

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abdelhafid Boussouf Mila

Institut des Sciences et Technologie

Département des Mathématiques et Informatique

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

En : Informatique

Spécialité : Sciences et Technologies de l'Information et de la
Communication (STIC)

Thème :

Une nouvelle approche à base de deep learning pour
estimer la durée de vie utile pour la maintenance
conditionnelle

Préparé par : Feyrouz Mezaine

NourElheyet Hadhoud

Soutenu devant le jury:

Président	Afri Faiza	Grade MAA
Examineur	Boumessata Meriem	Grade MAA
Encadreur	Abderrezak Samira	Grade MAA

Année Universitaire : 2020-2021

Remerciements

Louanges à Dieu le tout puissant, grâce à qui nous avons pu faire tout ce travail.

Nos remerciements s'adressent à toute les personnes qui, d'une manière ou d'une autre nous ont permis de mener à son terme ce mémoire et plus particulièrement :

A notre encadrent, Mme Abdrazak samira qui nous a grandement aidées à définir avec circonspection le sujet de notre travail, et qui par ses conseils et idées innovantes, nous a permis d'améliorer la qualité du contenu. Son aide technique a été primordiale pour avancer dans notre projet.

Nos vifs remerciements vont aux membres de Jury, pour avoir accepté de juger notre travail.

A nos enseignants, à qui nous devons notre formation, qu'ils trouvent ici l'expression de nos sentiments les plus respectueux et de notre profonde gratitude pour leur gentillesse, leur disponibilité et leur soutien.

A nos chers parents, nos frères et soeurs et nos familles qui nous ont toujours soutenus le long de notre formation.

A tous nos amis, pour leurs soutiens moraux et aide précieuse.

FerNour

Dédicace

J'ai l'honneur de dédier ce modeste travail réalisé grâce à l'aide d'Allah le tout puissant

Je dédie ce travail marquant de ma vie

A mes parents que je ne remercierai jamais assez

Ma mère Naima qui a su habilement guider mes premiers pas dans ce monde. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

Mon cher père Salah, Tu représentes pour moi le Symbole de la bonté par excellence, la source de tendresse.

À mes chers frères : Lotfi, Abd elrahim, Sidali ,Ramzi,Anis.

A mon fiancé, Abd elmauman, merci de ton soutien sans faille et d'être toujours présent dans les bons moments comme dans les plus difficiles.

à mes chères amies :amina ,soumaya ,amina ,salima,aida,sihem,chahira,hassib

A toute ma famille et mes amis

A ma chère binôme Nour-El Heyet

et en fin à toutes mes collègues du promo 2020 et en fin à toutes mes collègues du promo 2021.

A vous tous merci

Feyrouz

Dédicace

*J'ai l'honneur de dédier ce modeste travail réalisé grâce à l'aide d'Allah le tout puissant
Je dédie ce travail marquant de ma vie
A mes parents que je ne remercierai jamais assez
Ma mère DJamila qui a su habilement guider mes premiers pas dans ce monde. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.
Mon cher père yahai, Tu représentes pour moi le Symbole de la bonté par excellence, la source de tendresse.
À mes chers frères : Salah, Amin
À mes chères sœurs : Ayat, soumai
A mes grand parent :ALI ,Mamar,Fatima et feyala.
A mes ancles et tantes et ses enfants
à mes chères amies :amina ,soumaia ,Djihan
A toute ma famille et mes amis
A ma chère binôme Feyrouz
et en fin à toutes mes collègues du promo 2020 et en fin à toutes mes collègues du promo 2021.
A vous tous merci*

NourElheyet

Résumé

Avec le développement de l'industrie intelligente et afin de détecter les conditions anormales des équipements le plus tôt possible, avant que la machine ne doive être remplacée ou réparée, et de les éviter, la durée de vie utile restante (RUL) est estimée. Elle est au coeur du domaine de PHM (Pronostic and health management) et la maintenance conditionnelle (CBM). Une haute précision de la prédiction de RUL permet de bien garantir la fiabilité et la sécurité d'un composant ou d'un système. Elle réduit non seulement la consommation de main-d'œuvre et de matériaux, mais réduit également le besoin de maintenance future.

Plusieurs approches ont été développées pour estimer RUL. Les approches basées sur les données utilisent des données de capteurs et des données opérationnelles pour estimer RUL. De nos jours, dans cette classe de problèmes, différents types de réseaux de neurones sont utilisés.

Dans ce travail, nous avons proposé un modèle hybride, combinant le réseau deep AutoEncoder convolutionnel caractérisé par sa capacité d'extraction des caractéristiques, avec le réseau deep Gru Bidirectionnel LSTM. Ce dernier est un type du réseau de neurones RNN, connu par sa capacité de gérer la dépendance de séquence.

Un tuning d'hyperparamètres est fait afin d'obtenir un modèle performant. Enfin, une comparaison avec les travaux similaires est faite et montre des résultats prometteurs. Pour évaluer les performances du modèle proposé, un jeu de données de simulation de dégradation de turboréacteur d'avion fourni par la NASA-CMAPSS a été utilisé.

Mots clés : Maintenance conditionnelle (CBM); Durée de vie restante utile (RUL); Le réseau deep Gru Bidirectionnel Long-Short Term Memory; Le réseau deep Auto-Encoder Convolutionnel; l'ensemble de données C-MAPSS.

Abstract

With the development of intelligent industry and in order to detect abnormal conditions of equipment as early as possible, before the machine needs to be replaced or repaired, and to avoid them, the remaining useful life (RUL) is estimated. It is at the heart of the field of PHM (Prognosis and health management) and conditional maintenance (CBM). High accuracy of RUL prediction helps ensure the reliability and safety of a component or system. It not only reduces the consumption of labor and materials, but also reduces the need for future maintenance.

Several approaches have been developed to estimate RUL. Data-driven approaches use sensor data and operational data to estimate RUL. Nowadays, in this class of problems, different types of neural networks are used.

In this work, we proposed a hybrid model, combining the convolutional AutoEncoder deep network characterized by its feature extraction capacity, with the LSTM Bidirectional deep Gru network. The latter is a type of the RNN neural network, known for its ability to manage the sequence dependency.

Hyperparameter tuning is done in order to obtain an efficient model. Finally, a comparison with similar works is made and shows promising results. To assess the performance of the proposed model, an aircraft turbojet degradation simulation dataset provided by NASA-CMAPSS was used.

Keywords Condition-based maintenance (CBM); Remaining Useful Life (RUL), Gru Bidirectional, Long-Short Term Memory neural network (LSTM), Convolutional auto-encoder neural network(CAE), C-MAPSS dataset .

تلخيص:

مع تطور الصناعة الذكية ومن أجل اكتشاف الظروف غير الطبيعية للمعدات في أقرب وقت ممكن ، قبل أن تحتاج الآلة إلى استبدالها أو إصلاحها ، ولتجنبها ، يتم تقدير العمر الإنتاجي المتبقي (RUL) إنه في قلب مجال PHM إدارة التشخيص والصحة) والصيانة المشروطة (CBM). تساعد الدقة العالية لتنبؤ RUL على ضمان موثوقية وسلامة مكون أو نظام. فهو لا يقلل من استهلاك العمالة والمواد فحسب ، بل يقلل أيضًا من الحاجة إلى الصيانة المستقبلية.

تم تطوير العديد من الأساليب لتقدير RUL تستخدم الأساليب القائمة على البيانات بيانات الاستشعار والبيانات التشغيلية لتقدير RUL في الوقت الحاضر ، في هذه الفئة من المشاكل ، يتم استخدام أنواع مختلفة من الشبكات العصبية.

في هذا العمل ، اقترحنا نموذجًا هجينًا ، يجمع بين شبكة التشفير التلقائي العميقة التلافيفية التي تتميز بقدرتها على استخراج الميزات ، مع شبكة Gru العميقة ثنائية الاتجاه LSTM. هذا الأخير هو نوع من الشبكة العصبية RNN ، والمعروفة بقدرتها على الإدارة تبعية التسلسل.

يتم إجراء ضبط Hyperparameter من أجل الحصول على نموذج فعال. أخيرًا ، تم إجراء مقارنة مع أعمال مماثلة وتظهر نتائج واعدة.

لتقييم أداء النموذج المقترح ، تم استخدام مجموعة بيانات محاكاة تحلل نفائة للطائرات مقدمة من NASA-CMAPSS

الكلمات المفتاحية :

صيانة الحالة (CBM) ؛ الحياة المفيدة المتبقية (RUL) ؛ شبكة Gru العميق للذاكرة ثنائية الاتجاه طويلة المدى ؛ الشبكة العميقة التلافيفية للتشفير التلقائي ؛ مجموعة بيانات C-MAPSS.

Table des matières

Remerciements	3
Dédicace	5
Dédicace	7
Table des figures	17
Liste des tableaux	19
Liste des abréviations	21
Introduction général	25
Première partie. La Maintenance conditionnelle	29
Chapitre 1. La Maintenance Conditionnelle	31
1.1. Introduction	31
1.2. Maintenance	31
1.2.1. Historique de la maintenance	31
1.2.2. Définition de la maintenance	33
1.2.3. La sûreté de fonctionnement (SDF)	33
1.2.3.1. Définition :	33
1.2.3.2. Les concepts de maintenance(F.M.D.S) ..	33
1.2.4. Les niveaux de la maintenance	34
1.2.5. Les différentes formes de la maintenance ..	36
1.2.5.1. La maintenance corrective :	36
1.2.5.2. La maintenance préventive :	37
1.2.6. Prognostics and Health Management (PHM) 41	
1.2.6.1. Architecture du PHM :	41
1.2.6.2. Diagnostic	43
1.2.6.3. Pronostic	44
1.3. Définition de RUL	45
1.4. Les approches de RUL	46

1.4.1.	Les approches basées sur model physique	46
1.4.1.1.	Les avantages	47
1.4.1.2.	Les inconvénients	47
1.4.2.	Les approches basées sur les donnés	47
1.4.2.1.	Les avantages	47
1.4.2.2.	Les inconvénients	48
1.4.3.	Les approches hybrides	48
1.4.3.1.	Les avantages	49
1.4.3.2.	Les inconvénients	49
1.5.	conclusion	49
Deuxième partie. L'Apprentissage profond		
(Le deep learning)		51
Chapitre 2. L'Apprentissage profond (Deep Learning)		53
2.1.	Introduction	53
2.2.	L'apprentissage automatique	54
2.2.1.	Définition.....	54
2.2.2.	les types de l'apprentissage automatique	55
2.2.2.1.	L'Apprentissage supervisé	56
2.2.2.2.	L'Apprentissage non supervisé :	57
2.2.2.3.	Apprentissage par renforcement	58
2.2.3.	Domaines d'application de l'apprentissage automatique	
	60	
2.3.	Le Réseaux de neurones	60
2.3.1.	Définition.....	60
2.3.2.	Les différents types de réseaux neuronaux.....	61
2.3.2.1.	Neurone Formel « mono couches »	61
2.3.2.2.	Neurones multicouche :	61
2.3.3.	Les différentes applications des réseaux de neurones .	63
2.3.3.1.	Reconnaissance de formes.....	63
2.3.3.2.	Traitement de la parole.....	63
2.3.3.3.	Détection d'anomalies.....	64
2.3.3.4.	Traitements dépendant du temps.....	64
2.4.	L'apprentissage profond	64
2.4.1.	Définition de l'apprentissage profond (deep learning)	64

2.4.2.	LES Différents types de réseaux de neurones profond.	65
2.4.2.1.	Perceptron Multi couches (MLP)	65
2.4.2.2.	Réseau de neurones récurrents (RNN).....	66
2.4.2.3.	Réseau de neurones à convolution (CNN)	71
2.4.2.4.	Les Auto-Encodeurs.....	76
2.4.3.	Fonctionnement du deep Learning.....	77
2.4.4.	Les Applications du deep Learning.....	78
2.4.5.	Les avantages de Deep Learning.....	79
2.5.	L'apprentissage automatique VS L'apprentissage profond :où le Deep Learning Fait la difference?	79
	Conclusion.....	79
Troisième partie. La durée de vie utile restante (RUL) et le deep learning		81
Chapitre 3. La durée de vie utile restante (RUL) avec le deep learning		83
3.1.	Introduction	83
3.2.	Les travaux existants de RUL avec le deep learning	83
3.2.1.	RUL avec Auto-encodeur.....	84
3.2.2.	RUL avec les réseau de croyances profondes (DBN)...	85
3.2.3.	RUL avec les réseaux de neurones convolutifs(CNN) .	86
3.2.4.	RUL avec les réseaux de neurones récurrents (RNN) ..	88
3.2.5.	RUL avec un modèle deep hybride	90
3.3.	Conclusion	90
Quatrième partie. CONTRIBUTION		91
Chapitre 4. CONTRIBUTION		93
4.1.	Introduction	93
4.2.	Motivation	93
4.3.	L'architecture du modèle proposé	94
4.4.	Expérimentation et discussions	95
4.4.1.	Les étapes de travail.....	95
4.4.2.	L'ensemble de données utilisé	97
4.4.3.	Fonctions de métriques	98

4.4.3.1.	L'Erreur Absolue Moyenne (Mean Absolute Error (MAE)) :	98
4.4.3.2.	Erreur quadratique moyenne racine (Root mean squared error (RMSE)) :	99
4.4.3.3.	Erreur quadratique moyenne (Mean Squared Error(MSE)) :	99
4.4.3.4.	Fonction de score de régression (R^2) :	99
4.4.4.	Le pré-traitement de données	100
4.4.4.1.	Le pré-traitement des données de l'apprentissage et de test :	100
4.4.4.2.	La normalisation des données :	101
4.4.4.3.	Le prétraitement de "time window"	101
4.4.5.	L'ajustement des hyperparamètres	101
4.4.5.1.	L'effet de modification des valeurs de windows size :	101
4.4.5.2.	L'effet des valeurs de fonctions d'activation et d'optimizers	102
4.4.5.3.	Effet des différentes valeurs de validation-split .	103
4.4.5.4.	L'effet de changement de batch size sur les résultats	103
4.4.5.5.	L'effet de modification des valeurs de learning rate :	104
4.4.5.6.	L'effet du changement de nombre de neurones de GRUBi sur les résultats :	104
4.4.5.7.	L'effet de changement du nombre de filtre :	105
4.4.6.	Les résultats finals du modèle proposé	107
4.4.7.	L'affichage des résultats finals	107
4.4.8.	Comparaison avec les approches similaires de la littérature utilisant CMAPSS	110
4.5.	Conclusion	112
Cinquième partie. Implémentation		113
Chapitre 5. IMPLEMENTATION		115
5.1.	Introduction	115
5.2.	Les outils de développement	115
5.2.1.	Le langage python	115

5.2.2.	Environnement Anaconda	116
5.2.3.	google colab	116
5.2.4.	Les bibliothèques Tensorflow et Keras	116
5.3.	Quelques Fragments de code	117
5.4.	Exemples d'exécution.....	117
5.5.	Conclusion	118
Conclusion générale		121
Conclusion générale		121
Bibliographie		123
Bibliographie		129

Table des figures

1	Les formes de maintenance.	39
2	Coûts associés à différentes approches de maintenance	40
3	Architecture OSA/CBM.	42
4	Classification des méthodes de diagnostic	45
1	La relation Entre L'IA et ML et le Deep Learning	54
2	Techniques de Machine Learning	55
3	Un jeu d'entraînement étiqueté pour un apprentissage supervisé.....	57
4	Un jeu d'entraînement non étiqueté pour un apprentissage non supervisé	58
5	Le clustering identifie des modèles cachés dans les données.	58
6	Apprentissage par renforcement	59
7	Les Types d'apprentissage.....	59
8	perceptron simple avec et sans le bias.	62
9	Perception multicouche.	62
10	MLP.....	66
11	transforme une couche cachée de ANN en RNN	67
12	Le schéma de réseau RNN.....	68
13	Une chaîne de cellules LSTM.....	69
14	Schéma d'un réseau GRU à une unité	70
15	Structure générale des réseaux de neurones récurrents bidirectionnels GRU.	71
16	Modèle Many2Many Seq2Seq	71
17	RNN déroulé.....	72
18	un réseau de neurone cnn	72
19	un exemple de convolution avec une matrice de taille 6 x 6	73
20	Sortie de convolution.....	73

21	CNN – Classification des images	74
22	Image convolante avec un filtre 1	75
23	Image convolante avec un filtre 2	75
24	Image montrant le fonctionnement d’un auto-encodeur. ...	76
25	un processus d’autoapprentissage	77
1	l’architecture du modèle proposé.	95
2	tableau détaillé des sous datasets de CMAPSS.	98
3	L’affichage des métriques R2 avec le modèle (CAEBIGRU) pour la sous donnée FD001 de CMAPSS.	108
4	Les résultats finals du modèle CAEBi	109
5	Les plots des métriques MAE avec le modèle (CAE BIGRU) pour la sous donnée FD001 de CMAPSS.	109
6	Les plots des métriques RMSE avec le modèle (CNNAEBIGRU) pour la sous donnée FD001 de CMAPSS.	110
7	Les plots de distribution d’erreur avec le modèle (CAEBIGRU) pour la sous donnée FD001 de CMAPSS.	111
8	Les plots de regression avec le modèle (CAEBIGRU) pour la sous données FD001 de CMAPSS.	111
1	Les bibliothèques Tensorflow et Keras	117
2	Normalisation des données.	117
3	Le Model CAEBIGRU.	118
4	Le Modèle CAEBIGRU.	118
5	La structure du Modèle CAEBIGRU.	118
6	L’exécution du modèle CAE-BIGRU.	119
7	L’exécution de training du modèle CAE-BIGRU.	119
8	L’exécution de test du modèle CAE-BIGRU.	119

Liste des tableaux

1	Classification des niveaux de maintenance.	35
1	L'apprentissage automatique VS L'apprentissage pro-fond :où le Deep Learning Fait la difference?	80
1	les résultats du modèle avec différentes valeurs de windows size	102
2	les résultats du modèle avec différentes valeurs de fonctions d'activations et d'optimiser	103
3	Résultats de modification de validation-split.....	103
4	Les résultats de modifications de batch size	104
5	les résultats de modification de learning rate.....	105
6	Les différents résultats du changement du nombre de num time steps à 300	105
7	Les différents résultats du changement du nombre de num time steps à 200	106
8	Les différents résultats du changement du nombre de num time steps à 400	106
9	Résultats pour filtre = [10;10;10,1]	106
10	Résultats pour filtre = [20;20;20,1]	107
11	Les résultats finals de modèle proposé(CAEBIGRU)	107
12	Comparaison de RMSE sur C-MAPSS sous le sous dataset FD001.....	112

Liste des abréviations

DL	Deep learning
ML	Machine learning
IA :	Artificial Intelligence.
SVM :	Support vector Machine .
DBN :	Deep Belief Network.
CBM :	condition based maintenance
RUL :	Remaining useful life
SDF :	La sureté de fonctionnement
AFNOR :	assocaition française de normalisation
CNN :	Convolutional Neural Network.
RNN :	Recurrent Neural Network.

RELU : Rectified Linear Unit.

RNA : Réseaux de Neurones Artificielles.

PHM : prognostics and health management .

MSE : mean square error.

NASA : National Aeronautics and Space Administration.

MLP : Multi Layer Perceptron.

MAE : mean absolute error.

LSTM : Long Short-Term Memorys.

C-MAPSS : Commercial Modular Aero-Propulsion System Simulation.

BILSTM : Bidirectional long short term memorys.

GRU : Gate recurrent unit .

BIGRU : Bidirectional gate recurrent unit.

AE : Auto encodeur.

OSA : Open system architecture .

RMSE : Root mean squared erreur.

R^2 : Fonction de score de regression.

SVM : Support vecteur machine .

MLP : Multi layer perception .

CAE : convolution auto encodeur.

LR : Logistique régression.

BM : Boltzman machine.

ACO-DDBN : Reseau de crayon profond discriminant avec une colonie de fourmis optimisation .

SSL : Single scale low.

SSH : Single scale high

CBLSTM : Convolutionnelle bidirectionnelle réseaux mémoire
longue à court terme

CAEBILSTM : Convolution auto encodeur bidirectional long short
term memory

Introduction général

La maintenance est une étape importante dans tout domaine de l'industrie. Les stratégies traditionnelles de la maintenance telles que la maintenance corrective en cas de panne et la maintenance préventive systématique programmée n'ont pas montré leurs fiabilité et efficacité dans le domaine. Les technologies intelligentes de gestion des pronostics et de la santé (PHM), également appelées maintenance conditionnelle (CBM), montrent des capacités prometteuses dans les industries. elle contribue à la maximisation de la disponibilité opérationnelle, la réduction des coûts de maintenance et l'amélioration de la fiabilité et de la sécurité du système en surveillant les conditions de l'installation. L'objectif de la maintenance conditionnelle est l'estimation de la durée de vie utile restante (en anglais : Remaining Useful Life, RUL) de la machine pour l'évaluation de la dégradation de ses performances.

La durée de vie utile restante (RUL) peut être estimée en fonction des données de trajectoire de l'historique, ce qui est très important pour améliorer les calendriers de maintenance afin d'éviter les pannes catastrophiques d'ingénierie et d'économiser les coûts résultants. L'estimation RUL a un rôle important dans différents domaines, y compris les industries aéronautiques, les équipements médicaux et les centrales électriques, ce qui a inspiré les chercheurs à développer une variété d'approches de prédiction RUL.

En général, les approches existantes pour l'estimation de RUL peuvent être regroupées en trois catégories principales, les approches basées sur les modèles, les approches basées sur les données et les approches hybrides.

Les approches basées sur les modèles ont tendance à être plus précises si la dégradation du système complexe est modélisée avec précision, elles nécessitent des connaissances préalables approfondies sur les systèmes physiques qui ne sont généralement pas disponibles dans la

pratique .

Les approches basées sur les données sont capables de modéliser les caractéristiques de dégradation basées sur les données historiques des capteurs. Les corrélations et causalités sous-jacentes dans les données de capteur collectées peuvent être révélées, et les informations du système de correspondances telles que RUL peuvent être déduites. Les approches basées sur les données nécessitent généralement des données historiques suffisantes pour la formation des modèles. Ces dernières années, de nombreux algorithmes basés sur les données ont été proposés, de bons résultats pronostiques ont été obtenus, notamment : les modèles d'apprentissage comme les réseaux de neurones, la machine à vecteurs de support (SVM), les modèles de Markov cachés, etc.

Récemment, un outil puissant émergée appelée Deep learning où des abstractions de haut niveau de données peuvent être bien modélisées à l'aide de structures profondes complexes, conduisant à une extraction de caractéristiques plus efficace par rapport aux réseaux peu profonds. Les méthodes d'apprentissage en profondeur ont suscité un grand intérêt et obtenu des résultats significatifs dans de nombreux domaines, notamment la reconnaissance d'images et la reconnaissance vocale. Étant donné que les données brutes obtenues à partir de la surveillance de la santé des machines partagent une dimensionnalité élevée similaire à celles des recherches sur le traitement d'images, l'architecture d'apprentissage en profondeur a un grand potentiel dans la surveillance de la santé du pronostic (PHM) et l'estimation RUL.

Les données décrivant les paramètres de la maintenance conditionnelle (CBM) sont classées en tant que types de séries chronologiques. Les problèmes de prédiction de séries chronologiques sont un type difficile de modélisation prédictive, qui contrairement à la modélisation prédictive de régression, ils ajoutent également la complexité d'une dépendance de séquence entre les variables d'entrée. Un type puissant de réseau de neurones conçu pour gérer la dépendance de séquence est appelé réseaux de neurones récurrents (RNN). Le LSTM à son tour gère les dépendances à long term. Un type de LSTM est le

GRU bidirectionnel Lstm (BiGruLSTM), Il a donné des résultats prometteurs en particulier dans la capture des dépendances temporelles cachées à long terme parmi les signaux de séquence temporelle. Un autre type de modèle d'apprentissage en profondeur est le réseau de neurones convolutifs (CNN) et l'autoencodeur (AE). Ils ont amélioré leurs performances dans le domaine, en particulier ils ont la capacité d'extraire les caractéristiques locales importantes à partir de données séquentielles. Profitant de leurs caractéristiques, nous avons choisi de les utiliser dans un modèle deep hybride pour l'estimation de RUL.

Notre objectif s'inscrit dans le cadre de l'estimation de RUL en utilisant le deep learning. Pour atteindre cet objectif, nous proposons un modèle hybrid basé sur l'AutoEncoder convolutionnel et le GRU Bidirectionnel Lstm. Ensuite, nous faisons varier plusieurs hyperparamètres, analyser l'effet à chaque fois sur les résultats. A la fin, une comparaison entre ce modèle et ceux de la littérature sera accomplie. Le modèle proposé sera entraîné, tester et valider sur l'ensemble de données CMAPSS de la NASA.

Organisation du mémoire

Ce mémoire est structuré de la manière suivante :

chapitre 01 : La maintenance conditionnelle :

Dans ce chapitre tout d'abord on définit la maintenance ,les différents formes et niveaux , l'architecture de la maintenance, le pronostic et le health management. Nous terminons par la définition du temps restant utile et les approches de développement.

chapitre 02 : L'Apprentissage profond (Deep learning) :

Dans ce chapitre nous introduirons premièrement les réseaux de neurones en général,ensuite nous nous intéressons au deep learning et ses différents types. Nous définissons plus en détail les sous types de Cnn etRnn qu'on va utiliser dans notre approche,et en fin nous citons quelques travaux existents de RUL avec le deep learning.

chapitre 03 :Le temps restant utile (Remaining useful life RUL) avec le deep learning :

Dans ce chapitre, nous présentons quelques travaux de l'estimation de RUL avec le deep learning.

chapitre 04 : Contribution :

Nous allons faire dans ce chapitre une étude expérimentale sur l'approche proposée avec une discussion de chaque résultat, une analyse et comparaison des résultats de la méthode utilisée ainsi qu'une comparaison avec les résultats de quelques travaux mentionnés dans la littérature.

chapitre 05 : Implémentation :

Dans ce chapitre, nous abordons les outils de développement utilisés dans notre application .

Nous présentons aussi quelques fragments de codes de l'exécution.

Nous terminons enfin, par une conclusion générale qui résume notre travail et nous donne une prévision sur des travaux en futur.

Première partie

La Maintenance conditionnelle

Chapitre 1

La Maintenance Conditionnelle

1.1. Introduction

Les activités de maintenance, au sens de dépannage d'un équipement, ont toujours existé. Mais ces activités étaient au départ peu ou pas formalisées : elles n'étaient pas nécessairement assurées par du personnel spécialisé, ni encadrées par des méthodes spécifiques. De plus, elles consistaient essentiellement à réparer un équipement une fois que celui-ci était défaillant, mais n'intégraient que peu la notion de "préventif", c'est-à-dire des interventions visant à prévenir la panne.

La notion formalisée de "maintenance" est relativement récente. Elle est apparue avec l'automatisation des systèmes de production, les enjeux économiques et industriels croissants, les réglementations strictes pour la protection de l'individu et de l'environnement.

La fonction de maintenance ne peut se réduire à la seule activité d'entretenir un parc de machine mais a vocation à intervenir dans tout le cycle de l'exploitation du système (choix et conception du matériel, mise en service, détermination des plans de maintenance, organisation et logistique des activités de maintenance, suivi de l'évolution du système, etc.). Dans ce chapitre, nous détaillerons la maintenance et ses différents types avec ses niveaux . Ensuite, nous définirons le pronostic et la gestion de santé, son architecture et ses outils qui sont utilisés dans le PHM .

