

Supervised



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre :RTIC03 /M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Réseaux et Technologies de l'Information et de la Communication (RTIC)

Développement d'une approche smart pour le PHM appliquée à l'industrie 4.0

Par :

HAMIDI HADIL

Soutenu le 01/07/2021 devant le jury composé de :

Bitam Salim

Professeur

Président

Kazar Okba

Professeur

Rapporteur

Terrissa Sadek Labib

Professeur

Examineur

Année universitaire 2020-2021

Remerciements

En premier lieu, je remercie le bon dieu de m'avoir donné la force et la patience nécessaire pour achever ce travail de thèse. Je tiens à remercier mon directeur monsieur **Okba kazar**, Professeur à l'Université de Biskra, pour son immense savoir, ses précieux conseils et son aide durant notre travail J'exprime ma reconnaissance à tous les enseignants du **Département d'Informatique** qui m'ont aidé dans mon éducation. Je tiens aussi à remercier toute ma promotion de master et tous mes collègues. Je ne peux terminer sans avoir une pensée pour ma famille. Sans elle, je n'aurais jamais pu arriver jusque-là et atteindre mes objectifs. Un grand merci à mes parents pour leur soutien indéterminé. Enfin, je remercie tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Hamidi Hadil

Abstract

In the field of Prognostics and Health Management (PHM), prognostics is a process to learn about the health status of a system in a factory 4.0 that is based especially at iot and to estimate its residual time to failure (RUL). A good maintenance decision necessarily involves a best estimation of the latter. Recently, the emergence of artificial intelligence in the industrial field and in particular deep learning Computing has strongly contributed to the improvement of the prognosis process. In this work, we propose a new prognostic approach based on the CNN model. Three data-driven prognostic methods have been implemented and tested on phm08 data. tested on phm08 data (NASA). Finally an evaluation of our solution and the results obtained.

Keywords : *Factory 4.0, Prognostics and Health Management (PHM), Residual Lifetime (RUL), CNN, Artificial Intelligence, Performance Measures.*

Résumé

Dans le domaine du Prognostics and Health Management (PHM), le pronostic est un processus permettant de se renseigner sur l'état de santé d'un système dans une usine 4.0 qui est basé spécialement à iot et d'estimer son temps résiduel avant la défaillance (RUL). Une bonne décision de maintenance passe forcément par une meilleure estimation de ce dernier. Récemment, l'émergence de l'intelligence artificiel dans le domaine industriel et en particulier l'apprentissage profond(deep learning) Computing a fortement contribué à l'amélioration du processus de pronostic. Dans ce travail, nous proposons une nouvelle approche de pronostic basé sur le modèle CNN. Trois méthodes de pronostic guidé par les données ont été implémentées et testées sur des données phm08 (NASA). Enfin une évaluation de notre solution et les résultats obtenus.

Mots clés : *Industrie 4.0 , iot, Prognostics and Health Management (PHM), Durée de vie résiduelle (RUL), CNN, Intelligence artificielle, Mesures de performance.*

Table des figures

2.1	représente l'historique de l'industrie	5
2.2	La cybersécurité est fondamentale pour protéger les systèmes automatiques dans l'industrie 4.0	7
2.3	La technologie 4.0 au service de l'industrie connectée	9
3.1	Figure : représente l'architecture de phm(7 couches)	16
3.2	point de vue phénoménologie	18
3.3	schéma de l'apprentissage supervisé	25
3.4	Schéma d'un modèle supervisé	25
3.5	Schéma d'un modèle non supervisé	26
3.6	schéma du perceptron simple	27
3.7	Schéma d'un perceptron multicouches	29
3.8	Les réseaux neuronaux récurrents ont des boucles	30
3.9	Un réseau neuronal récurrent déroulé.	31
3.10	Architecture standard d'un réseau de neurone convolutionnel.	32
4.1	l'architecture globale du système.	35
4.2	un petit exemple de la base de données.	36
4.3	base de données prétraitement.	36
4.4	architecture CNN.	37
5.1	Google Colab.	39
5.2	python.	40
5.3	TensorFlow.	40

5.4 Keras.	41
5.5 NumPy.	41
5.6 Matplotlib.	42
5.7 Le DataSet utilisé.	42
5.8 Le DataSet utilisé(suite).	43
5.9 Le résultat de RUL.	44
5.10 génération de la colonne max.	45
5.11 "run to failure" de chaque machine.	45
5.12 "time to failure" de chaque machine(test).	46
5.13 "time to failure" de chaque machine(train).	47
5.14 normalisation des données.	48
5.15 l'affichage etat normal du machine	50
5.16 "l'architecture CNN."	51
5.17 "Graphe de la variation moyenne de la précision."	52
5.18 "Graphe de la variation moyenne de la perte."	53
5.19 "Graphe de la variation moyenne de la validation de précision."	53
5.20 "Graphe de la variation moyenne de la validation de perte."	54
5.21 F1 Score.	55
5.22 matrice de confusion.	55
5.23 matrice de confusion.	55
5.24 Architecture LSTM.	56
5.25 Graphe précision et perte	56
5.26 F1 score LSTM.	57

List of source code

- 5.1 Calcul RUL 43
- 5.2 Calcul RUL 44
- 5.3 Calcul RUL 45
- 5.4 time to failure 46
- 5.5 time to failure 46
- 5.6 normalisation des données 47
- 5.7 Modèle CNN 49
- 5.8 Modèle LSTM 49
- 5.9 Compilation du modèle 50
- 5.10 Training de modèle 50

Liste des tableaux

3.1 Synthèse sur les approches de pronostic 23

Table des matières

Acknowledgements	i
Abstract	ii
Résumé	iii
List of Figures	vii
List of Tables	viii
1 General Introduction	1
1.1 General context	1
2 L'industrie 4.0	4
2.1 Introduction	4
2.2 Définition :	4
2.3 Histoire de l'industrialisation	5
2.4 L'Industrie 4.0 optimise la production	6
2.5 Les enjeux auxquels l'industrie 4.0 confronte les entreprises	6
2.5.1 L'augmentation de la compétitivité globale :	6
2.5.2 La nécessaire transformation digitale de l'industrie :	6
2.5.3 La formation et la gestion du capital humain :	7
2.5.4 Les nouveaux risques digitaux : la cybersécurité dans l'industrie 4.0 :	7
2.6 La technologie 4.0 au service de l'industrie connectée	7
2.6.1 Internet des objets ou IoT (Internet of Things)	8
2.6.2 Big data, intelligence artificielle (IA) et machine learning	8
2.6.3 Réalité augmentée	8

2.6.4	Jumeaux numériques	8
2.6.5	Chaînes de blocs ou blockchain	8
2.7	Comment fonctionne une usine 4.0?	9
2.7.1	L'usine est virtualisée	9
2.7.2	Les systèmes sont interopérables	9
2.7.3	Les décisions sont décentralisées	9
2.7.4	L'analyse et la prise de décision	10
2.7.5	Elle est orientée service	10
2.7.6	Elle est modulaire	10
2.8	Comment la mettre en oeuvre?	10
2.9	Les approches de base de l'industrie 4.0	11
2.10	Des exemples d'usine appliquées l'industrie 4.0	11
2.11	Conclusion	13
3	Prognostics and Health Management (PHM) et L'apprentissage profond	14
3.1	Introduction	14
3.2	Prognostics and Health Management (PHM)	14
3.2.1	Définition	14
3.2.2	l'historique de phm :	15
3.2.3	l'architecture de phm	16
3.2.4	Liens entre les modules de PHM	18
3.2.5	Le rôle du pronostic dans le processus de durée de vie	19
3.2.6	Formulation du problème	20
3.2.7	Les approches de pronostic :	21
3.2.8	Synthèse sur les approches de pronostic	23
3.3	L'apprentissage profond (Deep learning)	24
3.3.1	Définition de l'apprentissage profond (deep learning)	24
3.3.2	Apprentissage automatique	24
3.3.3	Réseaux de neurones (RN)	26
3.3.4	Les réseaux de neurones récurrents (RNN ou Recurrent Neural Networks)	29

3.3.5	la mémoire à long terme (LSTM)	31
3.3.6	Les réseaux de neurones convolutionnels (CNN ou Convolutional Neural Networks)	31
3.3.7	Domaines d'applications de Deep Learning	32
3.3.8	l'apprentissage profond dans phm	33
3.4	Conclusion	34
4	la conception	35
4.1	Introduction	35
4.2	L'architecture globale de système	35
4.3	l'architecture détaillées du système	36
4.3.1	Base de donnée	36
4.3.2	les capteurs	37
4.3.3	deep learning	37
4.4	Conclusion	38
5	Développement détaillée du modèle	39
5.1	Introduction	39
5.2	Environnements et outils de développement	39
5.2.1	Google Colab	39
5.2.2	Python	40
5.2.3	TensorFlow	40
5.2.4	Keras	41
5.2.5	Numpy	41
5.2.6	Matplotlib	42
5.3	Travail effectué	42
5.3.1	Construction de la base d'apprentissage	42
5.4	Évaluation empirique et résultats	51
5.4.1	Les performances du CNN	52
5.5	Evaluation empirique er résultat du modèle LSTM	56
5.5.1	les performances du LSTM	56

<i>TABLE DES MATIÈRES</i>	0
5.6 Conclusion	57
6 Conclusion and Perspectives	58
6.1 Conclusion	58
6.2 Perspectives	59
References	60

Chapitre 1

General Introduction

1.1 General context

Dans le domaine industriel, la maîtrise des coûts de production tout en assurant un niveau de qualité désiré constitue un défi clé dans ce domaine. L'un des objectifs majeurs des industriels est de minimiser au maximum les coûts élevés induits par les indisponibilités non planifiées et les pannes des équipements qui peuvent générer des pertes considérables. Aussi l'exigence de compétitivité nécessite l'amélioration de la disponibilité, de la fiabilité et de la sécurité tout en réduisant les coûts du cycle de vie des moyens de production. La maintenance industrielle, qui est l'ensemble de toutes les actions techniques, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise, représente la solution adéquate à ces défis.

Malheureusement, la maintenance traditionnelle, basée principalement sur le diagnostic après-panne, présente beaucoup d'inconvénients qu'on peut résumer dans les points suivants :

- Arrêt de la production pendant une période indéfinie,
- Remplacement inutile de pièces de rechange,
- Coût de maintenance très élevé,
- Dégâts matériels et humains imprévus.

On peut remédier à ces inconvénients en passant à un autre type de maintenance qui est la maintenance prédictive. Cette maintenance est basée principalement sur le pronostic qui est l'estimation du temps de fonctionnement avant défaillance d'un équipement ou d'un système.

Le Prognostics and Health Management PHM permet de garantir la qualité des produits,

d'effectuer la maintenance juste à temps, de réduire les temps d'arrêt des équipements et d'éviter les pannes catastrophiques avec ces avantages.

Le pronostic des systèmes industriels devient actuellement un objectif important pour les industriels, sachant que la panne, qui peut survenir soudainement, est généralement très coûteuse en termes de réparation et d'interruption de la production et est mauvais à la réputation. L'une des principales approches du pronostic est l'approche guidée par les données, qui offre l'avantage de pouvoir apprendre des modèles basés sur des données empiriques et d'utiliser des méthodes d'intelligence artificielle (CNN). La durée de vie utile restante (RUL) de l'actif surveillé est le résultat du pronostic et est utilisée dans l'évaluation pronostique en appliquant des mesures appropriées et des critères supplémentaires. Il existe un large éventail de méthodes qui traitent le calcul du RUL.

Nous proposons, dans ce travail, un système de pronostic basé sur l'apprentissage profond. De plus, pour tester l'efficacité de notre solution, nous avons étudié la performance de notre système pronostic. Le mémoire est organisé en quatre chapitres

Le premier chapitre est contient la nouvelle technologie l'industrie 4.0 et sa importance dans nos jours à cause de sa objectif de prendre des décisions plus rapidement (en temps réel). alors nous présentons comment l'industrie 4.0 optimise la production et les nouvelles enjeux ensuite le fonctionnement d'une usine 4.0 aussi les approches de base de l'industrie 4.0 . finalement comment la mettre en oeuvre.

Dans le deuxième chapitre consiste à présenter le pronostic and health management (phm) et sa architecture et le plus important le RUL ensuite nous avons montré les approches de pronostic. Une deuxième partie dans ce chapitre présente l'apprentissage profond sa définition aussi l'apprentissage automatique ensuite les réseaux de neurones. De plus il présente les architectures de l'apprentissage profond RNN CNN LSTM . Ce chapitre se termine par l'apprentissage profond dans phm.

Le troisième chapitre montre l'architecture générale du système et le fonctionnement global de notre système et finalement l'architecture détaillée.

Le quatrième chapitre présente l'implémentation de l'approche en commençant par la présentation des outils et l'environnement de développement. Ensuite, nous allons présenter quelques interfaces qui montrent les résultats obtenus depuis l'estimation du RUL les calculs nécessaires

et enfin l'évaluation empirique et résultats.

Dans une conclusion générale, nous repositionnons l'ensemble de nos développements au regard de l'objectif initial de l'étude. Nous résumons les principaux résultats de notre travail.

Chapitre 2

l'industrie 4.0

2.1 Introduction

Grâce aux percées dans les domaines des technologies de l'information, des communications mobiles et de la robotique, les technologies numériques sont de plus en plus utilisées dans les usines du monde entier. Cette transformation, appelée «industrie 4.0» ou «quatrième révolution industrielle», succède aux bouleversements déclenchés par l'introduction de la machine à vapeur, de la chaîne de montage et de l'automatisation (développement de l'électronique).

