

E.N.S.T

المدرسة الوطنية العليا للتكنولوجيا
Ecole Nationale Supérieure de Technologie

الجمهورية الجزائرية الديمقراطية الشعبية



Department of Industrial
Engineering & Maintenance

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

المدرسة الوطنية العليا للتكنولوجيا

Ecole nationale supérieure de technologie

Département : Génie Industriel Et Maintenance

Mémoire de fin d'études en vue de l'obtention du diplôme

D'Ingénieur d'état en

Génie industriel

- Thème -

**Exploration des techniques d'apprentissage supervisé
pour l'amélioration de la maintenance prédictive**

Réalisé par

CHERIFI Lamia

AISSANI Manel

Les membres de Jury :

GOURI Rabah	M.C.A	Président
BELAYADI Djahida	M.A.B	Promotrice
Rezgui Wail	M.C.B	Examineur

Alger, le 25/06/2023

Année universitaire 2022 –2023

“

Dédicace

Du fond de mon cœur, je souhaite dédier ce travail à tous ceux qui me sont chers,

À ma chère mère, B.Djamila, je tiens à exprimer ma profonde gratitude pour son soutien indéfectible, son amour inconditionnel et ses précieux conseils tout au long de mes études. Que ce modeste travail soit l'accomplissement de vos souhaits les plus chers, le fruit des innombrables sacrifices que vous avez consentis pour moi.

En mémoire de mon très cher père, j'espère que cet humble geste sera perçu comme une marque de reconnaissance de la part d'une fille qui a toujours prié pour le salut de son âme.

À mes chères sœurs et frères, je vous remercie pour vos conseils et vos encouragements constants tout au long de mon parcours académique.

À mes chères amies, A. Djawher, D. Hassiba, B. Manel Nesrine, B. Sonia, A. Nesrine, votre amitié précieuse a illuminé mes journées et m'a apporté un soutien indéfectible. Votre présence m'a rappelé la beauté de l'amitié et m'a donné la force de persévérer.

Et enfin, à mon binôme, Manel, je suis reconnaissante de ta patience et de ta compréhension tout au long de notre collaboration.

Que ces mots témoignent de toute l'affection que je vous porte et de la gratitude infinie que j'éprouve envers chacun de vous. Vous avez fait une différence dans ma vie et je suis honorée de vous avoir à mes côtés.

Lamia

”

“

Dédicace

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

A mon cher père, Abdenaceur, qui m'a inculqué le sens de la responsabilité et qui a toujours été présent à mes côtés pour me guider vers le bon chemin.

A ma chère maman, Lila, pour son soutien moral et physique quotidien pendant toutes ces années, elle qui a toujours su m'encourager et me pousser d'aller vers l'avant, Merci infiniment.

A mes chères sœurs, Nesrine et Yasmine, qui n'ont pas cessé de me conseiller, m'encourager et me soutenir tout au long de mes études.

A mon cher frère, Aymen, et mon beau-frère, Hamza, pour leurs appuis et leurs encouragements.

Sans oublier Mon binôme, Lamia, pour sa patience et son sens de la responsabilité tout le long des trois années que nous avons passé ensemble à étudier.

Mes chères amies, A. Nesrine, B. Sonia, Z. Romaiassa et O. Maroua, qui m'ont toujours encouragé, et à qui je souhaite plus de succès. Merci pour leurs amours et leurs encouragements.

Manel

”

“

Remerciement

Nous tenons à adresser nos remerciements à toutes les personnes qui nous ont assistés, aidés et contribués à la réalisation de ce mémoire de fin d'études.

En particulier, nous tenons à exprimer notre profonde gratitude envers les membres du jury d'avoir accepté d'évaluer notre travail.

Nous sommes profondément reconnaissants envers Mme Belayadi Djahida, notre promotrice dévouée. Nous lui sommes reconnaissants de sa présence inspirante et de son accompagnement bienveillant tout au long de notre projet. Son expertise et ses conseils précieux ont été essentiels à notre réussite.

À nos professeurs, nous sommes extrêmement reconnaissants de leur engagement envers notre formation. Leur passion pour l'enseignement et leur volonté de partager leurs connaissances nous ont grandement inspirés. Leurs conseils avisés, leurs encouragements et leurs commentaires constructifs nous ont permis de progresser et de repousser nos limites.

Nous tenons également à remercier chaleureusement l'ensemble de l'équipe pédagogique et administrative qui a contribué à notre formation.

Ce PFE a été une expérience enrichissante et nous sommes conscients que cela n'aurait pas été possible sans votre précieuse aide. Votre passion, votre expertise et votre dévouement ont été des sources d'inspiration pour nous tous.

Nous vous adressons nos remerciements les plus sincères pour avoir cru en nous, nous avoir guidés et nous avoir donné les outils nécessaires pour réussir. Nous sommes fiers de vous avoir comme mentors et professeurs.

Que nos réussites communes continuent d'inspirer de futurs étudiants et que vous soyez récompensés pour vos efforts inlassables.

Avec une gratitude infinie,

CHERIFI Lamia, AISSANI Manel

”

Table des matières

Introduction générale	2
1 État de l'art de la Maintenance Prédictive	4
1.1 Introduction	5
1.2 La maintenance	5
1.2.1 Les objectifs de la maintenance	6
1.3 Les types de la maintenance	6
1.3.1 Maintenance corrective	7
1.3.2 Maintenance Préventive	7
1.3.3 Maintenance prévisionnelle	8
1.3.4 La philosophie sous-tendant chaque type de maintenance	8
1.4 Analyse approfondie de la maintenance prédictive	9
1.4.1 Outils de la maintenance prédictive	9
1.4.2 Les 5 étapes cruciales de la maintenance prédictive	10
1.4.3 Techniques de la maintenance prédictive	11
1.5 Étude comparative des différents types de maintenance	12
1.5.1 Avantages et enjeux de la maintenance prédictive	13
1.6 Conclusion	14
2 Apprentissage automatique	15
2.1 Introduction	16
2.2 Intelligence Artificielle	16
2.3 Apprentissage Automatique	17
2.4 Types d'apprentissage automatique	18
2.5 Apprentissage Automatique supervisé	18
2.6 Algorithmes d'apprentissage automatique supervisé	20
2.6.1 KNN	20
2.6.2 Arbre de décision	20
2.6.3 Forêt aléatoire	21
2.6.4 Machine à vecteur de support	21
2.6.5 Naive Bayes	22
2.7 Apprentissage automatique non-supervisé	24
2.7.1 K-means	24
2.7.2 Principal component analysis - PCA	24
2.8 Apprentissage par renforcement	25
2.9 Apprentissage automatique pour la maintenance predictive	26
2.9.1 Positionnement distinctif de notre travail	28

2.10	Choix des algorithmes d'apprentissage supervisé : Evaluation et sélection . . .	28
2.10.1	Critères d'évaluation des algorithmes d'apprentissage supervisé . . .	29
2.11	Conclusion	33
3	Mise en pratique : Résultats et analyses	34
3.1	Introduction	35
3.2	Description des données d'expérimentation	36
3.2.1	Premier dataset	36
3.2.2	Deuxième dataset	37
3.2.3	Troisième dataset	38
3.2.4	Quatrième dataset	39
3.3	Approche statistique utilisée pour comparer les performances des trois al- gorithmes	40
3.4	Méthodologie expérimentale	41
3.4.1	Préparation de données	41
3.4.2	Visualisation des données	42
3.4.3	Entraînement et évaluation des modèles	43
3.4.4	Random Forest	43
3.4.5	DT	44
3.4.6	KNN	44
3.4.7	Visualisation et Analyse des résultats	45
3.5	Tests et résultats	45
3.5.1	Résultats et discussions de la première base de données	45
3.5.2	Résultats et discussions de la deuxième base de données	48
3.5.3	Résultats et discussions de la troisième base de données	50
3.5.4	Résultats et discussions de la quatrième base de données	52
3.5.5	Comparaison entre les résultats des 4 ensembles de données	54
3.5.6	Comparaison avec d'autres travaux	57
3.5.7	Méthodologie de comparaison	57
3.6	Conclusion	59
	Conclusion et perspectives	60
	Bibliographie	63
	Annexes	67
	A	68
	74 ملخص	
	Résumé	75
	Abstract	76

Table des figures

1.1	Gestion de la maintenance : Maintenir, Rétablir et Prévenir pour une efficacité maximale	5
1.2	Les différentes approches de maintenance	7
1.3	Démarche de diagnostic de la maintenance prédictive [24]	11
1.4	Avantages et enjeux de la maintenance prédictive	14
2.1	Les différents types d'apprentissage automatique	17
2.2	Problèmes de classification et de régression en apprentissage supervisé . .	19
2.3	Exemple d'application de l'algorithme KNN	20
2.4	Exemple d'application de l'algorithme RF	21
2.5	Exemple d'application de l'algorithme SVM	22
2.6	Exemple d'application de l'algorithme Naive Bayes	22
2.7	Types de problèmes en apprentissage automatique non supervisé	24
2.8	Revue littérature des critères d'évaluation	30
2.9	Les algorithmes d'apprentissage automatique les plus performants , résultats sur 8 critères	32
3.1	Etapes de préparation de données	42
3.2	Visualisation des données	42
3.3	Code du RF	43
3.4	Code de DT	44
3.5	Code de KNN	45
3.6	Matrices de confusion des trois algorithmes pour évaluer les prédictions réalisées sur la première base de données.	46
3.7	Comparaison des métriques des trois algorithmes pour évaluer les prédictions réalisées sur la première base de données.	46
3.8	Récapitulatif des métriques pour les trois algorithmes de la première base de données.	47
3.9	Matrices de confusion des trois algorithmes pour évaluer les prédictions réalisées sur la deuxième base de données.	48
3.10	Comparaison des métriques pour trois algorithmes pour évaluer les prédictions réalisées sur la deuxième base de données.	48
3.11	Récapitulatif des métriques pour les trois algorithmes de la deuxième base de données.	49
3.12	Matrices de confusion des trois algorithmes pour évaluer les prédictions réalisées sur la troisième base de données.	50
3.13	Comparaison des métriques pour les trois algorithmes pour évaluer les prédictions réalisées sur la troisième base de données.	50

3.14	Récapitulatif des métriques pour les trois algorithmes de la troisième base de données.	51
3.15	Matrices de confusion des trois algorithmes pour évaluer les prédictions réalisées sur la quatrième base de données.	52
3.16	Comparaison des métriques pour les trois algorithmes pour évaluer les prédictions réalisées sur la quatrième base de données.	52
3.17	Récapitulatif des métriques pour les trois algorithmes de la quatrième base de données.	53
3.18	Comparaison des performances du RF dans les 4 ensembles de données pour les différentes métriques	54
3.19	Comparaison des performances de DT dans les 4 ensembles de données pour les différentes métriques	55
3.20	Comparaison des performances du KNN dans les 4 ensembles de données pour les différentes métriques	55
A.1	68
A.2	69
A.3	69
A.4	69
A.5	70
A.6	71
A.7	71
A.8	72
A.9	72
A.10	72
A.11	73

Liste des tableaux

1	Tableau des abréviations	1
1.1	Différents types de maintenance et leurs philosophies associées	8
1.2	Comparaison des techniques de maintenance prédictive : Analyse des vibrations, analyse de l'huile et thermographie [23]	12
1.3	Comparaison des types de maintenance : Préventive, Corrective et Prédictive	13
2.1	Algorithmes d'apprentissage automatique supervisé , avantages et limites .	23
2.2	Algorithmes K-means vs PCA - Applications, avantages et limites	25
2.3	Critères d'évaluation	29
2.4	Comparaison des performances des algorithmes de machine learning selon 8 critères d'évaluation	31
3.1	Caractéristiques et interprétations du premier dataset	36
3.2	Caractéristiques et interprétations du deuxième dataset	37
3.3	Caractéristiques et interprétations du Troisième dataset	38
3.4	Caractéristiques et interprétations du Quatrième dataset	39
3.5	Matrice de confusion[4]	40
3.6	Métriques d'évaluation des modèles	41
3.7	Performances des trois approches selon les différentes métriques d'évaluation	58

Liste des abréviations

Les principales notations et abréviations utilisées sont explicitées dans le tableau ci-dessous 1, sous leur forme la plus couramment employée dans le domaine industriel. 8

AI	Intelligence artificielle
DT	Arbre de décision
FN	False Negative
FP	False Positive
GMAO	Gestion de la maintenance assistée par ordinateur
IoT	Internet des objets
KNN	K plus proches voisins
ML	Apprentissage automatique
NN	Réseaux neuronaux
PDM	Maintenance prédictive
RF	Forêt aléatoire
SVM	Machines à vecteurs de support
TN	True Negative
TP	True Positive

TAB. 1 : Tableau des abréviations

Introduction générale

Dans le contexte de l'Industrie 4.0, où l'automatisation et la numérisation des processus industriels sont en constante évolution [20], la maintenance joue un rôle essentiel pour garantir des opérations efficaces et réduire les coûts associés aux temps d'arrêt imprévus [11]. La PDM émerge comme une approche prometteuse pour anticiper les défaillances et planifier les opérations de maintenance de manière proactive, afin d'optimiser les performances des équipements industriels [43].