1.2. Maintenance

1.2.1. Historique de la maintenance

L'apparition du term "maintenance" dans l'industrie a eu lieu vers 1950 aux USA, il se superpose progressivement à l'entretien.[1]

Années 60 : Maintenance réactive / corrective Intervient après la détection et la localisation d'un défaut.[2]

Années 70 : Maintenance préventive (préventive systématique) Effectuée dans l'intention de réduire la probabilité de défaillance d'un bien ou la dégradation d'un service rendu. C'est une intervention de maintenance prévue, préparée et programmée avant la date probable d'apparition d'une défaillance. La maintenance systématique : effectuée selon un échancier établi suivant le temps ou le nombre d'unités d'usage.

Années 80 : Maintenance prédictive (prévisionnelle, préventive conditionnelle) Basée sur la surveillance en continu de l'évolution du système, afin de prévenir un dysfonctionnement avant qu'il n'arrive sans prendre en compte la loi de dégradation. La décision d'intervention préventive est prise lorsqu'il y a un défaut imminent survient, ou approche d'un seuil de dégradation prédéterminé.

Années 90 : Maintenance proactive Implique la surveillance et la vérification continues des causes primaires de défaillance du système surveillé.

Années 2000 : La Télémaintenance et E-maintenance Le développement actuel des technologies de l'information et de la communication et la distribution de l'intelligence aux niveaux les plus bas, permet de passer de la maintenance classique à la maintenance à distance et en temps réel la « Télémaintenance » et la « E-maintenance » .[2]

- La Télémaintenance est un concept de récupération de données à distance ou de prise de contrôle et de décision à distance.
- La e-Maintenance est un concept lié au principe de web-services, elle intègre le principe de base de la Télémaintenance en lui associant une dimension forte, constituée par la coopération et le partage des connaissances au niveau des informations mais aussi des hommes, des services (ingénierie, exploitation, maintenance, sûreté, achats, comptabilité, ...) et des sociétés (client / fournisseur, inter fournisseurs, inter clients, ...).

1.2.2. Définition de la maintenance

La maintenance est définie comme la combinaison de toutes les actions techniques, administratives et de gestion au cours du cycle de vie d'un bien destinées à le conserver ou à le restaurer.[1]

un état dans lequel il peut remplir la fonction requise. Les tâches de maintenance varient selon les environnements de travail, qui incluent par ex. contrôle visuel, essais, mesures, changements de consommables (graissage, graissage, filtres à huile), réglage, réparation, entretien, remplacement de pièces, entretien, prélèvement d'huile, graissage, resserrage des boulons, nettoyage, détection de défaut, diagnostic de défaut etc . La maintenance peut être classée en trois types principaux, à savoir corrective, basée sur le temps et basée sur les conditions.

1.2.3. La sûreté de fonctionnement (SDF)

1.2.3.1. Définition : Selon AFNOR « c'est l'ensemble des propriétés qui décrivent la disponibilité et les facteurs qui la conditionnent : fiabilité, maintenabilité et logistique de maintenance ». Les préoccupations dites de « sécurité » sont très présentes dans le monde des machines-outils. Dans les applications de type manufacturier ou batch, les préoccupations sont plutôt liées à la disponibilité. Dès lors que la sécurité ou la « disponibilité » d'un système est mise en panne, on incrimine sa « fiabilité ». Enfin, en cas de dysfonctionnement, il convient de remettre le système en conditions de fonctionnement initial : c'est là qu'intervient la « maintenabilité ». Ces quatre caractéristiques constituent la sûreté de fonctionnement d'un système [2].

1.2.3.2. Les concepts de maintenance(F.M.D.S).

[a] Fiabilité

Aptitude d'un système à accomplir une fonction requise, dans des conditions données, pendant un intervalle de temps déterminé »[2].

La fiabilité d'un système c'est la probabilité que ce système s'accomplisse une fonction requise, dans des conditions données, pendant l'intervalle de temps (0, t).

$$R(t) = P [S \text{ non défaillant sur } (0, t)] \dots\dots\dots (1)$$

[b] **Maintenabilité**

« Aptitude d'un système à être maintenu ou rétabli, en un temps donné, dans un état de fonctionnement bien défini lorsque les opérations de maintenance sont accomplies avec des moyens donnés, suivant un programme déterminé » [2].

La maintenabilité est la probabilité que la maintenance d'un système S accomplie dans des conditions données, avec des procédures et des moyens prescrits, soit achevée au temps t, sachant que le système est défaillant à t = 0.

$$M(t) = P [S \text{ est réparé sur } (0, t)] \dots\dots\dots (2)$$

[c] **Disponibilité**

« Aptitude d'un système à être en état d'accomplir une fonction requise, dans des conditions données, à un instant donné, en supposant que la fourniture des moyens extérieurs soit assurée » [2].

La disponibilité est la probabilité qu'un système S₁ soit en état d'accomplir une fonction requise dans des conditions données et à un instant donné.

$$D(t) = P [S_1 \text{ non défaillant à l'instant } t] \dots\dots\dots (3)$$

[d] **Sécurité**

« Aptitude d'un système à ne pas générer, dans des conditions données, des événements critiques ou catastrophiques » [2].

La sécurité est la probabilité qu'un système S évite de faire apparaître, dans des conditions données, des événements critiques ou catastrophiques.

1.2.4. Les niveaux de la maintenance

La maintenance d'un bien s'exerce à travers de nombreuses opérations, parfois répétitives, sinon occasionnelles, communément définies jusqu'alors en 5 niveaux de maintenance.[3] Les niveaux de maintenance définie par rapport à la nature de l'intervention ainsi la qualification de l'intervenant. Voilà une représentation des niveaux de maintenance (AFNOR) dans le tableau suivant (Tableau 1).

1. Le niveau 01

Operation	Niveaux de maintenance	Classification AFNOR
Action simple	1	Utilisateur
Operation courante	2	Personnel qualifié
Operation spécialisée	3	Technicien qualifié
Intervention spécifique	4	Technicien ou équipe spécialisé
Renovation /réconstruction	5	Constructeur ,service ou société spécialisé

TABLEAU 1. Classification des niveaux de maintenance.

Ce type d'intervention peut être effectué par l'exploitant du bien, sur place, sans outillage et à l'aide des instructions d'utilisation. Le stock de pièces consommables nécessaires est très faible. Il contient des réglages simples prévus par le constructeur au moyen d'éléments, comme il nécessite aucun démontage ou ouverture de l'équipement, ou échanges d'éléments consommables accessibles en toute sécurité tels que certains fusibles, etc.

2.Le niveau 02

Il consiste à prouver des dépannages par échange standard des éléments prévus à cet effet et des opérations mineures de maintenance préventive comme le graissage ou le contrôle de bon fonctionnement de système. On peut se procurer les pièces de rechange transportables nécessaires sans délai et à proximité immédiate du lieu d'exploitation. Ce type d'intervention peut être effectué par un technicien de moyenne qualification, sur Lieux, avec l'outillage portable et simple.

3.Le niveau 03

Ce type d'intervention doit être effectué par un technicien spécialisé, sur place ou dans le local de maintenance, à l'aide de l'outillage prévu dans les instructions de maintenance et éventuellement des bancs d'essais et de contrôle des équipements tout ça en utilisant l'ensemble de la documentation nécessaire à la maintenance des systèmes. Il consiste à faire l'identification et le diagnostic des pannes, les réparations par échange de composants ou d'éléments fonctionnels et toutes opérations courantes de maintenance préventive telles que le réglage général des appareils de mesure.

4.Le niveau 04

Ce type d'intervention peut être effectué par une équipe comprenant un encadrement technique très spécialisé dans un atelier plus spécialisé. Englobe Tous les travaux importants de maintenance corrective ou préventive à l'exception de la rénovation et de la reconstruction.

Ce niveau comprend aussi le réglage des appareils utilisés pour la maintenance.

5. Le niveau 05

Par définition, ce type de travaux est effectué par le constructeur, ou par le reconstruteur, avec des moyens définis par le constructeur et donc proches de la fabrication. C'est la rénovation, la reconstruction ou l'exécution des réparations importantes confiées à un atelier central ou à une unité extérieure.

1.2.5. Les différentes formes de la maintenance

La maintenance permet de garder un système dans un état préalable défini (bon état) mais permet aussi la reconstitution et l'amélioration. L'entretien est une condition nécessaire mais insuffisante pour la maintenance [2].

Avant les années 60, le service maintenance avait pour principale mission d'intervenir sur les équipements en panne afin de les réparer le plus tôt possible. Ce type de maintenance appelé corrective, ensuite été peu à peu complété par une approche plus anticipative des phénomènes de défaillance, c'est à dire par une maintenance réalisée avant que la défaillance ne se produise. Cette deuxième forme de maintenance, dite préventive. Elle a été initialement mise en œuvre lors du développement du Boeing 747 (avion) en 1960.

1.2.5.1. La maintenance corrective : « Maintenance corrective c'est la maintenance exécutée après défaillance et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise ». Ce concept couvre simplement les actions à engager après une défaillance. Elle correspond à une attitude défensive dans l'attente d'une défaillance subite. Elle correspond à deux types d'intervention ; le dépannage et la réparation [2].

a) Le dépannage

« Le dépannage c'est une action sur un bien en panne, en vue de le remettre en état de fonctionnement. » Compte tenu de l'objectif, une action de dépannage peut s'accommoder des résultats provisoires et de condition de réalisation hors règles de procédures, et dans ce cas il sera suivie avec la réparation [1][2].

b) La réparation

« Intervention définitive et limitée de maintenance corrective après panne ou défaillance. » L'application de la réparation peut être décidée soit immédiatement à la suite d'une défaillance, soit après un dépannage ou bien après une visite de maintenance préventive conditionnelle ou systématique [1][2].

1.2.5.2. La maintenance préventive : « Maintenance exécutée à des intervalles prédéterminés ou selon des critères prescrits et destinée à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien » [4].

Ce sont des actions de maintenances effectuées pour objectif de réduire la probabilité de défaillance d'un bien ou la dégradation d'un service de système rendu. Elle correspond à une attitude proactive c'est-à-dire on agit avant la défaillance. On peut distinguer deux formes particulières de maintenance préventive :

1.2.4.2.1 La maintenance systématique

« Maintenance préventive exécutée à des intervalles de temps préétablis ou selon un nombre défini d'unités d'usage mais sans contrôle préalable de l'état du bien » [4].

C'est une maintenance préventive effectuée après un échéancier établi selon le temps ou le nombre d'unités d'usage. A l'intérieur de cette maintenance, on trouve la maintenance effectuée pour satisfaire aux réglementations diverses (visites et contrôles obligatoires, épreuves, ...), la maintenance de ronde (vérifications précises, visuelles ou instrumentées, graissages périodiques, ...), la maintenance des deux premiers niveaux, de plus en plus intégrée dans les tâches des opérateurs les visites périodiques des machines importantes, le remplacement périodique des pièces. C'est une maintenance qui coûte très cher, en particulier si l'on veut fixer une périodicité de visite qui fasse courir peu de risque d'avoir une défaillance avant l'intervention.

1.2.4.2.2 La maintenance conditionnelle

« Maintenance préventive basée sur une surveillance du fonctionnement du bien et/ou des paramètres significatifs de ce fonctionnement intégrant les actions qui en découlent. La surveillance du fonctionnement et des paramètres peut être exécutée selon un calendrier, ou à la demande, ou de façon continue ».[5]

Pour les équipements importants, on remplace la maintenance préventive systématique par la maintenance préventive conditionnelle : on ne déclenche pas les opérations sauf si certains paramètres arrivent à un seuil limite. La maintenance prédictive est subordonnée à un type d'évènement prédéterminé (information donnée par un capteur, autodiagnostic, échauffement des équipements, ...) révélateur de l'état de dégradation d'un bon rendement.[6]

Une récapitulative des types de maintenance est présentée dans la Figure 1

Il peut y avoir confusion, et débat, entre maintenance conditionnelle et prévisionnelle, mais peut-on considérer que la mise en œuvre de fonctions mathématiques simples relève de la « Predictive Maintenance »

1.2.4.2.2.1. Architecture CBM (Condition-based maintenance)

La maintenance basée sur la condition implique une surveillance continue des données du système afin d'obtenir une évaluation précise de l'état d'un composant du système et d'effectuer une maintenance en fonction de son état. Cela implique l'utilisation de la surveillance en temps réel du système et du traitement des données.[7]

Une autre capacité qui peut faire partie d'un système CBM est une capacité à fournir une estimation de la durée de vie résiduelle (RUL) du système ou du composant surveillé. Ce type d'actions fonctionnelles sont connus sous le nom de pronostics, par opposition aux diagnostics utilisés pour évaluer l'état actuel d'un système surveillé. [8]

Une approche de maintenance basée sur la condition promet une gamme d'améliorations par rapport aux approches existantes, avec une réduction des coûts de maintenance globale étant l'un des principaux moteurs du développement des organisations. Le coût associé à chacune des différentes approches de maintenance est illustré à la figure 2 . Une approche de maintenance corrective a un coût de maintenance relativement faible (minimales), mais des coûts de haute performance associés au coût élevé des pannes opérationnelles.[9]

En revanche, la maintenance préventive a généralement un faible coût d'exploitation, associée à des cas réduits de défaillances en cours d'emploi, mais elle utilise souvent des estimations très prudentes concernant la probabilité de défaillances de composants elle a donc un coût de maintenance élevé associé à l'élimination des composants

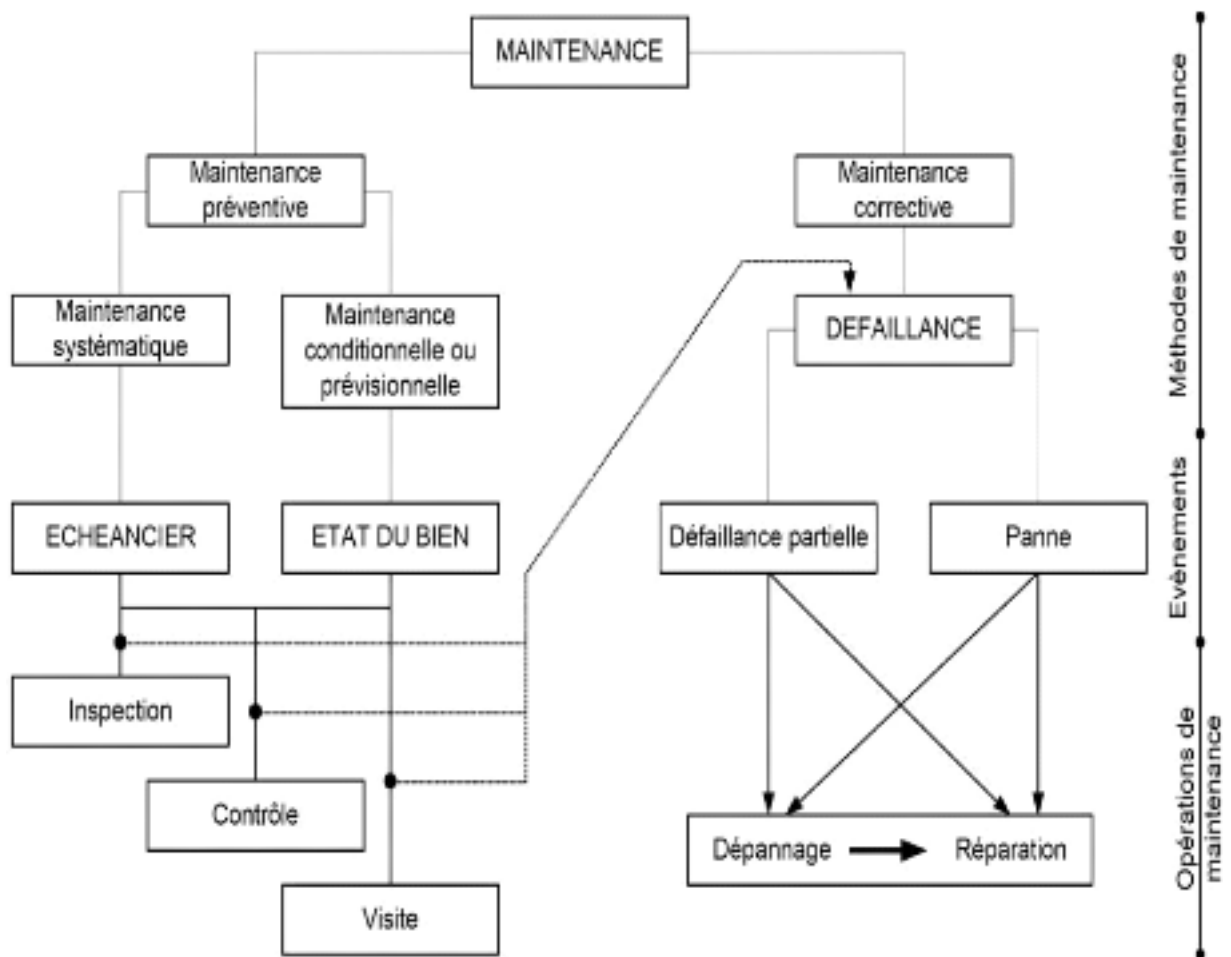


FIGURE 1. Les formes de maintenance.

avant qu'ils se dégradent. Donc la maintenance basée sur la condition c'est l'approche la plus rentable.

1.2.4.2.2. La maintenance prédictive

Le terme de maintenance prédictive n'est pas normalisé. On le trouve souvent employé pour définir les caractéristiques des prévisions de maintenance en fonction de la tendance d'évolution du dysfonctionnement ou de défaillance détectée, et de l'estimation du temps de fonctionnement correcte qui subsiste. C'est une autre forme de la

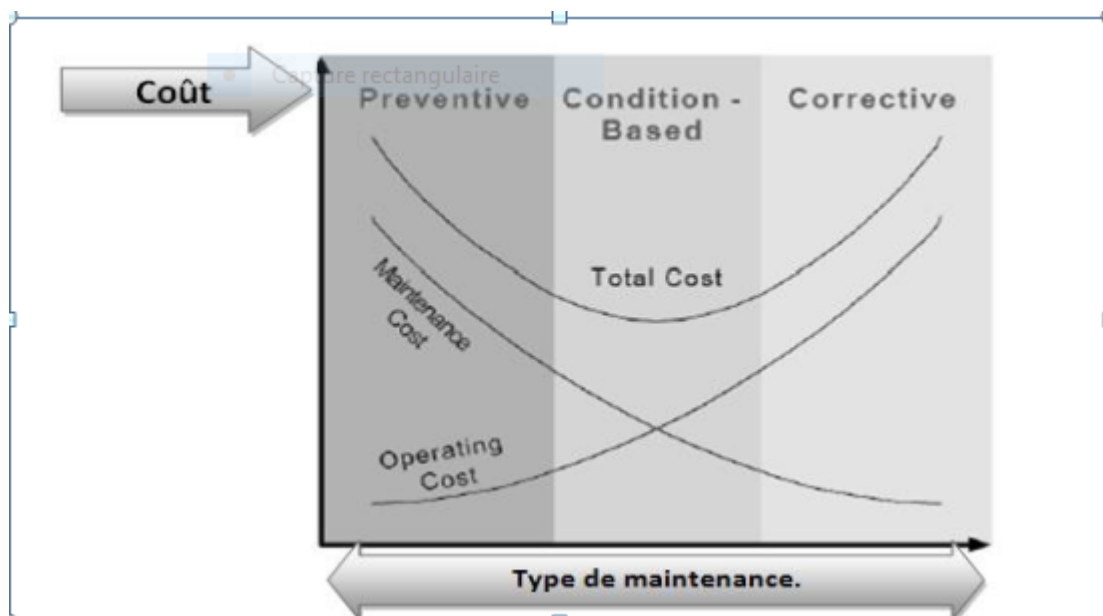


FIGURE 2. Coûts associés à différentes approches de maintenance

maintenance conditionnelle. La valeur initiale et la valeur limite du paramètre sont fixées, cette dernière correspond à un seuil significatif pour la perte de fonction de l'équipement et sa dangerosité. La périodicité de mesure est définie et elle est suffisante pour saisir toutes les évolutions du paramètre, la surveillance du paramètre représentatif pourra se faire d'une manière périodique. Les conditions de mise en place de la maintenance prédictive sur un équipement sont considérées. Cet équipement présente une dégradation remarquable et détectable, qui peut être saisie et mesurée à l'aide d'un paramètre. L'appareillage qui permet cette mesure doit être fiable et les résultats peuvent être interprétés par un personnel qualifié. La mise en œuvre de la maintenance prédictive repose généralement sur deux processus principaux qui sont le processus de supervision et le processus de pronostic.[10]

1.2.4.2.3. Les opérations de surveillance dans la maintenance préventive

Les opérations de surveillance (contrôles, visites, inspections) sont effectuées de manière continue ou à des intervalles prédéterminés ou non, calculés sur le temps ou le nombre d'unités d'usage comme ils sont nécessaires pour maîtriser l'évolution de l'état réel.

- **Les inspections :**

ce sont des activités de surveillance consistant à relever périodiquement des anomalies et exécuter des changements simples, pas d'outillage spécifique, ni d'arrêt de l'outil de production.

- **Visites :**

opérations de surveillance dans le cadre de la maintenance préventive systématique, qui s'opèrent selon une périodicité définies. Ces interventions correspondent à une liste d'opérations qui peuvent entraîner des démontages d'organes et une immobilisation du matériel. Une visite peut pousser à une action de maintenance corrective.

- **Contrôles :**

vérifications de conformité par rapport à des données préétablies suivies d'un jugement, le contrôle peut comporter une activité d'information et inclure une décision (acceptation, rejet, ajournement) aussi un déboucher comme les visites sur des opérations de maintenance corrective.

1.2.6. Prognostics and Health Management (PHM)

Le PHM (le pronostic et la gestion de santé en français) est au coeur du CBM. C'est une discipline focalisée sur la détection, la prédiction et le management de l'état de santé des systèmes complexes.[11] Cette discipline est transversale aux domaines de l'instrumentation, du traitement de données et des signaux, de la surveillance des systèmes jusqu'au développement d'un processus d'aide à la décision.[12] L'objectif de cette discipline est d'optimiser le compromis de disponibilité d'un système tout en assurant la sécurité des équipements et des personnes par une utilisation de la maintenance industrielle de manière adéquate.[13][14]

1.2.6.1. **Architecture du PHM :** L'une des architectures développées les plus connues est appelée OSA/CBM (Open System Architecture for Condition-Based-Maintenance)[15] . Cette architecture intègre la transversalité des domaines intervenant dans la discipline PHM. Elle établit un cadre de communication entre le système, le système de surveillance et les différents experts concernés au travers de la maintenance industrielle . Diverses architectures basées sur l'OSA/CBM ont été développées et appliquées à l'aérospatial, l'aéronautique, aux systèmes militaires, à l'électronique ou au nucléaire .

Une vision globale de cette architecture PHM[7] est donnée par la Figure 3 Cette architecture est constituée de 7 couches fonctionnelles :

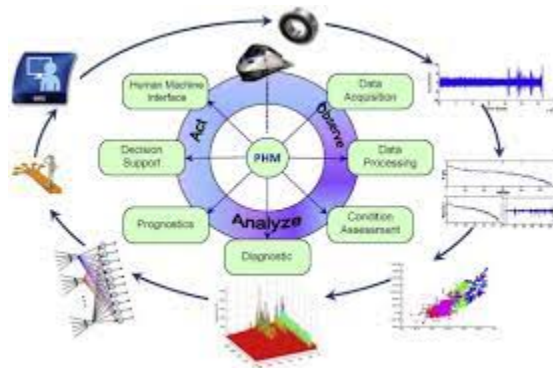


FIGURE 3. Architecture OSA/CBM.

Couche 1 : Module capteurs. Ce module fournit au système des données numériques issues de capteurs ou de transducteurs.

Couche 2 : Module traitement du signal. Ce module reçoit les données à partir des capteurs ou transducteurs ou d'autres processeurs de signaux et réalise des transformations de signaux et des extractions de caractéristiques ou de descripteurs.

Couche 3 : Module de surveillance. Le module de surveillance compare les données en ligne avec certaines valeurs attendues ou connues ; il doit également être capable de générer des alertes en fonction de seuils préalablement fixés.

Couche 4 : Module de diagnostic. Ce module détermine si l'état du système, sous-système ou composant surveillé est dégradé ou non et suggère les défaillances probables.

Couche 5 : Module de pronostic. Ce module prédit l'état futur du système, sous-système ou composant surveillé. Le module s'appuie sur les données issues des modules précédents.

Couche 6 : Module d'aide à la décision. Sa fonction principale est de recommander des actions de maintenance ou d'autres alternatives

pour continuer à faire fonctionner le système jusqu'à l'accomplissement de sa mission.

Couche 7 :Module de présentation. Ce module reçoit les informations de tous les modules précédents. Il peut être construit sous forme d'une IHM (Interface Homme-Machine).

[16] La connaissance nécessaire à priori de cette architecture dépend des mécanismes des relations de cause à effet menant à la dégradation et de leurs influences sur l'apparition des défaillances. Le deuxième élément concerne l'acquisition de données d'entrée, de données de sortie, ainsi que potentiellement des données sensibles à l'évolution de l'état de santé ou de dégradation d'un système. L'étape suivante consiste à extraire des indicateurs de qualité de l'état de santé ou de dégradation, précurseurs de l'apparition de défaillances à partir des données acquises. Ces indicateurs peuvent être traités séparément ou encore être fusionnés avec d'autres indicateurs, afin de concevoir de nouveaux indicateurs possédant certaines propriétés recherchées .

L'étape de surveillance, de diagnostic et de pronostic consiste à fusionner intelligemment ces indicateurs, de sorte à obtenir une estimation de l'état global de santé ou de dégradation d'un système. Puis, deux stratégies peuvent être envisagées suivant l'état de santé du système. Dans le cas de présence de défaillances sur le système, un processus de diagnostic est alors mis en place afin de détecter, de localiser et d'identifier le ou les composants en défaillance. Dans le cas d'absence de défaillances, un processus de pronostic est mis en place en vue d'estimer le temps de vie restant suivant les conditions opérationnelles, à partir d'un horizon de temps d'observation donné sur un système. Enfin, les deux dernières étapes concernent l'aide à la décision pour maintenir la disponibilité d'un système, tout en assurant un compromis entre la sécurité et les coûts d'exploitation. Certaines actions peuvent être entreprises comme la planification/réalisation d'actions de maintenance suivant les impératifs d'exploitation associés au système.

