sous le nom de Colab, est une plateforme de cloud computing proposée par Google. Il s'agit d'un environnement de développement et de collaboration basé sur le navigateur qui permet aux utilisateurs d'écrire, d'exécuter et de partager du code Python. Colab offre un accès gratuit à des ressources de calcul, notamment des processeurs et des GPU, ainsi qu'à des bibliothèques et des frameworks populaires tels que TensorFlow et PyTorch. Il permet également d'utiliser des notebooks interactifs, ce qui facilite le partage et la reproductibilité des travaux de data science et de machine learning. Colab est largement utilisé dans la communauté de l'apprentissage automatique et constitue un outil pratique pour expérimenter et exécuter du code Python dans le cloud.



FIG. 4.1 : Google Colab logo

### Langage de développement Python

est un langage de programmation flexible et interprété créé par Guido van Rossum, Sa syntaxe claire et lisible est l'une de ses principales caractéristiques.

Il est utilisé dans de nombreux domaines, tels que le développement web, l'analyse de données, l'intelligence artificielle, l'apprentissage automatique, l'automatisation des tâches, et bien d'autres. Python est accompagné d'une vaste bibliothèque standard qui offre de nombreuses fonctionnalités prêtes à l'emploi.

Grâce à sa simplicité, sa lisibilité et sa communauté dynamique, Python est devenu l'un des langages de programmation les plus populaires et les plus largement utilisés au monde. Il est apprécié pour sa facilité d'apprentissage, sa flexibilité et sa capacité à résoudre une grande variété de problèmes. plateformes.

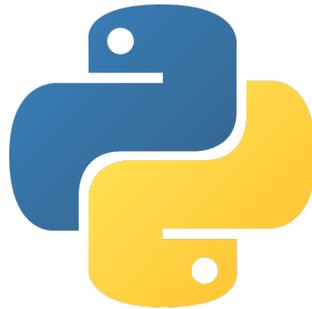


FIG. 4.2 : Python logo.

Nous avons utilisé les bibliothèques suivantes :

### TensorFlow

TensorFlow est une bibliothèque Disponible pour l'apprentissage automatique, il est utilisé dans la recherche, l'industrie développé par l'équipe Google Brain.

TensorFlow permet la création et la formation de modèles d'apprentissage automatique, y compris les réseaux de neurones, à l'aide de graphes informatiques. Ces graphiques représentent les calculs et le flux de données entre les différentes parties du modèle. La flexibilité de TensorFlow permet la création de modèles complexes et un traitement efficace des données.

La bibliothèque est optimisée pour tirer parti des ressources matérielles telles que les processeurs graphiques (GPU) et les unités de traitement de tenseur (TPU), pour accélérer les calculs et entraîner le modèle. Cela permet une utilisation maximale des capacités de calcul parallèle et améliore les performances des applications d'apprentissage automatique.



FIG. 4.3 : TensorFlow.

### Keras

Keras est une bibliothèque open source populaire et conviviale pour l'apprentissage automatique et le deep learning. Elle fournit une interface haut niveau pour construire, entraîner et évaluer des modèles d'apprentissage automatique.

Keras a été initialement développé comme un projet distinct, mais il est devenu une partie intégrante

de TensorFlow à partir de la version 2.0. Il offre une approche simple et intuitive pour créer des modèles d'apprentissage automatique, en utilisant une syntaxe claire et concise.

L'un des avantages clés de Keras est sa facilité d'utilisation, ce qui le rend idéal pour les débutants en apprentissage automatique. Il permet de créer des modèles en utilisant des blocs de construction appelés couches (layers), qui peuvent être empilées pour former des architectures de modèles complexes. Keras offre également une grande flexibilité pour personnaliser et ajuster les modèles selon les besoins spécifiques du projet.



FIG. 4.4 : Keras.

### Numpy

est une bibliothèque Python essentielle pour le calcul numérique et la manipulation de tableaux multidimensionnels. Elle offre des structures de données efficaces et des fonctions mathématiques rapides, ce qui en fait un outil de prédilection pour les applications scientifiques, l'analyse de données et d'autres domaines nécessitant des opérations numériques performantes.



FIG. 4.5 : Numpy.

### Pandas

Pandas est une bibliothèque open source sous licence BSD (une famille de licences de logiciels libres permissives, imposant des restrictions minimales sur l'utilisation et la distribution des logiciels couverts. Cela contraste avec les licences de copyleft, qui ont des exigences similaires) qui fournit des structures de données et des outils d'analyse de données haute performance et faciles à utiliser pour le langage de programmation Python.



FIG. 4.6 : Pandas.

### Sklearn

Aussi appelé apprentissage Scikit-learn, elle est une bibliothèque Python puissante et conviviale pour l'apprentissage automatique. Elle offre une grande variété d'algorithmes et de fonctionnalités pour la classification, la régression et le regroupement, ainsi que des outils pour le prétraitement des données et l'évaluation des modèles. C'est un choix populaire parmi les praticiens de l'apprentissage automatique en raison de sa facilité d'utilisation et de son efficacité.



FIG. 4.7 : Scikit-learn.

## NLTK

NLTK (Natural Language Toolkit) est une bibliothèque open source populaire pour le traitement automatique du langage naturel (NLP) en Python. Elle offre un ensemble complet d'outils et de ressources pour traiter et analyser le texte, que ce soit pour des tâches de tokenisation, de lemmatisation, de segmentation de phrases, d'analyse grammaticale, d'étiquetage de parties du discours, d'extraction d'entités nommées, de détection de collocations, etc.

## 4.3 Description du dataset

Il a été récupéré depuis la plate-forme [Kaggle] [43]. Les données incluses sont constituées d'un ensemble de tweets classés en deux catégories : les tweets positifs (liés à une véritable catastrophe) et les tweets négatifs (non liés à une véritable catastrophe). Il contient 3271 tweets positifs et 4342 tweets négatifs exactement.