Parallèlement, les avancées dans le domaine du ML ont ouvert de nouvelles perspectives pour la PDM. Le ML est une branche de l'IA qui permet aux systèmes informatiques d'apprendre à partir des données sans être explicitement programmés [31]. En exploitant les modèles et les algorithmes du ML, il devient possible de tirer des informations précieuses à partir de vastes ensembles de données, ce qui peut faciliter la prédiction des défaillances et l'optimisation des opérations de maintenance.

La question de recherche centrale de ce projet de fin d'études est de déterminer comment choisir le meilleur algorithme du ML pour la PDM. Avec une grande variété de techniques et d'approches disponibles, il est crucial de comprendre quel algorithme ou combinaison d'algorithmes peut offrir les meilleurs résultats.

Avant d'entamer notre démarche de travail, il est crucial de prendre en compte les recherches antérieures qui se sont intéressées à l'application des algorithmes du ML pour la PDM. Parmi ces études [33, 8, 26, 34, 17], ont exploré l'application des algorithmes du ML. Cependant, ces études présentent certaines lacunes, par exemple, elles n'ont pas fourni d'informations spécifiques sur les résultats de l'implémentation des modèles de ML. De plus, ces études ne discutent pas en profondeur du processus de sélection des algorithmes pour l'application souhaité.

En revanche, notre travail se distinguera des études de cas mentionnées précédemment. Nous nous concentrerons sur une évaluation théorique initiale des algorithmes, suivie d'une évaluation pratique sur quatre jeux de données distincts. Cette approche nous permettra de fournir des informations précieuses sur les performances spécifiques de ces algorithmes dans le contexte de la PDM.

Le PFE est organisé en trois chapitres

Le premier chapitre examine l'état de l'art de la maintenance prédictive, en mettant l'accent sur les concepts clés de la maintenance industrielle et ses différentes approches. Ce chapitre explore également les outils, les techniques et les méthodologies de la PDM, en mettant en évidence leurs avantages et enjeux pour les entreprises.

Le deuxième chapitre plonge dans le domaine du ML, en présentant ses différents types, tels que l'apprentissage supervisé, l'apprentissage non-supervisé et l'apprentissage par renforcement. Ce chapitre met également l'accent sur l'application du ML à la PDM et présente plusieurs algorithmes d'apprentissage automatique couramment utilisés dans ce domaine. L'objectif est de sélectionner les algorithmes les plus performants pour l'implémentation.

Le troisième chapitre se concentre sur l'évaluation des performances de ces algorithmes RF, DT et KNN pour la PDM. Ce chapitre décrit en détail les quatre ensembles de données utilisés dans l'expérimentation et présente une méthodologie rigoureuse pour évaluer les performances des algorithmes. Les résultats obtenus sont comparés à des études antérieures dans le domaine de la PDM, permettant ainsi d'évaluer la pertinence des algorithmes sélectionnés.

Chapitre 1

État de l'art de la Maintenance Prédictive

1.1 Introduction

Dans un contexte concurrentiel de plus en plus exigeant, les entreprises industrielles cherchent à minimiser leurs coûts de production pour rester compétitives. Les coûts liés aux arrêts de production causés par les pannes d'équipements, de lignes de production et de systèmes de production sont particulièrement préoccupants, car ils impactent directement le coût des biens produits et peuvent entraîner une baisse de compétitivité et une perte de parts de marché,[38].

Dans ce contexte, nous explorerons dans ce chapitre les concepts clés de la maintenance industrielle, en mettant l'accent sur la nécessité de minimiser les arrêts de production non programmés et sur l'importance des méthodes et outils de gestion de la maintenance pour assurer une performance optimale des équipements et une compétitivité accrue de l'entreprise.

Les deux approches principales de la maintenance sont : la maintenance corrective qui consiste à intervenir sur un équipement uniquement lorsqu'il tombe en panne, afin de le réparer et de le remettre en état de fonctionnement et la maintenance préventive qui vise à éviter les pannes en planifiant des opérations de maintenance en fonction de critères prédéterminés, afin d'anticiper les défaillances potentielles, [11].

1.2 La maintenance

La norme ISO 9000 définit la maintenance comme "l'ensemble des actions techniques, administratives et managériales au cours du cycle de vie d'un bien, visant à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir sa fonction requise" ISO 9000:2015. Cette définition englobe deux types principaux de maintenance : la maintenance préventive, qui consiste en des actions planifiées visant à prévenir les défaillances et à maintenir les performances, et la maintenance corrective, qui intervient lorsqu'une défaillance survient et nécessite une intervention pour rétablir le bon fonctionnement du bien [38], comme illustré dans la figure 1.1.

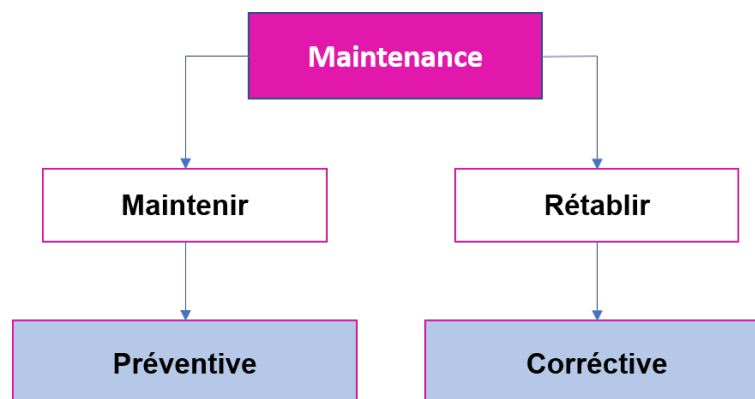


FIG. 1.1 : Gestion de la maintenance : Maintenir, Rétablir et Prévenir pour une efficacité maximale

1.2.1 Les objectifs de la maintenance

Le rôle de la fonction maintenance dans une entreprise est de garantir la plus grande disponibilité des équipements au meilleur rendement tout en respectant le budget alloué .[11]

La maintenance doit réaliser principalement deux types d'objectifs :

Objectifs Opérationnels

- Dans un état acceptable.
- Assurer la disponibilité maximale des installations à un prix raisonnable.
- Fournir un service qui élimine les pannes à tout moment.
- Penser à assurer une performance sûre et efficace à tout moment.
- Obtenir un rendement maximum pendant les prochaines années.
- Maintenir une installation d'une propreté absolue.

Objectif de coût

- Réduire au maximum les dépenses de la maintenance.
- Maximiser les profits.
- Assurer le service de maintenance dans les limites d'un budget.

1.3 Les types de la maintenance

La maintenance corrective et la maintenance préventive sont deux approches courantes de la gestion de la maintenance. La maintenance corrective intervient en cas de panne ou de dysfonctionnement, tandis que la maintenance préventive vise à anticiper les pannes en effectuant des activités de maintenance planifiées comme illustré dans la figure 1.2 . Chaque approche a ses avantages et ses limites, et une combinaison des deux peut être utilisée pour optimiser la gestion de la maintenance en fonction des besoins de chaque entreprise.[38]

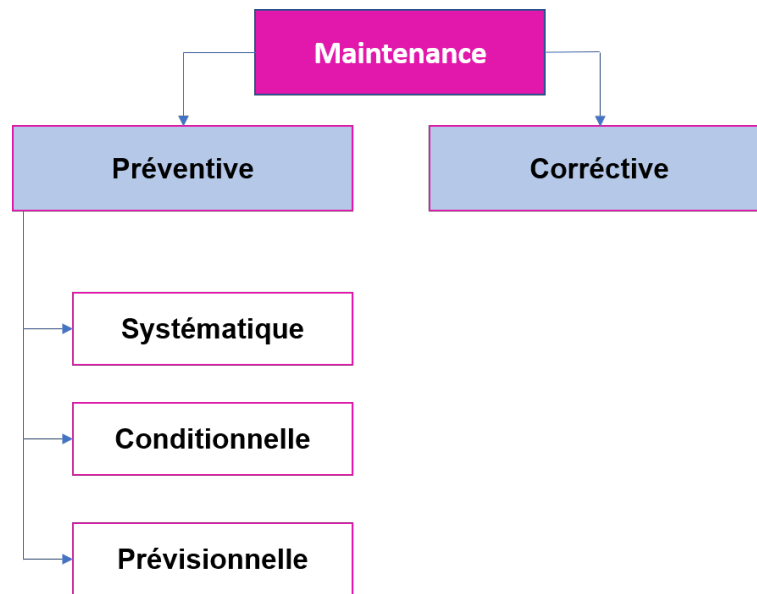


FIG. 1.2 : Les différentes approches de maintenance

1.3.1 Maintenance corrective

Elle est définie par la norme européenne NF EN 13306 X 60-319 comme suit : « Maintenance exécutée après détection d'une panne et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise », [6].

1.3.2 Maintenance Préventive

Elle est définie par l'AFNOR NF EN 13306 FD X 60-000 comme suit : « Maintenance exécutée à des intervalles prédéterminés ou selon des critères prescrits et destinés à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien »[14], elle est subdivisée en :

Maintenance systématique

Elle est définie par la norme européenne NF EN 13306 X 60-319 comme suit : Maintenance préventive exécutée à des intervalles de temps préétablis ou selon un nombre défini d'unités d'usage mais sans contrôle préalable de l'état du bien », [6].

Maintenance conditionnelle

Elle est définie par la norme européenne NF EN 13306 X 60-319 comme suit : « Maintenance préventive basée sur une surveillance du fonctionnement du bien et/ou des paramètres significatifs de ce fonctionnement intégrant les actions qui en découlent », [6].

1.3.3 Maintenance prévisionnelle

Parfois appelée **Maintenance Prédictive**, La maintenance prévisionnelle est, selon la norme NF EN 13306 X 60-319, une « maintenance conditionnelle exécutée en suivant les prévisions extrapolées de l'analyse et de l'évaluation de paramètres significatifs de la dégradation du bien ». Dans la section 1.4 nous allons examiner de plus près ce type de maintenance, [6].

1.3.4 La philosophie sous-tendant chaque type de maintenance

Différents types de maintenance existent, chacun avec sa propre philosophie. Le tableau suivant 1.1, présente les différents types de maintenance ainsi que leurs philosophies associées. Chaque type de maintenance offre une approche spécifique pour gérer les activités de maintenance et prévenir les défaillances des équipements.

Type de maintenance	La philosophie de chaque type
Systématique	La politique adoptée consiste à planifier les interventions de maintenance sur la base d'un calendrier prédéfini, en se basant sur les heures de fonctionnement de l'équipement ou un échancier établi à l'avance. Cette approche vise à anticiper les défaillances, mais peut présenter des limites si les interventions sont trop précoces ou trop tardives, entraînant ainsi l'arrêt de l'équipement sans défaillance réelle ou le remplacement prématuré de composants encore fonctionnels. [29]
Conditionnelle	La politique repose sur l'idée de se baser sur l'état de l'équipement plutôt que sur les indicateurs de surveillance, qui peuvent être peu fiables. Lorsque l'état de l'équipement se détériore au point d'atteindre un seuil critique, une intervention de maintenance est programmée pour éviter les arrêts. La maintenance conditionnelle permet de détecter les défaillances dès leur apparition, offrant ainsi une marge de manœuvre avant que l'équipement ne tombe en panne. [10]
Prévisionnelle (prédictive)	La politique de maintenance adoptée consiste à surveiller en temps réel les indicateurs pour anticiper les prévisions en extrapolant les tendances des données. Différente de la maintenance conditionnelle, elle permet d'obtenir un état prédictible plutôt qu'en temps réel. La surveillance peut être effectuée en ligne (télésurveillance) ou hors ligne (par ronde). [43]

TAB. 1.1 : Différents types de maintenance et leurs philosophies associées

1.4 Analyse approfondie de la maintenance prédictive

La maintenance prédictive repose sur la surveillance en temps réel des équipements en utilisant des capteurs pour collecter des données, qui sont ensuite analysées par des algorithmes de prédiction basés sur le Machine Learning.