1.2.6.2. **Diagnostic.** Le diagnostic est une fonction de la surveillance qui suit logiquement la phase de détection. Il peut être exécuté en ligne ou a posteriori sur la base des symptômes détectés sur le système. Les symptômes traduisent les effets observables des défaillances.[17]

Le Diagnostic est défini comme le processus permettant la localisation, le traitement et la compréhension (identification) des dépendances entre événements, ainsi que leurs causes et conséquences sur la disponibilité du système. [12][18]

Le traitement correct de ces événements constitue donc un outil clé pour rendre la fonction de surveillance efficace, réactive et adaptable. Son but est la détermination du type du mode de défaillance, de la localisation et de l'instant d'occurrence d'un défaut, en se basant sur une connaissance du système sous surveillance. Cette connaissance intègre, par exemple, une référence du comportement normal ou du comportement anormal du système sous surveillance. [19][18]

Il est l'un des facteurs contribuant à la disponibilité des systèmes qui est une composante de la sûreté de fonctionnement. [18] La littérature sur les méthodes de diagnostic industriel est vaste, allant des méthodes analytiques à l'intelligence artificielle et aux méthodes statistiques, du point de vue de la modélisation, on recense des méthodes de diagnostic basées sur des modèles, basées sur des règles (base de connaissances) et basée sur les modèles qui sont mentionné sur la figure 4. Les principales méthodes de diagnostic développées par les différentes communautés de recherche scientifique [17]

1.2.6.3. **Pronostic.** De nombreux travaux menés dans le cadre des activités de recherche dans le domaine du PHM se concentrent sur la conception de modèles dits « robustes » et précis pour évaluer l'état de santé des systèmes pour des applications particulières afin de soutenir la prise de décision. La fonction de pronostic a pour mission de choisir le meilleur moment pour la maintenance tout en réduisant les exigences d'inspection, le nombre d'occurrence de pannes du système et celui des défaillances secondaires. [12]

Le pronostic est spécifique à la maintenance prédictive et essentielle dans une approche CBM, car il permet d'anticiper au mieux les interventions de maintenance suivant la prédiction de l'état futur d'un équipement et la détermination de l'impact d'une défaillance sur les performances d'un système .

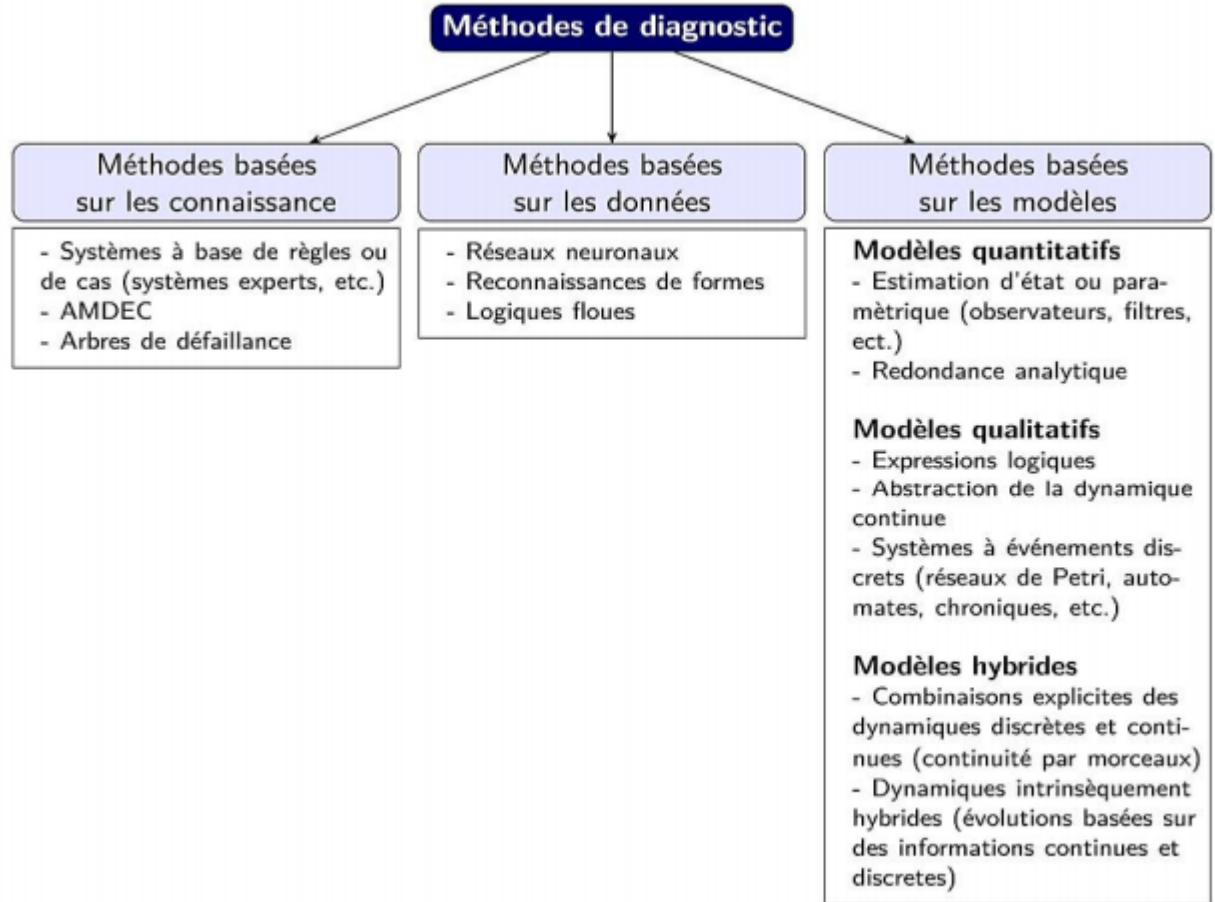


FIGURE 4. Classification des méthodes de diagnostic

Elle est donc associée à l'évolution de la défaillance ou à son comportement futur et notamment, à l'estimation de sa durée de vie résiduelle utile RUL (Remaining Useful Life) qui fait actuellement l'objet d'un nombre très élevé d'études et de recherches. 03.[18]

1.3. Définition de RUL

La durée de vie utile restante (RUL) est un élément crucial métrique utilisée dans de nombreux systèmes industriels et définie comme le temps entre l'instant courant après la détection de la dégradation et le moment où la dégradation atteint le seuil de défaillance. Sa prédiction précise permet de planifier la prochaine décision d'entretien

à l'avance qui diminue les coûts et le temps de maintenance en annulant les maintenances inutiles. Capitalisant sur le récent succès du Deep Learning (DL), dans ce titre en introduit une nouvelle approche de prédiction RUL hybride qui combine deux méthodes DL séquentiellement. Le modèle hybride utilise Réseau de neurones convolutifs (CNN) avec bidirectionnel long Réseaux de mémoire à court terme (BDLSTM) où CNN extrait caractéristiques spatiales tandis que BDLSTM extrait des caractéristiques temporelles. Notre vérification expérimentale effectuée sur le NASA Commercial Simulation de système d'aéropropulsion modulaire (C-MAPSS) ensemble de données, et les résultats ont révélé que l'approche proposée est supérieur aux autres modèles d'apprentissage automatique.[20]

1.4. Les approches de RUL

l'estimation du RUL est effectuée en utilisant différentes modèles qui peuvent être regroupés en trois grandes approches[21] :

- approche basée sur les données
- approche basée sur un modèle physique.
- approche hybride.

1.4.1. Les approches basées sur model physique

Elles estiment le RUL d'un système à partir d'une représentation du comportement physique de son processus de dégradation. Ces méthodes demandent une compréhension complète du comportement du système sur l'ensemble des niveaux, microscopique et macroscopique [22]. Les modèles comportementaux sont généralement décrits par des équations différentielles partielles ou des représentations d'état. Une fois le modèle physique disponible, le comportement du processus actuel est comparé à celui du modèle en fonctionnement normal pour détecter un début de dégradation. Après détection du début de dégradation, le modèle du système en fonctionnement dégradé est lancé pour simuler l'évolution de la dégradation dans le temps ainsi que son influence sur l'état du système. Le RUL correspond au temps que met la dégradation pour atteindre le seuil généralement prédéfini de défaillance.

Lorsque les modèles sont disponibles et précis, ces méthodes donnent une estimation exacte du RUL. En pratique, il existe des modèles

des systèmes en fonctionnement normal, par contre les modèles physiques des processus de dégradation sont difficiles à formaliser. Cette contrainte a favorisé l'émergence de méthodes hybrides qui utilisent le modèle physique du système en fonctionnement normal pour générer un indice de santé (résidu) associé à des modèles mathématiques (autorégressifs, stochastiques) dépourvus de sens physique pour modéliser la tendance de la dégradation et estimer ainsi le RUL.

1.4.1.1. Les avantages. • Efficace et descriptif.

- Haute précision.
- Facile à valider, certifier et vérifier.
- La dynamique des états peut être estimée à chaque intervalle de temps.
- L'extrapolation en dehors des données d'entraînement permet des prédictions plus robustes.

1.4.1.2. Les inconvénients. • Les modèles haute fidélité sont coûteux, longs et gourmands en calculs.

- Réutilisabilité limitée.
- Difficile de comprendre la physique des dommages pour les systèmes mécaniques complexes.
- Trop stochastique et complexe pour modéliser le défaut.
- Nécessite l'examen d'hypothèses simplificatrices.

1.4.2. Les approches basées sur les données

Elles estiment le RUL par l'analyse d'une grande masse de données observées[22]. Grâce au perfectionnement des systèmes de capteurs modernes ainsi que les technologies de traitement du signal et de stockage de données, ces méthodes sont largement utilisées aujourd'hui pour le pronostic. Elles peuvent être divisées en deux catégories :

- * les méthodes de l'Intelligence Artificielle (IA)
- * les méthodes d'analyse de tendance

1.4.2.1. Les avantages. • Implémentation et déploiement rapides • Faible coût de développement d'algorithmes et peu de connaissances requises sur la physique du système

- Les algorithmes peuvent être réglés pour être utilisés pour d'autres systèmes
- Aucune exigence sur les hypothèses et les estimations empiriques des paramètres physiques
- Capable de transformer des données bruyantes de haute dimension en informations de dimension inférieure pour les décisions pronostiques

1.4.2.2. Les inconvénients. • Exigence d'algorithme robuste

- Exigence de gestion des incertitudes
- Précision affectée par la sur généralisation et le sur apprentissage lors de l'apprentissage de l'algorithme
- Données inadéquates pour les systèmes nouvellement développés
- Résultats contre-intuitifs en raison de l'absence de connaissance physique du système
- Nécessite une grande quantité de données
- Avoir un horizon de prédiction court
- Faible précision avec des données dimensionnelles élevées

1.4.3. Les approches hybrides

Les approches présentées ci-dessus ont leurs avantages et leurs limites, et il n'est pas toujours possible de dissocier les deux. Ainsi, des chercheurs ont orienté leurs travaux vers le développement de méthodes hybrides dans le but d'exploiter les avantages des outils existants, et de compenser leurs limites, mais aussi pour réduire la complexité des calculs et faciliter l'implémentation dans le milieu industriel.[22] Dans la littérature, le mot Hybride est aussi utilisé pour signifier une combinaison de techniques différentes pour le pronostic de défaillance, sans se soucier de la nature de l'approche, par exemple : combine la régression logistique, le modèle autorégressif et moyenne mobile, qui sont des méthodes à base de données, pour le pronostic de défaillance des outils de forage ; combine le modèle de Markov caché et le modèle de prédiction gris (qui prend en charge les données incomplètes ou incertaines) pour la prédiction de défaillance. Les méthodes hybrides peuvent constituer une bonne alternative dans

un certain nombre d'applications sous réserve d'une bonne conception de la combinaison des outils.

1.4.3.1. Les avantages. • Éliminer les inconvénients des approches basées sur la physique et les approches basées sur les données et tirer parti de leurs avantages

- Ne nécessite pas de modèles très précis ou d'énormes quantités de données.
- Flexible et utile pour la gestion des incertitudes en conservant l'intuitivité du modèle
- Plus grande précision

1.4.3.2. Les inconvénients. • Porter les inconvénients des deux méthodes, mais dans une moindre mesure

- Ressource de calcul requise
- Nécessite à la fois des modèles et des données
- Imprécision causée par un modèle incorrect de données bruyantes

1.5. conclusion

Nous venons de présenter dans ce présent chapitre l'essentiel des concepts actuels attachés à la maintenance, en précisant les principaux types associés et en s'intéressant à la maintenance conditionnelle. On a parlé du PHM et ses architectures ainsi que les outils utilisés. Ensuite, on a défini le RUL et les approches effectuant l'estimation de RUL. Le deep learning est l'une des méthodes de l'IA partie des approches guidées par les données. Il est donc l'objet du chapitre suivant.

Deuxième partie

**L'Apprentissage profond
(Le deep learning)**

Chapitre 2

L'Apprentissage profond (Deep Learning)

2.1. Introduction

L'IA est basée sur une démarche d'apprentissage afin de reproduire une partie de l'intelligence humaine à travers une application, un système ou un processus. La reconnaissance faciale, la perception visuelle et autre sont des exemples de systèmes d'intelligence artificielle. Le machine Learning (ML) est un sous-domaine de l'IA qui utilise les réseaux neuronaux artificiels (ANN) pour imiter la façon dont les êtres humains prennent des décisions. Le machine Learning permet aux ordinateurs de développer des modèles d'apprentissage par eux-mêmes, sans aucune programmation, à partir de gros ensembles de données. La couche immédiatement inférieure est occupée par le Deep Learning (DL), est l'une des nombreuses approches du machine Learning qui connue un grand succès dans ces dernières années, Pour illustrer la relation entre ces termes, nous pouvons utiliser des cercles concentriques :

- Intelligence artificielle IA (**intelligence artificielle**) : le cercle plus large est l'idée qui a émergé en premier dans ce domaine
- Apprentissage automatique (**machine Learning**) : au milieu, il a prospéré plus tard après l'IA
- Apprentissage approfondie (Deep Learning) : le plus petit cercle est une expansion de l'IA actuellement la figure 1 représente la relation entre ML et IA [1 \[23\]](#)

Dans ce chapitre, nous allons présenter l'apprentissage profond, ses différents algorithmes ainsi que ses applications dans différents domaines. Une comparaison entre le Machine Learning et le Deep Learning sera présentée tout en montrant que l'apprentissage profond gagne en popularité en raison de ses performances dans différentes tâches.

ML between AI and DL

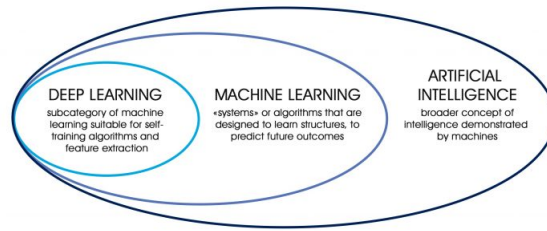


FIGURE 1. La relation Entre L'IA et ML et le Deep Learning

2.2. L'apprentissage automatique

2.2.1. Définition

L'apprentissage automatique (machine learning) est au cœur de la science des données et de l'intelligence artificielle[24]. Que l'on parle de transformation numérique des entreprises, de Big Data ou de stratégie nationale ou internationale, le machine learning est devenu incontournable.

A l'intersection des statistiques et de l'informatique, le machine learning se préoccupe de la modélisation des données. Les grands principes de ce domaine ont émergé des statistiques fréquentistes ou bayésiennes, de l'intelligence artificielle ou encore du traitement du signal.

L'apprentissage automatique (Machine learning) est un domaine de recherche en informatique. qui traite des méthodes d'identification et de mise en œuvre de systèmes et algorithmes par lesquels un ordinateur peut apprendre, ce domaine a souvent été associé à l'intelligence artificielle et plus spécifiquement l'intelligence computationnelle.[25] L'intelligence computationnelle est une méthode d'analyse de données qui pointe vers la création automatique de modèles analytiques. Autrement dit, permettant à un ordinateur d'élaborer des concepts, d'évaluer, prendre des décisions et prévoir les options futures.

Les applications du Machine learning sont nombreuses et variées, allant des moteurs de recherche et de la reconnaissance de caractères à la recherche en génomique, l'analyse des réseaux sociaux, la publicité ciblée, la vision par ordinateur, la traduction automatique ou encore le trading algorithmique. [26]

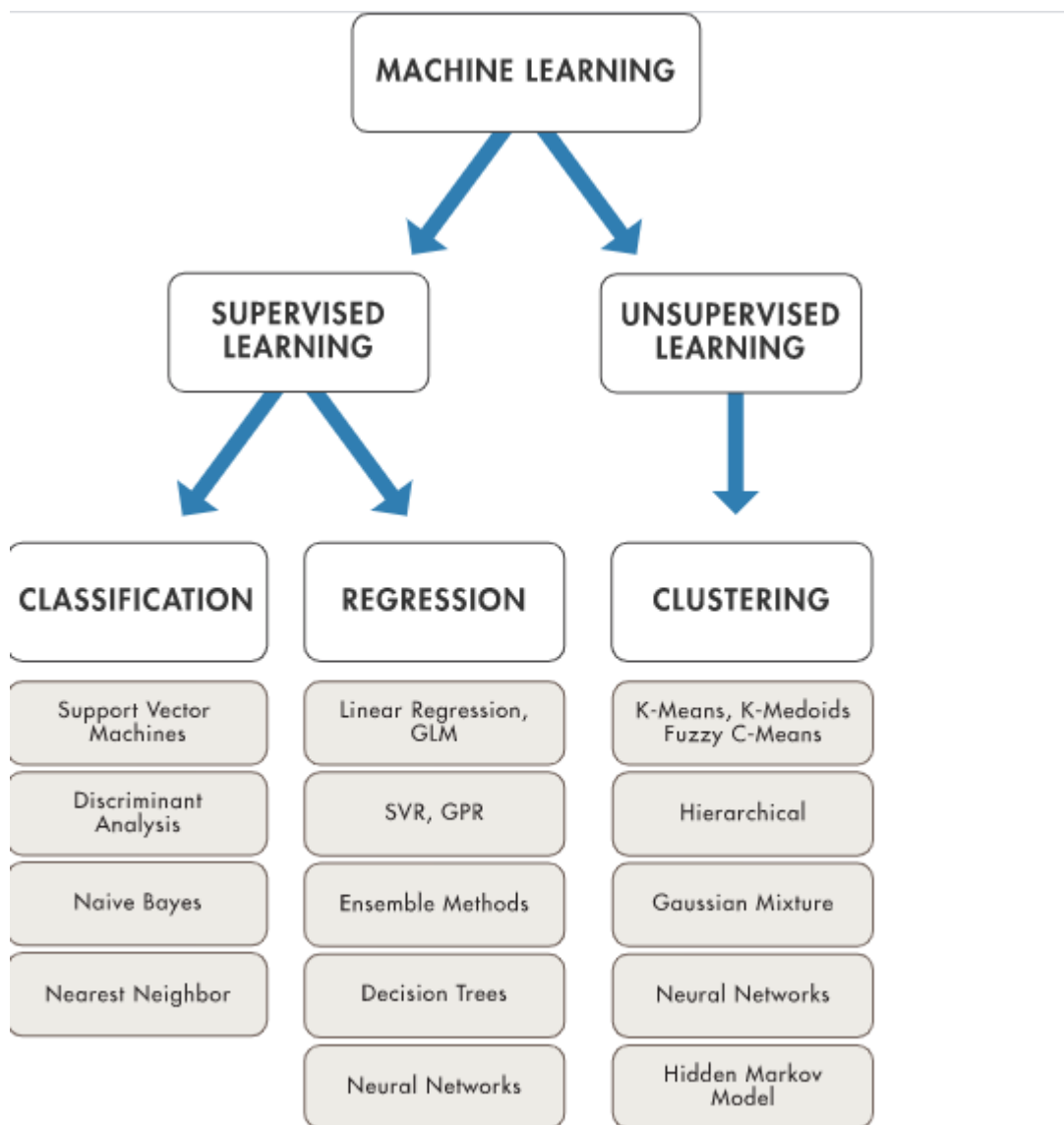


FIGURE 2. Techniques de Machine Learning

2.2.2. les types de l'apprentissage automatique

La théorie de l'apprentissage utilise des outils mathématiques dérivés de la théorie des probabilités et de la théorie de l'information, cela permet d'évaluer l'optimalité de certaines méthodes par rapport aux autres [3]. On peut citer trois types d'algorithme d'apprentissage automatique : la figure 2 se en charge les deferent type de ML 2

— Apprentissage supervisé.

- Apprentissage non supervisé.
- Apprentissage par renforcement.

2.2.2.1. L'Apprentissage supervisé. Le Machine Learning supervisé élabore un modèle qui établit des prédictions en s'appuyant sur des preuves en cas d'incertitude. Un algorithme d'apprentissage supervisé (figure 3) applique un ensemble connu de données d'entrée et de réponses connues aux données (résultats) et entraîne un modèle à produire des prévisions raisonnables pour les réponses aux nouvelles données. L'apprentissage supervisé est utilisé si on dispose de données connues pour les résultats à prédire. [26] Il développe des modèles prédictifs à l'aide des techniques de classification et de régression.

Les techniques de classification prévoient des variables discrètes [25]. Elles déterminent par exemple si un e-mail est authentique ou s'il s'agit d'un spam, ou encore si une tumeur est cancéreuse ou bénigne. Les modèles de classification classent des données d'entrée en catégories. L'imagerie médicale, la reconnaissance vocale et l'évaluation de crédit en sont des applications classiques. La classification est utilisée si les données peuvent être marquées, catégorisées ou divisées selon des groupes ou des classes spécifiques. Les applications de reconnaissance d'écriture manuscrite utilisent par exemple la classification pour reconnaître les lettres et les chiffres. Dans le traitement d'image et la vision par ordinateur, les techniques non supervisées de reconnaissance de formes servent pour la détection d'objet et la segmentation d'image. Les algorithmes couramment utilisés pour la classification incluent notamment les machines à vecteurs de support (SVM), le boosting/bagging d'arbres de décision, la méthode des k plus proches voisins, la classification naïve bayésienne, l'analyse discriminante, la régression logistique et les réseaux de neurones.

Les techniques de régression prévoient des variables continues, par exemple les variations de température ou les fluctuations de la demande en énergie. La prévision de charge d'électricité et le trading algorithmique en sont des applications classiques. Les techniques de régression sont de préférence employées si une plage de données doit être utilisée ou si les réponses à prédire sont des nombres réels, comme la température ou le délai jusqu'à la prochaine panne d'un équipement.

Les algorithmes de régression courants incluent notamment le modèle linéaire, le modèle non linéaire, la régularisation, la régression pas à pas, le boosting/bagging d'arbres de décision, les réseaux de neurones et l'apprentissage neuro-flou adaptatif.

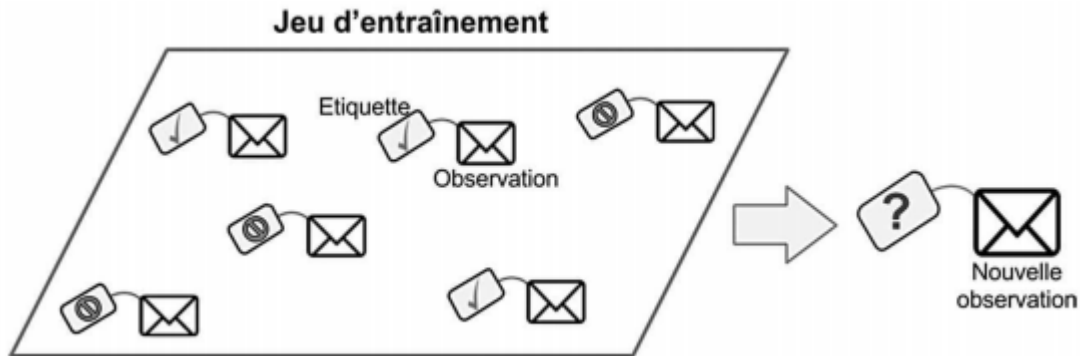


FIGURE 3. Un jeu d'entraînement étiqueté pour un apprentissage supervisé

2.2.2.2. L'Apprentissage non supervisé : L'apprentissage non supervisé identifie des modèles cachés ou des structures intrinsèques dans les données[25]. Il est utilisé pour tirer des conclusions à partir d'ensembles de données composés de données d'entrée sans réponses labellisées (figure 4) .

Le clustering est la technique d'apprentissage non supervisé la plus répandue. Il est utilisé pour effectuer une analyse exploratoire des données afin de trouver des modèles cachés ou des regroupements dans les données (figure 5). L'analyse de séquence génomique, l'étude de marché et la reconnaissance d'objets sont des exemples d'applications d'analyse de cluster.[27] Par exemple, si une société de téléphonie mobile souhaite optimiser les emplacements de construction de ses antennes-relais, elle peut estimer le nombre de clusters d'utilisateurs qui ont recours à leurs antennes à l'aide du Machine Learning. Un téléphone ne peut communiquer qu'avec une seule antenne à la fois. Par conséquent, l'équipe utilise des algorithmes de clustering pour déterminer le meilleur emplacement de la tour de relais, afin d'optimiser la qualité de réception du signal pour les groupes (ou clusters) d'utilisateurs.

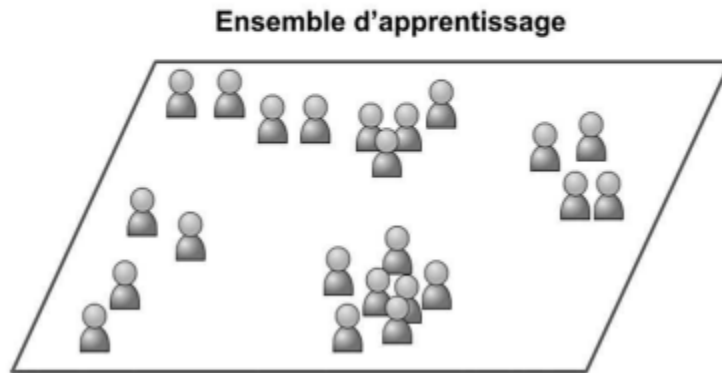


FIGURE 4. Un jeu d'entraînement non étiqueté pour un apprentissage non supervisé

Les algorithmes couramment utilisés pour le clustering incluent la méthode des k-moyennes et k-médoïdes, le clustering hiérarchique, les modèles de mélanges gaussiens, les modèles de Markov cachés, les cartes auto-organisatrices, le clustering c-moyennes flou et le clustering soustractif.



FIGURE 5. Le clustering identifie des modèles cachés dans les données.

2.2.2.3. Apprentissage par renforcement. L'apprentissage par renforcement est une approche de l'intelligence artificielle qui met l'accent sur l'apprentissage du système à travers ses interactions avec l'environnement[25]. Avec l'apprentissage par renforcement, le système adapte ses paramètres en fonction des réactions reçues de l'environnement, qui fournit ensuite un retour d'information sur les décisions prises (figure 6). Par exemple, un système qui modélise un joueur d'échecs qui utilise le résultat des étapes précédentes pour

améliorer ses performances, est un système qui apprend avec le renforcement. La recherche actuelle sur l'apprentissage avec renforcement est hautement interdisciplinaire et comprend des chercheurs spécialisés dans les algorithmes génétiques, les réseaux de neurones, la psychologie et les techniques de contrôle.

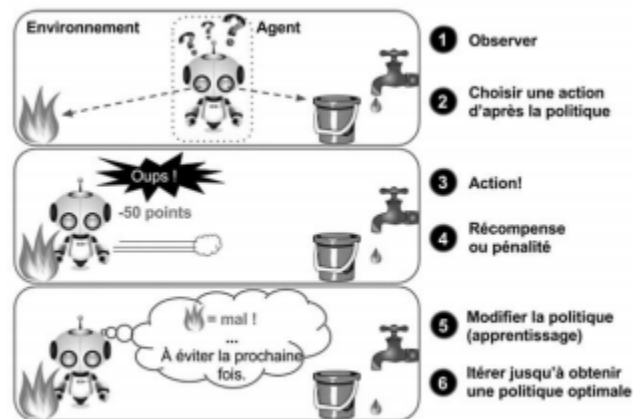


FIGURE 6. Apprentissage par renforcement

La figure 7 suivante résume les trois types d'apprentissage avec les problèmes connexes à résoudre.