2.2 Définition :

Le projet « Industrie 4.0 » vise à développer de nouvelles organisations de production sur toute la chaîne de valeur. « L'Industrie 4.0 » est annoncée comme la 4ème révolution industrielle. L'industrie 4.0 ou quatrième révolution industrielle fait référence à la transformation de l'industrie et des systèmes de production grâce à l'introduction des nouvelles technologies comme l'Internet des objets connectés et le cloud computing, pour fabriquer des produits grâce à des systèmes intelligents, tels que les systèmes de simulation et les capteurs. L'objectif de cette industrie est de prendre des décisions plus rapidement, presque en temps réel. En connectant les différentes ressources de l'entreprise entrent-elles, il est plus facile de déterminer les perturbations possibles des opérations, et ainsi de prendre une décision ou de poser une action pour corriger la situation. Les processus deviennent alors réactifs.[17]

2.3 Histoire de l'industrialisation

La révolution industrielle 1.0 : débute vers 1780 au Royaume-Uni, alors que la mécanisation fait basculer une société autrefois majoritairement agraire et artisanale vers une société commerciale et industrielle. Celle-ci repose sur l'exploitation du charbon avec l'invention de la machine à vapeur.

La révolution industrielle 2.0 : est amenée par l'introduction de l'électricité, de la mécanique et du développement du transport vers 1870. Cette deuxième révolution, l'industrie 2.0, se caractérise par une production de masse grâce aux chaînes de montage.

La révolution industrielle 3.0 : cette fois grâce à l'électronique, aux télécommunications et à l'informatique. Il existe alors une convergence de la production industrielle avec les technologies de l'information et de la communication. On introduit également le concept d'automatisation grâce à l'électronique et à l'informatique. Vers 1980.

une nouvelle révolution industrielle 4.0 amorce : c'est l'avènement de la mondialisation. Certains considèrent cette phase comme une industrie 3.5, puisqu'elle amène son lot de changements. La production est transférée vers des pays à faibles coûts grâce à une réduction des coûts de communication et de conteneurisation. Aujourd'hui, on parle de l'industrie 4.0, caractérisée principalement par l'introduction des technologies numériques à la production.

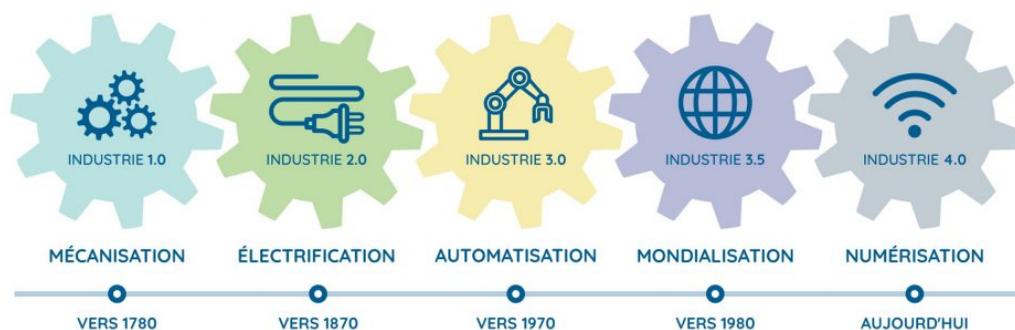


FIGURE 2.1 – représente l'historique de l'industrie

2.4 L'Industrie 4.0 optimise la production

Avec l'industrie 4.0, il devient possible de mettre au point des produits qui répondent parfaitement aux attentes du client. En parvenant à produire du premier coup des marchandises débarrassées de leurs éventuelles anomalies en amont, l'industrie 4.0 réduit la non-qualité (causant en moyenne 5 pourcent de perte de CA en 2017 selon l'AFNOR). Elle augmente la satisfaction client qui ne doit plus les renvoyer pour cause d'imperfection. Et ainsi, l'entreprise améliore ainsi sa performance globale.[3]

2.5 Les enjeux auxquels l'industrie 4.0 confronte les entreprises

Le processus de transformation vers l'industrie 4.0 confronte les entreprises à de nouveaux défis liés avec [10]

2.5.1 L'augmentation de la compétitivité globale :

Les entreprises doivent être en mesure d'ajuster leurs processus de production aux changements rapides de la demande tout en maximisant leur productivité. Pour cela, il est essentiel d'implémenter rapidement les progrès technologiques. Selon le rapport Digital Factories 2020 de PwC, 98 pourcent des entrepreneurs interrogés sont d'accord pour dire que la recherche d'efficacité en production est la raison principale de la digitalisation des usines.

2.5.2 La nécessaire transformation digitale de l'industrie :

Les investissements en RDi dans les processus logistiques sont devenus un atout essentiel pour la croissance économique à long terme d'une multitude d'entreprises et, dans de nombreux cas, leur survie en dépend. L'innovation dans l'environnement de l'industrie 4.0 se traduit par :

- la création de nouveaux produits pour une société digitalisée.
- l'adoption de systèmes de production avancés et intelligents (smart factories).
- la gestion correcte d'une Supply Chain complexe et globalisée.

2.5.3 La formation et la gestion du capital humain :

L'industrie 4.0 génère une demande de travailleurs ayant acquis de nouvelles aptitudes : le dénommé talent 4.0 qui munira les entreprises des connaissances nécessaires pour implanter des solutions technologiques dans leurs procédures.

2.5.4 Les nouveaux risques digitaux : la cybersécurité dans l'industrie 4.0 :

Le principal défi pour les plans de cybersécurité dans l'industrie 4.0 est la vitesse à laquelle évolue la nature des attaques informatiques. Vol d'informations, usurpation d'identité, blocage de systèmes informatiques... les menaces sont très diverses et peuvent devenir critiques dans le contexte de l'industrie connectée 4.0.

La cybersécurité est fondamentale pour protéger les systèmes automatiques dans l'industrie 4.0



FIGURE 2.2 – La cybersécurité est fondamentale pour protéger les systèmes automatiques dans l'industrie 4.0

2.6 La technologie 4.0 au service de l'industrie connectée

L'essor de la quatrième révolution industrielle ne peut être comprise sans mentionner les principales technologies qui la rendent possible : [10]

2.6.1 Internet des objets ou IoT (Internet of Things)

les machines industrielles ont maintenant la capacité de communiquer et de partager en temps réel les informations recueillies par de multiples capteurs. Cela permet une analyse plus approfondie de l'ensemble du système et une réponse immédiate à toute éventualité.

2.6.2 Big data, intelligence artificielle (IA) et machine learning

ces trois disciplines sont étroitement liées à la collecte et à la gestion d'un énorme volume de données de sources différentes (big data). Grâce au machine learning, l'IA permet aux ordinateurs d'apprendre et d'améliorer les processus par eux-mêmes, en se basant sur un historique de données et sur la répétition des opérations.

2.6.3 Réalité augmentée

elle permet d'ajouter des couches d'informations virtuelles à des environnements réels. Par le biais de dispositifs électroniques, comme des lunettes ou des écrans, les utilisateurs accèdent à une réalité mixte où se superposent des éléments digitaux à des éléments réels.

2.6.4 Jumeaux numériques

cette technologie part de données du monde physique pour recréer un scénario virtuel. Cette procédure permet d'étudier un problème ou de vérifier le fonctionnement d'un système de manière numérique pour ensuite appliquer la solution dans le monde réel.

2.6.5 Chaînes de blocs ou blockchain

il s'agit d'une nouvelle forme de structuration de la transmission de données, distribuées en blocs cryptés liés entre eux et identifiés avec des chiffres uniques et irrépétibles. Basé sur des algorithmes mathématiques complexes, ce système garantit l'inviolabilité des informations car il est en mesure de détecter et de refuser tout changement non autorisé.

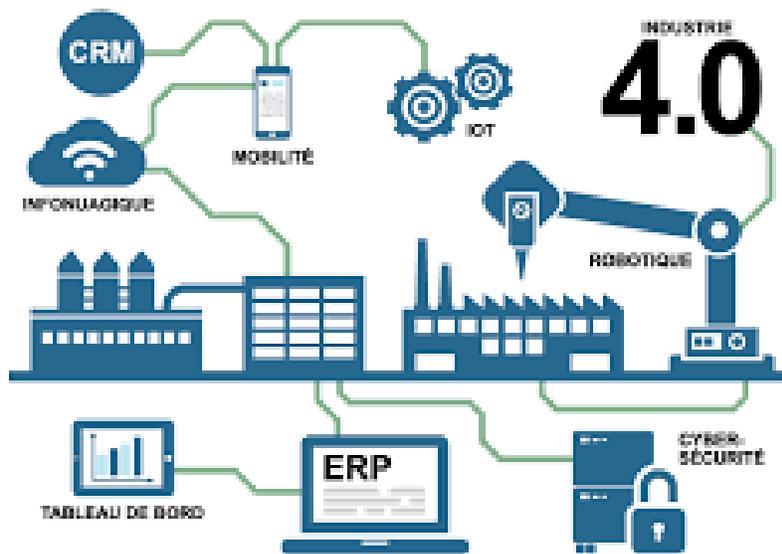


FIGURE 2.3 – La technologie 4.0 au service de l’industrie connectée

2.7 Comment fonctionne une usine 4.0?

Une usine 4.0 fonctionne selon six principes clés :[\[17\]](#)

2.7.1 L’usine est virtualisée

afin de pouvoir simuler et suivre en 3D les produits, les processus et l’environnement de production

2.7.2 Les systèmes sont interopérables

ils ont la capacité de communiquer et d’interagir entre eux.

2.7.3 Les décisions sont décentralisées

les systèmes cyber-physiques peuvent prendre des décisions de façon autonome.

2.7.4 L'analyse et la prise de décision

s'effectuent en temps réel, grâce à une communication permanente et instantanée.

2.7.5 Elle est orientée service

maintenance améliorée et offre de nouveaux services.

2.7.6 Elle est modulaire

elle s'adapte rapidement à une demande changeante.

2.8 Comment la mettre en oeuvre?

Alors pour être sur la ligne de départ de cette révolution il faut quoi? il est essentiel de disposer d'un modèle virtuel 3D produit / processus complet et cohérent, entièrement numérisé et partagé entre tous les métiers, les partenaires et les fournisseurs. Donc ce modèle virtuel gère l'ensemble des informations, sur tout le cycle de vie du produit. Il est aussi permet la simulation du comportement produit, du système de production, de sa maintenance et des services associés. En suite au cœur de ce modèle sont situés : Le PLM (Product Life cycle Management) l'objectif de PLM et d'assurer la définition, la gestion et l'accessibilité des informations produits et processus en support de la conception. Le MES (Manufacturing Execution System) il garantit la descente et la remontée d'informations en temps réel en support de la production /La réalité augmentée qui permet une assistance pour le contrôle qualité, la maintenance, la réparation ou la formation. L'intégration des solutions PLM/MES/réalité augmentée est un prérequis pour la continuité des modèles virtuels, de la conception à la production. La mise en oeuvre d'une usine 4.0 est un projet qui commence par la définition de la stratégie d'entreprise et le partage de celle-ci avec tous les métiers. Un tel projet nécessite en effet un effort collectif significatif qui passe par une vision partagée des enjeux et de la transformation . Pour aborder cette transformation et assurer le changement, une réflexion approfondie en amont, qui prenne en compte toutes les dimensions (technologique, organisationnelle, humaine et financière) est nécessaire. L'usine 4.0 a besoin d'outils et d'innovations technologiques, mais également d'ergonomie pour

proposer aux différents acteurs internes et externes, un environnement qui puisse les attirer. Pour construire un modèle compétitif et attractif, au-delà de l'aspect technologique, c'est l'aspect humain qu'il ne faut pas négliger.[17]

2.9 Les approches de base de l'industrie 4.0

Il existe 2 approches de base qui forment l'industrie 4.0 sont l'IOT et CPS .

1- l'IOT c'est l'Internet Industriel des Objets : Afin d'être opérationnelles, les technologies de l'industrie 4.0 ont avant tout besoin de données. Celles-ci permettent d'automatiser certaines tâches et d'aider à la prise de décisions. Elles proviennent principalement des capteurs IoT présents sur les lignes de production ou des informations transmises via l'IoT par les opérateurs. Ces données, récupérées dans l'environnement physique, sont ensuite envoyées sur une plateforme web puis traitées. Elles facilitent ainsi la prise de décision, notamment dans les changements de process.[1]

2- CPS c'est système cyber-physique de production est formé par un ou plusieurs systèmes embarqués dans des équipements industriels, systèmes qui communiquent les uns avec les autres. La commande et la régulation du système de production ne passe plus par une entité centrale, mais se trouve décentralisée au niveau des interactions entre l'ensemble des systèmes cyber-physiques.[15]

2.10 Des exemples d'usine appliquées l'industrie 4.0

Seat

SEAT est un point de repère international de l'Industrie 4.0. Déjà en 2018, le Financial Times identifiait cette entreprise comme leader européen de la transformation digitale. C'est le résultat quand on accorde la priorité à la digitalisation dans sa stratégie d'entreprise.

Cela fait déjà des années que l'entreprise automobile espagnole met en œuvre l'intelligence artificielle, réalité virtuelle, robots collaboratifs et big data, dans l'usine de Martorell. Cela a permis à l'entreprise d'être plus efficace et flexible. Autre secteur dans lequel elle travaille, c'est un laboratoire de biomécanique, où les équipes développent des postes de travail plus ergonomiques.

Grâce à des appareils qui analysent en 3D les caractéristiques musculosquelettiques des employés, ils peuvent prévenir des blessures dérivées du travail lors de la production et améliorer la réhabilitation dans le cas où se produisent des lésions.

General Electric

GE (General Electric) est la société qui est en train de transformer l'industrie grâce à des machines définies avec softwares et solutions connectées, et qui agit dans des secteurs clés comme celui de l'énergie, de la santé et du transport. Comme un des acteurs principaux dans le secteur d'Internet des Objets, GE a lancé Predix. Il s'agit d'une plateforme comme service (PaaS) basé dans le cloud qui facilite l'analyse à niveau industriel de la production et de l'optimisation des opérations. De l'autre côté, pour le secteur de l'énergie éolienne, GE a créé la technologie Digital Windfarm. Grâce à l'emploi des capteurs, réseaux des données et analyses, il crée des turbines personnalisées afin d'assurer l'efficacité maximale, en obtenant 20

Siemens

Siemens est un des grands points de référence de l'Industrie 4.0, bien comme prestataire de services pour d'autres entreprises que pour ses usines. Un des cas les plus remarquables est Siemens Electronic Works Amberg (EWA), une usine que la compagnie possède à Amberg (Allemagne).

Elle y fabrique divers produits, dont des automates programmables ou des PLC Simatic, dont elle produit environ 17 millions d'unités chaque année.