L'objectif de la maintenance prédictive est de maximiser l'intervalle entre deux interventions, réduire les arrêts non programmés et minimiser les pertes associées. Il s'agit d'un moyen d'optimiser la productivité et la qualité des produits tout en réduisant les coûts. Il est important de noter que certaines techniques telles que l'analyse vibratoire, l'analyse des huiles ou l'imagerie thermique sont souvent présentées à tort comme de la maintenance prédictive, alors qu'elles ne sont que des outils permettant de mettre en place une politique de maintenance prédictive.

Cette politique repose sur la surveillance des conditions de fonctionnement des équipements ou systèmes pour optimiser les opérations de l'entreprise. Un bon management de la maintenance prédictive utilise des outils de collecte de données tels que des capteurs pour proposer un planning de maintenance anticipée avant l'occurrence de la panne.

Un programme de maintenance prédictive consiste principalement en trois étapes : Acquisition de données, Traitement de données, Prise de décision.

1.4.1 Outils de la maintenance prédictive

La maintenance prédictive utilise divers outils pour optimiser la gestion des équipements industriels. Voici quelques-uns de ces outils [1] :

a. La GMAO (gestion de la maintenance assistée par ordinateur) : Il s'agit d'un logiciel qui permet de planifier, suivre et gérer les activités de maintenance de manière efficace. Il facilite la gestion des demandes d'intervention, l'organisation des tâches et la gestion des stocks de pièces de rechange.

b. L'IIOT (Internet Industriel des Objets) : Cette technologie permet de collecter des informations provenant de capteurs installés sur les machines et de les envoyer vers des plateformes Cloud. Cela permet d'analyser les données en temps réel, de détecter les anomalies et de prendre des décisions basées sur des informations précises. De plus, l'IIOT favorise l'interconnexion des équipements, ce qui facilite la communication entre eux.

c. Picomto : C'est une solution de digitalisation des instructions de travail. En plus de cela, Picomto a développé un interrupteur connecté intelligent appelé Check Start, qui améliore l'efficacité des équipements. Cet interrupteur permet de surveiller les machines et de recevoir des alertes en cas de dysfonctionnement.

d. Utilisation d'un PC portable, d'une tablette ou d'un smartphone connecté à Internet : Ces dispositifs permettent d'accéder à des données précises sur l'état des équipements et leur performance, même à distance. Ils permettent également de recevoir des alertes en cas de problème détecté. De plus, ils offrent la possibilité de régler certains paramètres, tels que les seuils d'alerte.

e. L'intelligence artificielle : Les avancées dans le domaine de l'intelligence artificielle permettent aujourd'hui de surveiller les différents composants d'une machine à l'aide de capteurs. Les données collectées sont interprétées par des applications informatiques qui prennent des décisions de manière automatique. Par exemple, il est possible de mesurer la température des équipements grâce à l'imagerie infrarouge, d'évaluer la viscosité du lubrifiant, de surveiller les états de pression pour le flux de fluide et d'air, ou encore de détecter les problèmes structurels tels que la fragilité, le désalignement ou les problèmes de roulement en mesurant la fréquence de vibration.

1.4.2 Les 5 étapes cruciales de la maintenance prédictive

La maintenance prédictive, basée sur l'analyse en temps réel des capteurs, suit un processus en plusieurs étapes pour optimiser la production industrielle :

1. **Surveillance continue des outils de production :** Les capteurs intégrés aux machines fonctionnent comme un système de surveillance, collectant en temps réel des données sur leur état et leur performance.
2. **Transmission des données :** Les données collectées par les capteurs sont ensuite transmises à un logiciel spécialisé, permettant leur traitement et leur analyse approfondie.
3. **Analyse des données :** Un technicien de maintenance utilise le logiciel pour analyser les données et déterminer la probabilité de défaillance d'une machine. Cette analyse permet également d'identifier les types de défaillances possibles.
4. **Anticipation des pannes :** Grâce à l'analyse des données, il devient possible d'anticiper les pannes potentielles. Le technicien peut ainsi prendre des mesures préventives pour éviter les interruptions coûteuses de la production.
5. **Planification de la maintenance :** En se basant sur les informations fournies par l'analyse des données, le technicien peut prévoir les opérations d'entretien nécessaires pour éviter les pannes. Cela permet de planifier de manière proactive les interventions de maintenance, évitant ainsi les arrêts imprévus de la production.

La maintenance prédictive, grâce à l'analyse en temps réel des capteurs, offre une approche proactive qui maximise l'efficacité et la fiabilité des outils de production. En anticipant les pannes et en planifiant la maintenance de manière préventive, elle permet d'optimiser la productivité tout en réduisant les coûts liés aux arrêts de production.[5]

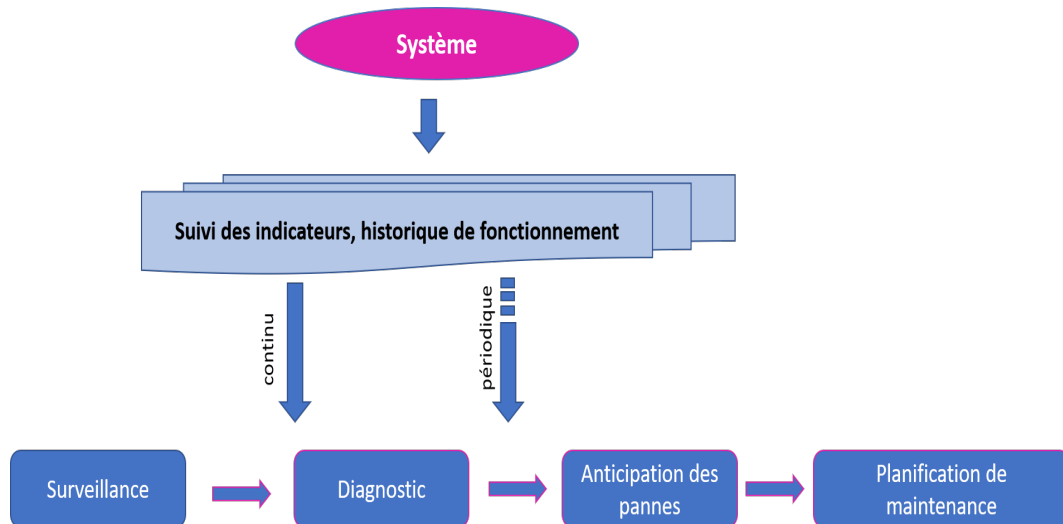


FIG. 1.3 : Démarche de diagnostic de la maintenance prédictive [24]

1.4.3 Techniques de la maintenance prédictive

La maintenance prédictive repose sur diverses techniques d'analyse pour détecter les défaillances potentielles des équipements avant qu'elles ne provoquent des pannes coûteuses. Trois méthodes couramment utilisées dans ce domaine sont l'analyse vibratoire, l'analyse thermique et l'analyse des huiles.

Analyse vibratoire

L'analyse vibratoire est une méthode de maintenance prédictive très efficace pour les systèmes électromécaniques en entreprise. Elle utilise des capteurs pour collecter les signaux de vibration et des programmes informatiques pour les évaluer. En comparant les données collectées à des données de référence saines, elle permet de détecter les défauts tels que le balourd, le désalignement, les problèmes de roulement, etc. Elle peut également détecter les vibrations engendrées par les fluides dans les pipelines. Cette technique peut prévenir les pannes plusieurs mois à l'avance avec un taux de fausses alarmes de 8

Analyse thermique

L'analyse thermique est fondée sur l'émission de lumière infrarouge par les objets en fonction de leur température, dont la relation est utilisée pour représenter les températures sous forme de couleurs. Toutefois, la forme, l'état de surface et l'angle de vue de l'objet peuvent affecter le taux de radiation. Cette méthode est couramment utilisée pour détecter les problèmes mécaniques et électriques, surtout pour les problèmes électriques tels que les connexions corrodées, les fuites de chauffage et de climatisation, les problèmes de charge et la défaillance des composants.

Analyse des huiles

L'analyse physique et chimique des huiles et lubrifiants est une méthode précieuse pour surveiller l'état des équipements et détecter les défaillances. Elle consiste en deux types d'analyses : l'analyse des particules qui vérifie l'état des composants mécaniques, et l'analyse de l'huile qui détermine s'il peut être utilisé ou remplacé en fonction de ses propriétés telles que la viscosité, le PH, la contenance en eau et le point d'inflammabilité. Cette méthode peut être effectuée en ligne ou hors ligne, et est souvent utilisée en complément de l'analyse vibratoire pour une maintenance prédictive plus complète.

Nous comparons l'analyse des vibrations, l'analyse de l'huile et la thermographie, trois techniques de maintenance prédictive largement utilisées, dans le tableau suivant.1.2

Technique de maintenance prédictive	Avantages	Inconvénients
Analyse des vibrations	Permet de détecter les défauts en temps réel et d'identifier les problèmes avec précision	Nécessite des compétences spécialisées et un équipement coûteux
Analyse de l'huile	Permet de détecter les défauts avant qu'ils ne se produisent, ne nécessite pas l'arrêt de l'équipement	Peut être coûteux, peut nécessiter une longue période d'analyse
Thermographie	Permet de détecter les problèmes avant qu'ils ne surviennent, ne nécessite pas de contact avec l'équipement	Peut être affectée par les conditions environnementales, ne peut pas détecter tous les types de défauts

TAB. 1.2 : Comparaison des techniques de maintenance prédictive : Analyse des vibrations, analyse de l'huile et thermographie [23]

1.5 Étude comparative des différents types de maintenance

Le choix d'un type de maintenance approprié est un élément essentiel pour maintenir la performance et la disponibilité des équipements industriels. [10] Les différents types de maintenance ont des coûts, des avantages et des limites distincts, et choisir le type de maintenance qui convient le mieux à l'entreprise peut s'avérer difficile.[18] Pour prendre une décision éclairée, nous avons établi une comparaison entre les types de maintenance les plus couramment utilisés. Cette étude comparative est présentée dans le tableau ci-dessous 1.3, avec des informations détaillées sur chaque type de maintenance.

Types	Caractéristiques	Avantages	Inconvénients	Ref
Préventive	Intervalles de maintenance programmés en fonction du temps, de l'utilisation ou de la surveillance de l'état	Réduit la probabilité de temps d'arrêt imprévus, prolonge la durée de vie des équipements, renforce la sécurité et améliore la productivité	Peut être coûteux et conduire à une maintenance inutile, ne tient pas compte des défaillances imprévues	[29], [10], [18]
Corrective	Approche réactive basée sur la défaillance d'un équipement	Coûts initiaux moins élevés, maintenance effectuée uniquement en cas de besoin	Les temps d'arrêt peuvent être importants, l'équipement peut être endommagé au point d'être irréparable, des problèmes de sécurité peuvent survenir	[29], [10], [18]
Prédictive	Contrôler l'équipement en temps réel pour détecter les changements de performance, identifier les problèmes potentiels et programmer la maintenance en conséquence	Minimise les temps d'arrêt imprévus, réduit les coûts de maintenance, optimise les performances et augmente la sécurité	Nécessite des équipements et des logiciels spécialisés, peut être complexe à mettre en œuvre, peut ne pas prendre en compte tous les modes de défaillance	[29], [39], [2], [43]

TAB. 1.3 : Comparaison des types de maintenance : Préventive, Corrective et Prédictive

1.5.1 Avantages et enjeux de la maintenance prédictive

La figure 1.4 illustre les principaux avantages de la maintenance prédictive par rapport à d'autres approches. Elle permet également de visualiser les enjeux inhérents à la mise en place de la maintenance prédictive.