Apprentissage supervisé	Apprentissage non supervisé	Apprentissage par renforcement
<ul style="list-style-type: none"> . Classification . Régression 	<ul style="list-style-type: none"> . Clustering . Segmentation . Réduction de dimension 	<ul style="list-style-type: none"> . Processus de décision . Systèmes de recommandation

FIGURE 7. Les Types d'apprentissage

2.2.3. Domaines d'application de l'apprentissage automatique

* **Automatisation :**

L'apprentissage automatique, qui fonctionne de manière entièrement autonome dans n'importe quel domaine sans aucune intervention humaine. Par exemple, des robots exécutant les étapes essentielles du processus dans les usines de fabrication.

* **Industrie de la finance :**

L'apprentissage automatique gagne en popularité dans le secteur financier. Les banques utilisent principalement le ML pour trouver des modèles dans les données mais aussi pour prévenir la fraude.

* **Organisation gouvernementale :**

Le gouvernement utilise le ML pour gérer la sécurité publique et les services publics. Prenons l'exemple de la Chine avec la reconnaissance faciale massive. Le gouvernement utilise l'intelligence artificielle pour empêcher le jaywalker.

* **L'industrie de la santé :**

La santé a été l'une des premières industries à utiliser l'apprentissage automatique avec détection d'images.

* **Transports :**

Dans l'industrie des transports, les données sont analysées pour identifier des patterns et des tendances. Ainsi, les itinéraires sont plus efficaces et les problèmes potentiels peuvent être prédits pour augmenter la rentabilité. L'analyse de données et les modèles du Machine Learning sont utilisés comme de précieux outils par les entreprises de livraison, les transports publics et les autres entreprises de transport.[28]

2.3. Le Réseaux de neurones

2.3.1. Définition

Un réseau de neurones est un modèle informatique dont la structure en couches est similaire à la structure en réseau des neurones du cerveau, avec des couches de nœuds connectés. Un réseau de neurones peut apprendre à partir de données ; il peut ainsi être entraîné à reconnaître des tendances, classer des données et prévoir de événements à venir.[2]

Un réseau de neurones décompose vos données en couches d'abstraction. Il peut être entraîné sur de nombreux exemples en vue de reconnaître des modèles au niveau de la parole ou des images, par exemple, à l'instar du cerveau humain. Son comportement est défini par la façon dont ses éléments individuels sont reliés et par la solidité (ou poids) de ces liaisons. Ces poids sont automatiquement ajustés au cours de l'entraînement selon une règle d'apprentissage spécifiée jusqu'à ce que le réseau de neurones exécute correctement la tâche souhaitée.

2.3.2. Les différents types de réseaux neuronaux

Il existe plusieurs types de réseaux de neurones, chaque type étant développé pour un objectif particulier.

2.3.2.1. Neurone Formel « mono couches ». Un neurone formel est une représentation mathématique et informatique d'un neurone biologique. Le neurone formel possède généralement plusieurs entrées et une sortie qui correspondent respectivement aux dendrites et au cône d'émergence du neurone biologique (point de départ de l'axone). Les actions excitatrices et inhibitrices des synapses sont représentées, la plupart du temps, par des coefficients numériques (les poids synaptiques) associés aux entrées. Les valeurs numériques de ces coefficients sont ajustées dans une phase d'apprentissage. Dans sa version la plus simple, un neurone formel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation, généralement non linéaire. La valeur finale obtenue est la sortie du neurone. Le neurone formel est l'unité élémentaire des réseaux de neurones artificiels dans lesquels il est associé à ses semblables pour calculer des fonctions arbitrairement complexes, utilisées pour diverses applications en intelligence artificielle, qui montré dans la figure 8

2.3.2.2. Neurones multicouche : Le perception multicouche fonctionne à la base comme un neurone formel. Les informations passent d'un point A (entrée) à un point B (sortie). Mais contrairement à un réseau monocouche, le perceptron multicouche comporte différents

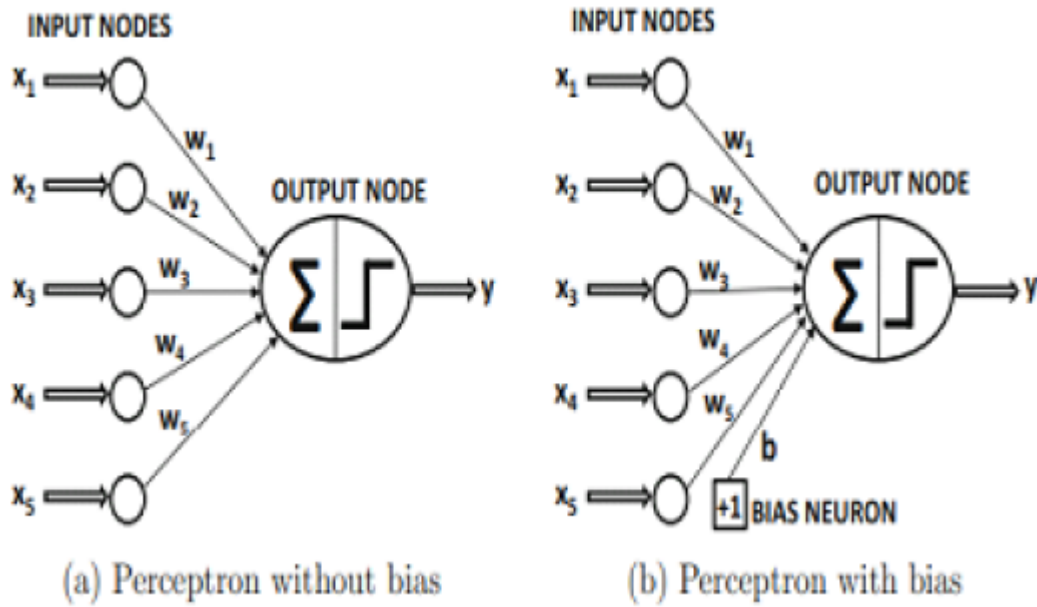


FIGURE 8. perceptron simple avec et sans le bias.

noyaux ou matrices de poids, qui permettent de traiter des informations plus complexe, qui a expliqué dans la figure 9 ci-dessous

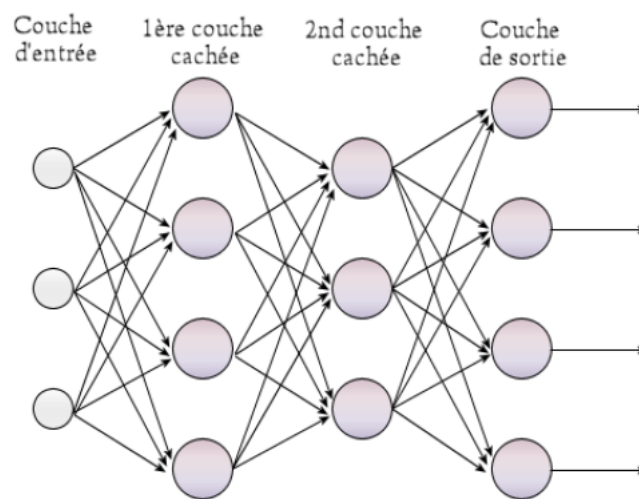


FIGURE 9. Perception multicouche.

2.3.3. Les différentes applications des réseaux de neurones

Les réseaux de neurones sont utilisés dans de plus en plus de domaines, citons la classification, la reconnaissance de formes, l'identification et la commande de processus. Le choix d'utiliser tel ou tel type de réseau de neurones dépend de l'application mais aussi des capacités de traitement du processeur sur lequel ils s'exécutent

2.3.3.1. Reconnaissance de formes. C'est un domaine privilégié d'application pour les RNAs et c'est lui qui a marqué leur début [13]. Le terme de reconnaissance est un terme général qui désigne les processus traitant des données pour en extraire des informations afin de parvenir à leur classification. Une partie non négligeable des applications neuronales actuelles appartient à cette catégorie, parmi lesquelles nous pouvons :

- Reconnaissance de caractères Latins, Chinois, Coréens, Japonais, Russes .
- Reconnaissance de visages .
- Classification d'images et de documents .
- Diagnostic des maladies .

Diagnostic des pannes dans les installations électriques, nucléaires, mécaniques, ainsi que dans les systèmes de télécommunications.

2.3.3.2. Traitement de la parole. Le traitement des signaux stochastiques (filtrage, classification) provenant des systèmes non linéaires est une tâche difficile avec les méthodes classiques de traitement. En effet, la résolution de ce type de problèmes implique la capacité de modéliser d'une façon adéquate, le système inconnu afin d'estimer sa sortie, ce qui n'est possible que pour des cas simples. La capacité d'apprentissage des réseaux de neurones leur permet d'extraire les propriétés statistiques des systèmes non linéaires, ce qui les rend aptes à traiter ce type de problèmes. Parmi les applications réalisées dans ce domaine nous pouvons : [13]

- Filtrage et suppression de bruits : amélioration du rapport signal/bruit.
- Analyse de signaux.
- Diagnostic de pannes
- Démodulation Multi-Utilisateurs.
- Classification des cibles radars dans les avions .

- Classification des bruits.

2.3.3.3. Détection d'anomalies. Ceci est une dérivation des reconnaissances de formes. On apprend à un réseau une image du fonctionnement (normal) d'un système et celui-ci sera ainsi capable d'indiquer tout état de dysfonctionnement quand certains paramètres engendrent une image (anormale). On trouve ce genre de système pour la surveillance de disjoncteurs à très haute tension (dans ce cas précis on écoute les bruits émis par le disjoncteur) ou encore dans la détection de colis pièges (examen du rayonnement gamma induit dans divers matériaux).

2.3.3.4. Traitements dépendant du temps. Dans ce cas-ci on se retrouve devant des problèmes de prédiction et d'identification et de commande de processus.

Prédiction:

Ici on va surtout utiliser des réseaux récurrents. Beaucoup de travaux sont consacrés à la prédiction de données financières et boursières et quelques résultats semblent devoir entretenir un optimisme mesuré.

Identification:

et commande de processus. : On retrouve ce genre d'applications dans l'aide au pilotage de systèmes spécifiques tel que les réacteurs chimiques, une automobile sans pilote etc. Ces applications demandent peu de ressources car les temps caractéristiques sont longs.

Nous pouvons citer également les applications bancaires et financières, le traitement de la parole et la robotique [13]. d'identification et de commande de processus.

2.4. L'apprentissage profond

2.4.1. Définition de l'apprentissage profond (deep learning)

Le deep learning ou apprentissage profond est un sous-domaine de l'intelligence artificielle (IA). Ce terme désigne l'ensemble des techniques d'apprentissage automatique (machine learning), autrement

dit une forme d'apprentissage fondée sur des approches mathématiques, utilisées pour modéliser des données. Pour mieux comprendre ces techniques, il faut remonter aux origines de l'intelligence artificielle en 1950, année pendant laquelle Alan Turing s'intéresse aux machines capables de penser. Cette réflexion va donner naissance au machine learning, une machine qui communique et se comporte en fonction des informations stockées. Le deep learning est un système avancé basé sur le cerveau humain, qui comporte un vaste réseau de neurones artificiels. Ces neurones sont interconnectés pour traiter et mémoriser des informations, comparer des problèmes ou situations quelconques avec des situations similaires passées, analyser les solutions et résoudre le problème de la meilleure façon possible. Comme chez les êtres humains, le deep learning consiste à apprendre des expériences vécues ou, dans le cas des machines, des informations enregistrées.[29]

2.4.2. LES Différents types de réseaux de neurones profond

Généralement il existe trois types importants de réseaux de neurones qui constituent la base de la plupart des modèles pré-entraînés en apprentissage en profondeur [12] :

- Perceptron Multi couches (MLP).
- Réseaux de neurones à convolution (CNN).
- Réseaux de neurones récurrents (RNN)
- Les auto-encodeurs

Discutons de chaque réseau de neurones en détail.

2.4.2.1. Perceptron Multi couches (MLP). Un seul perceptron (ou neurone) peut être imaginé comme une régression logistique. Le Perceptron Multi couches, ou MLP, est un groupe de plusieurs perceptrons/neurones à chaque couche. MLP est également connu sous le nom de réseau de neurones **Feed-Forward** car les entrées ne sont traitées que dans le sens direct figure 10 :

Comme le montre la figure 10, MLP se compose de 3 couches – Input, Hidden et Output. La couche d'entrée accepte les entrées, la couche cachée traite les entrées et la couche de sortie produit le résultat. Essentiellement, chaque couche essaie d'apprendre certains poids.

MLP peut être utilisé pour résoudre des problèmes liés à :

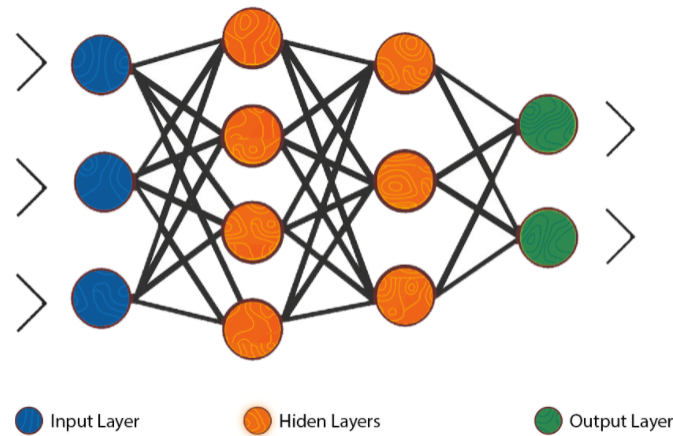


FIGURE 10. MLP

- Données tabulaires
- Données d'image
- Données texte

MLP ne peut pas capturer les informations séquentielles dans les données d'entrée qui sont nécessaires pour traiter les données de séquence.

Pour surmonter les limitations du MLP il y a deux autres architectures différentes qui sont :

Les réseaux de neurones récurrents (RNN) et les réseaux de neurones à convolution (CNN).

2.4.2.2. Réseau de neurones récurrents (RNN). Essayons d'abord de comprendre la différence entre un RNN et un ANN du point de vue de l'architecture [12] :

Une contrainte de bouclage sur la couche cachée de ANN se transforme en RNN.¹¹

RNN a une connexion récurrente sur l'état caché. Cette contrainte de bouclage garantit que les informations séquentielles sont capturées dans les données d'entrée. Au lieu de prendre en compte les données d'entrée de manière séparée (comme un CNN analyse image par image) le réseau récurrent lui prend en compte les données d'entrée passées.

Certaines architectures, dites bidirectionnelles, peuvent aussi prendre

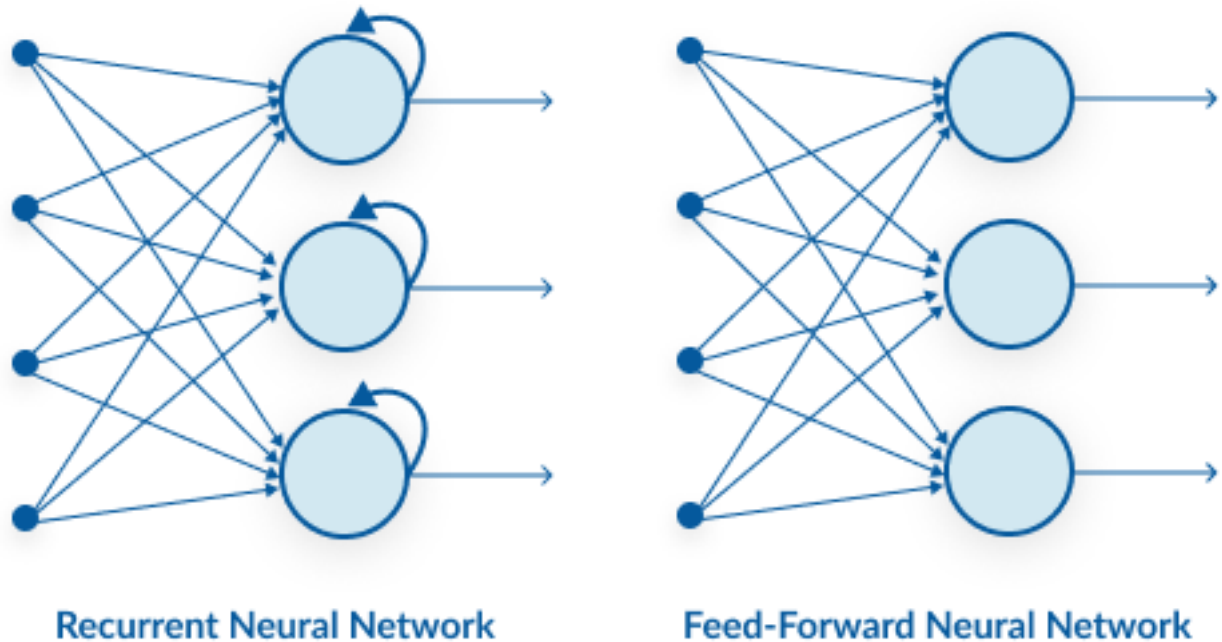


FIGURE 11. transforme une couche cachée de ANN en RNN

en compte les données futures. Par exemple lors d'une analyse de texte où on cherche à trouver des entités nommées (noms de personnes, sociétés, pays, etc.) nécessite de voir les mots de toute la phrase.

On peut utiliser des réseaux de neurones récurrents pour résoudre les problèmes liés à :

- Données de séries chronologiques
- Données texte
- Données audio

Les réseaux de neurones récurrents sont au coeur de bon nombre d'améliorations substantiels dans des domaines aussi divers que la reconnaissance vocale, la composition automatique de musique, l'analyse de sentiments, l'analyse de séquence ADN, la traduction automatique. La différence principal avec les autres réseaux de neurones vient du fait que ces derniers tiennent compte de l'enchaînement successif des données, bien souvent de leur enchaînement dans le temps. Par exemple dans le cas de l'analyse d'une série de mesures de capteurs (séries temporelles) le réseau aura encore en mémoire tout ou partie des observations précédentes.

Le schéma de ce réseau est produit ici¹² :

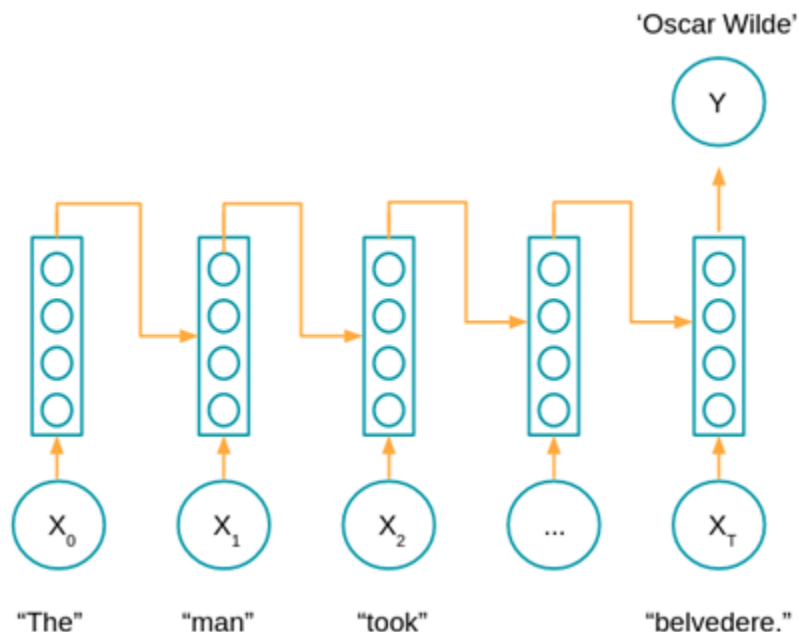


FIGURE 12. Le schéma de réseau RNN

2.4.2.2.1 Les types des réseaux RNN

Il existe plusieurs types de RNN qui ont une couche cachée spécialement conçue dont nous parlerons dans les sous titres suivants :

A. Long Short-Term Memory (LSTM)

Les réseaux de mémoire à long terme à court terme généralement appelés simplement LSTM, sont un type spécial de RNN.

Ils ont été introduits par Hochreiter Schmidhuber (1997). Les Réseaux neuronaux récurrents présentés sont capables d'apprendre des règles de mise à jour de séquence arbitraire en théorie. Dans la pratique, cependant, ces modèles oublient généralement rapidement le passé. C'est ce qu'on appelle le problème de la disparition de gradient et c'est pourquoi ils ont inventé le LSTM. La cellule LSTM est une adaptation de la couche récurrente qui permet aux signaux plus anciens des couches profondes de se déplacer vers la cellule du présent ¹³ La

figure suivante représente une chaîne de trois cellules LSTM [30] :

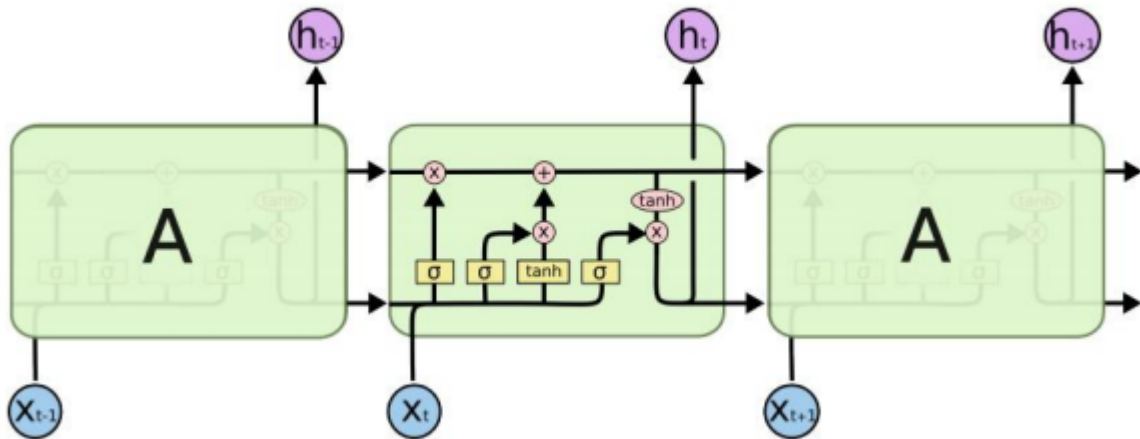


FIGURE 13. Une chaîne de cellules LSTM

B. BILSTM

Les réseaux de type BILSTM consistent à exécuter deux LSTM en parallèle : le premier réseau lit la séquence d'entrée de droite à gauche et le second réseau en sens inverse de gauche à droite.

Chaque LSTM engendre une représentation cachée : h (un vecteur allant de gauche à droite) et \bar{h} (un vecteur allant de droite à gauche) qui sont ensuite combinés afin de calculer la séquence de sortie.

Dans notre problème, saisir le contexte des mots de chaque direction permet de mieux comprendre la sémantique d'un tweet[21] .

C. Gated Recurrent Units (GRU)

Les cellules GRU (Gated Recurrent Unit) permettent au réseau récurrent de sauvegarder plus d'informations historiques pour une meilleure prédiction. Il introduit un update gate qui détermine la quantité d'informations à garder du passé ainsi qu'un reset gate qui

définit la quantité d'informations à oublier[31]. Le graphique 14 ci-dessous schématise la cellule

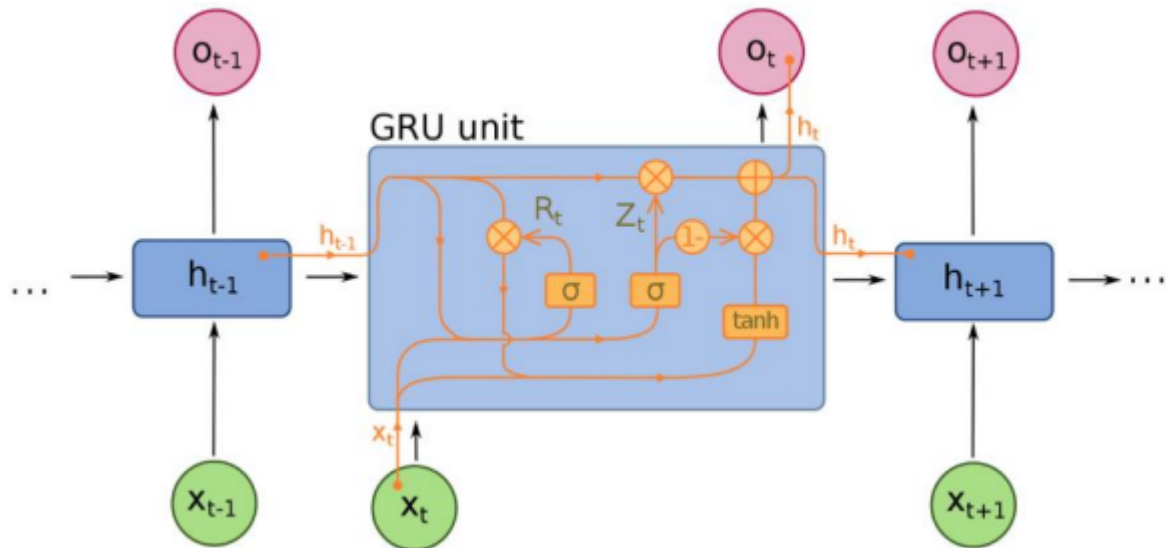


FIGURE 14. Schéma d'un réseau GRU à une unité

D.BIGRU

Les GRU bidirectionnels sont un type de réseaux de neurones récurrents bidirectionnels avec uniquement les portes d'entrée et d'oubli. Il permet d'utiliser les informations des pas de temps précédents et des pas de temps ultérieurs pour faire des prédictions sur l'état actuel. La figure 15 montre la structure typique d'un réseau de neurones récurrent bidirectionnel GRU.[32]

2.4.2.2 Les avantages du réseau neuronal récurrent (RNN)

- RNN capture les informations séquentielles présentes dans les données d'entrée, c'est-à-dire la dépendance entre les mots du texte tout en faisant des prédictions :

Comme vous pouvez le voir ici, la sortie (o_1, o_2, o_3, o_4) à chaque pas de temps dépend non seulement du mot courant mais aussi des mots précédents.

- Les RNN partagent les paramètres sur différents pas de temps. Ceci est communément connu sous le nom de partage de paramètres . Cela

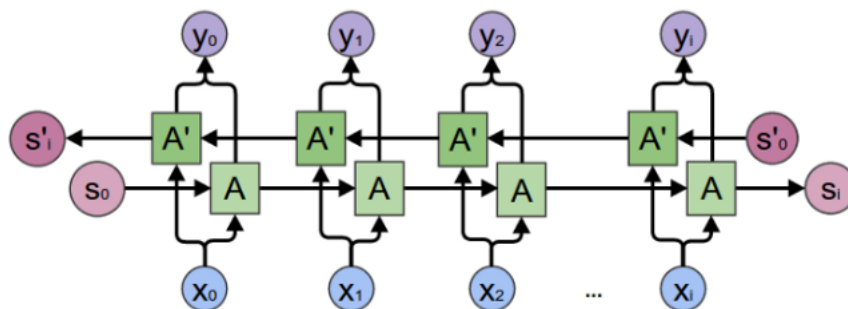


FIGURE 15. Structure générale des réseaux de neurones récurrents bi-directionnels GRU.