Ce sont des dispositifs utilisés pour contrôler usines, machines et automatiser la production industrielle. Cependant, la chose remarquable est que la production de l'usine est contrôlée par 2.800 pièces Simatic.

Cela permet de faire fonctionner la production sous forme automatisée, contrôlée par des machines et robots à 75

Airbus

Terminons la revue des exemples de l'industrie 4.0 en sillonnant le ciel. Airbus, point de référence dans la fabrication d'aéronefs, voulait diminuer la complexité de l'assemblage des sièges de cabine et, ainsi, diminuer le temps nécessaire pour terminer le travail. La solution fut d'utiliser des wearables pour améliorer la productivité et, en outre, réduire les connaissances nécessaires pour effectuer le travail. Pour ce faire, ils ont conçu une application pour des lunettes

intelligentes, qui montrent au travailleur les directives afin qu'il puisse marquer le sol plus rapidement et réduire les éventuelles erreurs.

Cette application offre aussi d'autres fonctionnalités comme la lecture de codes barres, récupérer des données cloud ou de réalité augmentée, grâce à lesquelles les positions des sièges des avions peuvent être indiqués au millimètre et en vérifier précision et qualité. Grâce à cette amélioration, le pourcentage de faute se réduit à zéro, Airbus a augmenté la productivité de 500 Schneider Electric

Schneider Electric est un point de repère global dans la gestion et automatisation d'énergie. Sur le plan de l'Industrie 4.0, il s'est associé avec le consultant Accenture pour créer Digital Services Factory (DSF) ou usine de services digitaux, un programme pour encourager l'innovation digitale et accélérer la commercialisation de nouveaux produits. Avec la mise en oeuvre de Digital Services Factory, Accenture aide Schneider Electric à réduire de 80% les coûts de structure. EcoStruxure est la plateforme pour IoT de Schneider Electric. Cela promeut le progrès d'internet des objets, la mobilité, l'analyse et la cybersécurité pour innover dans tous les niveaux, dès les produits connectés, aux applications et services. De cette manière, grâce à la combinaison de l'analyse en temps réel des technologies et des solutions connectées via IoT, Schneider peut réagir mieux ou anticiper les nécessités des clients. [1]

2.11 Conclusion

Dans ce chapitre on a présenté l'industrie 4.0 et montré l'importance de cette technologie dans plusieurs domaines. Le prochain chapitre, traite le concept de phm (pronostic and health management) et l'apprentissage profond.

Chapitre 3

Prognostics and Health Management (PHM) et L'apprentissage profond

3.1 Introduction

Dans ce chapitre, nous présentons l'évolution de la maintenance afin d'introduire le concept de pronostic et du PHM (Prognostics and Health Management) et le rôle de l'apprentissage profond au cours du cycle de vie du système. Il développe également l'état de l'art des approches de pronostic : l'approche basée sur le modèle physique, l'approche guidée par les données et l'approche hybride. Cet état de la technique ouvre la voie au travail actuel et à la contribution à ce domaine.

3.2 Prognostics and Health Management (PHM)

3.2.1 Définition

Le PHM (le pronostic et la gestion de santé en français) est une discipline focalisée sur la détection, la prédiction et le management de l'état de santé des systèmes complexes. 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. 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 mainte-

nance industrielle de manière adéquate.[14]

3.2.2 l'historique de phm :

Initialement, le PHM a commencé à réduire le taux d'accidents d'hélicoptères par l'Autorité de l'aviation civile du Royaume-Uni dans les années 1980 et a été développé en 1990 sur la base du système de surveillance HUMS qui mesure les conditions sanitaires et les performances de l'hélicoptère. HUMS a obtenu de bons résultats pour réduire le taux d'accidents de plus de moitié grâce à un hélicoptère en service. Au cours des années 90, le concept de surveillance de la santé des véhicules a été adopté par la recherche aérospatiale de la NASA aux États-Unis, qui consiste à surveiller la santé des véhicules spatiaux. Toutefois, il a rapidement été remplacé par un terme plus universel, Gestion intégrée de la santé des véhicules ou Gestion de la santé du système, pour intégrer les pronostics de divers systèmes spatiaux [12]. Dans les années 2000, la DARPA (Defense Advanced Research Projects Agency) aux États-Unis a mis au point le système de pronostic d'intégrité structurelle (SIPS) et la maintenance conditionnelle plus (CBM+), qui ont le même objectif. Le nom Prognostics and Health Management (PHM) a été adopté pour la première fois dans le programme de développement des avions de combat interarmées (JSF) (Joint Strike Fighter Program Office 2016). La technologie PHM a considérablement évolué sous différents aspects, notamment l'étude fondamentale de la physique des défaillances, le développement de capteurs, l'extraction de caractéristiques, le diagnostic pour la détection et la classification des pannes et les pronostics pour la prédiction des défaillances. Ces techniques ont été explorées et étendues à diverses industries. À mesure que les technologies sont appliquées et mûries dans l'industrie, de plus en plus d'articles abordent les applications réussies sous différents aspects [13, 14]. Depuis quelques années, des sociétés techniques ont été créées pour rassembler et améliorer les connaissances dans de nombreux domaines de recherche. En tant que représentants, la société PHM organise la conférence annuelle et publie le Journal international de pronostic et de gestion de la santé (IJPHM) depuis sa création en 2009 (PHM Society 2009). En outre, la société de fiabilité de l'IEEE organise chaque année une conférence sur le PHM depuis 2011 (IEEE Reliability Society 2011).

3.2.3 l'architecture de phm

Une vision globale de cette architecture PHM est donnée par la Figure 1.2 . Cette architecture est constituée de 7 couches fonctionnelles

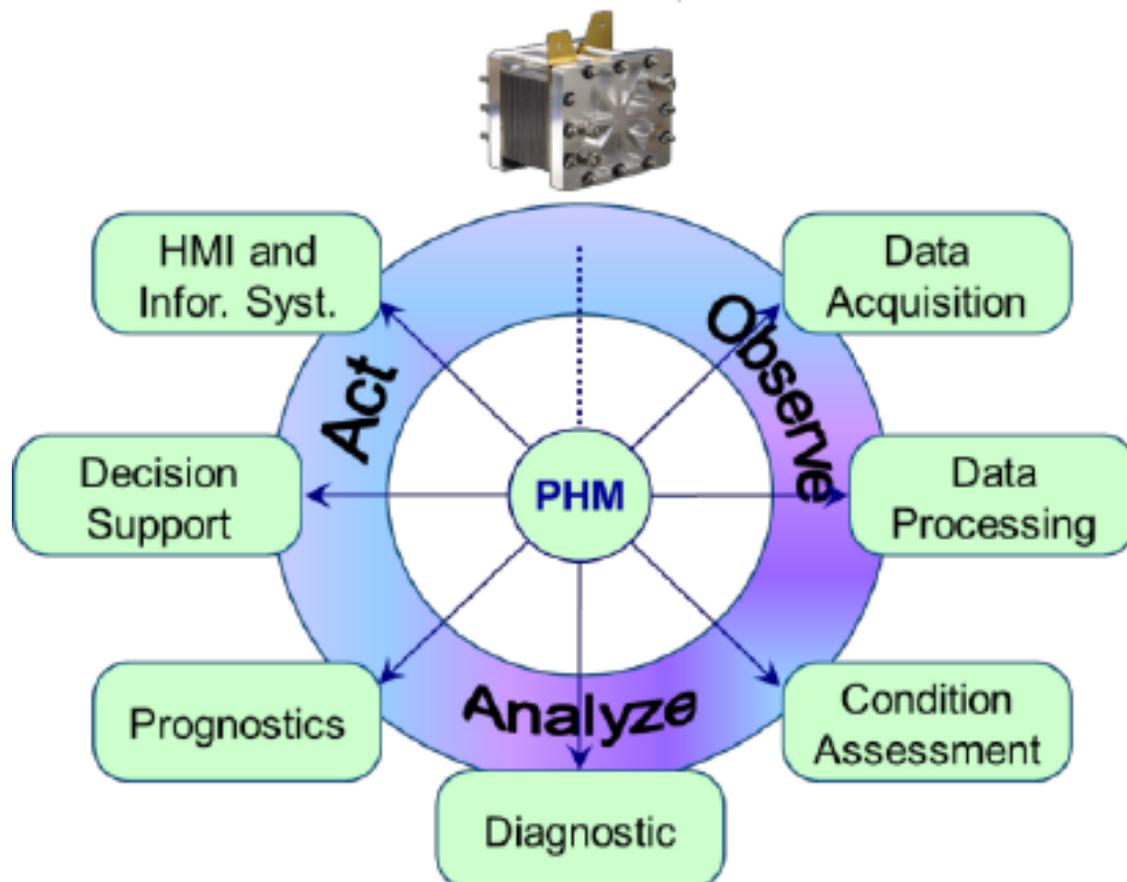


FIGURE 3.1 – Figure : représente l'architecture de phm(7 couches)

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).

La connaissance nécessaire a 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 [18] [19]. 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 [20].

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 [20].

3.2.4 Liens entre les modules de PHM

Les différents modules du PHM sont liés de manière complémentaire qui permet de maintenir le système en condition opérationnelle. Nous commençons en général par collecter des données concernant l'état de santé du système. Ces données sont ensuite traitées afin d'extraire des indicateurs de santé représentant le comportement du système. Ces indicateurs sont utilisés dans l'étape suivante pour détecter son état actuel. La détection peut conduire soit au diagnostic pour isoler une défaillance et identifier ses causes, soit au pronostic pour anticiper l'apparition d'une défaillance. Les résultats du diagnostic ou du pronostic permettent de décider et de programmer les interventions à faire. Le positionnement relatif entre détection, diagnostic et pronostic est présenté

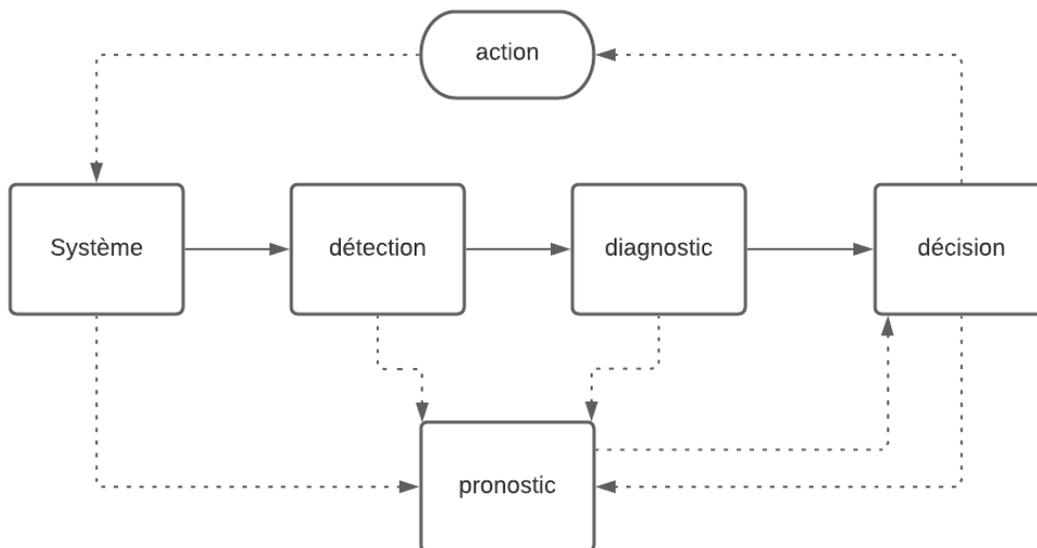


FIGURE 3.2 – point de vue phénoménologie

par la figure d'un point de vue phénoménologie

- la détection vise à identifier le mode de fonctionnement du système et son état de santé ;
- lorsqu'une défaillance est apparue, le diagnostic permet d'isoler et d'identifier le composant qui a cessé de fonctionner (des effets vers les causes) ;
- le pronostic vise la prédiction des états futurs du système (des causes vers les effets).

Parmi les modules du processus PHM, le pronostic est considéré comme le module clé qui doit

être implanté de manière efficace pour permettre de prendre les bonnes décisions afin d'améliorer la disponibilité et la sécurité, de réduire le temps d'arrêt et d'assurer la continuité de fonctionnement et l'efficacité de production.

3.2.5 Le rôle du pronostic dans le processus de durée de vie

Le pronostic est un processus englobant une capacité de prédiction. C'est la capacité à estimer la durée de vie utile restante (RUL) de l'équipement en termes de son historique de fonctionnement et de son utilisation future. Prédire la RUL des systèmes industriels devient actuellement un objectif important pour les industriels sachant que la panne, dont les conséquences sont généralement très coûteuses, peut se produire soudainement.

Les stratégies classiques de maintenance [31] ne sont pas plus efficaces et pratiques car elles ne prennent pas en compte l'état de fonctionnement instantané du produit. Adopter une maintenance systématique préventive par un remplacement fréquent pour augmenter la disponibilité du système est une stratégie coûteuse .

L'introduction d'une approche pronostique en tant que maintenance «intelligente» consiste en l'analyse, la surveillance et le suivi de la santé, basés sur des mesures physiques à l'aide de capteurs. Le RUL d'un système en service peut s'exprimer en heures de fonctionnement, en Kilomètres ou en cycles. Si nous pouvons prédire efficacement l'état des machines et des systèmes, des actions de maintenance peuvent être prises à l'avance. Un pronostic bon et fiable nécessite un diagnostic correct et fiable . La science et la technologie du PHM offrent le potentiel d'améliorer de manière significative la sécurité, la fiabilité et la disponibilité des ressources de grande valeur. Ce concept repose sur un processus en boucle fermée dont la mise en œuvre réussie dépend de l'intégration de plusieurs éléments multidisciplinaires, notamment :

- Détection à bord des paramètres opérationnels et de l'état des dommages matériels;
- Diagnostiquer les tendances, les défaillances et les dommages sous-jacents;
- Prédire la durée de vie utile restante en termes de probabilité de défaillance et de limites de performances fiables,
- Et décider des actions appropriées : chaque fois que la ressource est capable de réaliser une mission donnée ou bien a besoin d'être inspectée, entretenue ou remplacée. [22]

3.2.6 Formulation du problème

Considérant qu'il existe N machines du même type, comme le turboréacteur, chaque moteur contient des données à la fin de vie de la machine (cycles) collectés par plusieurs capteurs. Mathématiquement, l'ensemble des données peut être défini comme :

$$Dataset = (X^i, Y^i), i = 1, 2, 3, \dots, N(1)$$

Ainsi, X désigne la matrice de mesures des capteurs rassemblés d'un moteur dans laquelle Y correspond aux cycles de fonctionnement de l'équipement, comme indiqué dans l'Eq.2 et l'Eq.3, respectivement.

$$X^i = [x_1, x_2, x_t, \dots, x_{T_i}] \in R^{m \times T_i(2)}$$

$$Y^i = [y_1, y_2, y_t, \dots, y_{T_i}] \in R^{1 \times T_i(3)}$$

Où T_i est le nombre total de cycles de fonctionnement du i ème moteur et

$$x_t = [x_t^1, x_t^2, \dots, x_t^m] \in R^{m \times 1}$$

est un vecteur à m dimensions des mesures du capteur à l'instant t.