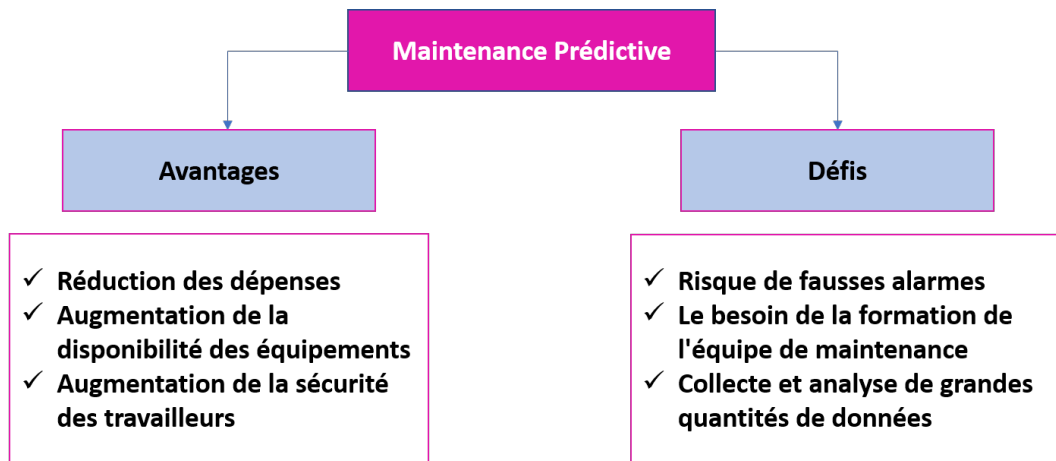


FIG. 1.4 : Avantages et enjeux de la maintenance prédictive

1.6 Conclusion

En conclusion, ce chapitre sur l'état de l'art de la Maintenance Prédicative a fourni une base solide pour comprendre les concepts fondamentaux liés à la maintenance. Nous avons exploré les différents types de maintenance, notamment la maintenance corrective, préventive et prévisionnelle, ainsi que la philosophie sous-tendant chaque type.

Une attention particulière a été accordée à la maintenance prédictive, avec une analyse approfondie de ses outils, étapes et de ses techniques. Nous avons également réalisé une étude comparative des différents types de maintenance, mettant en évidence les avantages et les enjeux spécifiques de la maintenance prédictive.

Ce chapitre nous a permis de prendre conscience de l'importance croissante de la maintenance prédictive dans la gestion efficace des actifs industriels. À travers notre projet de fin d'études, nous avons l'intention de se concentrer sur l'application de la maintenance prédictive et d'explorer comment les modèles de machine learning peuvent améliorer l'efficacité et la fiabilité de cette approche.

Chapitre 2

Apprentissage automatique

2.1 Introduction

Dans ce chapitre, nous allons plonger dans le monde de l'apprentissage automatique et explorer ses différentes facettes. Nous allons découvrir les types d'apprentissage automatique, notamment l'apprentissage supervisé, l'apprentissage non-supervisé et l'apprentissage par renforcement. Chacun de ces types offre des approches uniques pour traiter les données et tirer des conclusions précieuses.

Plus précisément, nous nous concentrerons sur l'application de l'apprentissage automatique à la maintenance prédictive. Nous explorerons également plusieurs algorithmes d'apprentissage automatique couramment utilisés dans la maintenance prédictive. Chacun de ces algorithmes possède ses propres caractéristiques et forces, et nous les évaluerons en fonction de critères spécifiques pour déterminer leur pertinence dans des contextes particuliers.

Enfin, nous aborderons l'évaluation et la sélection des algorithmes d'apprentissage supervisé. Nous examinerons les critères d'évaluation couramment utilisés pour mesurer les performances des algorithmes, tels que la précision, le rappel, la courbe ROC, etc. Nous présenterons également trois algorithmes spécifiques que nous avons choisis pour illustrer ces concepts.

2.2 Intelligence Artificielle

L'intelligence artificielle (IA) est un domaine scientifique qui vise à permettre à une machine d'accomplir des tâches nécessitant normalement l'intelligence humaine. Le terme "intelligence artificielle" a été introduit en 1956, mais on peut aussi utiliser le terme d'"informatique heuristique". Dans ce domaine, on ne parle pas de machines ou de programmes intelligents,[27].

L'informatique est la science du traitement de l'information, tandis que l'IA se concentre sur les cas où ce traitement ne peut pas être réduit à une méthode simple, précise et algorithmique. Un algorithme est une série d'opérations ordonnées, clairement définies et exécutables sur un ordinateur, permettant d'arriver à une solution dans un délai raisonnable (minutes, heures, etc., mais pas des siècles!,[12].

2.3 Apprentissage Automatique

Selon Michael Mitchell, Le Machine Learning est un sous-ensemble de l'intelligence artificielle qui permet aux programmes informatiques d'apprendre à partir de données et devenir plus intelligents au fil du temps. Les algorithmes de prédiction efficaces et précis utilisent les données pour découvrir des modèles et optimiser les opérations du programme. Pour cela, les données mises à la disposition de l'apprenant sont cruciales pour le succès des prévisions. Les techniques d'apprentissage sont des méthodes basées sur des données qui combinent des concepts informatiques fondamentaux avec des principes de statistiques, de probabilités et d'optimisation. L'apprentissage automatique utilise différents types de méthodes, telles que l'apprentissage supervisé, non supervisé et par renforcement, pour améliorer les performances et les prévisions précises en réponse à des questions posées avec des données. Ainsi, l'analyse des données et les statistiques sont intrinsèquement liées au Machine Learning, et cette technologie est devenue importante en raison de sa capacité à traiter d'énormes quantités de données et à en extraire des informations précieuses. Le schéma 2.3 met en évidence les différents types d'apprentissage automatique de manière visuelle et explicite.

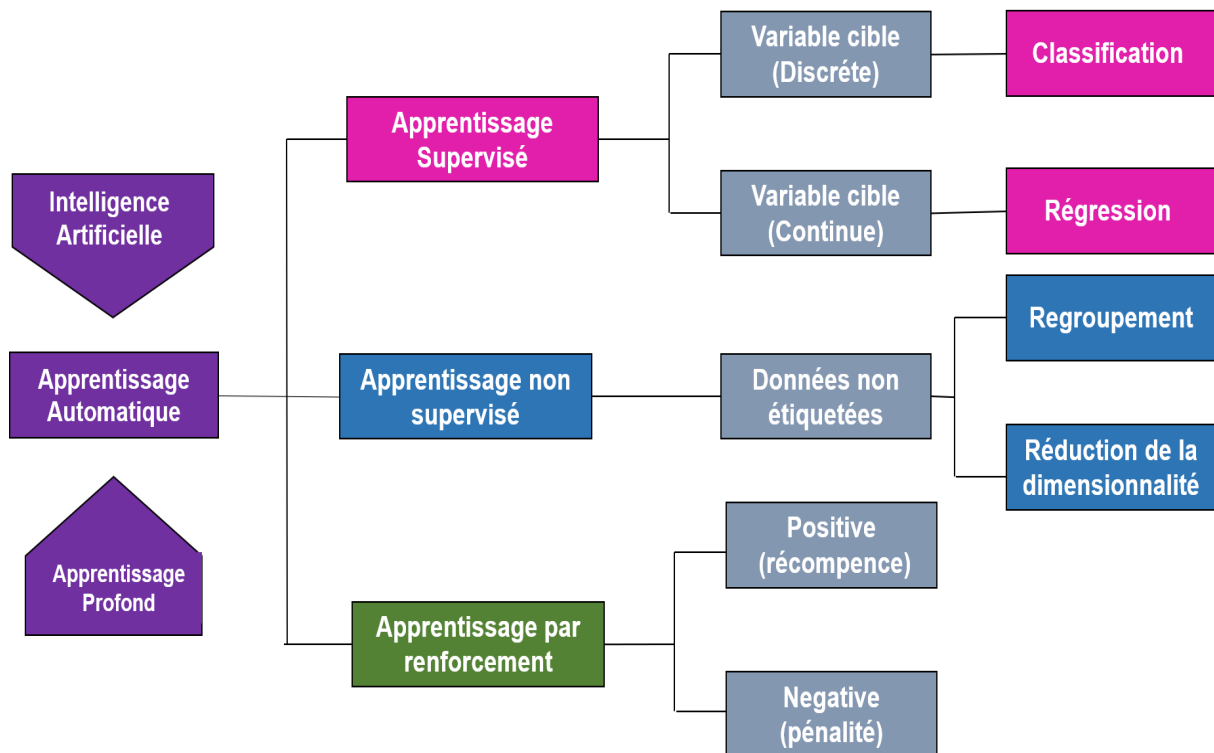


FIG. 2.1 : Les différents types d'apprentissage automatique

2.4 Types d'apprentissage automatique

Il existe différents types d'apprentissage automatique, chacun avec ses propres caractéristiques et applications. Les trois principaux types sont l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement. Chacun de ces types offre des approches uniques pour résoudre des problèmes complexes et contribue à l'avancement de l'intelligence artificielle et de nombreux domaines d'application.

2.5 Apprentissage Automatique supervisé

L'apprentissage automatique supervisé est une technique qui utilise des données étiquetées pour prédire des résultats. Au cours du processus de formation, les algorithmes d'apprentissage automatique développent une fonction capable de prédire les valeurs de sortie sur la base des données d'entrée[32]. Le modèle est affiné en comparant les résultats prédits aux résultats réels et en procédant à des ajustements en fonction des divergences. Essentiellement, l'apprentissage supervisé utilise les données existantes pour faire des prédictions sur des événements futurs[8].

Les tâches supervisées les plus courantes sont la "classification", qui sépare les données en catégories distinctes, et la "régression", qui ajuste les données à une fonction continue, comme illustré dans la figure 2.2. Par exemple, dans la classification, un modèle peut être formé pour distinguer les courriels indésirables des courriels légitimes en se basant sur des exemples étiquetés. Dans la régression, un modèle peut être utilisé pour prédire le prix d'une maison en fonction de ses caractéristiques telles que la superficie, le nombre de chambres, etc.

L'apprentissage supervisé est utilisé lorsque des objectifs spécifiques sont identifiés et peuvent être atteints à partir d'un ensemble d'entrées spécifié[33].

Il trouve des applications dans de nombreux domaines tels que la reconnaissance vocale, la détection de fraudes, la prédiction de demandes de produits, la prédiction de maladies et bien d'autres. Grâce à l'apprentissage supervisé, les modèles peuvent généraliser à partir des données d'entraînement pour effectuer des prédictions précises sur de nouvelles données non vues auparavant. Cela ouvre un large éventail de possibilités pour résoudre des problèmes complexes et prendre des décisions basées sur des modèles prédictifs fiables.

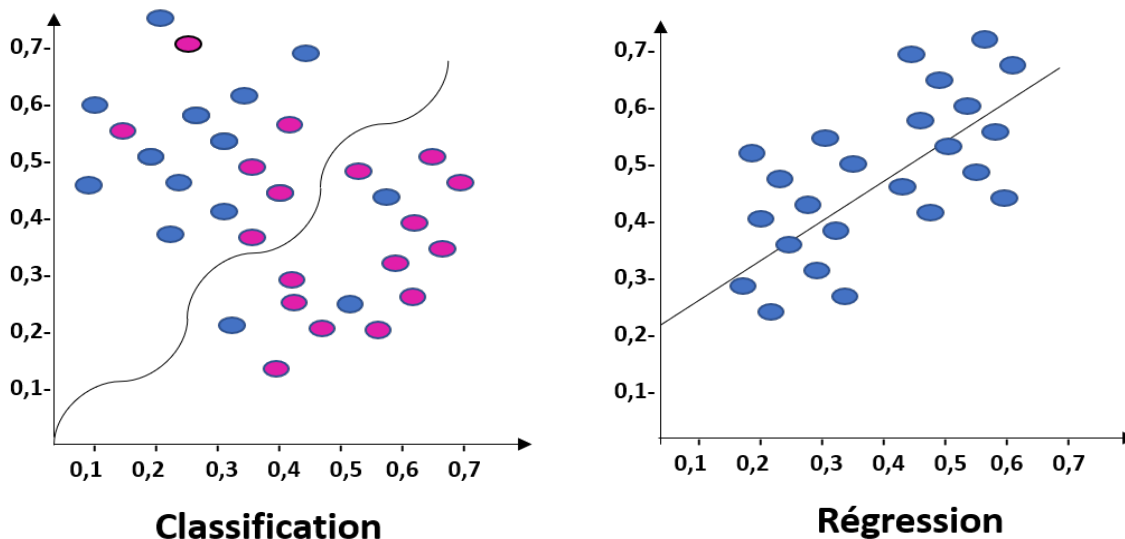


FIG. 2.2 : Problèmes de classification et de régression en apprentissage supervisé

La classification En apprentissage automatique, la classification consiste à prédire une étiquette ou une catégorie pour un exemple donné. Cela permet de regrouper les données en classes distinctes en fonction de leurs caractéristiques.