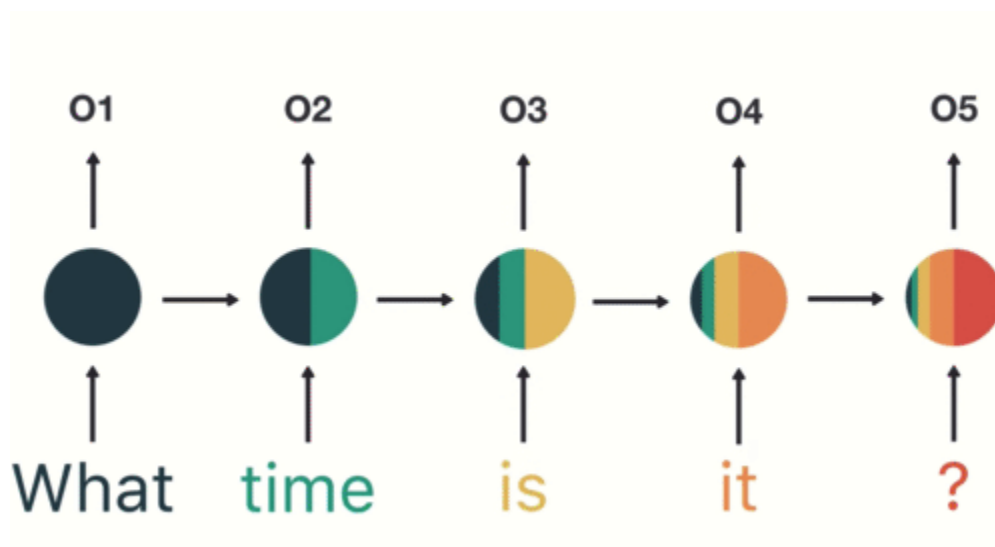


FIGURE 16. Modèle Many2Many Seq2Seq

se traduit par moins de paramètres à former et diminue le coût de calcul.

Comme le montre la figure ci-dessus, 3 matrices de poids - U , W , V , sont les matrices de poids qui sont partagées sur tous les pas de temps.

2.4.2.3. Réseau de neurones à convolution (CNN). Les réseaux de neurones convolutifs (CNN) font fureur dans la communauté de l'apprentissage en profondeur en ce moment. Ces modèles sont utilisés dans différentes applications et domaines, et ils sont particulièrement répandus dans les projets de traitement d'images et de vidéos dont fait partie la reconnaissance faciale ou encore la classification d'image.

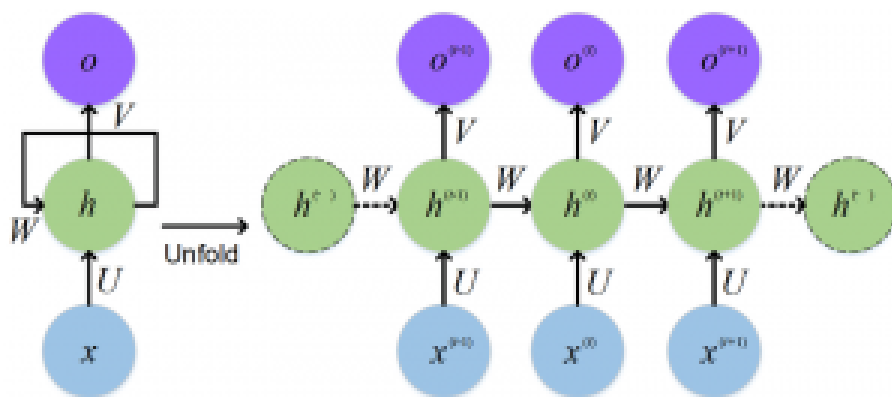


FIGURE 17. RNN déroulé

L'entreprise Bai Du (l'équivalent de Google en Chine) a par exemple mis en place des portiques actionnés par la reconnaissance visuelle qui laissent passer uniquement leurs employés. Snapchat et de nombreuses applications mobiles ont utilisé la percée de l'apprentissage et des CNN pour augmenter leurs fonctionnalités de "filtres" figure 18.

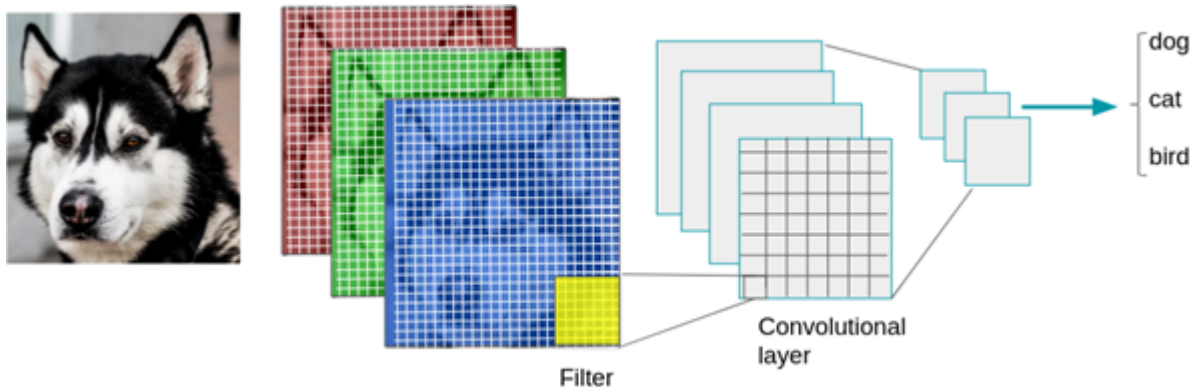


FIGURE 18. un réseau de neurone cnn

Le nom réseau convolutif renvoie à un terme mathématique : le produit de convolution. Les éléments constitutifs des CNN sont des filtres, également appelés noyaux. Les noyaux sont utilisés pour extraire les caractéristiques pertinentes de l'entrée à l'aide de l'opération de convolution. En termes simples, l'idée est qu'on applique un filtre à l'image d'entrée, les paramètres du filtre seront appris au fur et à mesure de l'apprentissage. Un filtre appris permettra par exemple de

détecter les angles dans une image si les angles servent à classifier au mieux l'image. L'image est d'abord décomposée dans les 3 canaux (R,G,B) pixels par pixels, on obtient donc 3 matrices de taille n x n (où n est le nombre de pixels).

Voici un exemple de convolution avec une matrice de taille 6 x 6 (figure 19) :

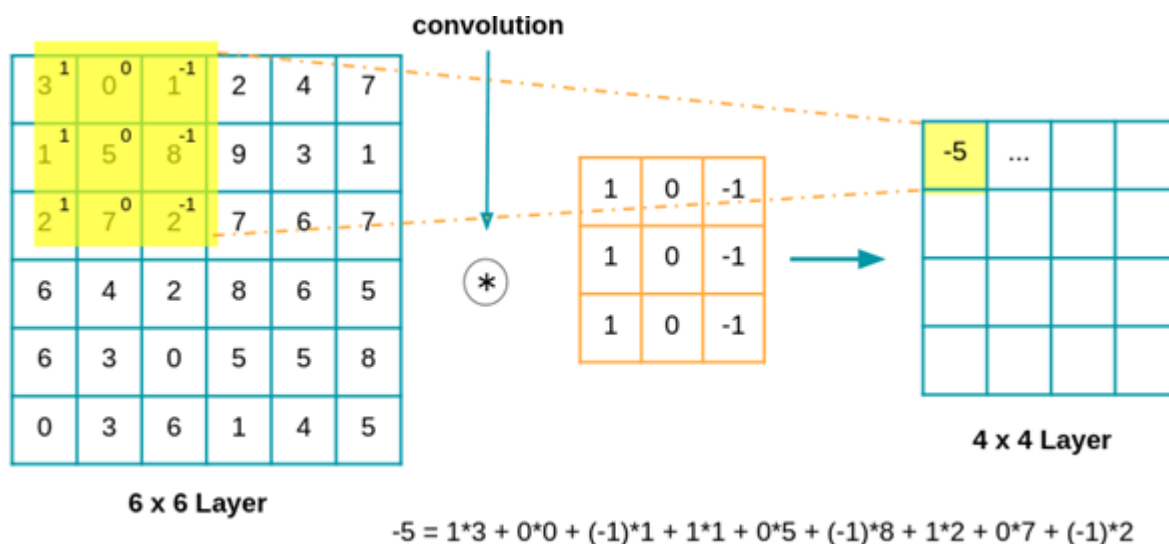


FIGURE 19. un exemple de convolution avec une matrice de taille 6 x 6

Essayons de saisir l'importance des filtres utilisant des images comme données d'entrée. La convolution d'une image avec des filtres donne une carte des caractéristiques figure 20 :



FIGURE 20. Sortie de convolution

Bien que les réseaux de neurones convolutifs aient été introduits pour résoudre les problèmes liés aux données d'image, ils fonctionnent également de manière impressionnante sur les entrées séquentielles.
Les Avantages du réseau de neurones à convolution (CNN)

- * Le réseau peut apprendre par étape à reconnaître les éléments caractéristiques d'une image. Pour reconnaître un visage par exemple : il apprendra à reconnaître d'abord des paupières, des pupilles, pour arriver à identifier des yeux.
- * Une fois un élément appris à un endroit de l'image le réseau sera capable de le reconnaître n'importe où d'autre dans l'image.
- * CNN apprend les filtres automatiquement sans le mentionner explicitement. Ces filtres aident à extraire les caractéristiques justes et pertinentes des données d'entrée figure 21 [12] .

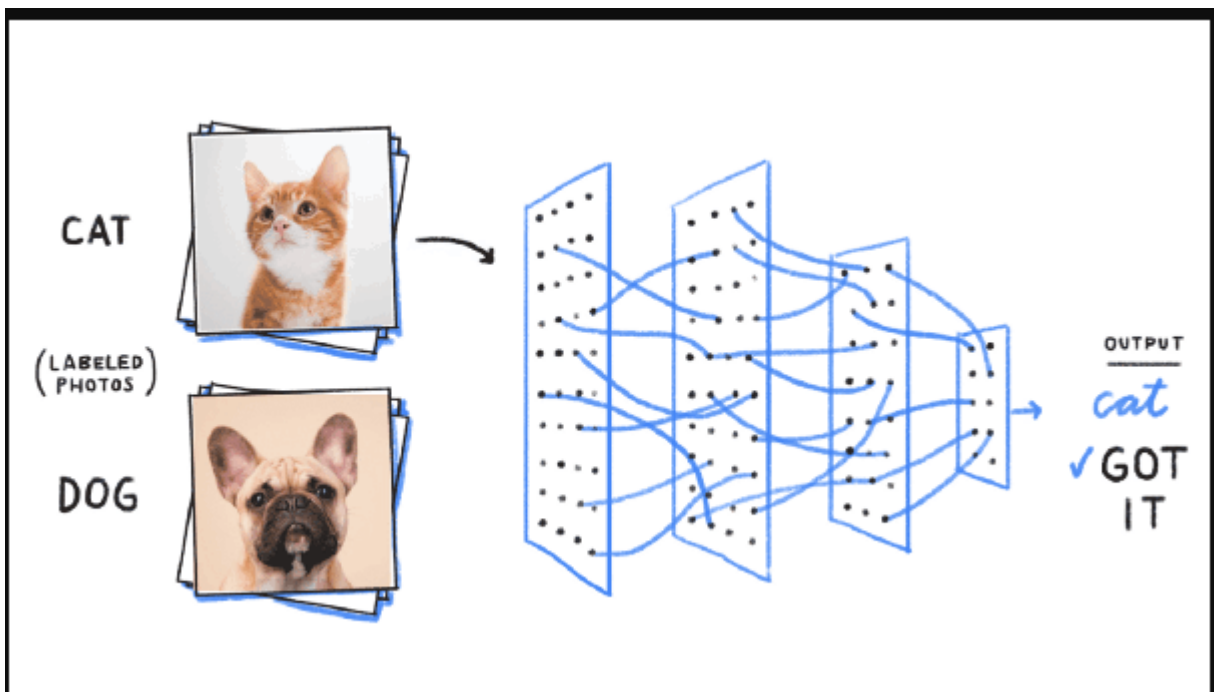


FIGURE 21. CNN – Classification des images

* CNN capture les caractéristiques spatiales d'une image. Les caractéristiques spatiales font référence à la disposition des pixels et à la relation entre eux dans une image. Ils nous aident à identifier avec précision l'objet, l'emplacement d'un objet, ainsi que sa relation avec d'autres objets dans une image.

Dans l'image de la figure 22 nous pouvons facilement identifier qu'il s'agit d'un visage humain en examinant des caractéristiques spécifiques telles que les yeux, le nez, la bouche, etc. On peut aussi voir

comment ces spécificités sont agencées dans une image. C'est exactement ce que les CNN sont capables de capturer.

* CNN suit également le concept de partage de paramètres. Un seul filtre est appliqué sur différentes parties d'une entrée pour produire une carte des caractéristiques :

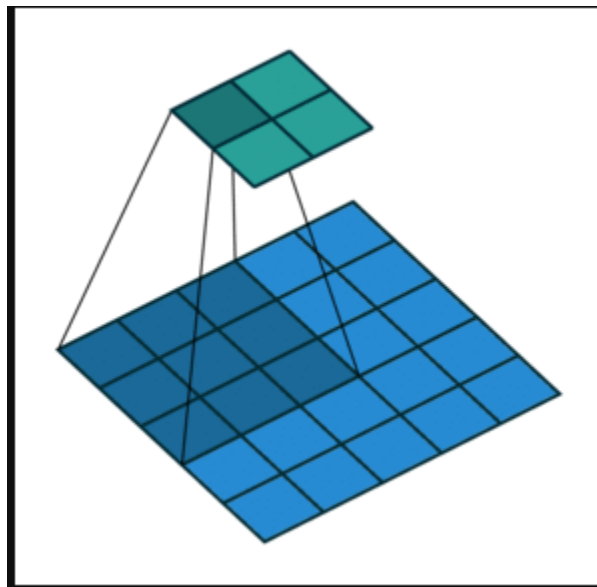


FIGURE 22. Image convolante avec un filtre 1

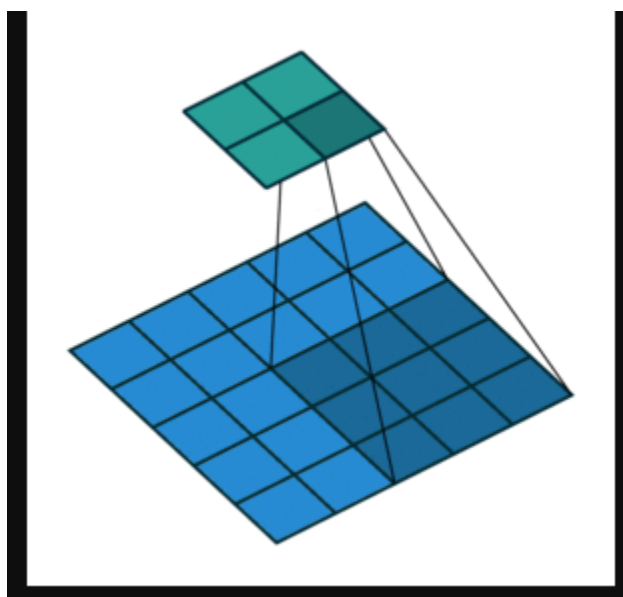


FIGURE 23. Image convolante avec un filtre 2

Notez que la carte de caractéristiques 2×2 est produite en faisant glisser le même filtre 3×3 sur différentes parties d'une image.

2.4.2.4. Les Auto-Encodeurs. Les auto encodeurs sont appliqués principalement à la détection d'anomalie (par exemple pour détecter la fraude en banque ou bien pour trouver des anomalies dans une ligne de production industrielle). Ils peuvent également servir à la réduction de dimension (proche dans l'idée d'une Analyse en Composante Principale). En effet le but des auto encodeurs est d'apprendre à la machine en quoi consiste des observations "normales".[24](#).[\[33\]](#)

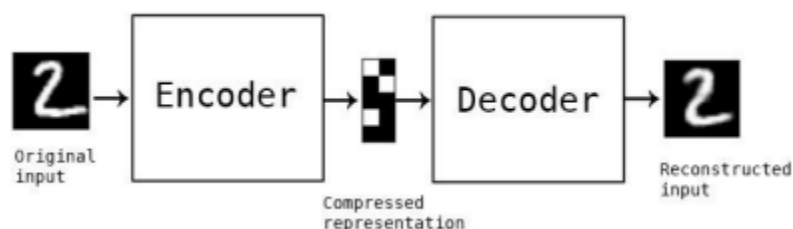


FIGURE 24. Image montrant le fonctionnement d'un auto-encodeur.

La forme la plus simple d'un auto-encodeur est un réseau de neurones non récurrents qui se propage vers l'avant, très semblable au perceptron multicouches - ayant une couche d'entrée, une couche de sortie ainsi qu'une ou plusieurs couches cachées les reliant -, mais avec toutefois une couche de sortie possédant le même nombre de nœuds que la couche d'entrée, son objectif étant de reconstruire ses entrées (prédire une valeur cible X' la plus proche des entrées X plutôt que de prédire une valeur cible Y étant donné les entrées X). Par conséquent, un auto-encodeur est un modèle d'apprentissage non supervisé. Un auto-encodeur se compose toujours de deux parties, l'encodeur et le décodeur. Le réseau va donc représenter les données au moyen d'une ou plusieurs couches cachées de sorte à ce qu'en sortie on retrouve les mêmes données qu'en entrée. L'objectif de retrouver en sortie les mêmes données qu'en entrée est caractéristique des auto-encodeurs (analogue à la fonction identité $f(x)=x$).

La phase d'encodage et de décodage n'est elle pas propre aux auto-encodeurs. En effet, on les retrouve dans la traduction automatique dans des réseaux de neurones récurrents. Après avoir entraîné le réseau avec suffisamment de données il sera possible d'identifier des

observations suspectes ou anormales lorsque celles-ci dépassent un certain seuil par rapport à la nouvelle "norme".

2.4.3. Fonctionnement du deep Learning

Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain.

Ce réseau est composé de dizaines voire de centaines de «couches» de neurones, chacune recevant et interprétant les informations de la couche précédente.

Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit (figure 25).[34]

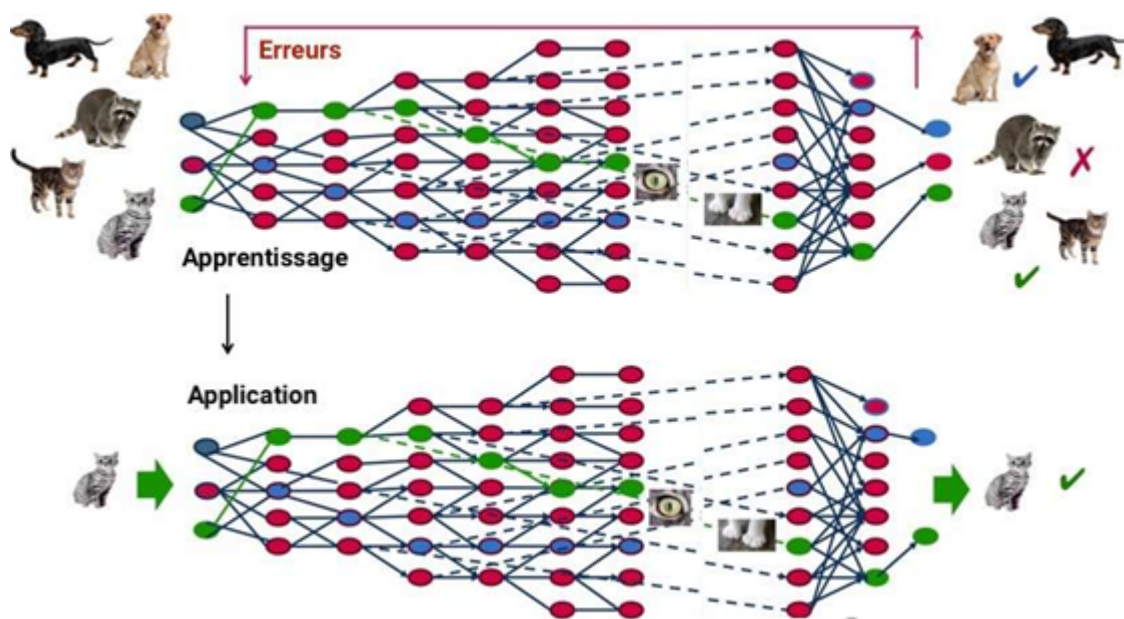


FIGURE 25. un processus d'autoapprentissage

À travers un processus d'autoapprentissage, le deep Learning est capable d'identifier un chat sur une photo. À chaque couche du réseau neuronal correspond un aspect particulier de l'image.

À chaque étape, les «mauvaises» réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique.

Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'ai jamais appris le concept de chat.

Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant.

2.4.4. Les Applications du deep Learning

Le deep Learning est utilisé dans de nombreux domaines :

- reconnaissance d'image
- traduction automatique
- voiture autonome
- diagnostic médical
- recommandations personnalisées
- modération automatique des réseaux sociaux
- prédiction financière et trading automatisé
- identification de pièces défectueuses
- détection de malwares ou de fraudes
- chatbots (agents conversationnels)
- exploration spatiale
- robots intelligents.[35]

2.4.5. Les avantages de Deep Learning

Permis les avantages qui distinguent le Deep Learning, on cite :

- La robustesse pour comprendre et utiliser de nouvelles données.
 - Il gère tout à un niveau d'abstraction beaucoup plus élevé que les réseaux de neurones standard.
 - Il obtient ses résultats plus rapidement. Il apprend au fil du temps plutôt qu'en un éclair.
 - Il peut gérer de grandes quantités de données pour de petits réseaux avec un coût d'apprentissage bien moindre.[36]

2.5. L'apprentissage automatique VS L'apprentissage profond :où le Deep Learning Fait la difference ?

tableau 01 [1](#) fait comparaisons entre DL et ML[13]

Conclusion

Nous avons présenté au cours de ce chapitre l'apprentissage automatique avec ses différents types, supervisé, non-supervisé et avec renforcement, ses domaines d'application et ses différents domaines d'applications ,ensuite nous avons présenté l'apprentissage profond, ses applications, les réseaux de neurones où on a parlé sur les différents types ,leur architecture,et on a cité quelques domaines des réseaux de neurones.À la fin de ce chapitre, on a fait une comparaison entre l'apprentissage automatique et l'apprentissage profond.Dans le troisième chapitre on s'intéressera à l'état de l'art du RUL.

MACHINE LEARNING	DEEP LEARNING
L'apprentissage automatique utilise des algorithmes pour analyser les données, tirer des leçons de ces données et prendre des décisions éclairées en fonction de ce qu'il a appris	L'apprentissage profond structure les algorithmes en couches pour créer un «réseau neuronal artificiel», capable d'apprendre et de prendre des décisions intelligentes par lui-même
Peut s'entraîner avec moins de données d'entraînement	Nécessite de grands ensembles de données pour l'entraînement
La sortie est sous forme numérique pour les applications de classification et de notation	La sortie peut être sous n'importe quelle forme, y compris des éléments de forme libre tels que du texte et du son libre
Réglage limité pour le réglage des hyperparamètres	Peut être réglé de différentes manières
S'entraîne sur CPU	S'entraîne sur GPU pour une optimisation accrue

TABLEAU 1. L'apprentissage automatique VS L'apprentissage profond : où le Deep Learning fait la différence ?

Troisième partie

La durée de vie utile restante (RUL) et le deep learning

Chapitre 3

La durée de vie utile restante (RUL) avec le deep learning

3.1. Introduction

La durée de vie utile résiduelle (RUL) est la prédiction du temps restant pendant lequel un système est capable d'exécuter sa fonction prévue; elle est mesurée à partir de l'instant présent jusqu'à la défaillance finale. En général, le but de la prévision de la RUL est d'influencer la prise de décision dans le système.

Parmi les approches puissantes pour l'estimation de RUL, on trouve les deep learning partie de l'IA. Afin de proposer un modèle performant basé sur cette technique, nous allons présenter dans ce chapitre quelques travaux existants dans la littérature concernant l'estimation de RUL avec les différentes algorithmes de deep learning.

3.2. Les travaux existants de RUL avec le deep learning

L'apprentissage profond est l'une des sous-branches de la machine learning, caractérisé par plusieurs couches de traitements non linéaires, et provient du réseau de neurones artificiels (ANN).[\[37\]](#) Comme le développement rapide de l'infrastructure informatique, DL est devenu l'un des principaux sujets de recherche dans le domaine de pronostics, étant donné sa capacité à saisir les relations hiérarchiques encastrées dans des structures profondes [\[38\]](#). La caractéristique de DL est son architecture réseau profonde où plusieurs couches sont empilées dans le réseau pour pleinement capturer les informations représentatives à partir des données d'entrée brutes [\[39\]](#). Les modèles DL ont attiré une grande attention et remarquable réalisation dans de nombreux domaines, tels que la reconnaissance d'images, la reconnaissance de la parole et traitement du langage naturel [\[40\]](#). Cependant, il n'a pas été pleinement exploité dans le domaine de la RUL

prédiction .

La littérature publiée sur le DL s'est principalement concentrée sur quatre architectures profondes suivantes [41] :

- Les réseaux Auto-encodeur.
- Les réseaux de croyances profondes (DBN).
- Les réseaux de neurones convolutifs (CNN).
- Les réseaux de neurones récurrents (RNN).

3.2.1. RUL avec Auto-encodeur

L'auto-encodeur (AE) peut apprendre une nouvelle représentation des données en reconstruisant les données d'entrée, qui contiennent deux phases [42] :

- encodeur.
- décodeur.

Ainsi, il est souvent utilisé pour préformation du réseau. L'auto-encodeur clairsemé empilé (SAE) est l'un des neurones profonds les plus couramment utilisés, approches de réseau (DNN) traitant des données parce qu'elles contiennent l'AE multicouches comme un encodeur automatique clairsemé et débruitage AE dans la surveillance de la santé des machines, les modèles AE sont principalement utilisés pour le diagnostic des pannes. Pour la prédiction RUL, AE est normalement utilisé pour extraire les caractéristiques de dégradation, des applications directes de l'AE très limité sur la prédiction RUL peuvent être trouvées dans la littérature.

Dans [43], les auteurs ont utilisé une architecture d'encodeur-décodeur LSTM uniquement dans le premier ensemble de données de simulation de système d'aéropropulsion modulaire commercial (C-MAPSS) obtenant un indice de santé non supervisé pour déterminer RUL.

Un décodeur Encodeur variationnel basé sur l'analyse visuelle des données pour prédire quand une machine en service tombera en panne est proposé dans [44]. Un encodeur variationnel convolutif profond est utilisé afin d'extraire automatiquement la fonction de dégradation des performances de plusieurs capteurs. De plus, un modèle encodeur-décodeur a été proposé dans [45]. Le BiLSTM et le CNN

sont utilisés dans l'encodeur, et les réseaux entièrement connectés sont utilisés comme décodeur des caractéristiques d'informations.

Ramin et al. [46] a développé une méthode de pronostic pour les roulements de machine appelée algorithme de pronostic basé sur la corrélation automatique (AEC). Ils ont collecté des expériences de test à échec sensoriel données et extraire des fonctionnalités non supervisées en formant un AE. Ensuite, un filtre de moyenne mobile a été utilisé pour passer à travers le résultat. L'état de santé du système peut être illustré après que l'algorithme AEC normalise la sortie de le filtre.