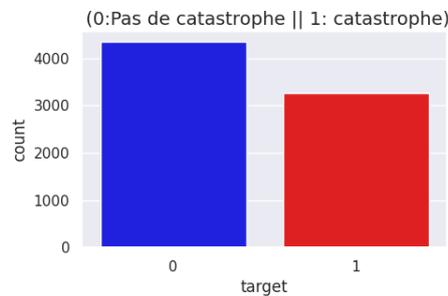


FIG. 4.8 : nombre de tweets positifs et négatifs

	text	target
0	Our Deeds are the Reason of this #earthquake M...	1
1	Forest fire near La Ronge Sask. Canada	1
2	All residents asked to 'shelter in place' are ...	1
3	13,000 people receive #wildfires evacuation or...	1
4	Just got sent this photo from Ruby #Alaska as ...	1

FIG. 4.9 : Les cinq premières lignes de dataset

## 4.4 Modèle LSTM

### 4.4.1 Le Prétraitement

Le prétraitement des données est une étape importante dans le NLP car il permet de nettoyer, normaliser et transformer les données textuelles brutes en une forme plus adaptée à l'analyse.

#### Nettoyage du texte :

Cela implique la suppression de tout texte indésirable, tel que les balises HTML, la ponctuation, les caractères spéciaux, les chiffres, etc. Cela permet de réduire le bruit dans les données et de se concentrer sur le texte réel

```

1 def remove_HTML(text) :
2     html=re.compile(r'<.*?>')
3     return html.sub(r'',text)
4
5 def remove_URL(text) :
6     url = re.compile(r'https? ://\S+|www\.\S+')
7     return url.sub(r'URL',text)
8
9 def remove_mention(text) :
10    at=re.compile(r'@\S+')
11    return at.sub(r'USER',text)
12
13 def remove_number(text) :
14    num = re.compile(r'[-+]?[\.\d]*[\d]+[ :,\.\d]*')
15    return num.sub(r'NUMBER', text)
16 def remove_emoji(text) :
17    emoji_pattern = re.compile("[
18        u"\U0001F600-\U0001F64F" # emoticons
19        u"\U0001F300-\U0001F5FF" # symbols & pictographs
20        u"\U0001F680-\U0001F6FF" # transport & map symbols
21        u"\U0001F1E0-\U0001F1FF" # flags (iOS)
22        u"\U00002702-\U000027B0"
23        u"\U000024C2-\U0001F251"
24        "]" +, flags=re.UNICODE)
25    return emoji_pattern.sub(r'EMOJI', text)

```

Listing 4.1 : Nettoyage du texte

#### Normalisation de la casse :

Il s'agit de convertir tout le texte en minuscules ou en majuscules, afin d'éviter les problèmes de casse lors de l'analyse. où les mots identiques mais écrits différemment peuvent être considérés comme différents.

```

1 def normalize_case(text) :
2     return text.lower()
3

```

Listing 4.2 : Normalisation

#### Tokenisation :

La tokenisation divise le texte en unités plus petites appelées "tokens". Les tokens peuvent être des mots, des phrases, des caractères individuels, etc. Cela permet de traiter le texte de manière plus granulaire et d'extraire des informations significatives.

```

1 def tokenize_text(text) :
2     tokens = nltk.word_tokenize(text)
3     return tokens

```

Listing 4.3 : Tokenisation

### Suppression des mots vides (Stop words) :

Les mots vides sont des mots courants tels que "le", "de", "et", etc., qui n'apportent généralement pas beaucoup de sens à un texte. En les supprimant, on réduit la dimensionnalité des données et on se concentre sur les mots clés et les informations importantes.

```
1 import nltk
2 nltk.download('stopwords')
3 print(stopwords.words('english'))
4
5 def toremove_stopword(text):
6     remove_stopword = [word for word in text.split()
7                         if word.lower() not in stopwords.words('english')]
8     return remove_stopword
9
10 train_data['text'] = train_data['text'].apply(toremove_stopword)
11
```

Listing 4.4 : Suppression des mots vides

### Lemmatisation et/ou racinisation (stemming) :

Ces techniques permettent de réduire les mots à leur forme de base pour regrouper les variantes d'un même mot. Par exemple, "joue", "joues" et "jouer" peuvent être ramenés à leur lemme "jouer" ou à leur racine "jou".

### Vectorisation :

La vectorisation consiste à représenter les mots ou les phrases sous forme de vecteurs numériques. Cela permet aux algorithmes de traitement automatique du langage naturel de manipuler les données textuelles en tant que nombres, ce qui facilite les calculs et les analyses.

## 4.4.2 Le traitement

### Fractionner l'ensemble de données

Nous avons utilisé la fonction `train_test_split` d'une bibliothèque ( `scikit-learn`) pour diviser les données en ensembles d'apprentissage et de test. Ensuite, nous divisons les données d'apprentissage en ensembles d'apprentissage et de validation.

```
1 X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3, random_state =41)
2
3 train_data, val_data, train_labels, val_labels = train_test_split( X_train, y_train, test_size
   =0.1, random_state=42)
```

- 'X' est ensemble de données d'entrée (features),
- 'y' est votre ensemble d'étiquettes (labels) correspondantes,
- `test_size` est le paramètre qui définit la proportion du dataset qui sera allouée à l'ensemble de test. Dans notre travail, Il est spécifié à 0,3, ce qui indique que 30 % des données seront réservées pour les tests. De même, il est réglé à 0,1, ce qui signifie que 10 % des données seront utilisées pour la validation.
- `random_state` est utilisé pour initialiser la graine (seed) du générateur de nombres aléatoires lors de la division du dataset en ensembles de train et de test. Il contrôle la manière dont les données sont mélangées avant la division ;
- 'train\_data' : l'ensemble de données d'entraînement qui contient une proportion de 60% des exemples d'entrée.