La classification binaire est utilisée lorsque l'on souhaite attribuer des étiquettes de classe "vrai" ou "faux" à des exemples.

La classification multi-classe , quant à elle, permet de classer les exemples parmi plusieurs classes distinctes sans distinction de vrai ou faux.

la classification multi-labels est une généralisation de la classification multi-classe où un exemple peut être associé à plusieurs étiquettes de classe simultanément. [33]

La régression La régression est une technique statistique et d'apprentissage automatique utilisée pour prédire une variable de réponse en fonction d'une ou plusieurs variables prédictives. Le processus implique l'estimation d'une valeur à partir de données d'entrée, et les modèles de régression peuvent être utilisés pour prédire des valeurs continues. Il existe de nombreux types de modèles de régression, notamment la **régression linéaire simple**, la **régression linéaire multiple**, la **régression polynomiale**, la régression logistique et la régression des séries chronologiques, chacun ayant ses propres avantages et limites. Le surajustement est un problème potentiel dans l'analyse de régression, qui peut être atténué par diverses techniques telles que la régularisation, la validation croisée et la sélection des caractéristiques.[40]

2.6 Algorithmes d'apprentissage automatique supervisé

2.6.1 KNN

Le K-voisin le plus proche (KNN) est l'un des premiers algorithmes d'apprentissage automatique supervisé qui peut être utilisé à la fois pour les problèmes de classification et de régression. Cependant, il est surtout utilisé comme algorithme de classification qui détermine la proximité des points de données pour classer ou prédire comment un point de données particulier sera regroupé. Dans l'algorithme KNN, les "k" voisins les plus proches sont déterminés sur la base de la métrique de distance, et la classe ou la prédiction de régression du point de données cible est déterminée par la classe la plus courante ou la valeur moyenne des "k" voisins les plus proches. L'algorithme KNN est non paramétrique, ce qui signifie qu'il ne fait aucune hypothèse sur la distribution sous-jacente des données. En outre, le KNN peut traiter des problèmes de classification multi-classes, bien qu'il nécessite une sélection minutieuse de la valeur de "k" et de la métrique de distance pour obtenir des performances optimales. [16]

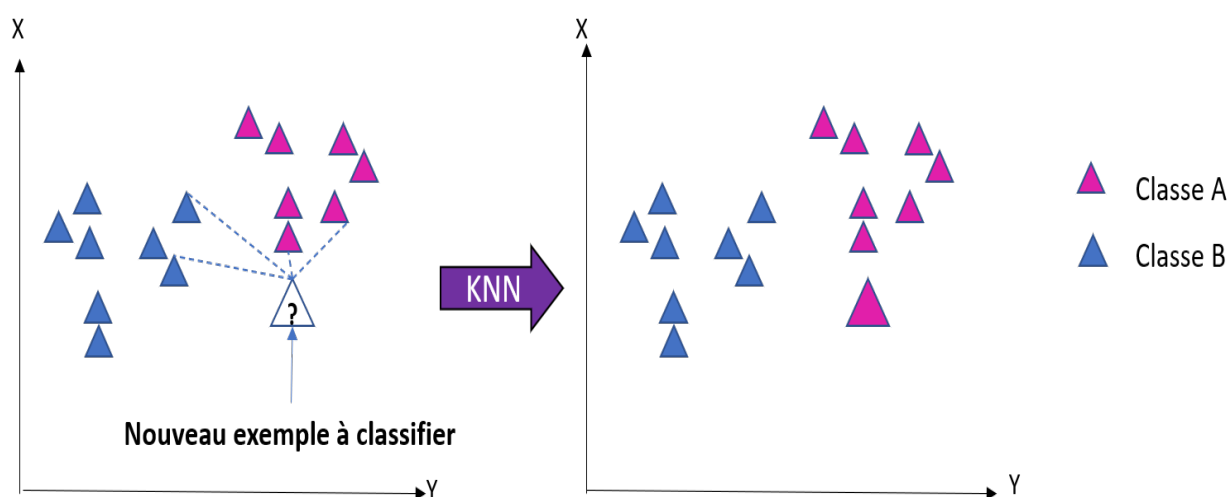


FIG. 2.3 : Exemple d'application de l'algorithme KNN

2.6.2 Arbre de décision

(*Decision Tree*) Un arbre de décision est une méthodologie populaire d'exploration de données utilisée pour créer des algorithmes de prédiction ou des systèmes de classification basés sur diverses variables d'entrée. La structure arborescente d'un arbre de décision ressemble à un organigramme, avec des nœuds internes rectangulaires et des nœuds foliaires ovales. Cette approche non paramétrique permet de traiter efficacement des ensembles de données vastes et complexes sans imposer un cadre paramétrique difficile. En utilisant cette technique, une population peut être divisée en branches qui forment un arbre inversé avec un nœud racine, des nœuds internes et des nœuds feuilles. Une fois que la taille de

l'échantillon est suffisante, les données de l'étude peuvent être divisées en ensembles de données de formation et de validation pour le développement et l'évaluation du modèle. [13]

2.6.3 Forêt aléatoire

(*Random Forest*) La forêt aléatoire est une méthode d'apprentissage ensembliste utilisée dans l'apprentissage automatique supervisé pour les tâches de classification et de régression. Elle a été introduite par Leo Breiman et Adele Cutler et est basée sur des arbres de décision comme classificateur principal. Dans cet algorithme, plusieurs arbres de décision sont formés sur différents sous-ensembles de l'ensemble de données original à l'aide d'une technique appelée agrégation bootstrap ou bagging. Les arbres ne sont ni élagués ni taillés, ce qui leur permet de s'adapter dans une certaine mesure à leurs données d'apprentissage. À chaque branche de l'arbre, un sous-ensemble aléatoire de caractéristiques est sélectionné pour diversifier davantage les classes, et ce processus est répété pour chaque point de ramification en utilisant un nouveau sous-ensemble aléatoire. La valeur suggérée pour la taille du sous-ensemble aléatoire est $\log_2(N+1)$, où N est la taille totale de l'ensemble de caractéristiques. [13]

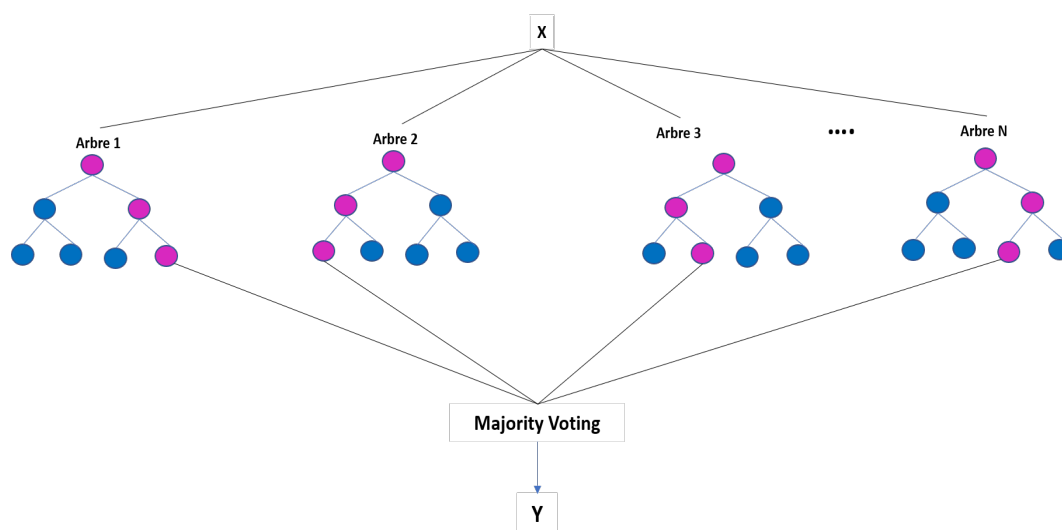


FIG. 2.4 : Exemple d'application de l'algorithme RF

2.6.4 Machine à vecteur de support

(*SVM*) La machine à vecteur de support (SVM) est l'une des méthodes d'apprentissage supervisé les plus efficaces, largement utilisée pour les tâches de classification et de régression. L'objectif de la SVM est de trouver la meilleure frontière de décision qui sépare les données en différentes classes tout en maximisant la marge entre la frontière et les points de données les plus proches. La frontière de décision est définie par un hyperplan, qui est un plan dans un espace à haute dimension qui sépare les données. Vapnik a été le premier à utiliser les SVM en 1992. [19]

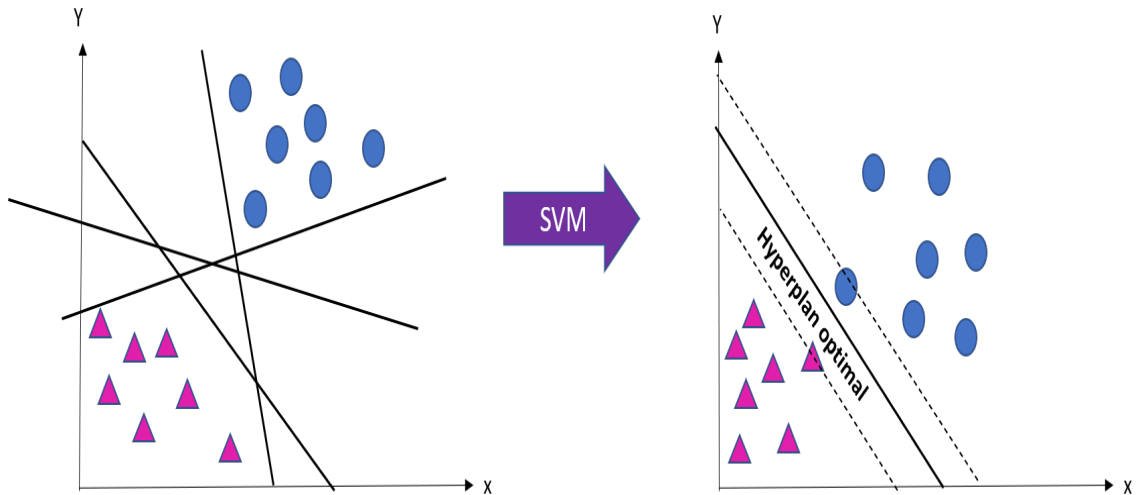


FIG. 2.5 : Exemple d'application de l'algorithme SVM

2.6.5 Naive Bayes

Naive Bayes est un algorithme probabiliste d'apprentissage automatique utilisé pour les tâches de classification. Il repose sur le théorème de Bayes, qui décrit la probabilité d'un événement en fonction de la connaissance préalable des conditions susceptibles d'être liées à l'événement. Naive Bayes part du principe que les caractéristiques des données sont conditionnellement indépendantes de l'étiquette de la classe, d'où son nom de "naïf". L'algorithme de Bayes naïf fonctionne en estimant d'abord les probabilités préalables de chaque classe, puis en estimant la probabilité de chaque caractéristique compte tenu de l'étiquette de la classe. Ces probabilités sont utilisées pour calculer les probabilités postérieures de chaque classe, et la classe ayant la probabilité la plus élevée est choisie comme classe prédite pour un nouveau point de données .[30]

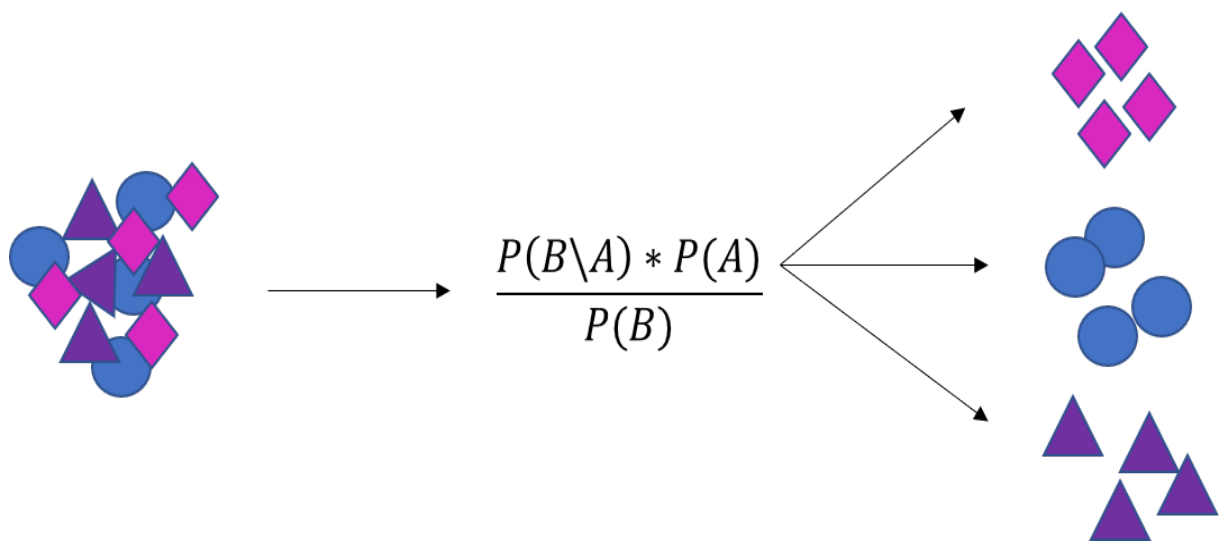


FIG. 2.6 : Exemple d'application de l'algorithme Naive Bayes

Ce tableau 2.1 présente un résumé des algorithmes d'apprentissage automatique supervisé que nous avons déjà abordés, en mettant en évidence leurs avantages et leurs limites. Chaque algorithme a ses propres caractéristiques et est adapté à des applications spécifiques dans divers domaines.