Jain et al. [47] ont prédit que le RUL d'un moteur d'avion utilisant un SAE empilé combiné avec la régression logistique (LR). Le SAE empilé a été utilisé pour extraire les fonctionnalités de dégradation des performances et la fusion de plusieurs fonctionnalités grâce à l'auto-apprentissage multicouche. Bien que les auteurs a affirmé que le RUL a été prédit sur la base d'un SAE empilé, il a en fait été réalisé par un modèle LR.

Lei et al.[42] ont développé un cadre de prédiction basé sur l'apprentissage en profondeur pour prédire le RUL des roulements combinant AE profond et DNN. L'AE profond a été utilisé pour sélectionner la dégradation des roulements fonctionnalités et réduire le nombre de réseau de prédiction paramètres.

3.2.2. RUL avec les réseau de croyances profondes (DBN)

(DBN) est une pile de Machines Boltzmann (RBM) qui inclut Boltzmann Machines (BM) avec une seule couche de détection de caractéristiques unités et BM d'ordre supérieur . Les gourmands couche par couche L'algorithme d'apprentissage des RBM peut pré-entraîner le modèle dans un manière non surveillée sans exigence stricte sur le montant de données d'entraînement.

Zhang et al. [48] [49] a présenté un ensemble de réseaux de croyances profondes multi-objectifs (MODBNE) pour l'estimation RUL, qui emploie un puissant algorithme évolutif (EA) multi-objectifs sur la base de la décomposition intégrée au traditionnel Technique de formation DBN. L'approche proposée peut évoluer plusieurs DBN simultanément, sous réserve de deux conflits objectifs : précision et diversité.

Ma et al. [50] ont appliqué un réseau de croyances profondes discriminant avec la colonie de fourmis (ACO-DDBN) pour prédire l'état de santé du machine. DDBD utilise une architecture profonde pour combiner les avantages du DBN et capacité discriminante de la stratégie de rétro propagation. L'ACO peut découvrir la meilleure combinaison de paramètres lors de la sélection des paramètres pour DDBN. La structure du modèle DDBN est déterminée automatiquement sans connaissance préalable grâce à l'optimisation.

Deutsch et He [51] ont présenté un réseau de neurones DBN feedforward (DBN-FNN) pour prédire RUL de composants de rotation utilisant un capteur de vibration. Ils ont développé l'approche DBNFNN combinant les avantages de la fonctionnalité autodidacte d'apprentissage de la DBN et de la capacité de prédiction du FNN, qui peut soit prendre caractéristiques de vibration ou extraire des caractéristiques des données de vibration pour prédire le RUL. Bien que la prédiction RUL les performances du DBF-FNN n'étaient pas aussi bonnes que celles du filtre à particules, la méthode proposée a accompli l'objectif d'extraction automatique des caractéristiques et la prédiction de RUL sans l'intervention de l'homme à l'ère de Big Data.

Deutsch et al. [52] ont ensuite développé une nouvelle méthode intégrée qui combine un DBN avec une particule filtre pour la prédiction RUL des roulements en céramique hybride utilisant signaux vibratoires. Données vibratoires réelles de la céramique hybride des tests de fonctionnement jusqu'à la défaillance des roulements ont été collectés et utilisés pour valider l'approche pronostique proposée. La proposition l'approche intégrée présente une meilleure prédiction RUL performances que l'approche basée sur le filtre à particules et la Méthode RBN.

3.2.3. RUL avec les réseaux de neurones convolutifs(CNN)

Récemment, CNN a été introduit avec succès dans la prédiction RUL pour sa forte capacité d'extraction de caractéristiques locales [53]. Par exemple, le travail présenté dans [37], un réseau de neurones à convolution profonde (DCNN) est proposé pour le pronostic. Il a utilisé l'approche de la fenêtre temporelle pour la préparation

des échantillons afin d'obtenir une meilleure extraction des caractéristiques par le DCNN. Les données brutes collectées avec la normalisation sont directement utilisées comme entrées de réseau proposées sans aucune expertise préalable sur le pronostic et le traitement du signal, ce qui facilite l'application de la méthode proposée.

Navathe et al. [53] ont développé le premier modèle CNN qui est spécifiquement appliqué pour résoudre un problème d'estimation RUL. C'était la première tentative de tirer parti de l'apprentissage en profondeur approche dans la prédiction de RUL. Dans leur travail, CNN était principalement utilisé pour conduire différentes unités de traitement alternativement comme la convolution, la mise en commun, écrasement tangent sigmoïde/hyperbolique, redresseur et normalisation. Les auteurs ont affirmé que la profondeur proposée L'approche de régression basée sur CNN pour l'estimation du RUL est à la fois efficace et précis par rapport à plusieurs algorithmes de pointe sur deux ensembles de données accessibles au public.

Inspirés par ces travaux, Li et al.[37] ont développé une nouvelle architecture d'apprentissage profonde pour l'estimation RUL dans les pronostics. Pour améliorer l'extraction de caractéristiques par CNN, une fenêtre temporelle est utilisée pour la préparation des échantillons. La proposition a adopté les mesures brutes du capteur directement comme les entrées du modèle. Ils ont affirmé qu'aucune expertise préalable sur le pronostic et le traitement du signal étaient nécessaires à l'aide de cet approche, qui facilite son application dans l'industrie des surfaces. Les performances pronostiques de l'approche proposée ont été validée sur la base des expérimentations réalisées sur l'ensemble de données C-MAPSS populaire. Les résultats de l'expérience de cette l'approche proposée a été comparée à l'approche CNN fourni par Navathe et al.[53] , et l'approche LSTM fourni par Zheng et al.[54] Les résultats de la comparaison a démontré que l'approche proposée atteint une grande précision sur la prédiction RUL.

Li et al.[55] ont développé une nouvelle approche intelligente de prédiction RUL des relèvements, où les informations du domaine temps-fréquence a été exploré et le CNN a été utilisé pour l'extraction de la fonctionnalité multi-échelle . Pour démontrer la supériorité de la méthode, le jeu de données sur les roulements préparé à partir

de La plate-forme PRONOSTIA a été utilisée pour les expériences. De plus, différentes implémentations du système basé sur DNN des approches ont été comparées telles que DNN, Single Scale-Low (SSL) et Single Scale-High (SSH). Les résultats de l'approche proposée suggèrent son potentiel dans les applications industrielles.

3.2.4. RUL avec les réseaux de neurones récurrents (RNN)

Le réseau neuronal récurrent (RNN) est une architecture profonde qui contient des connexions des couches de retour ou de couches de sortie aux couches précédentes, ainsi il est capable de traiter informations dynamiques [56]. Alors que RNN est l'un des plus techniques de modélisation séquentielle courantes, elle a des limites sur les prévisions RUL à long terme en raison de l'absence de tendance identification par les poids du résultat du réseau entraîné à partir du poids mis à jour sur la présentation de chaque schéma d'entrée . Pour cette raison, le réseau de mémoire à long terme (LSTM), qui conquiert la dépendance temporelle à long terme problèmes en contrôlant le flux d'informations à l'aide de la porte d'entrée, La porte d'oubli et la porte de sortie ont été développées. Le RNN et sa déclinaison la Mémoire à Long Court Terme (LSTM) ont acquis une grande attraction dans de nombreux applications qui ont un caractère séquentiel. Des chercheurs travaillent sur la prédiction RUL ont commencé à explorer la valeur de RNN dans ce domaine ces dernières années, en particulier LSTM.

Heimes [41] a proposé une approche RNN pour résoudre l'IEEE 2008 Challenge de la conférence Pronostic et Management de la Santé problème qui est lié à la prédiction RUL d'un système complexe. L'algorithme d'entraînement RNN consiste en une propagation arrière tronquée à travers le gradient de temps calcul, une méthode d'apprentissage du filtre de Kalman étendu et algorithmes évolutionnaires.

Yuan et al.[57] ont proposé un LSTM approche pour le diagnostic et la prédiction RUL de systèmes complexes comme un moteur d'avion. Alors que le LSTM n'a été utilisé que pour diagnostic de panne, l'estimation du RUL reposait toujours sur un Modèle SVM.

De même, Guo et al.[58] ont développé un RNN indicateur de santé basé sur l'indicateur de santé (RNN-HI) afin d'améliorer la précision

de la prédiction RUL des roulements. Cependant, le RUL des roulements est calculé à l'aide d'un modèle exponentiel avec seuil de défaillance prédéfini de RNN-HI au lieu du RNN directement.

Zheng [54] développé une approche basée sur LSTM pour l'estimation RUL en utilisant les données du capteur. La recherche a enquêté sur les modèles de capteurs et de données opérationnelles avec plusieurs conditions de fonctionnement, modèles de pannes et de dégradation par combinaison de plusieurs couches de cellules LSTM avec une alimentation standard couches avant. La supériorité du modèle LSTM en RUL la prédiction a été validée sur trois ensembles de données largement utilisés, CMAPSS Data Set, PHMo8 Challenge Data Set et Milling Base de données.

Wu et al.[41] a implémenté le LSTM vanille réseaux pour obtenir une bonne précision de prédiction RUL dans le cas d'opérations compliquées, conditions de travail, modèle dégradations et bruits forts. Un vecteur de pertinence Machines (RVM) a été utilisé pour détecter l'heure de début de dégradation et vanille LSTM a été utilisé pour calculer la RUL. L'inconvénient de cette approche est que la RUL nécessite un étiquetage à chaque pas de temps pour chaque échantillon et une certaine connaissance expérimentielle est requise puisqu'un seuil approprié doit être défini avant la mise en œuvre de la machine à vecteurs de support (SVM.) Inspirés des réseaux Vanilla LSTM, Wu et al.[59] développé un autre réseau LSTM axé sur les défauts pronostic avec séquence de dégradation des équipements, en dont le RUL peut être prédit sans aucun prédéfini au seuil. Pour atteindre cet objectif, un vecteur one-hot a été utilisé comme indicateur d'entrée à partir duquel l'heure d'arrêt a été calculé par le modèle.

Zhang et al.[54] ont présenté un réseau LSTM bidirectionnel pour découvrir les modèles intégrés dans des séries temporelles et suivre le système dégradation et, par conséquent, de prédire le RUL. Néanmoins, le réseau LSTM bidirectionnel n'était mis en œuvre pour suivre la variation de l'indice de santé, et la RUL a été prédit par la méthode récursive à un pas d'avance.

Ahmed et al.[60] a construit une nouvelle architecture LSTM pour la prédiction de RUL lorsque de courtes séquences d'observations surveillées ont été donnés avec une usure initiale aléatoire. Le LSTM

proposé a pu prédire le RUL avec des démarrages aléatoires, ce qui le rend plus adapté aux cas du monde réel en tant que premier l'état des systèmes physiques est généralement inconnu, en particulier en termes de ses défauts de fabrication. Une nouvelle fonction objectif asymétrique qui pénalise les prédictions tardives plutôt que les précédents a également été présenté afin de garantir des prévisions plus sûres.

3.2.5. RUL avec un modèle deep hybride

Une méthode hybride de prédiction RUL pour les vannes électriques a été présentée dans [?]. L'hybridation a utilisé un encodeur automatique convolutif pour extraire des caractéristiques plus profondes et un LSTM pour traiter efficacement les données de séries chronologiques. Les auteurs ont conçu une structure parallèle entre les sorties de l'auto-encodeur convolutif et les données d'origine pour enrichir les fonctionnalités introduites dans le LSTM.

Zhao et al.[61] ont présenté un approche du CNN et du LSTM bidirectionnel pour l'usinage prédiction de l'usure de l'outil nommée convolutionnelle bidirectionnelle Réseaux à mémoire longue à court terme (CBLSTM). CNN était d'abord utilisé pour extraire des caractéristiques locales robustes du entrée séquentielle. Ensuite, LSTM a été utilisé pour coder informations temporelles. La capacité du CBLSTM proposé de prédire le RUL de l'usure réelle de l'outil sur la base les données sensorielles ont été vérifiées avec un test d'usure d'outil réel. Il y a aussi beaucoup d'autres chercheurs qui ont payé leur attention sur les applications du LSTM.

3.3. Conclusion

Dans ce chapitre nous avons présenté le RUL avec une étude de ses différentes approches qui existent dans la littérature, ainsi que leurs avantages et leurs inconvénients.

nous avons terminé par une exploration de quelques travaux de RUL avec le deep learning.

Le prochain chapitre, s'intéresse à présenter notre contribution dans le domaine, le modèle que nous allons proposer ainsi que les différents résultats et comparaisons.

Quatrième partie

CONTRIBUTION

+

Chapitre 4

CONTRIBUTION

4.1. Introduction

Dans ce chapitre, nous allons utiliser le deep learning pour résoudre le problème de l'estimation de RUL pour la maintenance conditionnelle. En se basant sur les travaux connexes décrits dans le chapitre 3, nous essayerons dans la première partie de ce chapitre de proposer une nouvelle approche qui est une hybridation entre les réseaux de neurones convolutional autoencoder et Bidirectionnel GRU. Nous essayons ensuite de sélectionner les meilleures valeurs d'hyperparamètres afin d'aboutir à un modèle performant. Nous comparons enfin notre modèle avec quelques modèles de la littérature

.

4.2. Motivation

L'estimation de la durée de vie utile restante (RUL) et l'évaluation de la dégradation des performances des machines ont un rôle crucial dans la maintenance conditionnelle (CBM). Elles aident à réduire les coûts de maintenance, à améliorer la fiabilité et ainsi elles influencent la prise de décision dans le système.

Le deep learning est un outil puissant qui montre des capacités plus fortes dans le domaine de l'estimation de RUL, néanmoins dans la capture des relations non-linéaires complexes entre les entrées et les sorties.

Les problèmes résolus par les deep learning sont classés en deux types; classification et régression. La prédiction de RUL peut être vue comme un problème de régression linéaire. Les données décrivant les paramètres de la CBM sont de types séries chronologiques (time series). Un type puissant de réseau neuronal conçu pour gérer la dépendance de séquence à long terme est le LSTM et ses variantes.

Gated Recurrent Units(GRU) Bidirectionnel LSTM est choisit dans le modèle proposé pour cette raison et pour ses capacités d'apprendre les caractéristiques temporelles des données pour la prédiction de séries chronologiques.

Nous utilisons le modèle AUTO ENCODEUR Convolutionnel (CAE) pour la réduction des dimensions de données achevées en éliminant le maximum possible de bruit. L'hybridation est faite entre BIGRU et CAE pour profiter des avantages des deux modèles afin d'aboutir à une meilleure estimation de RUL.

4.3. L'architecture du modèle proposé

Notre modèle consiste en une hybridation entre le réseau profond "l'AutoEncoder Convolutionnel" et le "Gru Bidirectionnel Lstm". En premier lieu on a proposé une architecture bien déterminé. Ensuite on a fait varier à chaque fois un des hyper paramètres du modèle, analyser les résultats en fonction des valeurs de fonctions de métriques et prendre à chaque fois les valeurs aboutissant aux meilleurs résultats. La figure 1 présente les différentes couches utilisées dans notre modèle.

couche Conv2D : on ajout 4 couches convolutives de deux dimension avec 300 unités cachées pour chaque couche.

couchees Max-poling2D : nous utilisons une couche maxpooling deux dimensions après la couche convolutive. Elle a pour rôle de réduire les dimensions de l'entrée.

couches Up-Samling2D : Nous utilisons une couche Up-Sampling de deux dimensions après une couche convolutive pour décoder notre model .

couche Time-distrubuted : nous utilisons une couche de time-ditributed pour relier le modèle AEC avec le modèle BIGRU.

couchees Dropout : nous appliquons deux couches Dropout suivi de BIGRU et Time-distributed pour éviter le problème de overfitting.

couches Dense : nous ajoutons un couches entièrement connectées avec qui mappe la sortie de BIGRU à taille de sortie désirée qui est égale à 1 et prévue par la fonction d'activation linear.

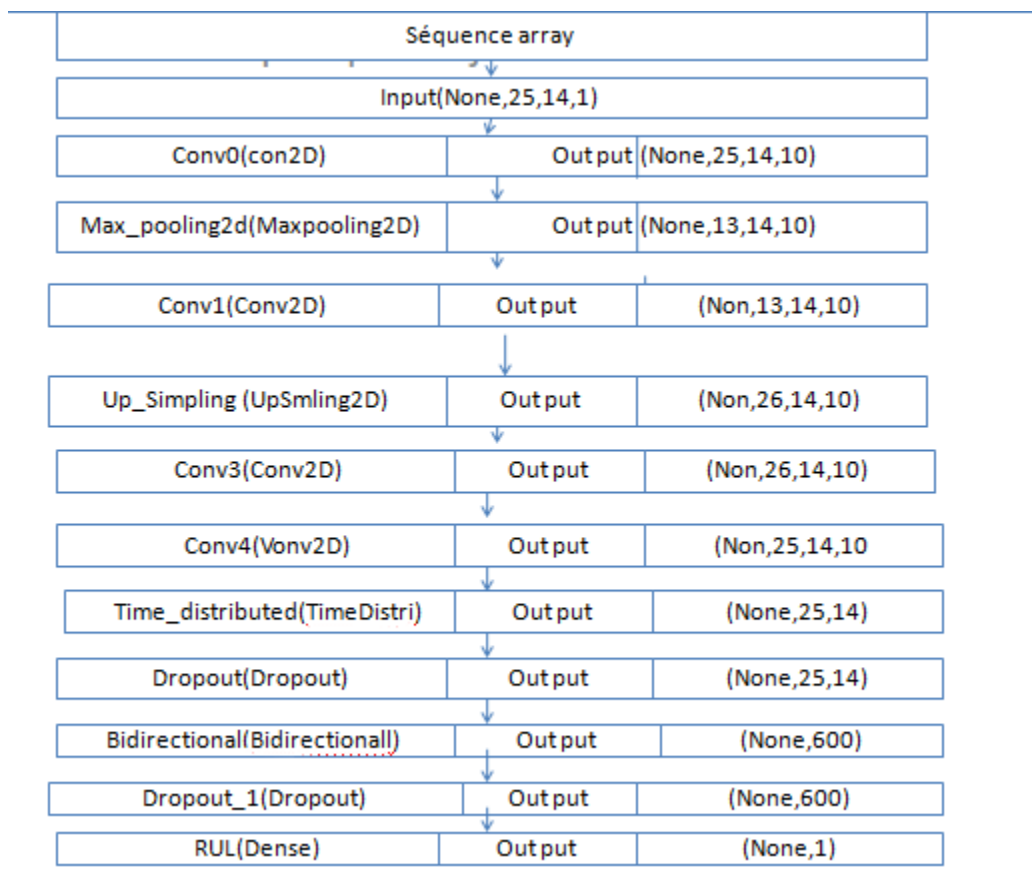


FIGURE 1. l'architecture du modèle proposé.

4.4. Expérimentation et discussions

4.4.1. Les étapes de travail

Dans notre travail, nous élaborons le modèle proposé basé sur le réseau de neurones hybride CAEBIGRU afin de prédire la durée de vie utile restante (RUL) des moteurs d'avion. Le réseau utilise des valeurs de capteur d'avion simulées pour prédire quand un moteur d'avion tombera en panne dans le futur afin que la maintenance puisse être planifiée à l'avance. Nous pouvons définir le problème comme un problème de régression, car il est donné un nombre en sortie (le nombre de cycles d'un moteur en service durera avant de tomber en panne).

Premierement ,on a chargé l'ensemble de données en tant que dataframe Pandas. (nous utiliserons les données C MAPSS dataset de

la NASA), L'ensemble de données est une série chronologique multivariée. Chaque entrée (ligne) dans l'ensemble de données reflète un cycle de fonctionnement d'un moteur spécifique identifié par l'ID du moteur et la durée du cycle. Il y a plusieurs entrées par moteur pour représenter différentes heures de rapport. Les autres colonnes représentent différentes fonctionnalités 3 paramètres opérationnels et 21 capteurs : L'ensemble de données CMAPSS est divisé en 4 ensembles chacun pour l'entraînement, le test et le RUL. Chaque sous-ensemble représente une condition opérationnelle différente et se compose d'un nombre différent de moteurs.

Tous les moteurs sont supposés être du même type de modèle et fonctionnant normalement au début de chaque série. Au cours de sa série, il développe une faille.

Le cycle est un nombre entier croissant de manière monotone et pour des raisons de modèle, il est supposé être équidistant et relatif pour chaque moteur. Les données sont ensuite divisées en un ensemble d'apprentissage et de test, chacun nécessitant une interprétation subtile lors de son traitement. Le dernier id, entrée de cycle, correspond au moment où le moteur est déclaré défectueux. Par exemple, si le premier moteur a 192 événements de séries chronologiques distincts, le cycle ira de 1 à 192, tandis que le RUL commencera par 192 et descendra à 1. Pendant la préparation des données, j'ajoute une colonne d'étiquette appelée 'rul'. C'est la vérité sur le terrain

Ensuite, Nous pouvons facilement normaliser l'ensemble de données à l'aide de la classe de pré traitement MinMaxScaler de la bibliothèque scikit-learn.

tant que on utilise AEC-BIGRU pour la modélisation de séries chronologiques donc on cree un fonction qui générera la séquence à envoyer au BIGRU selon la taille de la fenêtre.

Alors ,nous pouvons élaborer notre modele Une fois le modèle ajusté, nous pouvons estimer les performances du modèle sur les ensembles de données de train et de test. nous utilisons comme métriques d'évaluation de performance R^2 , Mean Squared Error (MSE) , mean absolute error (MAE) et root mean squared error (RMSE).

Enfin,nous pouvons générer des prédictions à l'aide du modèle pour l'ensemble de données de train et de test afin d'obtenir le vecteur de

RUL prédits.

4.4.2. L'ensemble de données utilisé

Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) [62] est un modèle de simulation de turboréacteur. C-MAPSS a été utilisé pour générer un jeu de données simulé de fonctionnement à l'échec à partir d'un moteur à double flux et il est publié dans le référentiel du centre d'excellence de pronostics de la NASA, qui est accessible au public. Un gros goulot d'étranglement dans les approches axées sur les données pour résoudre les problèmes dans le cadre de la maintenance prédictive est le manque d'ensembles de données de disponibilité jusqu'à l'échec. Cette ensemble de données simulées permet aux chercheurs de construire, tester et comparer différentes approches à ce problème. L'ensemble de données C-MAPSS est composé de quatre sous-ensembles de données avec différentes conditions de fonctionnement et de panne (figure 01). Chaque sous-ensemble de données est ensuite divisé en ensembles d'apprentissage et de test. Les données consistent en plusieurs mesures de séries chronologiques multivariées. Au sein de chaque jeu de données, chaque moteur est considéré à partir d'une flotte de moteurs du même type et où chaque série temporelle provient d'un seul moteur. Chaque moteur démarre avec différents niveaux d'usure initiale et de variation de fabrication qui sont inconnus. Ces usures et variations sont considérées comme un comportement normal pour chaque moteur. Le moteur fonctionne normalement au début de chaque série temporelle et développe un défaut à un moment donné. Dans l'ensemble d'apprentissage, le défaut prend de l'ampleur jusqu'à la panne du moteur. Dans l'ensemble de test, la série temporelle se termine quelque temps avant l'échec. L'objectif est de prédire le nombre de cycles opérationnels restants avant la défaillance dans l'ensemble de test, c'est-à-dire le nombre de cycles opérationnels restants dans le moteur. C'est ce qu'on appelle la durée de vie utile restante (RUL). Il y a 21 mesures de capteurs et trois réglages opérationnels. Chaque ligne des données est un instantané pris au cours d'un seul cycle de temps d'opération, pour un certain moteur. [11].

La figure 2 représenté bien les détails des 4 sous datasets de C-MAPSS.

Data	FD001	FD002	FD003	FD004
Engines: Training (N)	100	260	100	249
Engines: Testing	100	259	100	248
Operating Conditions	1	6	1	6
Fault Modes	1	1	2	2

FIGURE 2. tableau détaillé des sous datasets de CMAPSS.

Mentionnant que dans notre travail seules 14 données de capteur sont choisies ici comme données d'entrée du réseau comme cité dans [37] où certaines valeurs de capteur sont écartées car elles ne varient pas du tout tout au long du cycle de vie du moteur. Les colonnes sélectionnées sont 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 et 21.

4.4.3. Fonctions de métriques

Pour évaluer les performances de la méthode pronostique proposée, on utilise les fonctions de métriques suivantes : Fonction de score de régression(R_2) , mean squared error (MSE),mean absolute error(MAE) et root mean squared error(RMSE).

4.4.3.1. **L'Erreur Absolue Moyenne (Mean Absolute Error (MAE)) :** la fonction MAE mesure l'amplitude moyenne des erreurs dans un ensemble de prédictions, sans tenir compte de leur direction. C'est la moyenne sur l'échantillon de test des différences absolues entre la prédiction et l'observation réelle où toutes les différences individuelles ont le même poids. Si S_i la valeur estimée de l'ième exemple et la valeur réelle correspondante, alors l'erreur absolue moyenne (MAE) estimée est définie comme suit :[28]

$$MAE(y, \hat{y}) = \frac{1}{n_{samples}} \times \sum_{i=1}^{n_{samples}-1} |y_i - \hat{y}_i| \quad (1)$$

Le MAE est plus robuste aux valeurs aberrantes et ne pénalise pas les erreurs aussi fortement que mse. Le MAE est un score linéaire, ce

qui signifie que toutes les différences individuelles sont pondérées de manière égale.

4.4.3.2. **Erreur quadratique moyenne racine (Root mean squared error (RMSE))** : RMSE est une règle de notation quadratique qui mesure également l'ampleur moyenne de l'erreur. C'est la racine carrée de la moyenne des différences au carré entre la prédiction et l'observation réelle. Elle a la formule suivante :

$$RMSE = \sqrt{\frac{\sum_{i=1}^N e_i^2}{n}} = \sqrt{\frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{n}} \quad (2)$$

Où n est le nombre total d'exemples dans l'ensemble de test et $e_i = \hat{y}_i - y_i$ est l'erreur entre les valeurs de RUL estimées \hat{y} , et les valeurs de RUL actuelles y pour chaque engin dans l'ensemble de test. Les erreurs sont d'abord mises au carré avant le calcul de la moyenne, ce qui pénalise fortement les erreurs importantes [63].

4.4.3.3. **Erreur quadratique moyenne (Mean Squared Error(MSE))** : C'est une mesure d'évaluation de modèle souvent utilisée avec les modèles de régression. L'erreur quadratique moyenne d'un modèle par rapport à un ensemble de test est la moyenne des erreurs de prédiction au carré sur toutes les instances de l'ensemble de test. L'erreur de prédiction est la différence entre la valeur vraie et la valeur prédite pour une instance. Elle a pour formule [64] :

$$MSE = \frac{\sum_{i=0}^n (a_i - p_i)^2}{n} \quad (3)$$

4.4.3.4. **Fonction de score de régression (R^2)** : Le R^2 , ou R-carré est appelé coefficient de détermination. C'est un indicateur utilisé en statistiques pour juger de la qualité d'une régression linéaire. Mathématiquement, il s'agit de la proportion de la variance d'une variable dépendante qui s'explique par une ou plusieurs variables indépendantes dans le modèle de régression. On l'exprime soit entre 0 et 1, soit en pourcentage. Alors que la corrélation explique la force de la relation entre une variable indépendante et une variable dépendante, le R au carré explique dans quelle mesure la variance d'une variable explique la variance de la seconde variable. Le R^2 se calcule à partir

de la formule suivante :[5]

$$R^2(y, \bar{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4)$$

où $\bar{y} = \frac{1}{n} \times \sum_{i=1}^n y_i$ et $\sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n e_i^2$ Avec y_i la valeur du point i , \hat{y}_i la valeur prédite pour le point i par la régression linéaire, \bar{y} la moyenne empirique des points donnés.