- 'X\_test' : l'ensemble de données de test qui contient une proportion de 30% des exemples d'entrée.
- 'train\_labels' : l'ensemble d'étiquettes correspondantes pour l'ensemble d'entraînement, associé à train\_data.
- 'y\_test' : l'ensemble d'étiquettes correspondantes pour l'ensemble de test, associé à X\_test.
- 'val\_data' : l'ensemble de données de validation qui contient une proportion de 10% des exemples d'entrée.
- 'val\_labels' : l'ensemble d'étiquettes correspondantes pour l'ensemble de validation, associé à val\_data.

### La structuration du modèle LSTM

Nous avons utilisé un modèle séquentiel avec une couche d'embedding, trois couches de dropout, deux couches LSTM et une couche dense avec une activation sigmoid comme le montre (Fig : 4.10).

- La couche d'embedding : Elle est utilisée pour représenter et encoder des données textuelles sous forme de vecteurs. La sortie de la couche d'embedding a une forme de dimension (None, 20, 100), ce qui signifie qu'elle génère des sorties avec une taille de 20 séquences, chacune étant représentée par un vecteur de 100 dimensions. De plus, la couche d'embedding est composée de 300 000 paramètres au total.
- La couche de dropout : est une technique de régularisation utilisée dans la formation des réseaux de neurones. Elle consiste à désactiver aléatoirement un certain pourcentage de neurones pendant l'entraînement. Son objectif est de prévenir le surapprentissage (overfitting) et d'améliorer la capacité de généralisation du modèle.

```
1 model.add(Dropout(0.4))
```

Dans notre modèle, un taux de dropout de 0.4 signifie que 40 % des neurones sont désactivés de manière aléatoire pendant l'entraînement.

- La couche LSTM : Nous avons utilisé la couche LSTM pour capturer les dépendances à long terme dans les données séquentielles. cette couche génère des sorties avec une dimension de (None, 20, 200) , et elle est composée de 240 800 paramètres. la deuxième couche lstm\_1 génère des sorties avec une dimension de (None, 200) , et elle est composée de 320 800 paramètres.
- Partie dense :

Partie dense que nous avons employé est une partie de sortie adaptée aux tâches de classification binaire. Elle génère une seule valeur en sortie avec une dimension de (None, 1), indiquant que cette partie produit une prédiction unique. Cette partie dense est composée de 201 paramètres qui sont ajustés lors de l'entraînement du modèle pour améliorer les performances de la classification binaire.

```
1 model.add(Dense(1, activation='sigmoid'))
```

Dans notre modèle, nous avons appliqué la fonction d'activation sigmoid pour des problèmes de classification binaire. Cette fonction restreint les valeurs de sortie dans la plage de 0 à 1, ce qui nous permet d'estimer la probabilité que l'entrée appartienne à la classe positive. Une sortie proche de 0 indique une probabilité faible d'appartenance à la classe positive, tandis qu'une sortie proche de 1 indique une probabilité élevée.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 20, 100)	300000
dropout (Dropout)	(None, 20, 100)	0
lstm (LSTM)	(None, 20, 200)	240800
dropout_1 (Dropout)	(None, 20, 200)	0
lstm_1 (LSTM)	(None, 200)	320800
dropout_2 (Dropout)	(None, 200)	0
dense (Dense)	(None, 1)	201

=====  
 Total params: 861,801  
 Trainable params: 861,801  
 Non-trainable params: 0  
 =====  
 None

FIG. 4.10 : Résumé du modèle LSTM

### Entraînement du modèle

- Pour régler les poids du modèle pendant l’entraînement, nous avons opté pour l’algorithme d’optimisation Adam avec un taux d’apprentissage de 0.00001. Ce taux d’apprentissage joue un rôle essentiel en déterminant l’étendue des ajustements de poids effectués à chaque étape de l’entraînement.

```
1 adam = optimizers.Adam(learning_rate=0.00001)
```

- Le modèle a été compilé en utilisant la fonction de perte `binary_crossentropy`, qui est appropriée pour des problèmes de classification binaire avec deux classes distinctes à prédire (0 et 1). Nous avons également utilisé l’optimiseur Adam . - En spécifiant la liste de métriques , nous indiquons à Keras d’utiliser ces métriques pour évaluer les performances du modèle pendant l’entraînement et l’évaluation. Cela nous permettra d’obtenir les valeurs d’exactitude, de précision et de rappel comme métriques de performance pour notre modèle lors de son entraînement et de son évaluation

```
1 model.compile(loss = 'binary_crossentropy', optimizer=adam , metrics = [tf.keras.metrics.
  BinaryAccuracy(name='accuracy'),
2   tf.keras.metrics.Precision(name='precision'),
3   tf.keras.metrics.Recall(name='recall'),
4 ])
```

- La ligne de code suivante a été utilisée pour entraîner le modèle :

```
1 history=model.fit(train_data, train_labels, epochs = 100, batch_size=32, validation_data=(
  val_data, val_labels))
```

nous avons spécifié un `batch_size` de 32, ce qui signifie que chaque lot d’entraînement se compose de 32 échantillons. Pendant l’entraînement, nous avons utilisé un ensemble de données de validation distinct (`val_data`, `val_labels`) pour évaluer les performances du modèle et prendre des décisions d’arrêt prématuré afin de prévenir l’overfitting. Cet ensemble de données de validation est séparé de l’ensemble d’entraînement et permet d’évaluer objectivement la capacité du modèle à généraliser sur de nouvelles données.