Le RUL peut être calculé entre l'heure actuelle (yt) après la détection de la dégradation (td) et l'heure de défaillance (T). Le RUL peut être décrit comme suit.

$$RUL_t^i = T_i - y^i, t = 1, 2, \dots, T_i(4)$$

Pour traiter la fonction de non-linéarité, des méthodes d'apprentissage en profondeur sont proposées (') dans cet article. Laissez de noter son entrée, et la RUL observée est sa sortie.

$$RUL^i = \varphi(x^i, RUL^i)(5)$$

Pour minimiser l'erreur entre la valeur RUL prédite et la RUL cible observée à l'instant t .

$$\text{Minimiser : } RUL_t^i, RUL_t^i(6)$$

[16]

3.2.7 Les approches de pronostic :

Diverses approches du pronostic ont été développées. Mettre au travail un processus de pronostic consiste à exécuter un ensemble de traitements à partir des informations saisies. Les différentes approches du pronostic sont regroupées en fonction de leur applicabilité et de leur rendement économique. Ce sont trois familles [21] :

- Une approche basée sur le modèle physique.
- Une approche guidée par les données.
- Une approche basée sur l'expérience

Approches basées sur un modèle physique Les approches pronostiques les plus adaptées utilisent les modèles physiques de l'échec du système sous observation, dérivés des premiers principes. Le principal domaine d'application de telles approches, à ce jour, a impliqué l'utilisation de modèles de fatigue pour la modélisation de l'initiation et la propagation des fissures dans les composants structurels.

Approches guidées par les données Dans de nombreuses situations, la complexité des systèmes sous observation rend impossible de dériver des modèles robustes et précis qui peuvent être utilisés à des fins de pronostic. Cependant, il arrive souvent que des données historiques, qui capturent le comportement du signal, des signaux mesurés ou des fonctions extraites de l'étape de défaut infantile à l'échec de l'équipement, sont disponibles. Dans de tels cas, des méthodes basées sur les données qui modèlent comment ces signaux et ces caractéristiques évoluent peuvent être utilisées pour générer des prédictions de RUL.

Les approches pronostiques basées sur les données suivent généralement l'une des deux stratégies. La première stratégie est un processus en deux étapes. Tout d'abord, une réduction appropriée de la dimensionnalité, une extraction de caractéristiques ou des techniques de correspondance de motifs sont utilisées pour cartographier les signaux ou les caractéristiques du

ystème sur une dégradation ou un indice de santé de dimension unique. Techniquement, cette première étape relève du domaine des diagnostics de faute car elle concerne l'analyse d'événements postérieure. Une fois que le niveau actuel de dégradation est identifié, il est ensuite extrapolé dans le futur jusqu'à ce qu'une limite de seuil critique préalable soit dépassée. Une gamme de techniques peut être appliquée dans ces deux étapes. La deuxième stratégie consiste à modéliser directement la relation entre les signaux ou les caractéristiques surveillés et la durée de vie restante du système.

Le pronostic basé sur l'expérience est basé sur la formalisation des mécanismes physiques de détérioration des composants par modèles stochastiques (loi de fiabilité, processus markoviens ou non-markoviens) initiés par connaissances a priori et jugement d'expert.

3.2.8 Synthèse sur les approches de pronostic

TABLE 3.1 – Synthèse sur les approches de pronostic

	Basée sur les Modèles	Guidées par les Données	Basées sur l'expérience
Modèle du système	Nécessaire	Utile	Pas nécessaire
Historique des défaillances	Utile	Pas nécessaire	Nécessaire precipitation
Conditions passées	Nécessaire	Pas nécessaire	Utile
Conditions actuelles	Nécessaire	Nécessaire	Utile
Méthodes de reconnaissance des défaillances	Nécessaire	Nécessaire	Pas nécessaire
Historique des défaillances	Utile	Pas nécessaire	Utile
Général	Capteurs et modèle	Capteurs, pas de Modèl	Pas de capteur, pas de modèle

3.3 L'apprentissage profond (Deep learning)

3.3.1 Définition de l'apprentissage profond (deep learning)

L'apprentissage profond (« deep learning ») est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années.

Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données. L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », composés de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite. Il existe différents algorithmes de Deep Learning. Nous pouvons ainsi citer :

3.3.2 Apprentissage automatique

-L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.[13]

Apprentissage Supervisé : Cette approche a pour objectif la conception d'un modèle reliant des données d'apprentissage à un ensemble de valeurs de sortie (un comportement). Un expert est employé pour étiqueter correctement des exemples. L'apprenant doit alors trouver ou approximer la fonction qui permet d'affecter la bonne étiquette à ces exemples. [13]

La Figure représente le schéma de l'apprentissage supervisé, le processus se déroule en deux phrases.

La première est appelée la phase d'apprentissage durant laquelle le professeur fournit une étiquette à chaque exemple disponible dans l'environnement en les associant à une classe : x_1, x_2, \dots, x_n et x_0 dans le cas présenté ici. Ensuite il classe les exemples en élaborant un échantillon de données étiquetées noté : $S_m = (x_1, U_1), (x_2, U_2), \dots, (x_m, U_m)$. Il guide ainsi l'apprenant en lui

fournissant des exemples sur ce qu'il doit apprendre à savoir faire.

La seconde étape est nommée phase de test. Elle consiste à tenter de prédire l'étiquette d'un nouvel exemple en utilisant une fonction apprise à partir de S_m notée h . Avec cette hypothèse, l'apprenant peut associer $Y_n=h(x_n)$ à chaque x_n , $Y_o=h(x_o)$ pour chaque x_o , et etc... Plus ces valeurs seront proches de celles que le professeur aurait fournies à sa place, meilleur sera l'apprentissage.

L'apprenant apprend donc par lui-même à classer ou à décider d'une action comme le fait le professeur. [12]

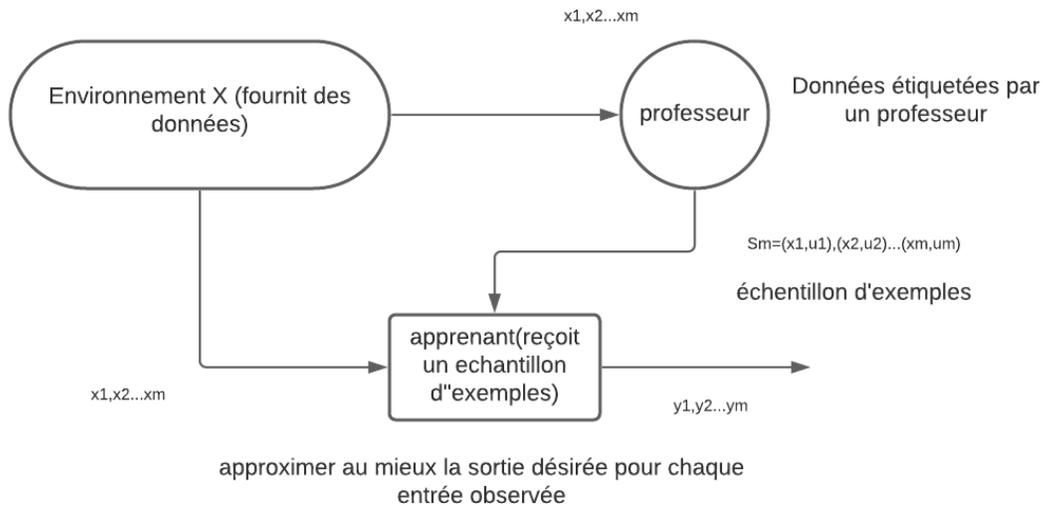


FIGURE 3.3 – schéma de l'apprentissage supervisé

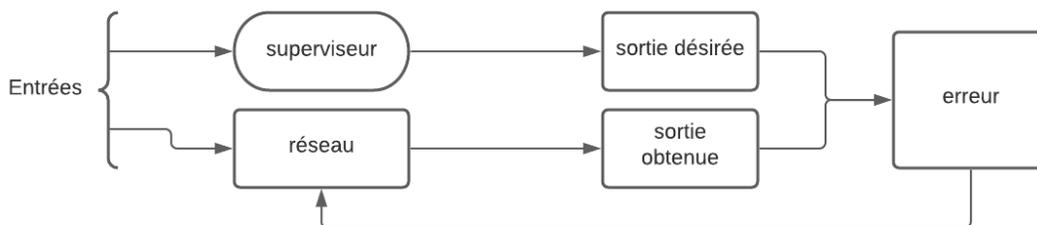


FIGURE 3.4 – Schéma d'un modèle supervisé

Apprentissage Non-Supervisé : Aucun expert n'est disponible. Il vise à concevoir un modèle structurant l'information. La différence ici est que les comportements (ou catégories ou encore les classes) des données d'apprentissage ne sont pas connus, c'est ce que l'on cherche à trouver. [6]

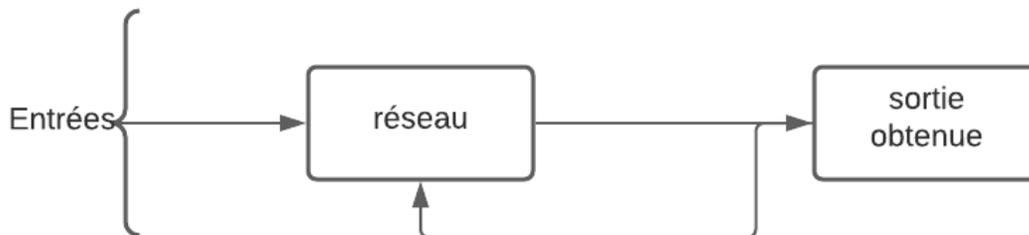


FIGURE 3.5 – Schéma d'un modèle non supervisé

Apprentissage par Renforcement : Les données en entrée sont les mêmes que pour l'apprentissage supervisé, cependant l'apprentissage est guidé par l'environnement sous la forme de récompenses ou de pénalités données en fonction de l'erreur commise lors de l'apprentissage. [6]

3.3.3 Réseaux de neurones (RN)

Les réseaux de neurones fonctionnent en répartissant les valeurs des variables dans des automates (les neurones). Ces unités sont chargées de combiner entre elles leurs informations pour déterminer la valeur du paramètre de discrimination. C'est de la connexion de ces unités entre elles qu'émerge la capacité de discrimination du RN. Chaque neurone reçoit des informations numériques en provenance de neurones voisins ; à chacune de ces valeurs est associé un poids représentatif de la force de la connexion. Chaque neurone effectue localement un calcul dont le résultat est transmis ensuite aux neurones avals. [19]

Dans la littérature, on parle de perceptron simple et de perceptron multicouche.

Perceptron simple Il s'agit d'un neurone binaire, c'est-à-dire dont la sortie vaut 0 ou

1. Pour calculer cette sortie, le neurone effectue une somme pondérée de ses entrées (chaque entrée possède un poids) : $= (1 \times X1 + W2 \times X2)$, puis applique une fonction d'activation à seuil :

si la somme pondérée dépasse une certaine valeur, la sortie du neurone est 1, sinon elle vaut 0. Il ne peut donc faire que de la classification, puis de la prédiction. La figure II.4 montre le schéma d'un perceptron simple.

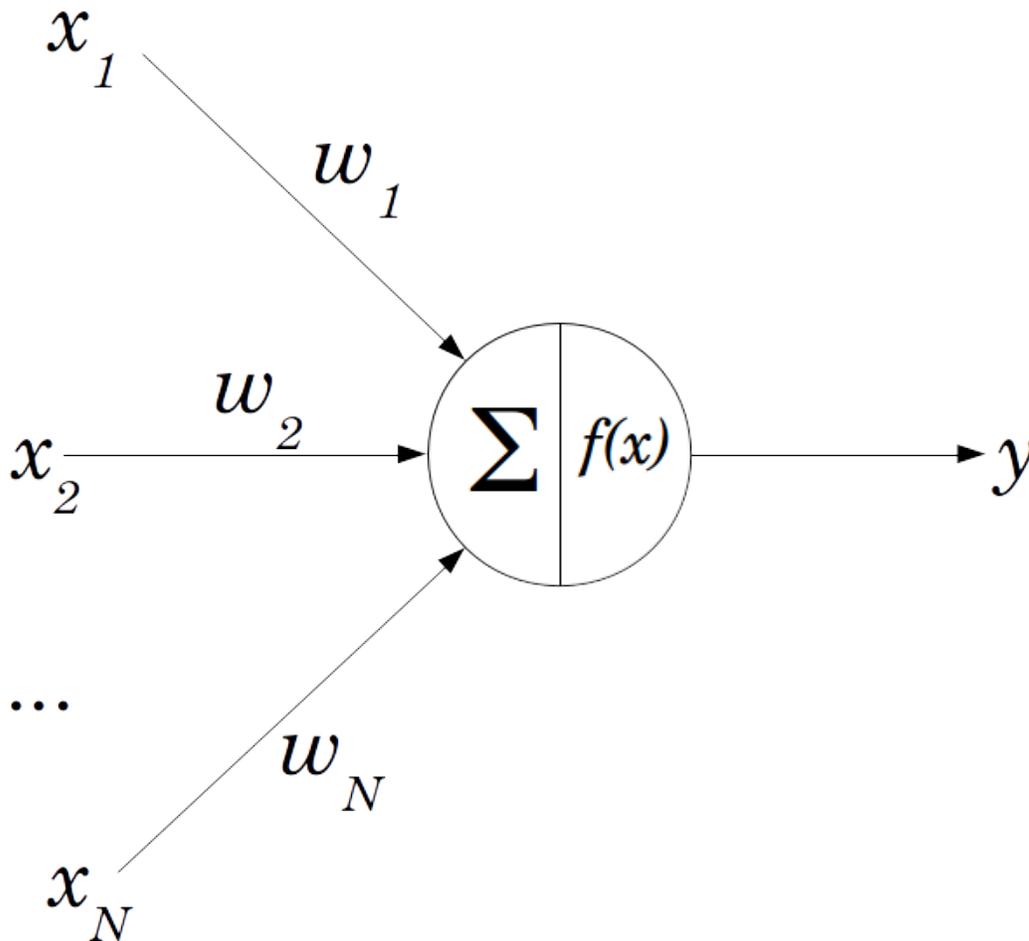


FIGURE 3.6 – schéma du perceptron simple

Perceptron multicouches (PMC)

La famille de réseau de neurones majoritairement employée est le perceptron multicouches. À lui seul ce type de réseau recouvre plus de 95 industrielles. Il comporte quelques dizaines à quelques centaines de neurones dans les cas usuels, voir plusieurs milliers pour les applications graphiques.