4.4.4. Le pré-traitement de données

4.4.4.1. **Le pré-traitement des données de l'apprentissage et de test** : Premièrement, on charge l'ensemble de données en tant que dataframe Pandas (C MAPSS dataset), L'ensemble de données est une série chronologique multivariée. Chaque entrée (ligne) dans l'ensemble de données reflète un cycle de fonctionnement d'un moteur spécifique identifié par l'ID du moteur et la durée du cycle. Il y a plusieurs entrées par moteur pour représenter différentes heures de rapport. Les autres colonnes représentent différentes fonctionnalités, 3 paramètres opérationnels et 21 capteurs : L'ensemble de données CMAPSS est divisé en 4 ensembles chacun pour l'entraînement, le test et le RUL. Chaque sous-ensemble représente une condition opérationnelle différente et se compose d'un nombre différent de moteurs. Tous les moteurs sont supposés être du même type de modèle et fonctionnant normalement au début de chaque série. Au cours de sa série, il développe une faille. Le cycle est un nombre entier croissant de manière monotone et pour des raisons de modèle, il est supposé être équidistant et relatif pour chaque moteur. Les données sont ensuite divisées en un ensemble d'apprentissage et de test, chacun nécessitant une interprétation subtile lors de son traitement. Le dernier id, entrée de cycle, correspond au moment où le moteur est déclaré défectueux. Par exemple, si le premier moteur a 192 événements de séries chronologiques distincts, le cycle ira de 1 à 192, tandis que le RUL commencera par 192 et descendra à 1. Pendant la préparation des données, on ajoute une colonne d'étiquette appelée 'rul'. C'est la vérité sur le terrain

4.4.4.2. **La normalisation des données** : Pour assurer une contribution égale de toutes les caractéristiques dans toutes les conditions de fonctionnement [65], une normalisation avec la méthode Min-Max est appliquée à un intervalle de $[-1, 1]$ en utilisant l'équation 5 de normalisation . décrite dans [37]

$$x_{norm}^{i,j} = \frac{2 \times x^{i,j} - x_{min}^{i,j}}{x_{max}^j - x_{min}^j} - 1, \forall i, j \quad (5)$$

où $x^{i,j}$ désigne le ième point de données d'origine du jème capteur, et $x_{norm}^{i,j}$ norm est la valeur normalisée de $x^{i,j}$. x_{max}^j et x_{min}^j désignent respectivement les valeurs maximales et minimales des données de mesure d'origine du jème capteur [37].

4.4.4.3. **Le prétraitement de "time window"**. Les données sont traitées avec une approche de fenêtre temporelle glissante "sliding time window approach" de taille $N_f = 25$ et un pas "stride" de 1. La fenêtre glissante signifie que le premier échantillon d'entrée sur le réseau prend des mesures des cycles 1-25, le second 2-26 et ainsi de suite pour chaque unité de la flotte. L'étiquette RUL pour un échantillon est alors simplement le nombre total de cycles du moteur est capable de fonctionner moins le cycle où la fenêtre se termine. [37]. Suivant la procédure standard adoptée par d'autres chercheurs, l'horizon maximal de prédiction pour RUL (R_{early}) a été limité à 125 cycles. Cela a un impact sur la précision du modèle et rend les modèles plus stables.

4.4.5. L'ajustement des hyperparamètres

Dans cette partie, nous présentons les différents résultats et effets de chaque changement d'hyper paramètre sur le modèle CAE_BIGRU pour FD001.

4.4.5.1. **L'effet de modification des valeurs de windows size** : Lors de l'utilisation de modèles d'apprentissage en profondeur pour la prédiction RUL, une approche de fenêtre temporelle (Time window) est largement utilisée pour convertir le flux de données de

surveillance des conditions en échantillons d'entrée pour l'apprentissage supervisé. Plus précisément, l'approche de la fenêtre temporelle prend les caractéristiques des pas de temps précédents comme variables d'entrée, tandis que RUL étiquette le pas de temps suivant comme variable de sortie [66]. Nous utilisons les valeurs (20,25,30) pour 20 epochs ,Activation=linear et optimizer=rmsprop. Nous obtenons les résultats du tableau 1 suivant :

TABLEAU 1. les résultats du modèle avec différentes valeurs de windows size

WS	train				test			
	mse	mae	rmse	r2	mse	mae	rmse	r2
20	205.33	10.69	13.257	0,69	245.63	10.436	15.42	0.85
25	165.96	9.34	11.79	0,92	172.89	9.39	12.76	0.90
30	235.99	12.21	14.61	0.71	248.38	11.66	15.46	0.85

Après avoir comparé les différents résultats des fonctions de métriques on trouve que le meilleur résultat de windows size est 25. Une plus grande fenêtre de temps peut couvrir plus d'informations brutes, qui est la base pour une extraction supplémentaire des fonctionnalités [64]. En comparant les résultats de test avec ceux de training on ne trouve pas une grande différence, et donc il n'ya pas un overfitting des résultats. Pour le rest du document on va travaillé avec cette valeur.

4.4.5.2. L'effet des valeurs de fonctions d'activation et d'optimizers. Les fonctions d'activations utilisées sont : Relu, sigmoid et tanh et les optimizers sont : adam, rmsprop et adadelta. les résultats sont illustrés dans le tableau suivant :2.

Après comparaison des différents résultats de fonctions de métriques, on trouve que le meilleur résultat de fonction d'activation est tanh et pour l'optimizer est rmsprop.

"tanh" est fréquemment utilisé dans les couches cachées d'un ANN car ses valeurs se situent entre -1 et 1 . Par conséquent, la moyenne de la couche cachée est très proche de 0, ce qui aide à centrer les

TABLEAU 2. les résultats du modèle avec différentes valeurs de fonctions d'activations et d'optimiser

CAEBiGRU	train					test			
Activatio function	Optimizer	mse	mae	rmse	r2	mse	mae	rmse	r2
Relu	adam	324.36	14.41	17.04	0,86	294.36	13.71	16.86	0.82
	rmsprop	277.95	13.29	15.84	0,85	339.48	14.52	18.14	0.79
	adadelta	250.37	12.90	15.13	/	289.03	13.72	18.39	0.82
sigmoid	adam	336.70	14.06	16.59	/	338.59	13.05	18.40	0.80
	rmsprop	412.14	14.01	17.85	/	438.48	16.35	20.94	0.74
	adadelta	6274.35	68.50	77.55	/	61.87	73.760	0.80	/
tanh	adam	165.96	9.34	11.79	0,92	172.89	9.39	12.76	0.90
	<i>rmsprop</i>	152.09	8.62	11.35	/	137.29	8.769	11.39	0.92
	adadelta	187.38	9.27	12.42	/	242.6	11.40	15.27	0.85

données. Cela rend l'apprentissage pour la couche suivante beaucoup plus facile et plus efficace, car il dispose d'une gamme plus large pour un apprentissage et une notation plus rapides. [67].

4.4.5.3. **Effet des différentes valeurs de validation-split.** Le tableau 3 illustre les différents résultats en faisant varier les valeurs de validation-split à 0.05, 0.1, 0.15 et 0.2.

On constate que 0.1 comme valeur d'exemples de validation par rap-

TABLEAU 3. Résultats de modification de validation-split

CAEBiGRU	train				test			
Validation splite	mse	mae	rmse	r2	mse	mae	rmse	r2
0,05	122.83	7.52	10.34	0,93	175.84	9.83	13.26	0.89
0,1	152	8,62	11,35	0,9	137,29	8,77	11,39	0,92
0,15	184.75	8.99	12.32	0,88	274.95	12.29	16.58	0.84
0,2	102.21	6.52	9.23	0,94	233.30	10.99	15.27	0.86

port à l'ensemble de training donne les meilleurs résultats des metrics.

4.4.5.4. **L'effet de changement de batch size sur les résultats.** Le batch size représente la taille de l'intervalle (fenêtre) pour prendre un

max sur le kernel_size. Les valeurs prises de batch size sont 100,128,200 et 300. Le tableau 5 illustre les résultats.

Après comparaison des différents résultats de fonctions de métriques,

TABLEAU 4. Les résultats de modifications de batch size

CAEBiGRU	train				test			
les valeurs batch-size	mse	mae	rmse	r2	mse	mae	rmse	r2
100	102.67	7.10	9.47	/	272.65	11.80	16.51	0.84
128	152	8,62	11,4	/	137,3	8,77	11,39	0,92
200	86.35	6.41	8.64	/	274.95	12.29	16.58	0.84
300	168.88	9.31	12.22	/	217.46	10.73	14.75	0.88

on trouve que le meilleur résultat est donné par la valeur de batch-size égale à 128.

4.4.5.5. **L'effet de modification des valeurs de learning rate** : Le taux d'apprentissage "Le learning rate" est un hyper paramètre important, il contrôle la quantité de mise à jour des poids en réponse au gradient estimé à la fin de chaque lot. Cela peut avoir un impact important sur le compromis entre la rapidité ou la qualité d'apprentissage du problème par le modèle. Les valeurs de learning rate à utiliser sont : 0.001, 0.00001, 0.0001, 0.000001 et 0.0000001. Les résultats sont illustrer par le tableau 5.

Après comparaison des différents résultats, on trouve que le meilleur résultat est donné par le learning rate égale à 0.0001.

4.4.5.6. **L'effet du changement de nombre de neurones de GRUBi sur les résultats** : Dans cette section nous modifions le nombre de

TABLEAU 5. les résultats de modification de learning rate.

CAEBiGRU	train				test			
	mse	mae	rmse	r2	mse	mae	rmse	r2
les valeurs Learning Rate								
0,001	126.44	7.89	10.40	/	273.705	12.23	16.54	0.84
0,00001	75.15	5.81	7.97	/	255.81	11.68	15.99	0.85
0,0001	152	8,62	11,35	/	137,29	8,77	11,39	0,92
0,000001	148.48	8.57	11.44	/	205.40	10.32	14.33	0.88
0,0000001	74.92	5.81	7.96	/	254.66	11.63	15.95	0.85

neurones "num timesteps" avec différent nombre de couches. Les tableaux 6, 7 et 8 résument les différents résultats.

TABLEAU 6. Les différents résultats du changement du nombre de num time steps à 300

CAEBiGRU	train				test			
	mse	mae	rmse	r2	mse	mae	rmse	r2
les couches								
5 couches cnn et 2BI-GRU,2Max-pooling	33.95	2.82	3.93	/	288.77	12.63	16.99	0.83
4 CNN et 2BI-GRU	32.88	2.71	3.89	/	342.67	13.20	18.51	0.80
4 cnn et 1 couches BI-GRU	152	8,62	11,35	/	137,3	8,77	11,39	0,92

4.4.5.7. **L'effet de changement du nombre de filtre** : Les valeurs de filtres prises sont : [10;10;10,1] et [20;20;20,1]. Les deux tableaux

TABLEAU 7. Les différents résultats du changement du nombre de num time steps à 200

CAEBiGRU	train				test			
les couches	mse	mae	rmse	r2	mse	mae	rmse	r2
5 cnnAE , 2BI-GRU, 2 Max-pooling	47.09	3.99	5.64	/	288.25	12.06	16.97	0.83
4 CNNAE, 2 BI-GRU	44.58	3.72	5.27	/	313.46	12.91	17.70	0.83
4 cnnAE,1 BI-GRU	144	8.16	11.2	/	280.82	12.274	16.75	0.83

TABLEAU 8. Les différents résultats du changement du nombre de num time steps à 400

CAEBiGRU	train				test			
les couches	mse	mae	rmse	r2	mse	mae	rmse	r2
5 cnnAE , 2BI-GRU, Max-pooling	33.43	2.68	3.77	/	388.75	14.48	19.71	0.77
4 CNNAE, 2 BI-GRU	108.47	6.46	9.41	/	368.98	14.14	19.20	0.79
4 cnnAE,1 BI-GRU	65.93	5.32	7.28	/	339.94	13.62	18.46	0.81

suivants font illustrer les résultats .9.10

TABLEAU 9. Résultats pour filtre = [10;10;10,1]

CAEBiGRU	train				test			
les couches	mse	mae	rmse	r2	mse	mae	rmse	r2
5 couches cnn et 2BI-GRU, 2Max-pooling	33.95	2.82	3.93	/	288.77	12.63	16.99	0.83
4 CNNet 2 BI-GRU	32.88	2.71	3.89	/	342.67	13.20	18.51	0.80
4 couches cnn et 1 couchesBI-GRU	152	8,62	11,35	/	137,29	8,77	11,39	0,92

On remarque que les valeurs de filtres 10, 10, 10 et 1 donnent une meilleure performance du modèle.

TABLEAU 10. Résultats pour filtre = [20;20;20,1]

CAEBiGRU les couches	train				test			
	mse	mae	rmse	r2	mse	mae	rmse	r2
5 couches cnn et 2BI-GRU	35.03	2.75	3.91	/	375,7	13,5	16,3	0,76
4 CNNet 1 BI-GRU	34.85	2.32	3.33	/	323.55	13.60	17.98	0.81
4 couches cnn et 1couches BI-GRU	65.5	5.16	7.21	/	359.96	13.903	18.97	0.79

4.4.6. Les résultats finals du modèle proposé

Les résultats finaux de notre modèle proposé sous le sous data set (FD001) après 20 époques et avec utilisation des paramètres que nous avons ajusté (Epoch=20, Activation=tanh, Optimizer= Rmsprop, WS=25, validation_split=0,1 ;batch_size=128,num_timesteps=300,Learning Rate=0.0001) et qui donne le modèle le plus performant sont illustré par la figure 11.

TABLEAU 11. Les résultats finals de modèle proposé(CAEBIGRU)

les résultats final	train				test			
	mse	mae	rmse	r2	mse	mae	rmse	r2
CAE BiGRU	152	8,62	11,35	-5,041E+09	137,29	8,77	11,39	0,92

4.4.7. L'affichage des résultats finals

La figure 3 nous donne les résultats finaux des fonctions de métriques pour le modèle (CAEBIGRU) sous le sous dataset FD001 de CMAPSS.

montre une comparaison entre les valeurs de training R^2 et de test R^2 . On peut voir que les valeurs augmentent au fil des époques et ont tendance à se rapprocher de 1. Les valeurs d'entraînement et de test sont proches.

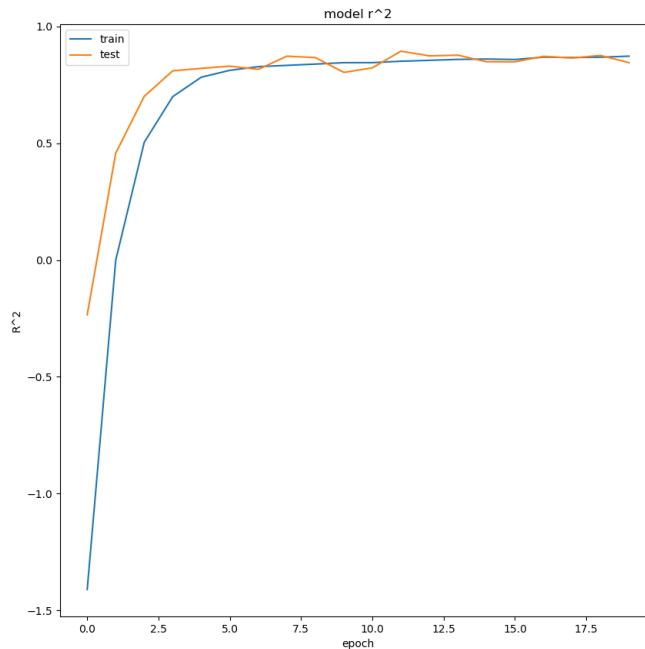


FIGURE 3. L’affichage des métriques R2 avec le modèle (CAEBIGRU) pour la sous donnée FD001 de CMAPSS.

La figure 4 nous donne les résultats finaux des métriques MSE avec le modèle (CAEBIGRU) pour la sous données FD001 de CMAPSS.

La figure 5 nous donne les résultats finaux des métriques MAE avec le modèle (CAEBIGRU) pour la sous donnée FD001 de CMAPSS.

La figure 6 nous donne les résultats finaux des métriques RMSE avec le modèle (CAEBIGRU) pour la sous donnée FD001 de CMAPSS. Elle compare les valeurs d’entraînement et de test rmse. On peut voir que l’erreur diminue avec le temps. la valeur efficace d’apprentissage et la valeur efficace de validation diminuent avec l’époque du processus d’apprentissage. En outre, la valeur efficace a tendance à être constante au stade tardif.

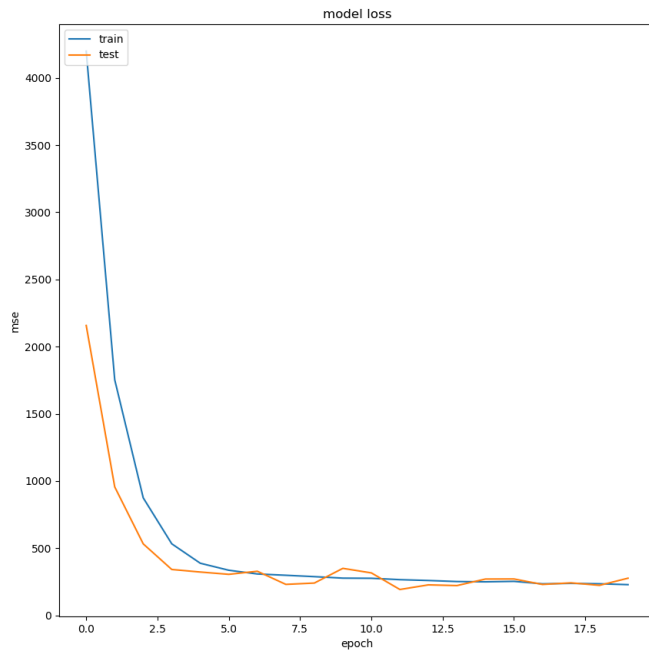


FIGURE 4. Les résultats finals du modèle CAEBi

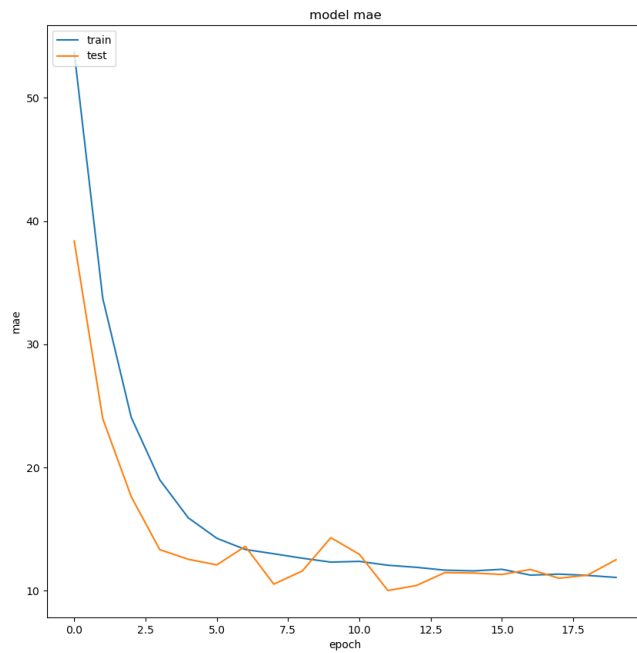


FIGURE 5. Les plots des métriques MAE avec le modèle (CAE BIGRU) pour la sous donnée FD001 de CMAPSS.

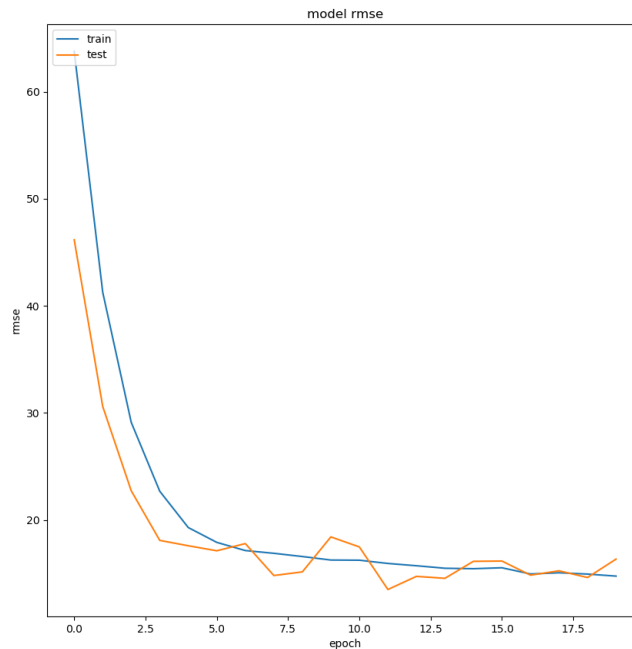


FIGURE 6. Les plots des métriques RMSE avec le modèle (CNNAEBI-GRU) pour la sous donnée FD001 de CMAPSS.

La figure 7 nous donne les résultats finaux de la distribution d’erreur avec le modèle(CAEBIGRU) pour la sous donnée FD001 de CMAPSS.

On constate que le rmse d’apprentissage a la même tendance de changement avec le rmse de validation et la même chose avec les valeurs R^2 et les autres métriques. Par conséquent, il n’y a pas de problème de surapprentissage dans le processus d’apprentissage, et la sélection d’hyper-paramètres d’apprentissage est adaptée à ce problème.

4.4.8. Comparaison avec les approches similaires de la littérature utilisant CMAPSS

Le tableau 12 montre que les résultats obtenues par le modèle proposé sont encourageantes.

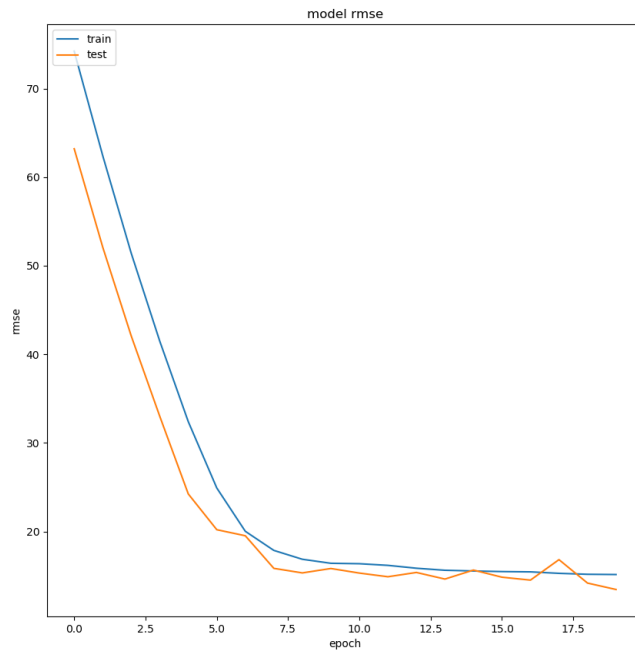


FIGURE 7. Les plots de distribution d’erreur avec le modèle (CAEBIGRU) pour la sous donnée FD001 de CMAPSS.

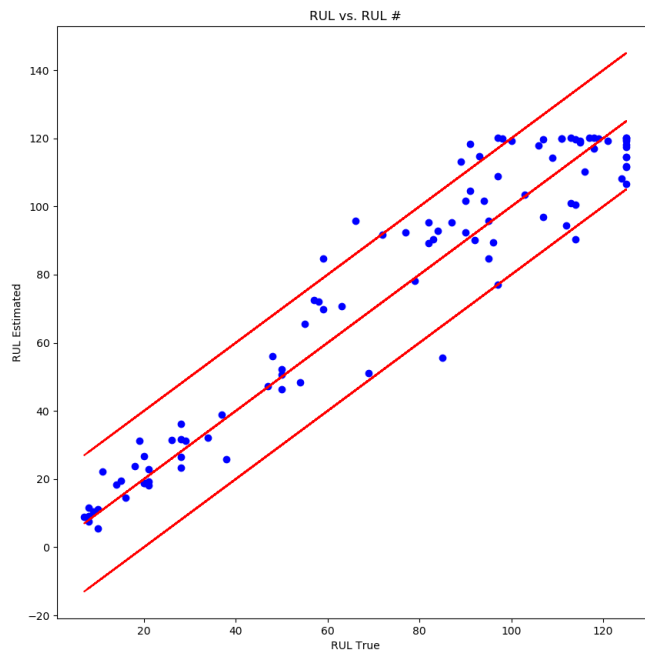


FIGURE 8. Les plots de regression avec le modèle (CAEBIGRU) pour la sous données FD001 de CMAPSS.

TABLEAU 12. Comparaison de RMSE sur C-MAPSS sous le sous dataset FD001.

Approach	FD001
CNN BiLSTM [68]	10.74
BD-LSTM [54]	15.42
LSTM [54]	18.07
BD-RNN [54]	20.04
CNN [53]	18.45
MLP [54]	20.84
DCNN [37]	12.61
LSTM [37]	13.52
RNN [37]	13.44
[69]	12.18
Parallel LSTM CNN [65]	13.017
Noisy Parallel CNN BiGRU and FC (NPBGRU) [70]	10.44
Double channel CNN and bidirectional LSTM [71]	12.58
Gated graph neural networks [72]	12.14
BiLSTM MultiscaleCNN [73]	12,75
CNNBDGRU [74]	11.085
The proposed method	11.39

4.5. Conclusion

Nous avons proposé dans ce chapitre un modèle hybride basé deep learning pour la prédiction de RUL. Le modèle est le CAEBiGru, une hybridation entre CNN, Auto Encoder et Gru Bidirectionnel . Ensuite pour aboutir à une meilleure configuration, nous avons fait varier différents hyper paramètres et étudier l'effet sur le résultat.

A la fin, nous avons comparé le modèle avec les autres approches similaires dans la littérature. L'implémentation de ce modèle sera présenté dans le prochain chapitre.

Cinquième partie

Implémentation

+

Chapitre 5

IMPLEMENTATION

5.1. Introduction

Dans ce chapitre nous avons présenté les outils de développement utilisés pour la réalisation du modèle proposé ainsi que quelques fragments de code associés.

5.2. Les outils de développement

Pour réaliser notre travail, nous avons utilisé python comme langage de programmation , google colab comme outil de développement , Anaconda comme Environment de programmation .