### 4.4.3 Évaluation du modèle

- Ensuite, nous évaluons les performances du modèle en utilisant la méthode `evaluate` avec les données de test `X_test` et les étiquettes de test `y_test`. Cela nous donne une estimation de la performance du

modèle sur des données qu'il n'a pas vues pendant l'entraînement.

```
1 model.evaluate(X_test, y_test)
```

## 4.5 Modèle BERT

### 4.5.1 Analyse des ensembles de données

Supprimons les colonnes inutiles de notre ensemble de données, à savoir 'id', 'keyword', 'location'. Nous sommes principalement intéressés par 'text' et 'target' pour la construction du classifieur pour le moment.

```
1 df = df.drop(['id', 'keyword', 'location'], axis=1)
```

### 4.5.2 Fractionner l'ensemble de données

Comme observé dans le modèle précédent (modèle LSTM), nous avons employé la fonction `train_test_split` afin de diviser l'ensemble de données en ensembles distincts pour l'apprentissage et les tests.

### 4.5.3 Préprocesseur et Encodeur BERT

L'étape suivante consiste à télécharger le préprocesseur et l'encodeur BERT.

```
1 preprocess = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3")
```

Le lien renvoie vers un modèle TensorFlow Hub appelé "bert\_en\_uncased\_preprocess", qui est utilisé pour automatiser la préparation des données pendant l'utilisation du modèle BERT afin de traiter le langage naturel (NLP). Cette approche élimine l'obligation de faire manuellement les étapes de Prétraitement que nous avons vues précédemment dans le modèle LSTM.

ce modèle fournit des opérations de prétraitement pour convertir du texte brut en entrée utilisable par le modèle BERT. Le prétraitement des données est une étape essentielle dans l'utilisation de modèles basés sur BERT, car il nécessite une mise en forme spécifique pour être efficace.

Le modèle "bert\_en\_uncased\_preprocess" offre une interface pratique pour effectuer les étapes de prétraitement requises, telles que la tokenisation du texte, la conversion en identifiants de tokens BERT, la création de masques d'attention, etc. Ces opérations permettent de préparer les données avant de les fournir à un modèle BERT pré-entraîné pour l'apprentissage ou l'inférence.

```
1 encoder = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4")
```

Le lien "[https://tfhub.dev/tensorflow/bert\\_en\\_uncased\\_L-12\\_H-768\\_A-12/4](https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4)" : pointe vers un modèle BERT en anglais qui utilise une architecture à 12 couches d'encodeurs, chaque encodeur ayant une taille de 768 unités cachées et 12 têtes d'attention. Le numéro "4" à la fin du lien représente la version spécifique du modèle BERT.

L'instruction `hub.KerasLayer` est utilisée pour créer une couche Keras qui encapsule le modèle BERT pré-entraîné. Cela vous permet d'intégrer cette couche dans votre propre modèle Keras et de l'utiliser pour des tâches de traitement du langage naturel.

### 4.5.4 Création du modèle

Le modèle est conçu pour traiter des données textuelles brutes. Il commence par appliquer une fonction de prétraitement appelée "preprocess" pour préparer le texte avant de le soumettre à un encodeur nommé "encoder". Une fois que l'encodeur a généré des représentations de sortie, une couche de dropout est appliquée pour régulariser le modèle en désactivant certains neurones pendant l'entraînement. Enfin, les prédictions sont générées à l'aide d'une couche dense comme indiqué ci-dessous :

```
1 text_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text-layer')
2 preprocessed_text = preprocess(text_input)
3 outputs = encoder(preprocessed_text)
4 d_layer = tf.keras.layers.Dropout(0.1, name="dropout-layer")(outputs['pooled_output'])
5 d_layer = tf.keras.layers.Dense(1, activation='sigmoid', name="output")(d_layer)
6 model = tf.keras.Model(inputs=[text_input], outputs = [d_layer])
```

- La structure du modèle est la suivante :

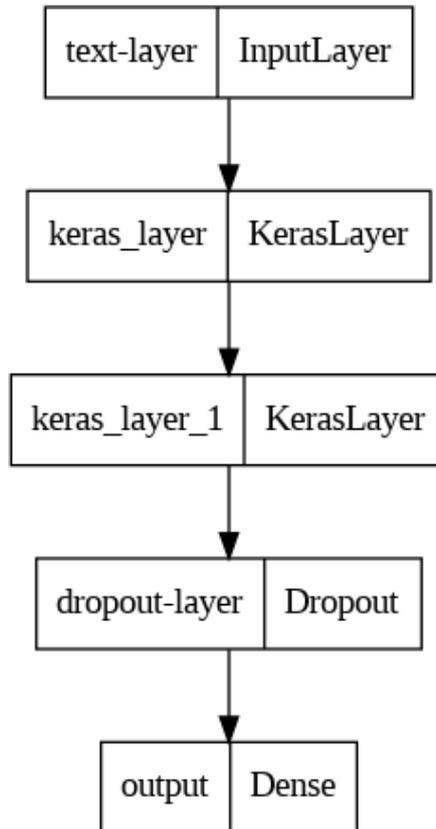


FIG. 4.11 : Structure du modèle BERT

### 4.5.5 Entraînement et Evaluation le modèle

on a utilisé les mêmes étapes précédentes de modèle LSTM, y compris le même batch-size, le même learning-rate et le même algorithme d'optimisation.