Le PMC est un modèle de réseau à propagation par couche (Figure II. 1). Les neurones y sont or-

ganisés en couches successives : une couche d'entrée, une couche de sortie et entre les deux une ou plusieurs couches intermédiaires, appelées aussi couches cachées. Il n'existe pas de connexion entre les neurones d'une même couche, en revanche tout neurone d'une couche est connecté à tous les neurones de la couche suivante.

La « couche » d'entrée n'est pas une réelle couche de neurones car elle se contente de coder les variables d'observation. La couche de sortie code la variable de discrimination. Les valeurs d'activité des neurones sont propagées dans le réseau, de l'entrée vers la sortie, sans retour arrière. La présence d'une couche cachée permet de modéliser des relations non linéaires entre les entrées et la sortie. En théorie une seule couche cachée suffit, mais le fait de disposer d'une seconde couche cachée permet de modéliser plus facilement une fonction de discrimination non continue. En pratique, la plupart des problèmes sont résolus avec un ou deux niveaux, trois au maximum. [19]

La figure II.5 illustre l'estimation de l'âge au décès à partir de l'observation de critères osseux de la surface sacro-pelvienne iliaque. Les entrées correspondent à l'observation de critères morphologiques sur la surface sacro.

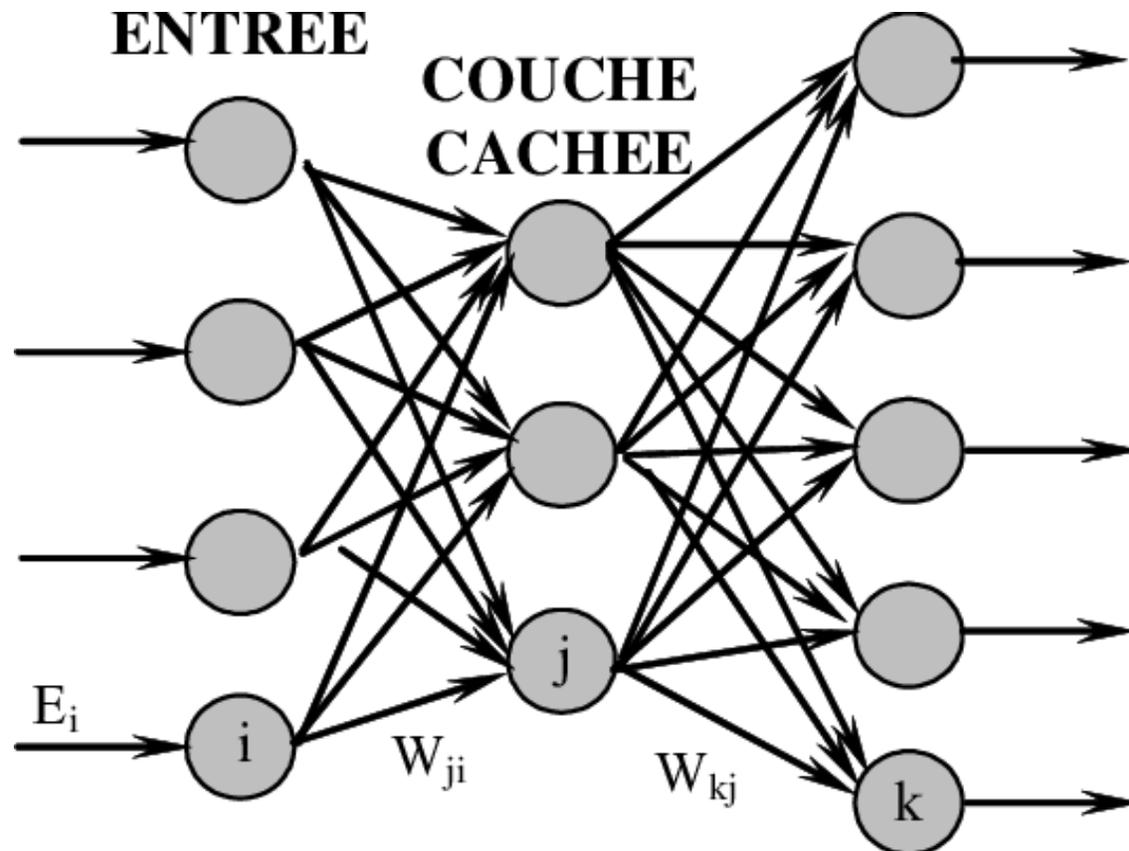


FIGURE 3.7 – Schéma d'un perceptron multicouches

3.3.4 Les réseaux de neurones récurrents (RNN ou Recurrent Neural Networks)

Les humains ne commencent pas leur réflexion à partir de zéro chaque seconde. En lisant cet essai, vous comprenez chaque mot en fonction de votre compréhension des mots précédents. Vous ne jetez pas tout et recommencez à penser à partir de zéro. Vos pensées ont de la persévérance. Les réseaux de neurones traditionnels ne peuvent pas le faire, et cela semble être une lacune majeure. Par exemple, imaginez que vous souhaitez classer le type d'événement qui se produit à chaque étape d'un film. On ne sait pas comment un réseau de neurones traditionnel pourrait utiliser son raisonnement sur des événements antérieurs dans le film pour les informer plus tard.[11]

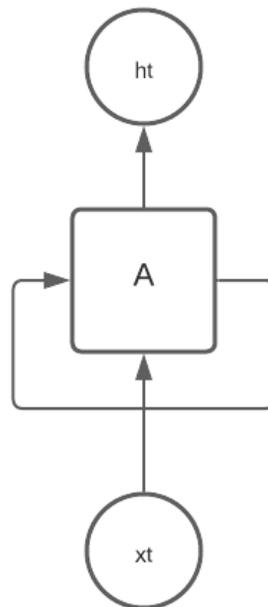


FIGURE 3.8 – Les réseaux neuronaux récurrents ont des boucles

Les réseaux de neurones récurrents résolvent ce problème. Ce sont des réseaux avec des boucles qui permettent à l'information de persister. Dans la figure II.6 ci-dessus, un segment de réseau neuronal; "A" regarde une entrée x_t et fournit une valeur h_t . Une boucle permet de passer des informations d'une étape du réseau à l'autre. Ces boucles rendent les réseaux neuronaux récurrents un peu mystérieux. Cependant, si vous pensez un peu plus, il s'avère qu'ils ne sont pas tous différents d'un réseau de neurones normal. Un réseau de neurones récurrent peut être considéré comme des copies multiples du même réseau, chacune transmettant un message à un successeur comme le montre la figure. Considérez ce qui se passe si nous déroulons la boucle.

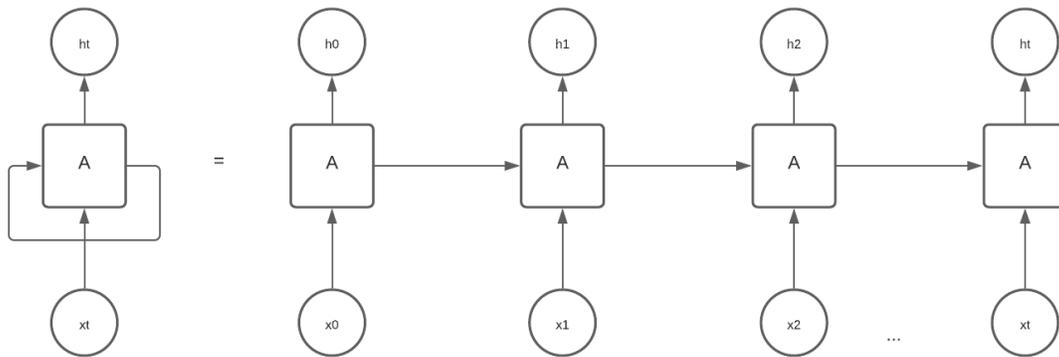


FIGURE 3.9 – Un réseau neuronal récurrent déroulé.

3.3.5 la mémoire à long terme (LSTM)

Le réseau neuronal récurrent utilise des blocs de mémoire à long terme pour fournir un contexte à la façon dont le programme reçoit les entrées et crée les sorties. Le bloc de mémoire à long terme est une unité complexe comportant divers composants tels que des entrées pondérées, des fonctions d'activation, des entrées provenant de blocs précédents et des sorties éventuelles.

L'unité est appelée bloc de mémoire à long terme parce que le programme utilise une structure fondée sur des processus de mémoire à court terme pour créer une mémoire à plus long terme. Ces systèmes sont souvent utilisés, par exemple, dans le traitement du langage naturel. Le réseau neuronal récurrent utilise les blocs de mémoire à long court terme pour prendre un mot ou un phonème particulier et l'évaluer dans le contexte des autres dans une chaîne, où la mémoire peut être utile pour trier et catégoriser ces types d'entrées. En général, LSTM est un concept accepté et commun dans les réseaux neuronaux récurrents pionniers.[2]

3.3.6 Les réseaux de neurones convolutionnels (CNN ou Convolutional Neural Networks)

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images [19]. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme

d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN. La figure II.8 illustre l'architecture standard d'un CNN.[11]

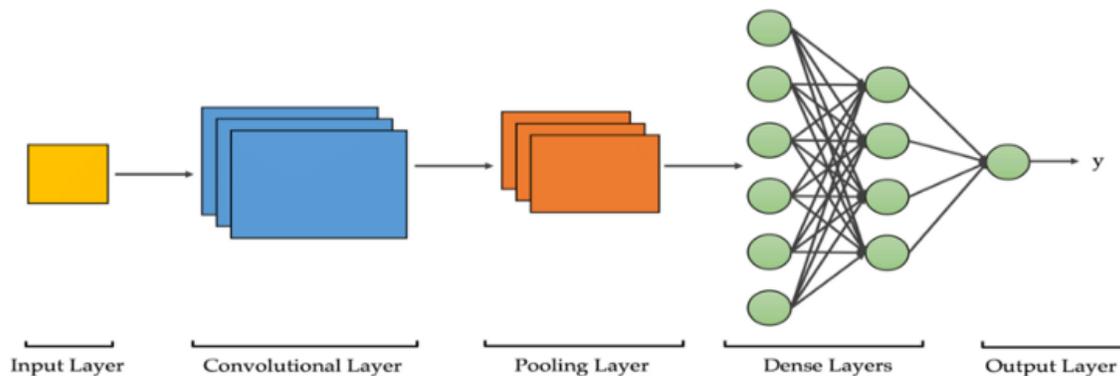


FIGURE 3.10 – Architecture standard d'un réseau de neurone convolusionnel.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image. La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories. [11]

3.3.7 Domaines d'applications de Deep Learning

Des applications du Deep Learning sont utilisées dans divers secteurs, de la conduite automatisée aux dispositifs médicaux.[11]

Conduite automatisée : Les chercheurs du secteur automobile ont recours au Deep Learning pour détecter automatiquement des objets tels que les panneaux stop et les feux de circulation.

Le Deep Learning est également utilisé pour détecter les piétons, évitant ainsi nombre d'accidents.

Aérospatiale et défense : Le Deep Learning sert à identifier des objets à partir de satellites utilisés pour localiser des zones d'intérêt et identifier quels secteurs sont sûrs ou dangereux pour les troupes au sol.

Recherche médicale : À l'aide du Deep Learning, les chercheurs en cancérologie peuvent dépister automatiquement les cellules cancéreuses. Des équipes de l'Université de Californie à Los Angeles (UCLA) ont conçu un microscope qui génère un ensemble de données de grande dimension afin d'entraîner une application de Deep Learning à identifier avec précision des cellules cancéreuses.

Automatisation industrielle : Le Deep Learning participe à l'amélioration de la sécurité des employés travaillant à proximité d'équipements lourds, en détectant automatiquement les situations dans lesquelles la distance de sécurité qui sépare le personnel ou les objets des machines est insuffisante.

Électronique : Le Deep Learning est utilisé pour la reconnaissance audio et vocale. Par exemple, les appareils d'assistance à domicile qui répondent à votre voix et connaissent vos préférences fonctionnent grâce à des applications de Deep Learning

3.3.8 l'apprentissage profond dans phm

Alors a fin de collecter les données necessaire des machines de notre usine 4.0 pour faire la pronostic à elles il faut que calculer RUL pour connaitre le temps ou la machine vas tomber en panne alors apres tout ça notre base de données et prete pour la enter dans l'apprentissage profond plus précisément l'architecture CNN (modèle CNN) pour traiter les données afin de tirer des règle qui permettront de tirer des conclusions et nous obtenons une prédiction de classe à partir du modèle appris.