5.2.1. Le langage python

Le langage Python est un langage de programmation open source multi-plateformes et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données, ou la gestion d'infrastructures. Il n'est donc pas, comme le langage HTML par exemple, uniquement dédié à la programmation web. Python est principalement utilisé pour le scripting et l'automatisation de tâches simples mais fastidieuses, c'est-à-dire l'interaction avec les navigateurs web. Mais Python est aussi utilisé pour : programmer des applications, générer du code, créer des services web , faire de la métaprogrammation. Langage principalement utilisé pour le machine learning et la data science, Python a fortement contribué à l'essor du big data. Grâce à ses nombreuses bibliothèques telles Panda, Bokeh, Numpy, Scipy, Scrapy, Matplotlib, Scikit-Learn ou encore TensorFlow, Python offre une grande flexibilité dans les tâches à effectuer et une grande compatibilité quelle que soit la plateforme utilisée[7]

5.2.2. Environnement Anaconda

Anaconda est une plateforme de science des données pour les scientifiques des données. C'est une distribution de Python, etc. Avec plus de 300 packages pour la science des données, elle est rapidement devenue l'une des meilleures plateformes pour tout projet. Anaconda aide à simplifier la gestion et le déploiement des packages. Ce est livré avec une grande variété d'outils pour collecter facilement des données à partir de diverses sources en utilisant divers apprentissage automatique et algorithmes d'IA. Il aide à obtenir un environnement facilement gérable configuration qui peut déployer n'importe quel projet en un seul clic.[75]

5.2.3. google colab

Colaboratory, souvent raccourci en "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation. En termes plus techniques, Colab est un service hébergé de notebooks Jupyter qui ne nécessite aucune configuration et permet d'accéder gratuitement à des ressources informatiques, dont des GPU.[10]

5.2.4. Les bibliothèques Tensorflow et Keras

TensorFlow est une bibliothèque open source, permettant d'exécuter des applications de machine learning et de deep learning. Cet outil dédié à l'apprentissage automatique a été développé par Google, et est fortement utilisé dans le domaine de l'intelligence artificielle (IA). Ainsi, des professionnels comme des novices peuvent créer des modèles de machine learning ou de deep learning pour optimiser les capacités de leur matériel.[23] Keras est un framework de deep learning open source. Cet outil peut fonctionner au-dessus de TensorFlow, Theano, Microsoft Cognitive Toolkit et PlaidML. L'USP de Keras est sa vitesse – il est livré avec un support intégré pour le parallélisme des données, et donc, il peut traiter des volumes massifs de données tout en accélérant le temps de formation pour les modèles. Comme il est écrit en Python, il est incroyablement facile à utiliser

et extensible. Si Keras se comporte brillamment pour les calculs de haut niveau, les calculs de bas niveau ne sont pas son point fort. Pour les calculs de bas niveau, Keras utilise une autre bibliothèque appelée backend[9]



FIGURE 1. Les bibliothèques Tensorflow et Keras

5.3. Quelques Fragments de code

La figure 2 représente la normalisation des données du modèle CAEBIGRU

```
# Normalize the data
ub = train_set_x_org.max(0)
lb = train_set_x_org.min(0)
train_set_x = 2 * (train_set_x_org - lb) / (ub - lb) - 1

N_ft = sel.shape[0] # Number of features (N_ft)
m, N_en = sliding_window(train_set, N_tw, stride) # Number of training data & engines

train_x = np.empty((m, N_tw, N_ft, 1), float)
train_y = np.empty((m, 1), float)

k = 0
for i in range(N_en):
    idx = train_set[:,0] == i+1 # Index for engine number 'i'
    train_i_x = train_set_x[idx,:] # Engine 'i' training data
    train_i_c = train_set_c[idx] # Engine 'i' cycles (c)
    train_i_y = train_i_c[-1] - train_i_c # RUL: Remaining Useful Lifetime for engine 'i'
    train_i_y[train_i_y > R_early] = R_early # R_early = 125
    N_sw = int((train_i_x.shape[0] - N_tw) / stride + 1) # Number of sliding windows for engine 'i'
    for h in range(N_sw):
        k = k + 1
        vert_start = h * stride
        vert_end = h * stride + N_tw
        train_i_x_slice = train_i_x[vert_start:vert_end,:] # Training input data for engine 'i' on time window 'h'
        train_i_y_slice = train_i_y[vert_start:vert_end,:] # Training output data for engine 'i' on time window 'h'
        train_i_x_slice.shape = (N_tw, N_ft, stride) # Reshape training set input (N_tw, N_ft, 1)
        train_i_y_slice.shape = (1, 1) # Reshape training set output (1, 1)
        train_x[k-1,:,:,:] = train_i_x_slice
        train_y[k-1,:] = train_i_y_slice
```

FIGURE 2. Normalisation des données.

La figure 3 montre un fragment de code de l'implémentation du modèle CONVAEBI et la figure 4 présente l'affichage de la structure du modèle hybride par python. le modele CAEBIGRU

5.4. Exemples d'exécution

La figure 6 ainsi que la figure 7 et la figure 8 montrent des exemples d'exécution du code pendant les phases d'apprentissage et de test respectivement.


```

CWL_2d_fit(X_train, Y_train, epochs = 20, batch_size = 128, validation_split=0.10) ##1
CWL_2d_fit(X_train, Y_train, epochs = 20, batch_size = 200, validation_split=0.10)
CWL_2d_fit(X_train, Y_train, epochs = 20, batch_size = 300, validation_split=0.10)
CWL_2d_fit(X_train, Y_train, epochs = 20, batch_size = 128, validation_split=0.15)

Epoch 1/20
129/129 [#####] - 1785 15/step - loss: 1767.5490 - mae: 33.2290 - rmse: 39.6779 - r2_keras: -0.0279 - val_loss: 585.8658 - val_mae: 20.8758 - val_rmse: 22.7310 - val_r2_keras: -0.6469768192.0000
Epoch 2/20
129/129 [#####] - 1785 15/step - loss: 404.9213 - mae: 15.9063 - rmse: 19.8473 - r2_keras: 0.7653 - val_loss: 328.7017 - val_mae: 13.8817 - val_rmse: 16.2118 - val_r2_keras: -0.737489408.0000
Epoch 3/20
129/129 [#####] - 1785 15/step - loss: 312.2321 - mae: 13.3527 - rmse: 17.5251 - r2_keras: 0.8193 - val_loss: 318.7096 - val_mae: 14.5926 - val_rmse: 16.7290 - val_r2_keras: -32757493760.0000
Epoch 4/20
129/129 [#####] - 1785 15/step - loss: 263.9244 - mae: 12.1569 - rmse: 16.1426 - r2_keras: 0.8458 - val_loss: 334.1859 - val_mae: 13.5886 - val_rmse: 16.9637 - val_r2_keras: -45390168064.0000
Epoch 5/20
129/129 [#####] - 1785 15/step - loss: 235.8828 - mae: 11.3200 - rmse: 15.2571 - r2_keras: 0.8629 - val_loss: 338.1600 - val_mae: 13.5893 - val_rmse: 16.5123 - val_r2_keras: -34245670912.0000
Epoch 6/20
129/129 [#####] - 1785 15/step - loss: 218.8263 - mae: 10.8361 - rmse: 14.7145 - r2_keras: 0.8731 - val_loss: 289.0078 - val_mae: 13.1429 - val_rmse: 15.6864 - val_r2_keras: -33155823616.0000
Epoch 7/20
129/129 [#####] - 1699 15/step - loss: 209.6530 - mae: 10.5604 - rmse: 14.4210 - r2_keras: 0.8779 - val_loss: 210.4124 - val_mae: 11.7004 - val_rmse: 13.8559 - val_r2_keras: -14943270800.0000
Epoch 8/20
129/129 [#####] - 1715 15/step - loss: 206.8116 - mae: 10.4568 - rmse: 14.2291 - r2_keras: 0.8818 - val_loss: 194.1190 - val_mae: 9.6645 - val_rmse: 12.5230 - val_r2_keras: -3729988864.0000
Epoch 9/20
129/129 [#####] - 1785 15/step - loss: 194.3149 - mae: 10.0567 - rmse: 13.8746 - r2_keras: 0.8869 - val_loss: 400.4615 - val_mae: 14.8104 - val_rmse: 17.9387 - val_r2_keras: -34989510656.0000
Epoch 10/20
129/129 [#####] - 1785 15/step - loss: 187.2934 - mae: 9.8946 - rmse: 13.6141 - r2_keras: 0.8913 - val_loss: 280.1188 - val_mae: 12.0282 - val_rmse: 15.3408 - val_r2_keras: -10140173312.0000
Epoch 11/20
129/129 [#####] - 1715 15/step - loss: 181.1144 - mae: 9.6697 - rmse: 13.3825 - r2_keras: 0.8948 - val_loss: 222.9832 - val_mae: 10.7370 - val_rmse: 13.8612 - val_r2_keras: -10755919072.0000
Epoch 12/20
129/129 [#####] - 1715 15/step - loss: 172.3639 - mae: 9.4264 - rmse: 13.0510 - r2_keras: 0.8992 - val_loss: 232.1826 - val_mae: 10.0560 - val_rmse: 13.4821 - val_r2_keras: -7156245504.0000
Epoch 13/20
129/129 [#####] - 1715 15/step - loss: 164.6575 - mae: 9.1437 - rmse: 12.7770 - r2_keras: 0.9041 - val_loss: 365.8536 - val_mae: 14.8048 - val_rmse: 18.2465 - val_r2_keras: -49253588992.0000
Epoch 14/20
129/129 [#####] - 1715 15/step - loss: 157.8523 - mae: 8.9655 - rmse: 12.4882 - r2_keras: 0.9081 - val_loss: 307.8716 - val_mae: 13.2475 - val_rmse: 16.6217 - val_r2_keras: -17311886592.0000
Epoch 15/20
129/129 [#####] - 1785 15/step - loss: 144.5923 - mae: 8.5540 - rmse: 11.9525 - r2_keras: 0.9159 - val_loss: 267.7347 - val_mae: 11.6520 - val_rmse: 15.1714 - val_r2_keras: -14684624896.0000
Epoch 16/20
129/129 [#####] - 1725 15/step - loss: 134.8265 - mae: 8.2281 - rmse: 11.5167 - r2_keras: 0.9221 - val_loss: 298.5138 - val_mae: 12.0422 - val_rmse: 15.8810 - val_r2_keras: -35888236544.0000

```

FIGURE 6. L'exécution du modèle CAE-BIGRU.

```

## 4.2 Training set

preds = CWL_2d.evaluate(x = X_train, y = Y_train, verbose=1, batch_size=128)
print()
#print ("Train " + str(preds))
#print ("Test RMSE = " + str(np.sqrt(preds)))
print("\nMSE: {}".format(preds[0]))
print("\nmse: {}".format(preds[1]))
print("\nmse: {}".format(preds[2]))
print("\nr^2: {}".format(preds[3]))

y_hat_tr = CWL_2d.predict(x = X_train)
score_1_tr = score_cal(y_hat_tr, Y_train)

143/143 [#####] - 89s 624ms/step - loss: 108.4689 - mae: 6.4651

MSE: 108.46886444091797
mae: 6.465122222900391
rmse: 9.414093971252441
R^2: -2763108864.0

```

FIGURE 7. L'exécution de training du modèle CAE-BIGRU.

```

## 4.3 Test set

y_hat = CWL_2d.predict(x = X_test)
score_1 = score_cal(y_hat, Y_test)
score = print("Score = " + str(sum(score_1)))

y_hat = CWL_2d.predict(x = X_test)
score_1 = score_cal(y_hat, Y_test)
score = "Score = " + str(sum(score_1))
score

preds = CWL_2d.evaluate(x = X_test, y = Y_test, batch_size=128)
print()
print ("MSE = test " + str(preds))
#print ("RMSE = " + str(np.sqrt(preds)))
print("\nMSE: {}".format(preds[0]))
print("\nmse: {}".format(preds[1]))
print("\nmse: {}".format(preds[2]))
print("\nr^2: {}".format(preds[3]))

1/1 [#####] - 15 520ms/step - loss: 368.9082 - mae: 14.1448 - rmse: 19.

MSE = test [368.90816357421875, 14.144826889038086, 19.208856826416, 0.7863308442695618]

MSE: 368.90816357421875
mae: 14.144826889038086
rmse: 19.208856826416
R^2: 0.7863308442695618

```

FIGURE 8. L'exécution de test du modèle CAE-BIGRU.

Conclusion générale

Le travail présenté dans ce mémoire est une estimation de temps restant utile (RUL) pour la maintenance conditionnelle.

Le but générale était de proposer un modèle performant à base de deep learning, la méthode d'apprentissage qui a montré ses performances ces dernières années, pour une bonne estimation de RUL. Un modèle nommé **CAEBIGRU**, hybridation entre CNN, AutoEncoder et BiGru est proposé. Ce choix est justifié par l'efficacité de la méthode pour les problèmes de prédiction de séries chronologiques (time series) qui sont un type difficile de problème de modélisation prédictive ainsi que leurs capacités dans l'extraction des caractéristiques. Un ajustement des hyperparamètres est faites puis une comparaison avec les autres modèles similaires de la littérature est effectuée afin d'étudier l'efficacité de la méthode par rapport au problème et par rapport aux autres méthodes proposées dans le domaine.

Ce projet s'est avéré bénéfique sur plusieurs plans. Il nous a permis d'acquérir de nouvelles connaissances et de renforcer d'autres. Il nous a permis de découvrir notamment le domaine de la maintenance, la maintenance conditionnelle et le deep learning qui nous en font nécessiter une recherche bibliographique profonde. Il nous a permis aussi d'apprendre un nouveau langage de programmation qui est Python et ses différentes technologies.

Durant la réalisation de ce travail, nous avons rencontré des difficultés, mentionnant :

1. Les optimisations manuels des hyper paramètres.
2. La difficulté d'accès aux les ressources qui ne sont pas toujours disponible (payant) .
3. Plus important encore, le manque d'équipements physiques suffisamment puissants pour faciliter l'apprentissage et le test des méthodes appliquées. Ce problème rend également le temps d'exécution

sur les autres ensembles (FD002, FD003, FD004) de la base des données CMAPSS.

A la fin nous pouvons dire que notre travail a atteint ses objectifs fixés au début, nous avons arrivé à proposer un modèle basé deep learning "CAEBIGRU" avec une configuration optimal, l'implémenter, ajuster ses hyper paramètres puis comparer les résultats et les analyser.

Les résultats de notre proposition ont justifié l'efficacité de modèle CAEBIGRU .Mais comme n'importe quel travail scientifique il reste ouvert à d'autres améliorations.

Comme perspectives de notre travail, nous pouvons citer :

- L'application du modèle sur le reste des sous dataset.
- L'application du modèle sur un autre type de dataset pour voir la performance du modèle dans un autre domaine.
- L'utilisation d'une méthode d'optimisation pour optimiser les hyper-paramètres du modèle.
- La proposition d'une autre hybridation possible de deep learning.

Bibliographie

- [1] J. Héng, "Pratique de la maintenance préventive : mécanique, pneumatique, hydraulique, électricité, froid," 2005.
- [2] R. Zemouri, *Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance*. PhD thesis, Université de Franche-Comté, 2003.
- [3] F. Monchy, "Maintenance : Méthodes et organisation» édition dunod," 2000.
- [4] F. Guérard, "Contribution à la maintenance proactive par la formalisation du processus de pronostic des performances de systèmes industriels,"
- [5] D. DASCAU, "Méthodes probabilistes pour la modélisation de la maintenance préventive. ut de compiègne," 2001.
- [6] G. Zwingelstein, "La maintenance basée sur la fiabilité : Guide pratique d'application de la rcm, hermes, coll," *Diagnostic et maintenance, Paris*, 1996.
- [7] E. Ramasso, *Méthodologies d'analyse de séries temporelles sous incertitudes aléatoires et épistémiques pour le suivi et le pronostic de l'état de systèmes et structures-De l'estimation d'une cinétique d'endommagement à son contrôle*. PhD thesis, UBFC, 2016.
- [8] K. Medjaher and N. Zerhouni, "Residual-based failure prognostic in dynamic systems," *IFAC Proceedings Volumes*, vol. 42, no. 8, pp. 716–721, 2009.
- [9] K. Medjaher, R. Gouriveau, and N. Zerhouni, "A procedure for failure prognostic in dynamic systems," *IFAC Proceedings Volumes*, vol. 42, no. 4, pp. 1597–1602, 2009.
- [10] M. Bengtsson, *Condition based maintenance systems : an investigation of technical constituents and organizational aspects*. Citeseer, 2004.
- [11] P. A. Sandborn and C. Wilkinson, "A maintenance planning and business case development model for the application of prognostics and health management (phm) to electronic systems," *Microelectronics Reliability*, vol. 47, no. 12, pp. 1889–1901, 2007.
- [12] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems—reviews, methodology and applications," *Mechanical systems and signal processing*, vol. 42, no. 1-2, pp. 314–334, 2014.
- [13] J. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mechanical systems and signal processing*, vol. 25, no. 5, pp. 1803–1836, 2011.
- [14] I. Geanta, B. Iung, D. Theilliol, M. Schieber, and Y. Fusero, "Investigation d'une cartographie multicritère pour la sélection de techniques de diagnostic et de pronostic dédiée à la maintenance en ligne de véhicules complexes," in *5e Journées Doctorales/Journées Nationales MACS, JD-JN-MACS 2013*, p. CDR0M, 2013.
- [15] M. J. Daigle and K. Goebel, "A model-based prognostics approach applied to pneumatic valves," *International journal of prognostics and health management*, vol. 2, no. 2, pp. 84–99, 2011.
- [16] H. Meng and Y.-F. Li, "A review on prognostics and health management (phm) methods of lithium-ion batteries," *Renewable and Sustainable Energy Reviews*, vol. 116, p. 109405, 2019.
- [17] K. Javed, R. Gouriveau, and N. Zerhouni, "State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels," *Mechanical Systems and Signal Processing*, vol. 94, pp. 214–236, 2017.

- [18] A. Soualhi, *Du diagnostic au pronostic de pannes des entraînements électriques*. PhD thesis, Université Claude Bernard-Lyon I, 2013.
- [19] J. W. Sheppard, M. A. Kaufman, and T. J. Wilmering, "Ieee standards for prognostics and health management," in *2008 IEEE AUTOTESTCON*, pp. 97–103, IEEE, 2008.
- [20] X.-S. Si, W. Wang, C.-H. Hu, D.-H. Zhou, and M. G. Pecht, "Remaining useful life estimation based on a nonlinear diffusion degradation process," *IEEE Transactions on reliability*, vol. 61, no. 1, pp. 50–67, 2012.
- [21] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *2008 international conference on prognostics and health management*, pp. 1–6, IEEE, 2008.
- [22] K. Medjaher, D. A. Tobon-Mejia, and N. Zerhouni, "Remaining useful life estimation of critical components with application to bearings," *IEEE Transactions on Reliability*, vol. 61, no. 2, pp. 292–302, 2012.
- [23] P. P. Shinde and S. Shah, "A review of machine learning and deep learning applications," in *2018 Fourth international conference on computing communication control and automation (ICCU-BEA)*, pp. 1–6, IEEE, 2018.
- [24] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "An overview on application of machine learning techniques in optical networks," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1383–1408, 2018.
- [25] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b*, vol. 4, pp. 51–62, 2017.
- [26] T. Dietterich, C. Bishop, D. Heckerman, M. Jordan, M. Kearns, *et al.*, "Adaptive computation and machine learning."
- [27] I. Vlahavas, A. Likas, and G. Paliouras, "Machine learning and neural networks," *Progressive*, no. 1, 2013.
- [28] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine learning : algorithms and applications*. Crc Press, 2016.
- [29] O. A. Bensiah and M. Berkane, "La proposition d'une nouvelle approche basée deep learning pour la prédiction du cancer du sein," 2020.
- [30] Y. Mercadier, *Classification automatique de textes par réseaux de neurones profonds : application au domaine de la santé*. PhD thesis, Université Montpellier, 2020.
- [31] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv :1412.3555*, 2014.
- [32] S. Hao, Z. Miao, J. Wang, W. Xu, and Q. Zhang, "Labanotation generation based on bidirectional gated recurrent units with joint and line features," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 4265–4269, IEEE, 2019.
- [33] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 37–49, JMLR Workshop and Conference Proceedings, 2012.
- [34] S. Mezzah, S. Kasmi, M. Alilouche, S. Loubar, *et al.*, *Implémentation de réseaux de neurones profonds à l'aide d'un processeur ARM Cortex-M*. PhD thesis, université Abderahmene Mira. Bejaia, 2020.
- [35] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2063–2079, 2018.
- [36] R. L. DAHAH, "La détection de la colère chez le conducteur en utilisant le deep learning," 2020.
- [37] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

- [38] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr : a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.
- [39] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system : A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.
- [40] Z. Kang, C. Catal, and B. Tekinerdogan, "Remaining useful life (rul) prediction of equipment in production lines using artificial neural networks," *Sensors*, vol. 21, no. 3, p. 932, 2021.
- [41] Y. Wang, Y. Zhao, and S. Addepalli, "Remaining useful life prediction using deep learning approaches : A review," *Procedia Manufacturing*, vol. 49, pp. 81–88, 2020.
- [42] L. Ren, Y. Sun, J. Cui, and L. Zhang, "Bearing remaining useful life prediction based on deep autoencoder and deep neural networks," *Journal of Manufacturing Systems*, vol. 48, pp. 71–77, 2018.
- [43] P. Malhotra, V. TV, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder," *arXiv preprint arXiv :1608.06154*, 2016.
- [44] R. Zemouri, Z. Al Masry, I. Remadna, S. L. Terrissa, and N. Zerhouni, "Hybrid architecture of deep convolutional variational auto-encoder for remaining useful life prediction," in *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference* (F. D. M. Piero Baraldi and E. Zio, eds.), (Venice, Italy), pp. 3591–3598, Copyright c ESREL2020-PSAM15 Organizers, Research Publishing Services, Singapore, 2017.
- [45] H. Liu, Z. Liu, W. Jia, and X. Lin, "A novel deep learning-based encoder-decoder model for remaining useful life prediction," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.
- [46] R. M. Hasani, G. Wang, and R. Grosu, "An automated auto-encoder correlation-based health-monitoring and prognostic method for machine bearings," *arXiv preprint arXiv :1703.06272*, 2017.
- [47] J. Ma, H. Su, W.-l. Zhao, and B. Liu, "Predicting the remaining useful life of an aircraft engine using a stacked sparse autoencoder with multilayer self-learning," *Complexity*, vol. 2018, 2018.
- [48] L. Liao, W. Jin, and R. Pavel, "Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7076–7083, 2016.
- [49] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2306–2318, 2016.
- [50] M. Ma, C. Sun, and X. Chen, "Discriminative deep belief networks with ant colony optimization for health status assessment of machine," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 12, pp. 3115–3125, 2017.
- [51] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, vol. 48, no. 1, pp. 11–20, 2017.
- [52] J. Deutsch, M. He, and D. He, "Remaining useful life prediction of hybrid ceramic bearings using an integrated deep learning and particle filter approach," *Applied Sciences*, vol. 7, no. 7, p. 649, 2017.
- [53] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *International conference on database systems for advanced applications*, pp. 214–228, Springer, 2016.
- [54] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5695–5705, 2018.

- [55] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Reliability engineering & system safety*, vol. 182, pp. 208–218, 2019.
- [56] T. Salunkhe, N. Jamadar, and S. Kivade, "Prediction of remaining useful life of mechanical components-a review," *International Journal of Engineering Science and Innovative Technology (IJE-SIT)*, vol. 3, no. 6, pp. 125–135, 2014.
- [57] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network," in *2016 IEEE international conference on aircraft utility systems (AUS)*, pp. 135–140, IEEE, 2016.
- [58] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, 2017.
- [59] Q. Wu, K. Ding, and B. Huang, "Approach for fault prognosis using recurrent neural network," *Journal of Intelligent Manufacturing*, vol. 31, no. 7, pp. 1621–1633, 2020.
- [60] A. Elsheikh, S. Yacout, and M.-S. Ouali, "Bidirectional handshaking lstm for remaining useful life prediction," *Neurocomputing*, vol. 323, pp. 148–156, 2019.
- [61] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional lstm networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.
- [62] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 international conference on prognostics and health management*, pp. 1–9, IEEE, 2008.
- [63] D. Mishra, "Regression : An explanation of regression metrics and what can go wrong."
- [64] C. Staudemeyer, Ralf Omlin, "Evaluating performance of long short term memory recurrent neural networks on intrusion detection data," in *CM International Conference Proceeding*, 218, 2013.
- [65] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "A multimodal and hybrid deep neural network model for remaining useful life estimation," *Computers in Industry*, vol. 108, pp. 186–196, 2019.
- [66] T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi, "An ensemble framework based on convolutional bi-directional lstm with multiple time windows for remaining useful life estimation," *Computers in Industry*, vol. 115, p. 103182, 2020.
- [67] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions : Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv :1811.03378*, 2018.
- [68] I. Remadna, S. L. Terrissa, R. Zemouri, S. Ayad, and N. Zerhouni, "Leveraging the power of the combination of cnn and bi-directional lstm networks for aircraft engine rul estimation," in *2020 Prognostics and Health Management Conference (PHM-Besançon)*, pp. 116–121, IEEE, 2020.
- [69] H. Yang, F. Zhao, G. Jiang, Z. Sun, and X. Mei, "A novel deep learning approach for machinery prognostics based on time windows," *Applied Sciences*, vol. 9, no. 22, p. 4813, 2019.
- [70] A. Al-Dulaimi, A. Asif, and A. Mohammadi, "Noisy parallel hybrid model of nbgru and ncnr architectures for remaining useful life estimation," *Quality Engineering*, vol. 32, no. 3, pp. 371–387, 2020.
- [71] C. Zhao, X. Huang, Y. Li, and M. Yousaf Iqbal, "A double-channel hybrid deep neural network based on cnn and bilstm for remaining useful life prediction," *Sensors*, vol. 20, no. 24, p. 7109, 2020.
- [72] J. Narwariya, P. Malhotra, V. TV, L. Vig, and G. Shroff, "Graph neural networks for leveraging industrial equipment structure : An application to remaining useful life estimation," *arXiv preprint arXiv :2006.16556*, 2020.

- [73] Y. Jiang, Y. Lyu, Y. Wang, and P. Wan, "Fusion network combined with bidirectional lstm network and multiscale cnn for remaining useful life estimation," in *2020 12th International Conference on Advanced Computational Intelligence (ICACI)*, pp. 620–627, IEEE, 2020.
- [74] I. Remadna, L. S. Terrissa, S. Ayad, and N. Zerhouni, "Rul estimation enhancement using hybrid deep learning methods," *International Journal of Prognostics and Health Management*, vol. 12, no. 1, 2021.
- [75] K. Yonas, *Hate speech detection for amharic language on social media using machine learning techniques*. PhD thesis, ASTU, 2019.

Bibliographie

- [1] AFNORa, Juin 2001 :« Norme X 60-319 norme européenne ». Terminologie de la maintenance.
- [2] AFNOR, Mai 2002 :«NF X60-000, Maintenance industrielle, Fonction maintenance.
- [3] <https://www.mathworks.com/discovery/machine-learning.html>
- [4] <https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d>
- [5] <https://blog.nalo.fr/lexique/r-carre-ou-r2>
- [6] <https://www.theses.fr/2018GREATo86>
- [7] <https://www.futura-sciences.com/tech/definitions/informatique-python-19349>
- [8] <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501861-tensorflow-tout-savoir-sur-la-plateforme-de-deep-learning-de-google>
- [9] <https://mobiskill.fr/blog/conseils-emploi-tech/quels-sont-les-frameworks-utilises-en-deep-learning>
- [10] <https://research.google.com/colaboratory/faq.html?hl=fr>
- [11] https://github.com/RodrigoNeves95/C-MAPSS_Problem
- [12] <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/>
- [13] <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in->