## 4.6 Résultats du modèle LSTM

- la phase d'entraînement et de validation

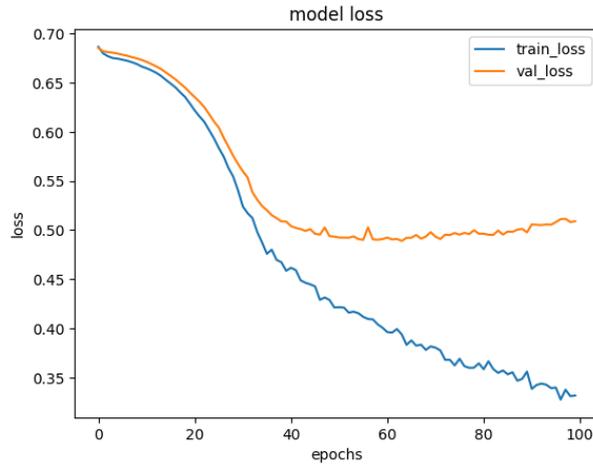


FIG. 4.12 : Variation de la perte(loss) en fonction du nombre d'époques

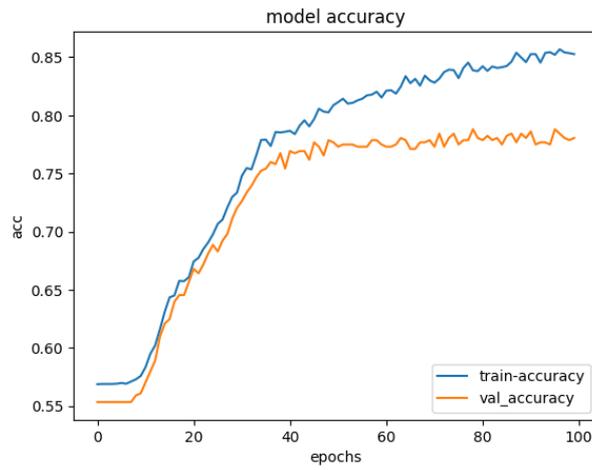


FIG. 4.13 : Accuracy en fonction du nombre d'époques

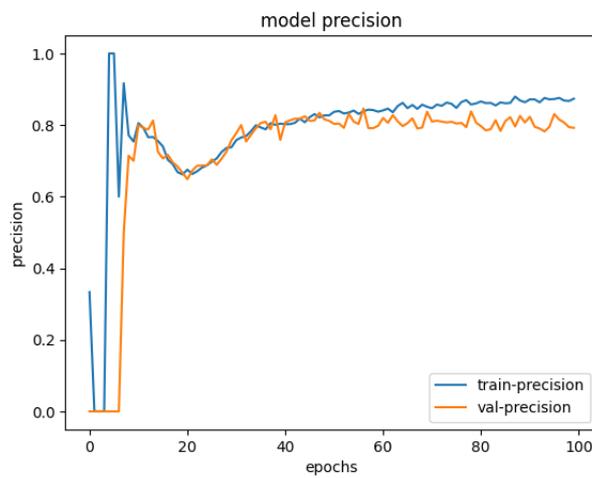


FIG. 4.14 : Variation de la précision en fonction du nombre d'époques

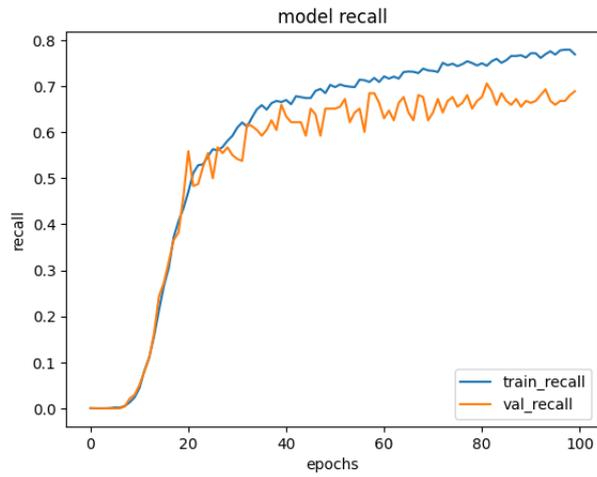


FIG. 4.15 : Evolution du rappel(recall) en fonction du nombre d'époques

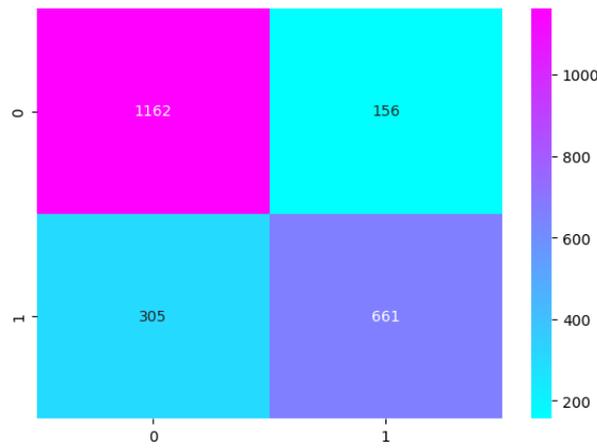


FIG. 4.16 : visualisation graphique de la matrice de confusion

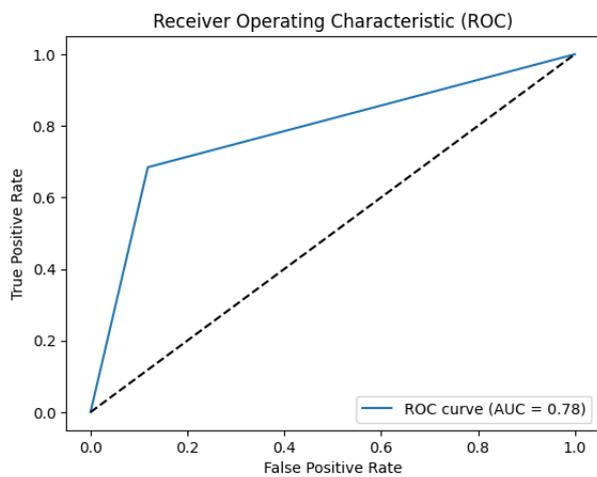


FIG. 4.17 : Receiver Operating Characteristic (ROC)