3.4 Conclusion

Nous avons présenté au cours de ce chapitre Prognostics and Health Management (définition ,l'architecture...) et l'apprentissage automatique avec ses différents types, supervisé, non-supervisé et avec renforcement, les réseaux de neurones tel que le perceptron simple et le perceptron multicouches, ensuite nous avons présenté l'apprentissage profond et ses différentes méthodes : RNN, CNN, LSTM et quelques domaines d'applications dont il est utilisé, nous nous sommes aussi intéressés à l'utilisation pratique à travers quelques travaux de l'utilisation des différentes méthodes d'apprentissage profond dans la Dans le chapitre suivant, nous allons appliquer le deep learning dans un scénario bien précis.

Chapitre 4

la conception

4.1 Introduction

dans ce chapitre nous présentons la conception globale et détaillées de notre système et l'architecture de l'apprentissage profond utilisé.

4.2 L'architecture globale de système

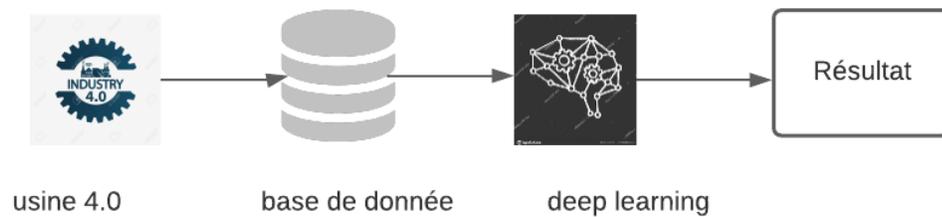


FIGURE 4.1 – l'architecture globale du système.

cette figure représente l'architecture globale de notre système , alors on a : usine 4.0 : contient plusieurs machines connectées entre eux avec iot plus les capteurs placés dans les machines pour l'extraction des informations nécessaire ensuite le stockage de données dans une base de donnée pour faire les opérations nécessaire après la préparation des données ce suite la transmission des données au CNN(deep learning) en fin le résultat .

4.3 L'architecture détaillées du système

4.3.1 Base de donnée

La préparation des données est la première étape à appliquer avant de transmettre les données au CNN -la collection des données comme elle montre dans la figure **prétraitement**

```

1,1,10.0047,0.2501,20.0,489.05,604.13,1499.45,1309.95,10.52,15.49,394.88,2318.87,8770.20,1.26,45.40,372.15,2388.13,8120.83,8.6216,0.03,368,2319,100.00,28.58,17.1735
1,2,0.0015,0.0003,100.0,518.67,642.13,1584.55,1403.96,14.62,21.61,553.67,2388.01,9045.76,1.30,47.29,521.81,2388.15,8132.87,8.3907,0.03,391,2388,100.00,38.99,23.3619
1,3,34.9986,0.8401,60.0,449.44,555.42,1368.17,1122.49,5.48,8.00,194.93,2222.86,8343.91,1.02,41.92,183.26,2387.95,8063.84,9.3557,0.02,334,2223,100.00,14.83,8.8555
1,4,20.0031,0.7005,0.0,491.19,607.03,1488.44,1249.18,9.35,13.65,334.82,2323.85,8721.53,1.08,44.26,314.84,2388.07,8052.30,9.2231,0.02,364,2324,100.00,24.42,14.7832
1,5,42.0041,0.8405,40.0,445.00,549.52,1354.48,1124.32,3.91,5.71,138.24,2211.80,8314.56,1.02,41.79,130.44,2387.89,8083.67,9.2986,0.02,330,2212,100.00,10.99,6.4025
1,6,20.0032,0.7017,0.0,491.19,607.37,1480.46,1258.90,9.35,13.65,334.51,2323.94,8711.44,1.08,44.40,315.36,2388.05,8053.17,9.2276,0.02,364,2324,100.00,24.44,14.7019
1,7,41.9998,0.8400,40.0,445.00,549.57,1354.43,1131.44,3.91,5.71,139.11,2211.82,8316.88,1.02,42.09,130.16,2387.88,8082.01,9.3753,0.02,331,2212,100.00,10.53,6.4254
1,8,0.0011,0.0000,100.0,518.67,642.08,1589.55,1407.59,14.62,21.61,553.48,2388.10,9050.35,1.30,47.50,521.74,2388.03,8133.29,8.4339,0.03,391,2388,100.00,38.98,23.2337
1,9,0.0011,0.0020,100.0,518.67,642.70,1586.18,1399.01,14.62,21.61,553.90,2388.10,9051.35,1.30,47.30,521.72,2388.08,8132.72,8.3922,0.03,392,2388,100.00,38.99,23.2412
1,10,42.0066,0.8400,40.0,445.00,549.83,1353.19,1125.09,3.91,5.71,138.37,2211.84,8320.04,1.02,41.91,130.32,2387.91,8085.94,9.3667,0.02,330,2212,100.00,10.75,6.4268
1,11,25.0051,0.6200,80.0,462.54,537.41,1260.50,1051.56,7.05,9.02,175.86,1915.28,8009.91,0.94,36.66,165.53,2028.24,7866.43,10.8941,0.02,307,1915,84.93,14.28,8.5087
1,12,35.0029,0.8413,60.0,449.44,555.85,1360.54,1130.69,5.48,8.00,195.05,2222.88,8341.87,1.02,41.87,183.41,2387.95,8063.79,9.2878,0.02,335,2223,100.00,14.69,8.7988
1,13,42.0029,0.8409,40.0,445.00,549.91,1350.90,1123.53,3.91,5.71,139.09,2211.78,8321.20,1.02,42.14,130.60,2387.88,8081.77,9.3490,0.02,329,2212,100.00,10.46,6.3382
1,14,25.0073,0.6203,80.0,462.54,537.22,1262.80,1051.01,7.05,9.02,175.35,1915.24,8014.89,0.94,36.88,164.63,2028.14,7870.29,10.9160,0.02,306,1915,84.93,14.28,8.5684
1,15,9.9988,0.2500,20.0,489.05,604.63,1497.87,1302.51,10.52,15.50,394.93,2318.90,8771.08,1.26,45.39,371.84,2388.12,8124.37,8.6496,0.03,369,2319,100.00,28.84,17.0805
1,16,9.9987,0.2500,20.0,489.05,604.61,1500.47,1311.39,10.52,15.49,394.44,2318.87,8775.86,1.26,45.52,372.19,2388.08,8127.07,8.6609,0.03,369,2319,100.00,28.71,17.1759
1,17,0.0003,0.0000,100.0,518.67,642.30,1584.57,1410.11,14.62,21.61,554.00,2388.06,9056.59,1.30,47.44,522.17,2388.10,8133.38,8.4340,0.03,392,2388,100.00,38.81,23.3487
1,18,10.0066,0.2507,20.0,489.05,604.23,1502.57,1304.56,10.52,15.49,394.31,2318.93,8778.55,1.26,45.10,372.03,2388.06,8119.39,8.6646,0.03,367,2319,100.00,28.60,17.1096
1,19,25.0018,0.6200,80.0,462.54,536.66,1255.21,1053.69,7.05,9.03,174.94,1915.22,8005.22,0.94,36.44,165.27,2028.13,7868.80,10.8969,0.02,308,1915,84.93,14.42,8.5456
1,20,0.0004,0.0017,100.0,518.67,642.31,1581.33,1399.44,14.62,21.61,553.41,2388.09,9049.58,1.30,47.38,521.62,2388.06,8132.25,8.4044,0.03,393,2388,100.00,38.97,23.2494
1,21,42.0030,0.8404,40.0,445.00,549.12,1349.16,1117.33,3.91,5.72,138.51,2211.82,8310.18,1.02,41.98,130.48,2387.90,8080.31,9.3488,0.02,330,2212,100.00,10.54,6.3614
1,22,0.0004,0.0000,100.0,518.67,642.34,1589.25,1401.52,14.62,21.61,553.90,2388.12,9051.82,1.30,47.45,521.55,2388.06,8131.87,8.4423,0.03,392,2388,100.00,38.73,23.3303
1,23,20.0025,0.7011,0.0,491.19,607.10,1482.68,1252.00,9.35,13.66,334.49,2323.91,8716.75,1.08,44.44,314.87,2388.10,8050.94,9.2149,0.02,366,2324,100.00,24.58,14.6688

```

FIGURE 4.2 – un petit exemple de la base de données.

preprocessing

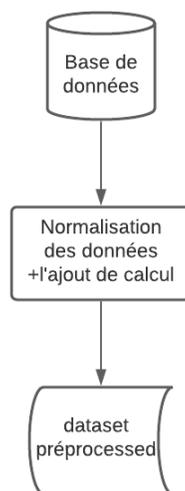


FIGURE 4.3 – base de données prétraitement.

Après tout ça la base de données prétraitement vas sur 3 étapes

training set

validation set

test set

4.3.2 les capteurs

Dispositif sensible à certains phénomènes physiques qu'il capte ou détecte. dans notre cas il capte la variation en temps de cycle de machine. on ces dernier sont placés dans les machines de notre usine pour donner des informations précis .

4.3.3 deep learning

dans l'apprentissage profond nous discuterons la méthode de détection la plus courante CNN. Convolution Neural Network (CNN ou ConvNet) est l'une des structures réseau les plus représentatives de la technologie d'apprentissage en profondeur, ont été introduits pour la première fois par Fukushima [18] et a connu un grand succès dans le domaine du traitement et de la reconnaissance d'images. mais cette fois si avec des valeurs c'est à dire l'entré n'est pas une image c'est ensemble des valeurs à tester voici notre architecture cnn de notre modèle dans la figure suivante :

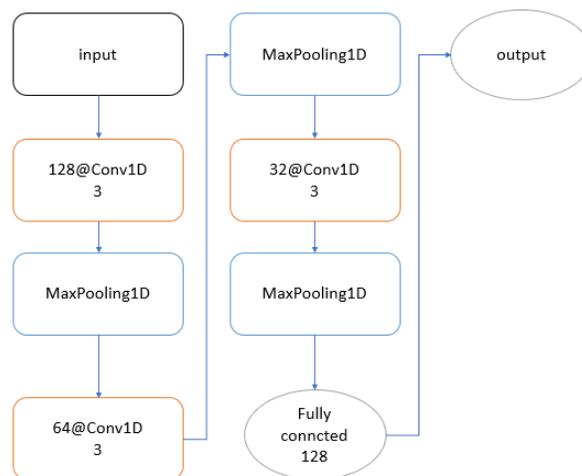


FIGURE 4.4 – architecture CNN.

tant que l'entrée de notre base des données sont des valeurs nous avons travailler avec Conv1D c'est à une seule dimension la raison d'utiliser conv1D parceque la convolution est une opération mathématique où vous " résumez un tenseur ou une matrice ou un vecteur en un plus petit.

alors apres le modèle c'est à dire training le modèle ce suivre le test. Un ensemble de test utilisé une seule fois à la fin du projet afin d'évaluer les performances du modèle final qui est affiné et sélectionné sur le processus d'apprentissage avec des ensembles d'apprentissage et de validation. Les prédictions d'ensemble sur l'ensemble de tests ont été analysées pour les vrais positifs (V P), les vrais négatifs (V N), les faux positifs (F P) et les faux négatifs (F P). Ces paramètres ont été utilisés pour calculer une variété de mesures de performance, notamment " precision " (PR), " sensitivity " (SE), " accuracy " (ACC), " F1-score " (F1), "specificity" (SP) selon les équations suivantes :

$$1)PR = \frac{TP}{TP + FP}$$

$$2)SE = \frac{TP}{TP + FN}$$

$$3)ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$4)SP = \frac{TN}{TN + FP}$$

$$5)F1 = \frac{2TP}{2TP + FP + FN}$$

4.4 Conclusion

Nous avons présenté au cours de ce chapitre la conception et l'architecture CNN utilisé alors dans le prochain chapitre nous allons appliquer ce modèle et on vas voir l'évaluation et le résultat obtenue.

Chapitre 5

Développement détaillée du modèle

5.1 Introduction

Dans ce chapitre nous allons présenter les outils et les plateformes utilisés pour développer notre approche. Enfin, les résultats obtenus seront discutés

5.2 Environnements et outils de développement

5.2.1 Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.[8]



FIGURE 5.1 – Google Colab.

5.2.2 Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.[6]



FIGURE 5.2 – python.

5.2.3 TensorFlow

TensorFlow est un open source développé par les chercheurs de Google pour exécuter l'apprentissage automatique, l'apprentissage profond et d'autres charges de travail d'analyse statistique et prédictive. Comme les plateformes similaires, il est conçu pour rationaliser le processus de développement et d'exécution d'applications analytiques avancées pour les utilisateurs tels que les scientifiques des données, les statisticiens et les modélisateurs prédictifs.[7]



FIGURE 5.3 – TensorFlow.

5.2.4 Keras

Keras est une bibliothèque open-source de composants de réseaux neuronaux écrits en Python. Keras est capable de fonctionner au-dessus de TensorFlow, Theano, PlaidML et autres. La bibliothèque a été développée pour être modulaire et conviviale, mais elle a initialement commencé dans le cadre d'un projet de recherche pour le système d'exploitation intelligent neuro-électronique ouvert ou ONEIROS. L'auteur principal de Keras est François Chollet, un ingénieur de Google qui a également écrit Xception, un modèle de réseau neuronal profond. Bien que Keras ait été officiellement lancé, il n'a été intégré à la bibliothèque centrale TensorFlow de Google qu'en 2017. Un support supplémentaire a également été ajouté pour l'intégration de Keras avec Microsoft Cognitive Toolkit.[9]



FIGURE 5.4 – Keras.

5.2.5 Numpy

NumPy est le paquetage fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que les tableaux masqués et les matrices), ainsi qu'un assortiment de routines permettant d'effectuer des opérations rapides sur les tableaux, notamment des opérations mathématiques, logiques, de manipulation de formes, de tri, de sélection, d'E/S, de transformées de Fourier discrètes, d'algèbre linéaire de base, d'opérations statistiques de base, de simulation aléatoire et bien plus encore.[4]



FIGURE 5.5 – NumPy.

5.2.6 Matplotlib

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques⁵. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy. Matplotlib est distribuée librement et gratuitement sous une licence de style BSD. Sa version stable actuelle (la 2.0.1 en 2017) est compatible avec la version 3 de Python.^[5]



FIGURE 5.6 – Matplotlib.