Algorithme	Avantages	Inconvénients	Applications dans la maintenance	Ref
KNN	Simple et facile à comprendre	Coûteux et sensible au choix de la valeur K	Prévision des défaillances, diagnostic des défauts, estimation de la durée de vie restante	[16], [36]
Arbres de décision	Facilité d'interprétation et de visualisation, possibilité de traiter des données numériques et catégorielles	Instable avec de petites variations dans les données	Diagnostic des pannes, prédiction des défaillances des équipements, maintenance prédictive	[16]
Forêts aléatoires	Réduire les surajustements, gérer les données manquantes, travailler avec de nombreuses caractéristiques d'entrée	difficile à interpréter et à visualiser	Prévision des défaillances de l'équipement, diagnostic des pannes	[9]
SVM	Efficace dans les espaces de grande dimension, l'astuce du noyau permet de gérer la non-linéarité	Sensible au choix du noyau, l'apprentissage peut être lent avec de grands ensembles de données	Détection des anomalies, surveillance des conditions, diagnostic des défaillances	[42], [36]
Naive Bayes	Rapide, fonctionne bien avec des données de haute dimension	Suppose l'indépendance des caractéristiques, peut être sensible aux caractéristiques non pertinentes	Détection des défaillances, contrôle de la qualité, prédiction des pannes d'équipement	[16]

TAB. 2.1 : Algorithmes d'apprentissage automatique supervisé , avantages et limites

2.7 Apprentissage automatique non-supervisé

L'apprentissage automatique non supervisé est une technique dans laquelle la structure de l'entrée est décidée par l'algorithme lui-même. Cette méthode ne met pas l'accent sur l'établissement de liens entre l'entrée et la sortie, contrairement à l'apprentissage supervisé. Son objectif premier est d'assembler des caractéristiques connexes en un seul groupe. L'objectif principal de l'apprentissage non supervisé, comme indiqué précédemment, est de regrouper des valeurs similaires les unes aux autres. Le regroupement et l'association sont les deux catégories de méthodes d'apprentissage automatique non supervisé comme illustré dans la figure 2.7. Ses algorithmes sont conçus pour trouver des modèles et des connexions cachés dans les données.[7],[40].

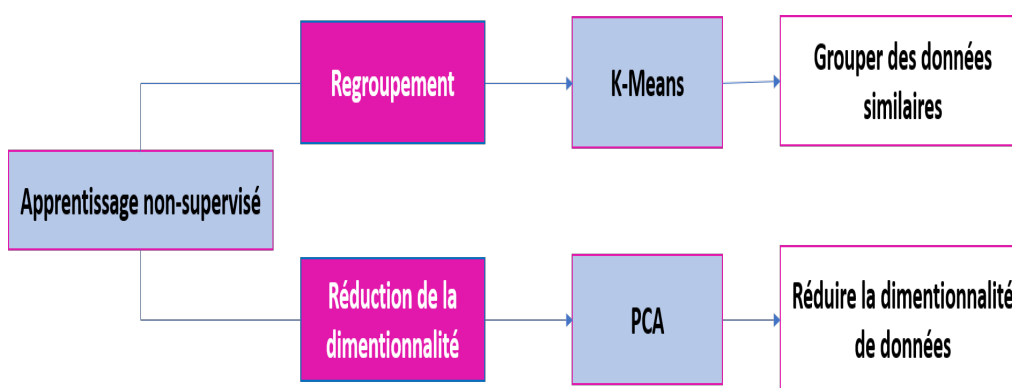


FIG. 2.7 : Types de problèmes en apprentissage automatique non supervisé

Des algorithmes spécifiques ont été créés pour résoudre certaines difficultés liées à l'apprentissage automatique non supervisé. Parmi ces algorithmes : L'analyse en composantes principales (ACP) pour les questions de réduction de la dimensionnalité et les k-means pour les questions de regroupement.

2.7.1 K-means

K-means est un algorithme d'apprentissage automatique non supervisé utilisé pour résoudre les problèmes de regroupement[25]. Cet algorithme itératif vise à classer un ensemble de données en le divisant en un nombre prédéterminé de clusters (k-clusters). Chaque cluster est un groupe de points de données qui partagent des caractéristiques similaires. [3] Les K-means sont largement utilisées dans diverses applications, telles que la segmentation d'images, la détection d'anomalies et la segmentation de la clientèle [16].

2.7.2 Principal component analysis - PCA

L'analyse en composantes principales (ACP) est l'un des principaux algorithmes d'apprentissage automatique non supervisé utilisés pour résoudre les problèmes de réduction de la dimensionnalité [15]. Elle est particulièrement efficace pour les bases de données

d'images qui n'ont pas d'étiquettes de classe sur les images. Dans l'ACP, une image d'apprentissage est traitée comme un vecteur aléatoire avec une distribution spécifique pour identifier les composantes primaires d'une image de visage. Ces composantes sont ensuite utilisées pour exprimer l'image du visage et atteindre l'objectif de réduction de la dimension [21].

Ce tableau 2.2 compare les avantages, les limites et les applications des algorithmes d'apprentissage non supervisé

Algorithme	Avantages	Limites	Applications dans la maintenance	Ref
K-means	Simple et rapide à mettre en œuvre Efficace sur les grands ensembles de données	Sensible à la sélection du centroïde initial Peut converger vers des optimums locaux	Détection des anomalies, surveillance des performances surveillance de la santé des machines	[35], [25]
PCA	Permet la réduction des données Identifie les principaux éléments de la complexité des données	Nécessite des relations linéaires entre les variables Ne peut pas traiter les données catégorielles ou nominales	Détection des anomalies, surveillance des performances, optimisation des processus	[15], [21]

TAB. 2.2 : Algorithmes K-means vs PCA - Applications, avantages et limites

2.8 Apprentissage par renforcement

L'apprentissage par renforcement est un type d'apprentissage automatique dans lequel un agent apprend à prendre des décisions dans un environnement en effectuant des actions et en recevant un retour d'information sous forme de récompenses ou de punitions [37]. Son objectif ultime est de prendre des mesures pour maximiser la récompense ou diminuer le risque en utilisant les informations obtenues des activistes de l'environnement. Il s'agit d'une technique efficace pour construire des modèles d'IA qui peuvent stimuler l'automatisation ou améliorer l'efficacité opérationnelle de systèmes complexes tels que les robots, la conduite autonome, la fabrication et la logistique de la chaîne d'approvisionnement, mais il n'est pas recommandé de l'utiliser pour s'attaquer à des problèmes simples ou fondamentaux. [33]

2.9 Apprentissage automatique pour la maintenance prédictive

Ces dernières années ont vu une augmentation significative de l'activité de recherche dans le domaine de la maintenance prédictive, qui a des applications dans plusieurs industries. Dans cet aperçu de la littérature, nous parlerons de certaines des recherches importantes qui ont évalué l'efficacité de divers modèles d'apprentissage automatique pour la maintenance prédictive.

Nous présentons ici les résultats essentiels, la méthodologie et les modèles d'apprentissage automatique de certains des travaux les plus importants dans ce domaine.

Étude de cas 1

Predictive maintenance enabled by machine learning : Use cases and challenges in the automotive industry(2021), [8]

Synthèse Dans cette étude de cas, l'application de l'apprentissage automatique à la PDM a été examinée dans le contexte de l'industrie automobile. Les chercheurs ont évalué l'efficacité de plusieurs modèles d'apprentissage automatique dans la prédiction des pannes de machines et de la durée de vie utile restante. Les modèles étudiés comprenaient les DT, les forêts aléatoires, les SVM et les NN.

Les résultats clés de cette étude ont démontré que l'utilisation des modèles du ML peuvent considérablement améliorer l'efficacité des opérations et réduire les coûts dans l'industrie automobile.

Cependant, pour évaluer la robustesse de ces résultats, il aurait été bénéfique de comprendre comment ils ont été évalués. Des métriques spécifiques telles que la précision et le rappel auraient permis d'obtenir une évaluation quantitative des performances des modèles.

Étude de cas 2

Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0 (2020) ,[31]

Synthèse Dans cette étude, les chercheurs se sont concentrés sur l'application du ML à la maintenance prédictive. Ils ont examiné différents modèles du ML, notamment RF, kNN et SVM. L'objectif était de prédire les pannes de machines et la durée de vie utile restante.

Les résultats essentiels de l'étude ont montré que le choix de l'algorithme dépendait du cas d'utilisation spécifique et des données disponibles. Chaque modèle du ML avait ses propres forces et limitations, et il était crucial de sélectionner le bon modèle en fonction des besoins du système.

L'un des points importants soulevés par cette étude était l'importance de la collecte et du traitement précis des données. Des données de qualité sont essentielles pour une mise en œuvre réussie de la maintenance prédictive.

Cependant, l'étude n'a pas fourni d'informations spécifiques sur les résultats de l'implémentation des modèles du ML.

Étude de cas 3

Predictive Maintenance using Machine Learning (2022) ,[17].

Synthèse Les résultats de cette étude mettent en lumière l'efficacité remarquable de deux modèles du ML spécifiques : NN et SVM . En utilisant ces modèles, les chercheurs ont pu anticiper avec précision les défaillances des machines, ouvrant ainsi la voie à une planification plus efficace de la maintenance. Ces découvertes offrent des perspectives prometteuses pour améliorer la gestion de la maintenance dans divers secteurs de l'industrie.

Cependant, des réflexions critiques ont été soulevées quant à la nature des données utilisées dans cette étude. Malheureusement, l'absence de détails spécifiques sur les caractéristiques des données, telles que leur volume, leur qualité, leur représentativité et leur cohérence, limite notre compréhension de la fiabilité et de la généralisabilité des résultats obtenus.

Étude de cas 4

Machine Learning approach for Predictive Maintenance in Industry 4.0 (2018),[26].

Synthèse Dans cette étude , les chercheurs se sont penchés sur l'application du ML à la PDM dans le contexte de l'industrie 4.0. Ils ont examiné spécifiquement les modèles RF et DT.

Les résultats clés de cette recherche démontrent que le ML offre la capacité de prédire avec une précision accrue les défaillances d'équipement dans l'industrie 4.0. Cela a un impact significatif sur l'efficacité opérationnelle, contribuant ainsi à réduire les coûts.

Cependant, une réflexion critique soulevée est l'importance d'explorer l'impact des différents hyperparamètres des modèles du ML utilisés. Par exemple, le nombre d'arbres dans les RF ou la profondeur des DT peuvent avoir une influence considérable sur les performances des modèles. Une recherche plus approfondie dans ce domaine est donc recommandée afin d'obtenir des résultats encore meilleurs.