## 4.7 Résultats du modèle BERT

- la phase d'entraînement et de validation

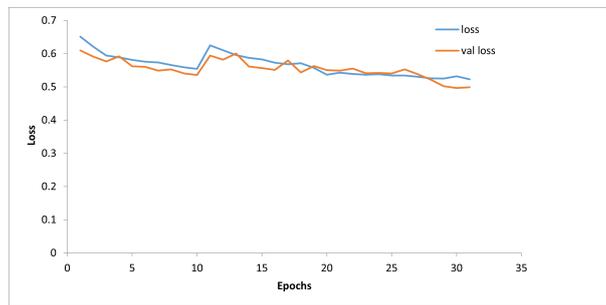


FIG. 4.18 : Variation de la perte(loss) en fonction du nombre d'époques

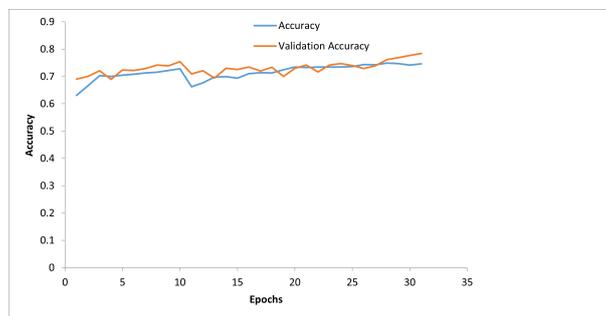


FIG. 4.19 : Accuracy en fonction du nombre d'époques

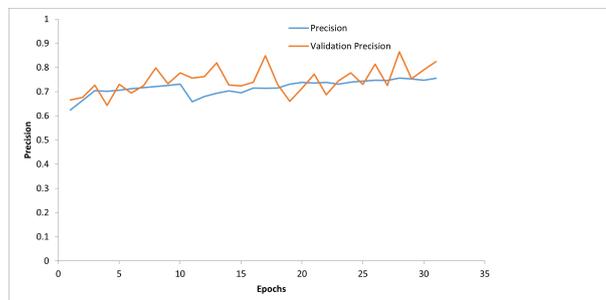


FIG. 4.20 : Variation de la précision en fonction du nombre d'époques



FIG. 4.21 : Evolution du rappel(recall) en fonction du nombre d'époques

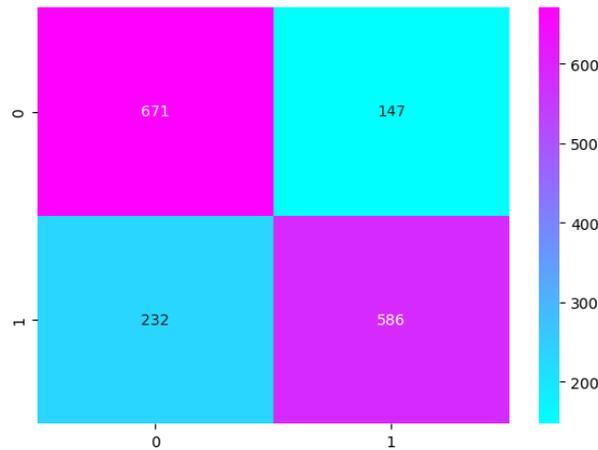


FIG. 4.22 : Visualisation graphique de la matrice de confusion

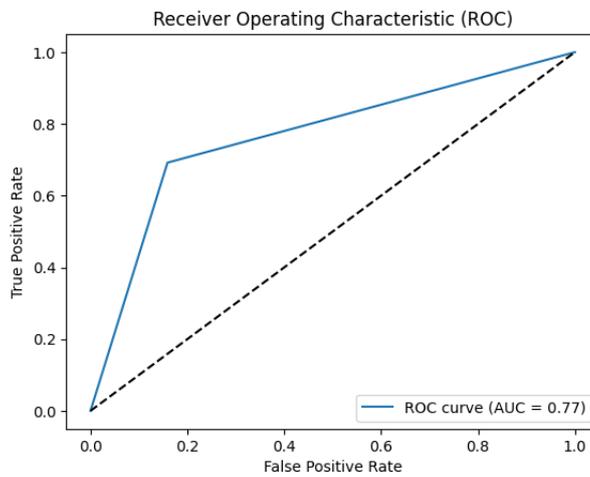


FIG. 4.23 : Receiver Operating Characteristic (ROC)

## 4.8 Comparaison et analyse des performances

Le modèle LSTM est entraîné pendant 100 époques, tandis que le modèle BERT est entraîné pendant 31 époques.

Model	Loss	Accuracy	Precision	Recall
LSTM	0.4576	0.7982	0.8091	0.6843
BERT	0.5166	0.7665	0.8132	0.6919

TAB. 4.2 : Les Résultats

Voici une analyse des différentes métriques :

**Loss (Perte) :** Il s'agit d'une mesure de la déviation entre les valeurs prédites par le modèle et les valeurs réelles. Un score de perte plus faible indique une meilleure performance du modèle. Dans ce tableau, le modèle LSTM affiche une perte de 0.4576, tandis que le modèle BERT a une perte légèrement plus élevée de 0.5166. Cela indique que le modèle LSTM a une meilleure performance en termes de perte que le modèle BERT.

**Accuracy (Exactitude) :** Il s'agit d'une mesure de l'exactitude globale du modèle, c'est-à-dire du pourcentage de prédictions correctes par rapport au total des prédictions. Le modèle LSTM a une

exactitude de 0.7982, tandis que le modèle BERT a une exactitude légèrement inférieure de 0.7665. Cela indique que le modèle LSTM a une meilleure performance en termes de exactitude globale que le modèle BERT.

**Precision (Précision) :** Il s'agit d'une mesure de la proportion de résultats positifs prédits correctement par le modèle parmi tous les résultats positifs prédits. Un score de précision plus élevé indique une meilleure capacité du modèle. le modèle LSTM a une précision de 0.8091, tandis que le modèle BERT a une précision légèrement supérieure de 0.8132. Cela indique que le modèle BERT a une meilleure performance en termes de précision des résultats positifs que le modèle LSTM.