5.3 Travail effectué

5.3.1 Construction de la base d'apprentissage

Dans le présent travail, nous avons utilisé « PHM08 Challenge Data Set » Les données surveillées du système d'un moteur de machine sont tirées du Répertoire des données du Centre d'excellence en pronostics de la NASA .Alors on a 218 machines avec 25 caractéristiques pour chaque entrée . voilà l'entête de notre base de données

	unit number	time in cycle	operational setting1	operational setting2	operational setting3	sensor1	sensor2	sensor3	sensor4	sensor5	sensor6	sensor7	sensor8	sensor9	sensor10	sensor11
0	1	1	10.0047	0.2501	20.0	489.05	604.13	1499.45	1309.95	10.52	15.49	394.88	2318.87	8770.20	1.26	45.40
1	1	2	0.0015	0.0003	100.0	518.67	642.13	1584.55	1403.96	14.62	21.61	553.67	2388.01	9045.76	1.30	47.29
2	1	3	34.9986	0.8401	60.0	449.44	555.42	1368.17	1122.49	5.48	8.00	194.93	2222.86	8343.91	1.02	41.92
3	1	4	20.0031	0.7005	0.0	491.19	607.03	1488.44	1249.18	9.35	13.65	334.82	2323.85	8721.53	1.08	44.26
4	1	5	42.0041	0.8405	40.0	445.00	549.52	1354.48	1124.32	3.91	5.71	138.24	2211.80	8314.56	1.02	41.79

FIGURE 5.7 – Le DataSet utilisé.

r5	sensor6	sensor7	sensor8	sensor9	sensor10	sensor11	sensor12	sensor13	sensor14	sensor15	sensor16	sensor17	sensor18	sensor19	sensor20	sensor21
52	15.49	394.88	2318.87	8770.20	1.26	45.40	372.15	2388.13	8120.83	8.6216	0.03	368	2319	100.0	28.58	17.1735
62	21.61	553.67	2388.01	9045.76	1.30	47.29	521.81	2388.15	8132.87	8.3907	0.03	391	2388	100.0	38.99	23.3619
48	8.00	194.93	2222.86	8343.91	1.02	41.92	183.26	2387.95	8063.84	9.3557	0.02	334	2223	100.0	14.83	8.8555
35	13.65	334.82	2323.85	8721.53	1.08	44.26	314.84	2388.07	8052.30	9.2231	0.02	364	2324	100.0	24.42	14.7832
91	5.71	138.24	2211.80	8314.56	1.02	41.79	130.44	2387.89	8083.67	9.2986	0.02	330	2212	100.0	10.99	6.4025

FIGURE 5.8 – Le DataSet utilisé(suite).

Le suivi et la prédiction de la progression des dommages dans les moteurs des machines. Donc on vas calcules RUL Ils estiment l'efficacité et la déviation du débit en fonction des données opérationnelles, et utilisent cette information pour la détection des défaillances.

méthode RUL

```

1 def remaining_useful_life(df):
2     return len(df) - 1
3     lm = cp.LabelMaker(
4         target_entity='unit number',
5         time_index='time in cycle',
6         labeling_function=remaining_useful_life,
7     )
8     label_times = lm.search(
9         data.sort_values('time in cycle'),
10        num_examples_per_instance=1,
11        minimum_data=100,
12        verbose=True,
13    )
14
15 label_times.head()

```

Listing 5.1 – Calcul RUL

les résultats sont stockés dans une tableau séparer.voici la figure suivante

	rull	unit number
0	122	1
1	63	2
2	49	3
3	58	4
4	256	5

FIGURE 5.9 – Le résultat de RUL.

ensuite le calcul de rtf(run to failure) il faut d'abord générer le max depuis testdata pour faire l'opération comme la figure suivante montre .

```
1  
2  
3 # generate column max for test data  
4 rul = pd.DataFrame(dataset_test.groupby('unit number')['time in cycle'].  
5     max()).reset_index()  
6 rul.columns = ['unit number', 'max']  
7 rul.head()
```

Listing 5.2 – Calcul RUL

	unit number	max
0	1	54
1	2	157
2	3	116
3	4	74
4	5	218

FIGURE 5.10 – génération de la colonne max.

```

1
2
3 # run to failure
4 pm_truth['rtf']=pm_truth['rull']+rul['max']
5 pm_truth.head()

```

Listing 5.3 – Calcul RUL

	rull	unit number	rtf
0	122	1	176
1	63	2	220
2	49	3	165
3	58	4	132
4	256	5	474

FIGURE 5.11 – "run to failure" de chaque machine.

finalement le calcul de ttf(time to failure)c'est à dire la machine vas tomber en panne et le ajouter dans traindata et testdata à travers de ce dernier nous avons déterminer "la normalisation de notre données"en 2 catégories

```

1
2
3 pm_truth.drop('rull', axis=1, inplace=True)
4 dataset_test=dataset_test.merge(pm_truth,on=['unit number'],how='left')
5 dataset_test['ttf']=dataset_test['rtf'] - dataset_test['time in cycle']
6 dataset_test.drop('rtf', axis=1, inplace=True)
7 dataset_test.head()

```

Listing 5.4 – time to failure

sensor6	sensor7	sensor8	sensor9	sensor10	sensor11	sensor12	sensor13	sensor14	sensor15	sensor16	sensor17	sensor18	sensor19	sensor20	sensor21	ttf
21.61	553.33	2388.12	9042.41	1.30	47.57	521.44	2388.10	8124.04	8.4363	0.03	393	2388	100.00	38.97	23.3029	175
9.02	175.04	1915.12	7983.34	0.94	36.81	164.32	2027.95	7865.20	10.8935	0.02	308	1915	84.93	14.29	8.6686	174
21.61	554.31	2388.18	9037.61	1.30	47.49	521.15	2388.19	8121.12	8.4620	0.03	393	2388	100.00	38.91	23.2693	173
15.49	393.88	2318.94	8761.83	1.26	45.41	371.42	2388.21	8123.53	8.6815	0.03	369	2319	100.00	28.60	17.0930	172
21.61	553.53	2388.13	9041.98	1.30	47.66	520.86	2388.13	8118.30	8.4617	0.03	392	2388	100.00	38.82	23.3124	171

FIGURE 5.12 – "time to failure" de chaque machine(test).

```

1
2
3 dataset_train['ttf'] = dataset_train.groupby(['unit number'])['time in
   cycle'].transform(max)-dataset_train['time in cycle']
4 dataset_train.head()

```

Listing 5.5 – time to failure

sensor6	sensor7	sensor8	sensor9	sensor10	sensor11	sensor12	sensor13	sensor14	sensor15	sensor16	sensor17	sensor18	sensor19	sensor20	sensor21	tff
15.49	394.88	2318.87	8770.20	1.26	45.40	372.15	2388.13	8120.83	8.6216	0.03	368	2319	100.0	28.58	17.1735	222
21.61	553.67	2388.01	9045.76	1.30	47.29	521.81	2388.15	8132.87	8.3907	0.03	391	2388	100.0	38.99	23.3619	221
8.00	194.93	2222.86	8343.91	1.02	41.92	183.26	2387.95	8063.84	9.3557	0.02	334	2223	100.0	14.83	8.8555	220
13.65	334.82	2323.85	8721.53	1.08	44.26	314.84	2388.07	8052.30	9.2231	0.02	364	2324	100.0	24.42	14.7832	219
5.71	138.24	2211.80	8314.56	1.02	41.79	130.44	2387.89	8083.67	9.2986	0.02	330	2212	100.0	10.99	6.4025	218

FIGURE 5.13 – "time to failure" de chaque machine(train).

Normalisation des données

on a définie le point de failure = 30

Alors Si label bc=1 donc etat normal

si label bc =0 dégradation jusqu'à l'arrêt de la machine (ne fonctionne pas) comme la figure suivante

```

1 df_train=dataset_train.copy()
2 df_test=dataset_test.copy()
3 period=30
4 df_train['label_bc'] = df_train['tff'].apply(lambda x: 1 if x <= period
      else 0)
5 df_test['label_bc'] = df_test['tff'].apply(lambda x: 1 if x <= period else
      0)
6
7 print(df_train)

```

Listing 5.6 – normalisation des données

```
      unit number  time in cycle  ...  ttf  label_bc
0          1          1  ...  222          0
1          1          2  ...  221          0
2          1          3  ...  220          0
3          1          4  ...  219          0
4          1          5  ...  218          0
...      ...      ...  ...  ...      ...
45913      218      129  ...    4          1
45914      218      130  ...    3          1
45915      218      131  ...    2          1
45916      218      132  ...    1          1
45917      218      133  ...    0          1

[45918 rows x 28 columns]
```

FIGURE 5.14 – normalisation des données.

creation de modèle CNN

```

1 nb_features =X_train.shape [2]
2 timestamp=seq_length
3 model = Sequential()
4 model.add(Conv1D(128, kernel_size=(3),activation='relu',input_shape= (
    timestamp, nb_features),padding='same'))
5 model.add(LeakyReLU(alpha=0.1))
6 model.add(MaxPooling1D((2),padding='same'))
7 model.add(Conv1D(64, (3), activation='relu',padding='same'))
8 model.add(LeakyReLU(alpha=0.1))
9 model.add(MaxPooling1D(pool_size=(2),padding='same'))
10 model.add(Conv1D(32, (3), activation='relu',padding='same'))
11 model.add(LeakyReLU(alpha=0.1))
12 model.add(MaxPooling1D(pool_size=(2),padding='same'))
13 model.add(Flatten())
14 model.add(Dense(128, activation='relu'))
15 model.add(LeakyReLU(alpha=0.1))
16 model.add(Dense(1, activation='sigmoid'))

```

Listing 5.7 – Modèle CNN

Aussi nous avons tester l'architecture LSTM sur notre base de données

voici le modèle LSTM

```

1 nb_features =X_train.shape [2]
2 timestamp=seq_length
3
4 model = Sequential()
5
6 model.add(LSTM(
7     input_shape=(timestamp, nb_features),
8     units=100,
9     return_sequences=True))
10 model.add(Dropout(0.2))
11
12 model.add(LSTM(
13     units=50,
14     return_sequences=False))
15 model.add(Dropout(0.2))
16
17 model.add(Dense(units=1, activation='sigmoid'))
18 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['
    accuracy'])
19 model.summary()

```

Listing 5.8 – Modèle LSTM

Compilation du modèle

```

1 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['
  accuracy'])
2 model.summary()

```

Listing 5.9 – Compilation du modèle

train du modèle

```

1 # fit the network
2
3 hist = model.fit(X_train, y_train, epochs=20, batch_size=200,
  validation_split=0.25, verbose=1)
4 print(hist.history)
5 epo = range(len(hist.history['accuracy']))
6 acc = hist.history['val_accuracy']
7 loss = hist.history['val_loss']
8 plt.plot(epo, acc, 'g+', label='Training accuracy')
9 plt.plot(epo, loss, 'r--', label='training loss')
10 #plt.plot(epochs, val_accuracy, 'b', label='Validation accuracy')
11 plt.title('Training accuracy and loss')
12 plt.legend()
13 plt.show()
14 model.save('mymodel.h5')

```

Listing 5.10 – Training de modèle

Prédiction sur le modèle

```

1 def prob_failure(machine_id):
2     machine_df=df_test[df_test.id==machine_id]
3     machine_test=gen_sequence(machine_df,seq_length,seq_cols)
4     m_pred=model.predict(machine_test)
5     failure_prob=list(m_pred[-1]*100)[0]
6     return failure_prob

```

```

machine_id=30
print('Probability that machine will fail within 30 days: ',prob_failure(machine_id))

```

```
Probability that machine will fail within 30 days: 0.43491423
```

FIGURE 5.15 – l’affichage etat normal du machine .

5.4 Évaluation empirique et résultats

Architecture de CNN

on a proposé une architecture cnn de l'apprentissage profond pour garantir meilleur performance pour notre projet

la figure suivante représente l'architecture cnn

Layer (type)	Output Shape	Param #
conv1d_35 (Conv1D)	(None, 50, 128)	9344
leaky_re_lu_43 (LeakyReLU)	(None, 50, 128)	0
max_pooling1d_33 (MaxPooling)	(None, 25, 128)	0
conv1d_36 (Conv1D)	(None, 25, 64)	24640
leaky_re_lu_44 (LeakyReLU)	(None, 25, 64)	0
max_pooling1d_34 (MaxPooling)	(None, 13, 64)	0
conv1d_37 (Conv1D)	(None, 13, 32)	6176
leaky_re_lu_45 (LeakyReLU)	(None, 13, 32)	0
max_pooling1d_35 (MaxPooling)	(None, 7, 32)	0
flatten_10 (Flatten)	(None, 224)	0
dense_20 (Dense)	(None, 128)	28800
leaky_re_lu_46 (LeakyReLU)	(None, 128)	0
dense_21 (Dense)	(None, 1)	129

FIGURE 5.16 – "l'architecture CNN."

5.4.1 Les performances du CNN

Pour visualiser les performances de notre CNN de deep learning, voici les graphes qu'ils nous avons créé :

- un graphe de "accuracy" sur l'ensemble de données train « acc » sur les époques d'entraînement.
- un graphe de "loss" sur l'ensemble de données train «loss » au cours des époques d'entraînement.
- un graphe de "val-accuracy" sur l'ensemble de données de validation 'Val acc' sur les époques d'entraînement.
- un graphe de "val-loss" sur l'ensemble de données de validation «val-loss» sur les époques d'entraînement.

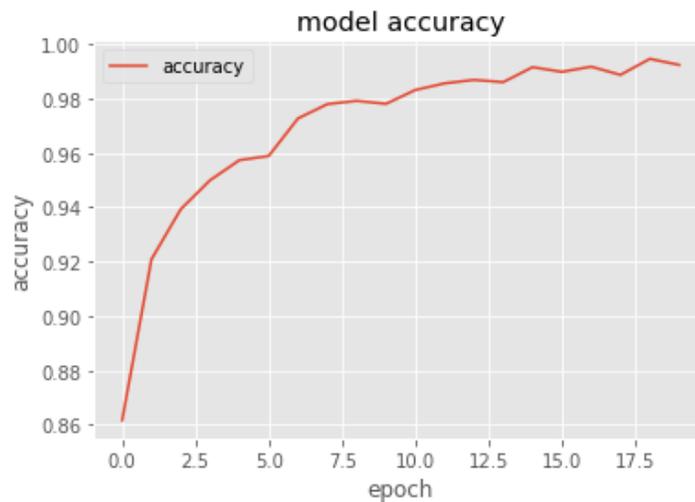


FIGURE 5.17 – "Graphe de la variation moyenne de la précision."