Étude de cas 5

Machine Learning for Predictive Maintenance : A Multiple Classifiers Approach (2021),[33]

Synthèse Cette étude se concentre sur l'utilisation de deux modèles du ML : kNN et SVM. Les chercheurs ont adopté une approche innovante en utilisant des classificateurs multiples pour améliorer l'exactitude des modèles du ML dans différents scénarios. Les résultats ont montré que cette approche était prometteuse et offrait de meilleures performances prédictives.

Cependant, malgré les résultats encourageants, des réflexions critiques peuvent être formulées. L'étude ne fournit pas de discussion approfondie sur le processus de sélection

des paramètres pour les algorithmes de classification. Une exploration plus détaillée de ces aspects aurait permis une meilleure compréhension de la validité des résultats et de la reproductibilité de l'étude.

Étude de cas 6

Machine learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines (2021),[34]

Synthèse Cette étude a examiné l'efficacité de différents modèles du ML, notamment RF et NN.

Les résultats essentiels de cette enquête ont démontré que les modèles du ML permettaient de détecter avec précision les défauts des roulements. Parmi les algorithmes étudiés, RF s'est révélé être le plus précis dans la plupart des cas, offrant ainsi des perspectives prometteuses pour la PDM. Cependant, il est important de noter que dans l'étude les caractéristiques spécifiques utilisées pour entraîner les modèles de ML n'ont pas été mentionnés .

2.9.1 Positionnement distinctif de notre travail

Dans notre travail, nous entreprendrons une sélection rigoureuse d'algorithmes pour la mise en œuvre de la maintenance prédictive dans la section 2.10 . Cette sélection sera basée sur une évaluation théorique approfondie, en nous référant à des articles de recherche qui classifient et compareront ces algorithmes. Notre objectif sera de choisir les algorithmes les plus performants et les mieux adaptés à notre domaine d'application.

Une fois les algorithmes sélectionnés, nous procéderons à des expérimentations en utilisant plusieurs jeux de données pertinents dans le chapitre qui suit. Notre travail se distinguera des études de cas mentionnées précédemment, car nous nous concentrerons sur une évaluation théorique initiale des algorithmes et une évaluation pratique sur plusieurs jeux de données. Cela nous permettra de fournir des informations précieuses sur les performances spécifiques de ces algorithmes dans le contexte de la maintenance prédictive.

2.10 Choix des algorithmes d'apprentissage supervisé : Évaluation et sélection

Le choix de l'algorithme d'apprentissage supervisé peut avoir un impact significatif sur les performances du modèle, il est donc important de prendre une décision éclairée. Dans cette étude, nous effectuons une revue de la littérature pour identifier les critères les plus importants dans le choix d'un algorithme d'apprentissage supervisé. Ensuite, nous évaluons théoriquement plusieurs algorithmes populaires en utilisant ces critères et en comparant leurs performances. Enfin, nous sélectionnons les algorithmes les plus performants pour une expérimentation plus approfondie.

2.10.1 Critères d'évaluation des algorithmes d'apprentissage supervisé

Pour choisir les meilleurs algorithmes d'apprentissage supervisé pour notre application, il est important de prendre en compte plusieurs critères d'évaluation. Dans ce tableau, nous avons répertorié les huit critères les plus importants pour évaluer les performances des algorithmes d'apprentissage supervisé.

Critère	Signification
Exactitude (Accuracy)	Mesure de la proportion de prédictions correctes.
Précision (Precision)	Mesure de la proportion de vrais positifs parmi tous les résultats positifs.
Rappel (Recall)	Mesure de la proportion de vrais positifs identifiés parmi tous les résultats réellement positifs.
Flexibilité(Flexibility)	Capacité d'un modèle à s'adapter à des changements ou à des variations dans les données d'entrée.
Facilité d'interprétation(Ease of interpretation)	Facilité avec laquelle un modèle peut être compris et interprété par les utilisateurs et les décideurs.
Vitesse d'apprentissage (Speed of learning)	Rapidité de traitement des données et de classification.
Tolérance aux attributs non pertinents (%) (Tolerance to irrelevant attributes)	Capacité à maintenir la performance du modèle tout en réduisant sa complexité en éliminant des attributs non pertinents.
Tolérance aux attributs redondant (%) (Tolerance to redundant attributes)	Capacité à maintenir la performance du modèle tout en réduisant sa complexité en éliminant des attributs redondants.
Capacité d'explication (Explanation ability)	Capacité à expliquer les décisions prises par un modèle pour une utilisation éthique et responsable.
Taille de mémoire (Memory size (GB))	Taille de la mémoire nécessaire pour stocker le modèle.
Paramétrisation (Parametrization (N))	Ajustement de la complexité du modèle en fonction de la quantité de données et des objectifs de performance.

TAB. 2.3 : Critères d'évaluation

Dans cette étude, nous avons évalué la performance de différents algorithmes de machine learning en utilisant 8 critères d'évaluation pertinents pour notre domaine d'application. Nous avons effectué une revue de la littérature pour attribuer une note à chaque algorithme sur chaque critère.

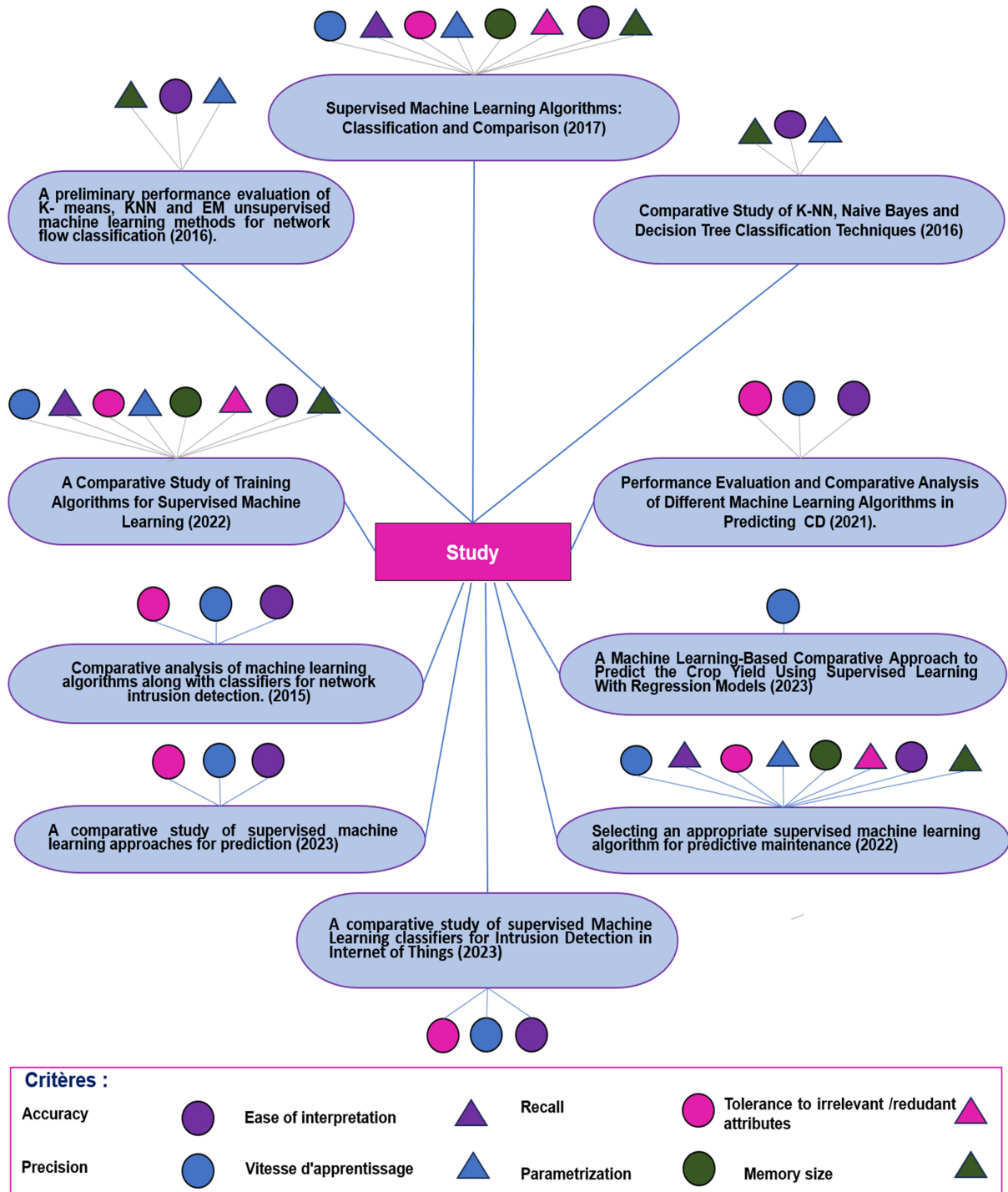


FIG. 2.8 : Revue littérature des critères d'évaluation

Dans cette partie, nous avons utilisé des articles de référence mentionnés précédemment qui ont déjà évalué et classifié les algorithmes. Ces articles fournissent des notes pour chaque algorithme sur chaque critère d'évaluation. Nous avons extrait ces notes des articles et les avons utilisées pour calculer la moyenne des critères pour chaque algorithme.

Ensuite, nous avons représenté les résultats sous forme d'un tableau. Chaque ligne de ce tableau correspond à un critère d'évaluation spécifique, tandis que chaque colonne représente un algorithme de machine learning. Les notes attribuées vont de 0 à 10, où 10 représente la note la plus élevée. Ce tableau nous permet de visualiser facilement les performances relatives des différents algorithmes sur chaque critère d'évaluation.

Critère	DT	RF	NBC	KNN	SVM	LR
Exactitude(/10)	7	8	8	7	8	6
Précision (/10)	8	9	7	8	8	6
Rappel(/10)	6	9	7	8	7	5.5
Facilité d'interprétation (/10)	9	8	7	8	8	9
Vitesse d'apprentissage/classification (/10)	9	7.5	9	9	9	9
Tolérance aux attributs non pertinents / redondants (/10)	7.5	8	5.5	7	8.5	5
Taille de la mémoire (/10)	7	8	6	7	6	6
Paramétrisation (/10)	8	8	8	8	6	8
Moyenne (/10)	7.7	8.19	7.19	7.75	7.563	6.81

TAB. 2.4 : Comparaison des performances des algorithmes de machine learning selon 8 critères d'évaluation

Afin de sélectionner les meilleurs algorithmes, nous avons calculé la Moyenne des notes de chaque algorithme . Nous avons ensuite représenté les résultats sous forme d'un histogramme, qui nous permet de voir les performances globales des différents algorithmes et de sélectionner les trois meilleurs. Ces résultats nous permettront de choisir l'algorithme de machine learning le plus performant pour notre application.

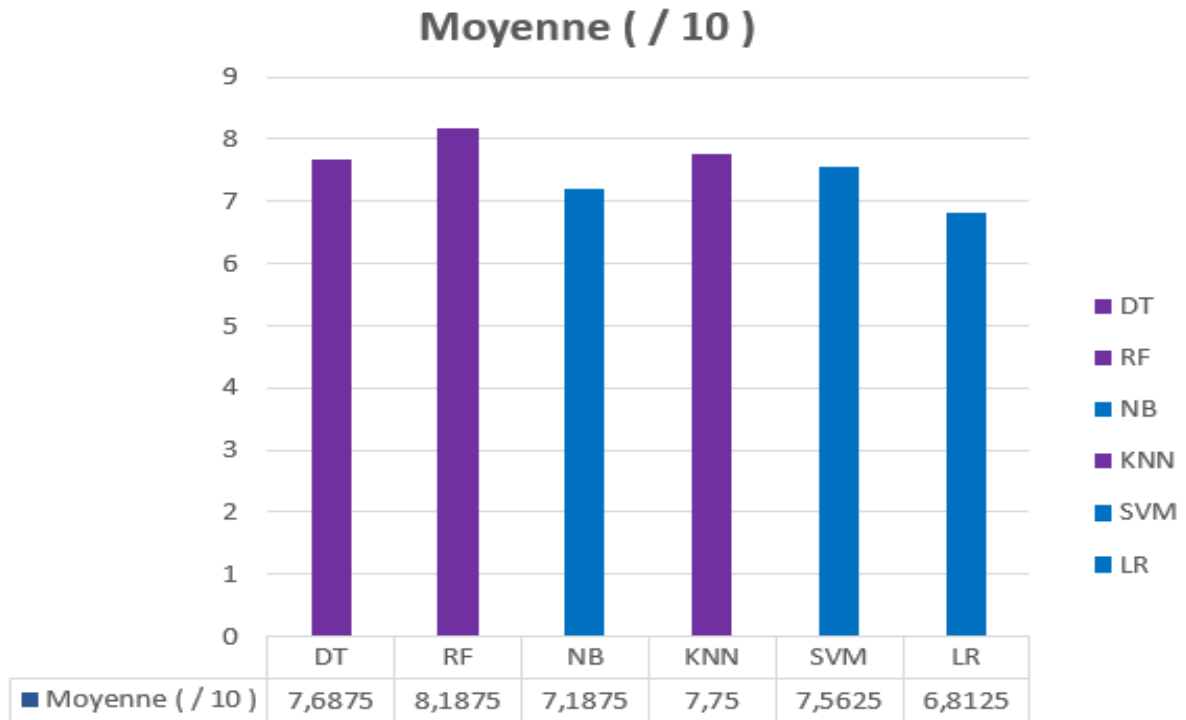


FIG. 2.9 : Les algorithmes d'apprentissage automatique les plus performants , résultats sur 8 critères

En conclusion, selon notre évaluation basée sur une revue de la littérature, les algorithmes de machine learning les plus performants pour notre domaine d'application sont le RF avec une moyenne de 8.2, le DT avec une moyenne de 7.68 et le K-Nearest Neighbors (KNN) avec une moyenne de 7.75 . Nous prévoyons de tester ces algorithmes sur des données réelles dans la partie expérimentale pour confirmer nos résultats et déterminer le meilleur algorithme pour notre application.