**Recall (Rappel) :** Il s'agit d'une mesure de la proportion des résultats positifs prédits correctement par le modèle parmi tous les résultats réels positifs. Un score de rappel plus élevé indique une meilleure capacité du modèle à trouver les résultats positifs réels. Le modèle LSTM a un rappel de 0.6843, tandis que le modèle BERT a un rappel légèrement supérieur de 0.6919. Cela indique que le modèle BERT a une meilleure performance en termes de rappel des résultats positifs que le modèle LSTM.

le modèle LSTM semble avoir de meilleures performances en termes de perte et de exactitude globale, tandis que le modèle BERT présente de meilleures performances en termes de précision des résultats positifs et de rappel des résultats positifs. Il est important de noter que cette analyse est basée uniquement sur les informations fournies dans le tableau et qu'il peut y avoir d'autres facteurs à considérer lors de l'évaluation de la performance globale des modèles.

**Temps d'entraînement** généralement le modèle LSTM peut être plus rapide en termes de temps d'entraînement par rapport au modèle BERT. Cela est dû à la différence de complexité et de taille des deux modèles.

D'autre part, le modèle BERT est beaucoup plus complexe et contient un grand nombre de paramètres. Il utilise des transformers et nécessite un entraînement plus intensif pour apprendre les représentations de langage bidirectionnelles. Cela peut entraîner des temps d'entraînement plus longs et nécessite des ressources informatiques plus importantes.

**La courbe ROC (Receiver Operating Characteristic)** est une représentation graphique de la performance d'un modèle de classification binaire. Elle illustre le compromis entre le taux de vrais positifs (TPR) et le taux de faux positifs (FPR) à différents seuils de classification. tel que :

$$TPR = \frac{TP}{(TP + FN)} \quad (4.1)$$

$$FPR = \frac{FP}{(FP + TN)} \quad (4.2)$$

Pour AUC-ROC (aire sous la courbe caractéristique de fonctionnement du récepteur) :

AUC-ROC > 0.9 Un excellent classificateur.

0.8 < AUC-ROC ≤ 0.9 Un bon classificateur.

0.7 < AUC-ROC ≤ 0.8 Classificateur acceptable.

0.6 < AUC-ROC ≤ 0.7 Un mauvais classificateur.

AUC-ROC ≤ 0.6 Échec du classificateur.

Dans la figure 4.17, l'Aire sous la courbe ROC (AUC-ROC) est de 0,78, ce qui indique d'une performance acceptable du modèle LSTM. De même, dans la figure 4.23, l'Aire sous la courbe ROC (AUC-ROC) est de 0,77, ce qui indique une performance acceptable du modèle BERT.

## 4.9 Conclusion

Après l'analyse des résultats obtenus et discussion de ceux-ci, il y a une convergence des performances des deux modèles dans la classification binaire, mais le modèle de BERT prend beaucoup de temps à apprendre, contrairement à LSTM, et cela parce que le modèle BERT contient un grand nombre de

paramètres. et il nécessite des appareils puissants, et à partir de là, On peut dire que le modèle LSTM peut prédire rapidement les catastrophes en temps réel.

# Conclusion et perspectives

### Conclusion générale

BERT excelle lorsque des quantités abondantes de données d'entraînement sont disponibles. Il s'est avéré être un outil précieux pour analyser les publications sur les réseaux sociaux en capturant des nuances sémantiques et contextuelles complexes. Avec une Exactitude de 76%, il peut identifier correctement une proportion importante de messages liés à une catastrophe, fournissant des informations précieuses aux équipes d'intervention.

D'autre part, LSTM est mieux adapté aux tâches spécifiques où la compréhension des dépendances à long terme est essentielle, grâce à sa capacité à conserver et à utiliser les informations contextuelles sur le long terme. Il a atteint une Exactitude de 79%.

L'utilisation des modèles BERT et LSTM dans la conception d'un modèle d'apprentissage basé sur les médias sociaux pour la gestion des catastrophes représente une avancée majeure. Ces modèles offrent une précision précieuse dans la classification des tweets en cas de catastrophe, contribuant ainsi à une meilleure gestion des crises et à une réponse plus efficace. Cependant, il est nécessaire de poursuivre l'exploration de ces approches et de continuer les efforts d'optimisation afin de maximiser leur utilité potentielle et leur applicabilité dans des situations de catastrophe réelles.

### Perspectives

Les modèles BERT et LSTM ont démontré des performances solides dans le traitement du langage naturel, ce qui rend leurs perspectives très prometteuses. Cependant, il est important de noter que chaque modèle présente des avantages et des limitations spécifiques.

Il est essentiel de choisir le modèle en tenant compte des spécificités de la tâche, des ressources disponibles et des exigences de performance. Dans certains cas, une combinaison des deux modèles peut être bénéfique, en exploitant les avantages de chaque approche pour obtenir des résultats améliorés.

Il est également important d'envisager l'intégration de nouvelles sources de données, telles que les capteurs environnementaux et les données géospatiales, pour une vue plus complète de la situation. De plus, le développement des modèles capables de traiter les informations en temps réel à partir des réseaux sociaux permettrait une réactivité accrue dans la prise de décision lors des situations de crise.

En conclusion, les modèles BERT et LSTM ouvrent de nouvelles perspectives passionnantes dans le domaine du traitement du langage naturel. Ils offrent des outils puissants pour la classification de texte, la génération de texte et de nombreuses autres tâches liées au langage. L'utilisation judicieuse de ces modèles en fonction des besoins spécifiques permettra de tirer le meilleur parti de leurs capacités et d'obtenir des résultats de haute qualité.