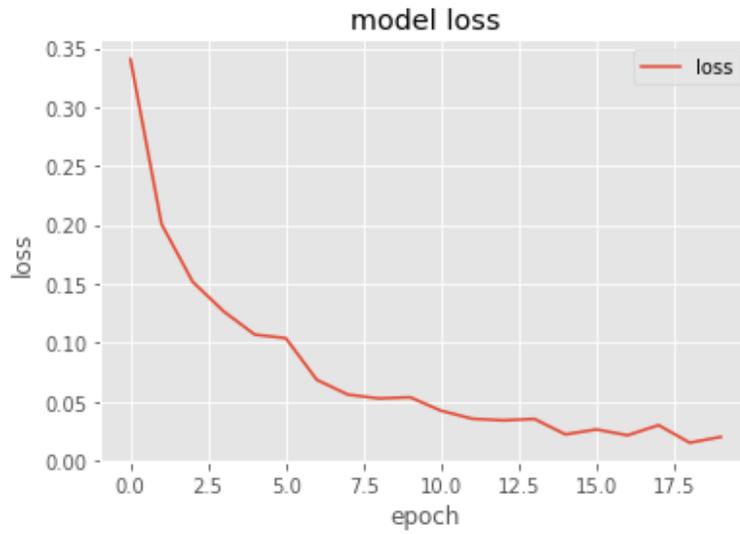


FIGURE 5.18 – "Graphe de la variation moyenne de la perte."

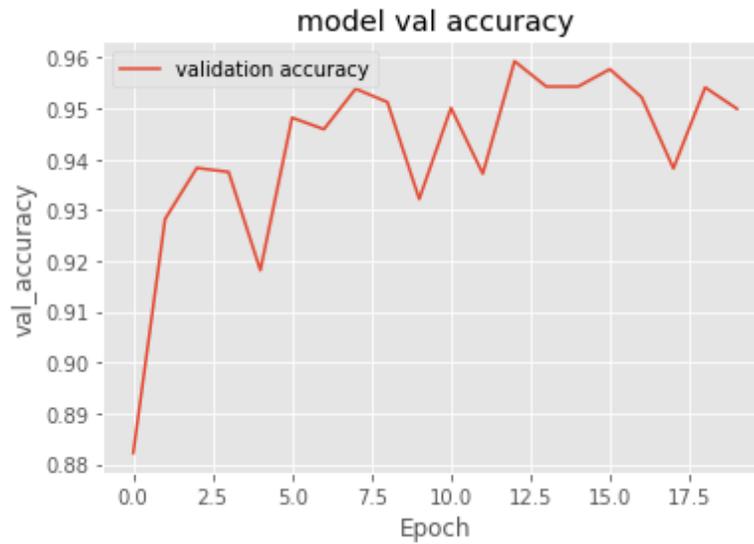


FIGURE 5.19 – "Graphe de la variation moyenne de la validation de précision."

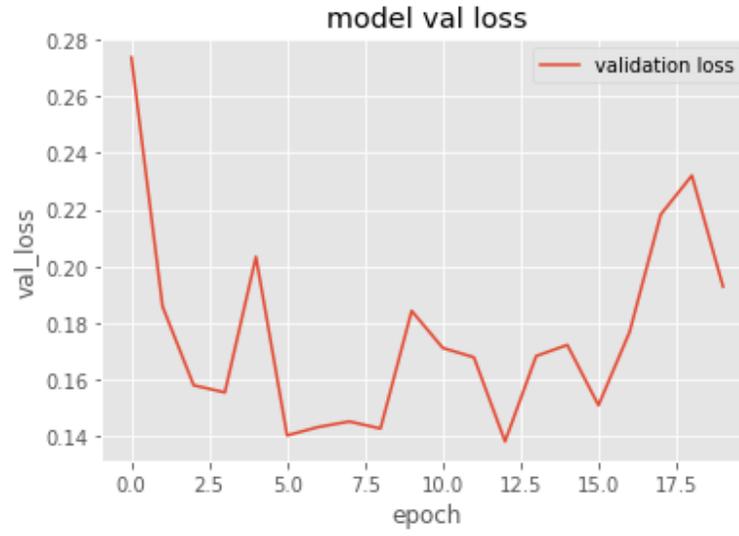


FIGURE 5.20 – "Graphe de la variation moyenne de la validation de perte."

f1 score

	precision	recall	f1-score	support
0.0	1.00	0.98	0.99	29598
1.0	0.01	1.00	0.01	4
accuracy			0.98	29602
macro avg	0.50	0.99	0.50	29602
weighted avg	1.00	0.98	0.99	29602

FIGURE 5.21 – F1 Score.

matrice de confusion

```

[[29036  562]
 [    0    4]]
Sensitivity : 0.9810122305561186
Specificity : 1.0
    
```

FIGURE 5.22 – matrice de confusion.

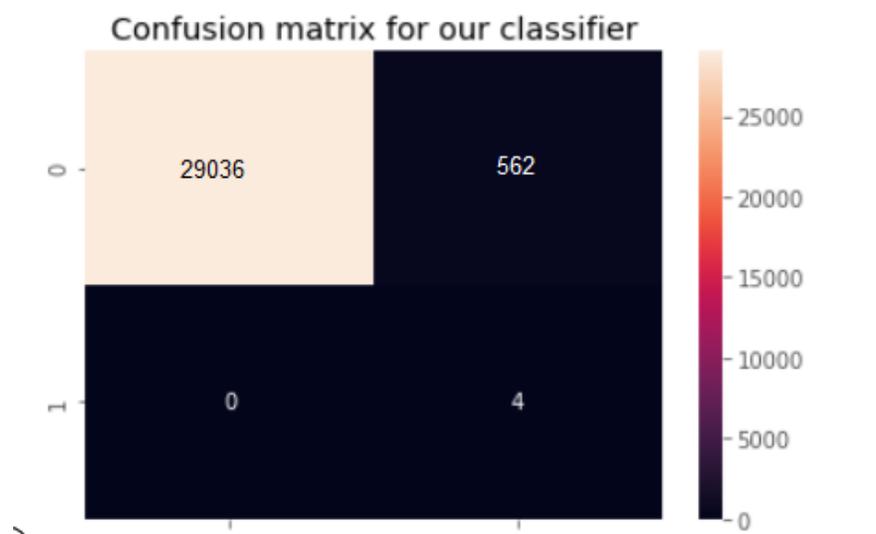


FIGURE 5.23 – matrice de confusion.

5.5 Evaluation empirique et résultat du modèle LSTM

l'architecture LSTM

```
Model: "sequential_2"
```

Layer (type)	Output Shape	Param #
lstm_4 (LSTM)	(None, 50, 100)	50000
dropout_4 (Dropout)	(None, 50, 100)	0
lstm_5 (LSTM)	(None, 50)	30200
dropout_5 (Dropout)	(None, 50)	0
dense_2 (Dense)	(None, 1)	51

```
Total params: 80,251
Trainable params: 80,251
Non-trainable params: 0
```

FIGURE 5.24 – Architecture LSTM.

5.5.1 les performances du LSTM

Même avec LSTM le résultat est excellent la figure suivante représente le graphe précision et perte

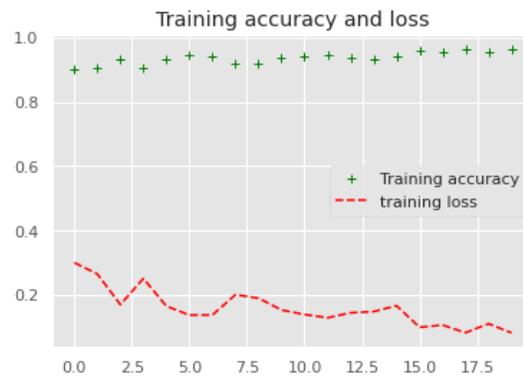


FIGURE 5.25 – Graphe précision et perte .

f1 score

```
Accuracy of model on test data: 0.9825343738387217
      precision    recall  f1-score   support
0.0         1.00      0.98      0.99     29597
1.0         0.00      0.00      0.00         4

 accuracy
macro avg      0.50      0.49      0.50     29601
weighted avg   1.00      0.98      0.99     29601
```

FIGURE 5.26 – F1 score LSTM.

5.6 Conclusion

on a vu dans ce chapitre les outils de programmation ensuite le travail effectué et finalement l'évaluation de notre modèle CNN proposé et notre résultat .

Chapitre 6

Conclusion and Perspectives

6.1 Conclusion

Dans cette mémoire, nous avons exprimé la qualité de pronostic comme étant un processus indispensable dans une stratégie de maintenance prédictive. L'objectif du pronostic est de suivre et d'anticiper le comportement des équipements afin d'être capable de réagir intelligemment tout en réduisant les coûts de maintenance inutiles et en évitant l'arrêt des équipements. Cette mémoire vise ainsi à contribuer au développement d'outils de pronostic industriel en introduisant les nouvelles technologies 4.0 l'industrie 4.0.

L'introduction du paradigme de l'apprentissage profond permet d'utiliser les architectures de l'apprentissage profond comme CNN donc une approche où on peut utiliser l'apprentissage profond et le pronostic phm ensemble.

Dans ce travail de mémoire, nous avons introduit les notions de Prognostics and Health Management et l'apprentissage profond.

Ainsi, nous avons conçu une architecture dédiée à cette solution puis détaillé les différents blocs de cette architecture. Après avoir choisi l'approche basée données dans notre mémoire, nous avons utilisé trois méthodes : le réseau de neurones artificiels, afin d'estimer le RUL d'un ensemble des machines. Les données utilisées proviennent du référentiel de données du centre d'excellence en pronostic de la National Aeronautics and Space Administration (NASA).

Dans le but d'étudier et discuter la performance de notre système, une étude comparative entre les méthodes a été réalisée en calculant la précision. Pour évaluer l'approche proposée

Le travail effectué nous a permis de proposer une nouvelle approche de la maintenance prédic-

tive qui est l'apprentissage profond.

6.2 Perspectives

Dans cette mémoire, nous avons contribué au pronostic industriel en développant un système «Prognostic dans l'apprentissage profond ». Plusieurs perspectives liées à ces travaux sont actuellement en phase d'amélioration :

- Implémenter une nouvelle solution « PHM dans l'apprentissage profond » pour fournir tout le cycle PHM
- Appliquer la solution proposée à un cas réel.

Bibliographie

- [1] Les applications de l'internet des objets dans l'industrie. <https://blog.iot-valley.fr/applications-internet-objets-industrie>.
- [2] Long short-term memory (lstm). <https://www.techopedia.com/definition/33215/long-short-term-memory-lstm>.
- [3] L'industrie 4.0 révolutionne la chaîne de production de a à z. <https://www.sfrbusiness.fr/room/internet-des-objets/industrie-4-0-revolution-chaine-production.html>.
- [4] What is numpy? <https://numpy.org/doc/stable/user/whatisnumpy.html>.
- [5] Matplotlib. <https://matplotlib.org/>.
- [6] Python : définition et utilisation de ce langage informatique. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/>.
- [7] Tensorflow. <https://www.lemagit.fr/definition/TensorFlow>.
- [8] Google colab. <https://ledatascientist.com/google-colab-le-guide-ultime/>.
- [9] What is keras. <https://deepai.org/machine-learning-glossary-and-terms/keras>.
- [10] Industrie 4.0 : la quatrième révolution industrielle. <https://www.mecalux.fr/blog/industrie-4-0>.
- [11] ALLAL Mohammed Anes. Utilisation du deep learning dans la radio cognitive. 52 :21–25, 2017-2018.

-
- [12] ALLAL Mohammed Anes. Utilisation du deep learning dans la radio cognitive. 52 :18, 2017-2018.
- [13] B. Benmammar. Apprentissage automatique. *rapport de recherche*, 1 :71.
- [14] Zohra bouzidi. Pronostic des systèmes industriels basé sur l'intelligence artificielle. 168(17), 2018-2019.
- [15] ean-Daniel WEISZ Dorothée KOHLER. Industrie 4.0 : comment caractériser cette quatrième révolution industrielle et ses enjeux? *Industrie 4.0 : comment caractériser cette quatrième révolution industrielle et ses enjeux?*, 1(1), Nov 2016.
- [16] Soheyb Ayad Nourddine Zerhouni Ikram Remadna, Labib Sadek Terrissa. Rul estimation enhancement using hybrid deep learning methods. *International Journal of Distributed Systems and Technologies (IJDST)*, 19 :4, 2021.
- [17] CAROLINE QUENEDEY PATRICIA VERRIÈRE. L'industrie 4 .0 la 4Ème rÉvolution industrielle sauvera-t- elle l'industrie franÇaise? *Kurt Salmon*, pages 1–7, 2015.
- [18] Alexandre Sarazin, Sébastien Truptil, Aurelie Montarnal, and Jacques Lamothe. Système de systèmes dans les architectures phm. In *MOSIM'18-12ème Conférence internationale de Modélisation, Optimisation et SIMulation*, pages 7–p, 2018.
- [19] A. Schmitt and B. Le Blanc. Les réseaux de neurones artificiels. *vol*, page 13.
- [20] zohra bouzidi. Pronostic des systèmes industriels basé sur l'intelligence artificielle maintenance prédictive. *Pronostic des systèmes industriels basé sur l'intelligence artificielle Maintenance prédictive*, 168 :19–20.
- [21] zohra bouzidi. Pronostic des systèmes industriels basé sur l'intelligence artificielle maintenance prédictive. 168 :25–32, 2018-2019.
- [22] zohra bouzidi. Pronostic des systèmes industriels basé sur l'intelligence artificielle maintenance prédictive. 168(23-24), 2018-2019.