# Bibliographie

- [1] Bouzidi Zair et al. (2018). *A new efficient alert model for disaster management. Proceedings of Conference AIAP.*
- [2] Louis Ngamassi et al. (2016). *Use of Social Media for Disaster Management : A Prescriptive Framework.*
- [3] Chikoto et al. (2013). *Disaster mitigation and preparedness : Comparison of nonprofit, public, and private organizations.*
- [4] Kawai et al. (2015). *Tsunami evacuation drill system using smart glasses.*
- [5] Si Mohamed and Ben Massou. *TS08B-Disaster Risk Management.*
- [6] Kigotho et al. (2012). *It's time to connect : Twitter for educational purposes.*
- [7] Mitsuhashi et al. (2015). *Web-based system for designing game-based evacuation drills.*
- [8] Lai Hang Dennis et al. (2016). *Everyday knowledge and disaster management : the role of social media.*
- [9] Robert et al. (2016). *Comparing felt reports and tweets about earthquakes.*
- [10] Lagmay et al. (2017). *Street floods in Metro Manila and possible solutions.*
- [11] *SIMPLIFYING THE DIFFERENCE : MACHINE LEARNING VS DEEP LEARNING.* (2020). <https://www.scs.org.sg/articles/machine-learning-vs-deep-learning/>.
- [12] George Seif. (2018). *I'll tell you why Deep Learning is so popular and in demand.* <https://medium.com/swlh/ill-tell-you-why-deep-learning-is-so-popular-and-in-demand-5aca72628780>.
- [13] Hochreiter. (1998). *The vanishing gradient problem during learning recurrent neural nets and problem solutions.*
- [14] Hinton et al. *Réseau neuronal artificiel.*
- [15] Shaistha fathima. (2019) *Not too deep INTRO to Neural Network and Perceptron!*. <https://www.ibm.com/topics/underfitting>.
- [16] Shaistha fathima. (2021). *What exactly is TensorFlow?*. <https://medium.datadriveninvestor.com/what-exactly-is-tensorflow-80a90162d5f1>.
- [17] Es-Sabery et al. (2021). *Sentence-level classification using parallel fuzzy deep learning classifier.*
- [18] Kishan Maladkar. (2018). *Types Of Activation Functions In Neural Networks.* <https://analyticsindiamag.com/most-common-activation-functions-in-neural-networks-and-rationale-behind-it/>
- [19] Pedro Borges. (2018). *Deep Learning :Recurrent Neural Networks.* <https://medium.com/deeplearningbrasil/deep-learning-recurrent-neural-networks-f9482a24d010>.
- [20] Patrick HairyTM. (2021). *Les réseaux de neurones récurrents pour les séries temporelles.* <https://metalblog.ctif.com/2021/09/06/les-reseaux-de-neurones-recurrents-pour-les-series-temporelles/>.

- [21] Cheng He. (2020). *Transformer in CV*. <https://towardsdatascience.com/transformer-in-cv-bbdb58bf335e>.
- [22] Vaswani et al. (2017). *Attention is all you need*.
- [23] . Computation, N. (2016). *Long short-term memory*.
- [24] Jacob Devlin et al. (2018). *Open Sourcing BERT : State-of-the-Art Pre-training for Natural Language Processing*. <https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>.
- [25] Dosovitskiy et al. (2020). *An image is worth 16x16 words : Transformers for image recognition at scale*.
- [26] Velickovic et al. (2019). *Deep graph infomax*.
- [27] Florian Stec. (2019). *Google BERT Update : tout sur la nouvelle mise à jour de l'algorithme Google*. <https://digitechnologie.com/google-bert-update/#:~:text=BERT>.
- [28] Shri Varsheni R. (2021). *Why and how to use BERT for NLP Text Classification ?*. <https://www.analyticsvidhya.com/blog/2021/06/why-and-how-to-use-bert-for-nlp-text-classification/>.
- [29] Juan G. *An Intuitive Explanation of Transformer-Based Models*. <https://factored.ai/transformer-based-language-models/>.
- [30] Devlin et al. (2018). *Bert : Pre-training of deep bidirectional transformers for language understanding*.
- [31] Allison Ragan. (2018). *-Taking the Confusion Out of Confusion Matrices*. <https://towardsdatascience.com/taking-the-confusion-out-of-confusion-matrices-c1ce054b3d3e>.
- [32] Louana Lelong. *Qu'est-ce que l'overfitting ?*. <https://www.jedha.co/formation-ia/overfitting>.
- [33] *What is underfitting ?*. <https://www.ibm.com/topics/underfitting>.
- [34] Ilyes Talbi. (2023). *7 méthodes pour éviter l'overfitting*. <https://larevueia.fr/7-methodes-pour-eviter-loverfitting/#:~:text=Une%20des%20m%C3%A9thodes%20les%20plus,d%27entra%C3%AEnements%20en%20plusieurs%20groupes>.
- [35] Vasilev et al. (2019). *Python Deep Learning : Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*.
- [36] Amar Budhiraja. (2016) *Dropout in (Deep) Machine learning*. <https://www.jedha.co/formation-ia/overfitting>.
- [37] Jain et al. (2015). *Real-time social network data mining for predicting the path for a disaster*.
- [38] Singh et al. (2019). *Event classification and location prediction from tweets during disasters*.
- [39] Kabir et al. (2019). *A deep learning approach for tweet classification and rescue scheduling for effective disaster management*.
- [40] Kumar et al. (2019). *A comparative analysis of machine learning techniques for disaster-related tweet classification*.
- [41] Devaraj et al. (2020). *Machine-learning methods for identifying social media-based requests for urgent help during hurricanes*.
- [42] Wang et al. (2021). *Transformer-based multi-task learning for disaster tweet categorisation*.
- [43] Addison Howard devrishi et al. (2019). *Natural Language Processing with Disaster Tweets*. <https://kaggle.com/competitions/nlp-getting-started>.