2.11 Conclusion

Dans ce chapitre, nous avons introduit les principes de base de l'apprentissage automatique et exploré les différents types d'apprentissage, tels que l'apprentissage supervisé, non supervisé et par renforcement. Nous avons également examiné plusieurs algorithmes populaires utilisés dans l'apprentissage supervisé, tels que KNN, arbre de décision, forêt aléatoire, machine à vecteur de support et Naive Bayes.

Dans le cadre de notre étude, nous avons choisi de nous concentrer sur trois algorithmes spécifiques pour l'implémentation de la de maintenance prédictive. Ces trois algorithmes, à savoir l'arbre de décision, la forêt aléatoire et le SVM, ont été sélectionnés en fonction de leur adaptabilité aux caractéristiques spécifiques de nos données et de leur performance prouvée dans des problèmes similaires.

Dans le chapitre suivant, nous allons mettre en pratique ces connaissances en implémentant ces algorithmes et en les évaluant sur notre jeu de données. Nous examinerons les performances de chaque algorithme en utilisant des critères d'évaluation tels que l'exactitude, la précision et le rappel. Cette évaluation nous permettra de déterminer lequel des trois algorithmes est le plus approprié pour notre problème de maintenance prédictive.

En résumé, ce chapitre nous a permis d'explorer les bases de l'apprentissage automatique, de comprendre les différents types d'apprentissage et de sélectionner les trois algorithmes les plus pertinents pour notre implémentation. Le chapitre suivant sera dédié à l'application de ces algorithmes dans notre système de maintenance prédictive et à l'évaluation de leurs performances.

Chapitre 3

Mise en pratique : Résultats et analyses

3.1 Introduction

Dans ce chapitre, notre objectif est de réaliser une évaluation complète et objective des performances des algorithmes RF, DT et KNN pour la maintenance prédictive. Dans cette perspective, nous vérifierons également notre hypothèse de départ concernant le classement de ces algorithmes.

Cette partie expérimentale repose sur l'évaluation de ces algorithmes sur quatre ensembles de données spécifiques à la maintenance prédictive, qui ont été soigneusement sélectionnés pour couvrir une gamme variée de scénarios. Notre objectif est d'analyser en détail les performances de chaque algorithme sur ces ensembles de données, en utilisant un ensemble de critères d'évaluation bien établis dans la littérature.

Pour mener à bien cette évaluation, nous avons mis en place une méthodologie rigoureuse. Tout d'abord, nous décrirons en détail chaque ensemble de données, en mettant en évidence leurs caractéristiques spécifiques et leur pertinence pour la maintenance prédictive. Ensuite, nous présenterons les différentes étapes de notre expérimentation, en expliquant clairement comment nous avons configuré chaque algorithme et sélectionné les paramètres optimaux.

De plus, pour évaluer plus précisément les performances de nos algorithmes par rapport aux approches décrites dans la littérature existante, nous comparerons les résultats obtenus sur notre deuxième ensemble de données avec ceux présentés dans deux articles de référence dans le domaine de la maintenance prédictive.

3.2 Description des données d'expérimentation

Dans cette section, nous décrirons en détail les différents datasets utilisés dans cette étude, en mettant en évidence leurs sources, leurs caractéristiques et les interprétations associées à chaque variable.

3.2.1 Premier dataset

(Données réelles)

Le premier dataset utilisé dans cette étude est extrait d'une compétition sur Zindy. Il représente une source de données réelles provenant d'une machine de conditionnement automatique. Ce dataset est spécifiquement utilisé pour évaluer la performance des algorithmes d'apprentissage automatique.

Il comprend des variables clés telles que les mesures des particules fines captées par les capteurs **Sensor1_PM2.5** et **Sensor2_PM2.5**, ainsi que des données sur la température et l'humidité relative. Ces informations sont d'une importance capitale pour prédire les défaillances potentielles de la machine. Le tableau 3.1 présente les caractéristiques du premier dataset, telles que la taille, le nombre de variables et un descriptif des variables, ainsi que les interprétations associées à chaque caractéristique.

Caractéristiques du premier dataset	Interprétation
taille	Significative : L'ensemble de données contient 297 177 enregistrements, ce qui indique une grande quantité de données
nombre de features	Moyen : Il y a seulement 5 variables dans cet ensemble de données, ce qui le rend relativement petit.
Descriptif des features	Sensor1_PM2.5 : Mesure des particules fines Sensor2_PM2.5 : Mesure des particules fines Temperature : Température Relative_Humidity : Humidité relative failure : Indicateur binaire de défaillance
corrélation de toutes les variables avec la variable cible	Moyenne

TAB. 3.1 : Caractéristiques et interprétations du premier dataset

3.2.2 Deuxième dataset

(Données réelles) Source :

<https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset>

Le dataset AI4I 2020 Predictive Maintenance Dataset a été créé et fourni par Stephan Matzka de l'École d'ingénierie - Technologie et vie, Hochschule für Technik und Wirtschaft Berlin, Berlin, Allemagne. Il s'agit d'un ensemble de données synthétiques conçu pour représenter les données réelles de maintenance prédictive dans l'industrie.

Le tableau 3.2 présente les caractéristiques du deuxième dataset, telles que la taille, le nombre de variables et un descriptif des variables, ainsi que les interprétations associées à chaque caractéristique.

Caractéristiques du deuxième dataset	Interprétation
taille	Importante : L'ensemble de données contient 10 000 enregistrements, ce qui indique une quantité importante de données.
nombre de features	Moyen : Il y a seulement 5 variables dans cet ensemble de données, ce qui le rend relativement petit.
Descriptif des features	Torque [Nm] : Mesure du couple appliqué dans l'unité de newton-mètre (Nm) Tool wear [min] : Temps d'usure de l'outil de production en minutes Air temperature [K] : Température de l'air en kelvins Process temperature [K] : Température du processus en kelvins Machine failure : Indicateur binaire de défaillance de la machine (0 : Pas de défaillance, 1 : Défaillance de la machine)
corrélation de toutes les variables avec la variable cible	Très forte

TAB. 3.2 : Caractéristiques et interprétations du deuxième dataset

3.2.3 Troisième dataset

(Données de simulation) Source : <https://gallery.azure.ai/Experiment/Predictive-Maintenance-Modelling-Guide-Experiment-1> Ce dataset est utilisé pour la maintenance prédictive. Il comprend des lectures de télémétrie, l'identification des erreurs, les opérations de maintenance et les défaillances. Il est conçu pour être utilisé dans des modèles de maintenance prédictive afin de prédire le temps restant avant une défaillance potentielle de la machine.

Le tableau 3.3 présente les caractéristiques du troisième dataset, telles que la taille, le nombre de variables et un descriptif des variables, ainsi que les interprétations associées à chaque caractéristique.

Caractéristiques du Troisième dataset	Interprétation
taille	Importante : L'ensemble de données contient 14 482 enregistrements, ce qui indique une quantité importante de données.
nombre de features	Il y a 7 variables dans cet ensemble de données, ce qui le rend relativement modéré en termes de nombre de caractéristiques.
Descriptif des features	vibrationmean_24hrs : Moyenne des lectures de vibrations sur une période de 24 heures voltage_24hrs : Moyenne des lectures de tension sur une période de 24 heures pressuremean_24hrs : Moyenne des lectures de pression sur une période de 24 heures vibrationmean : Moyenne des lectures de vibrations voltage : Moyenne des lectures de tension pressuremean : Moyenne des lectures de pression failure : Indicateur binaire de défaillance de la machine (0 : Pas de défaillance, 1 : Défaillance de la machine)
corrélation de toutes les variables avec la variable cible	Faible

TAB. 3.3 : Caractéristiques et interprétations du Troisième dataset

3.2.4 Quatrième dataset

(Données de simulation) Source : <https://www.kaggle.com/datasets/parvmodi/automotive-vehicles-engine-health-dataset>

Cet ensemble de données comprend diverses caractéristiques et mesures liées à la santé des moteurs de véhicules, telles que le régime moteur, la température, la pression et d'autres données de capteur. Il peut également contenir des métadonnées sur les véhicules, telles que la marque, le modèle, l'année et le kilométrage.

Le but de la création de ce dataset est de construire un modèle de maintenance prédictive pour les moteurs automobiles. En analysant les modèles et les tendances des données, les algorithmes d'apprentissage automatique peuvent être entraînés à prédire quand un moteur est susceptible de nécessiter une maintenance ou une réparation. Cela permet aux propriétaires de véhicules et aux mécaniciens de résoudre de manière proactive les problèmes potentiels avant qu'ils ne s'aggravent, ce qui se traduit par de meilleures performances du véhicule et une durée de vie du moteur plus longue.

Le tableau 3.4 présente les caractéristiques du quatrième dataset, telles que la taille, le nombre de variables et un descriptif des variables, ainsi que les interprétations associées à chaque caractéristique.

Caractéristiques du Quatrième dataset	Interprétation
taille	Importante : L'ensemble de données contient 19 535 enregistrements, ce qui indique une quantité importante de données.
nombre de features	Il y a 3 variables dans cet ensemble de données, ce qui le rend très petit en termes de nombre de caractéristiques.
Descriptif des features	Fuel pressure : Pression de carburant Lub oil pressure : Pression d'huile de lubrification Engine Condition : Condition du moteur (0 : Mauvaise condition, 1 : Bonne condition)
corrélation de toutes les variables avec la variable cible	Faible

TAB. 3.4 : Caractéristiques et interprétations du Quatrième dataset

3.3 Approche statistique utilisée pour comparer les performances des trois algorithmes

Pour évaluer et comparer les performances des algorithmes de machine learning que nous avons sélectionné, tels que le KNN, l'arbre de décision et la forêt aléatoire, nous avons utilisé la matrice de confusion.

La matrice de confusion est une mesure de performance utilisée en apprentissage automatique et en statistiques pour évaluer les performances d'un modèle de classification. Elle permet de visualiser les prédictions d'un modèle par rapport aux valeurs réelles des données. Elle est organisée en une grille qui contient quatre éléments importants ,[28]. Selon le tableau 3.5, on peut observer les différentes combinaisons de prédictions et de réalités.

	Prédiction : Négative	Prédiction : Positive
Réalité : Négative	TN	FP
Réalité : Positive	FN	TP

TAB. 3.5 : Matrice de confusion[4]

TN (**True Negative**) représente le nombre d'observations qui ont été correctement classées comme négatives.

FP (**False Positive**) représente le nombre d'observations qui ont été incorrectement classées comme positives.

FN (**False Negative**) représente le nombre d'observations qui ont été incorrectement classées comme négatives.

TP (**True Positive**) représente le nombre d'observations qui ont été correctement classées comme positives.

Ces mesures nous aident à évaluer la précision, la sensibilité et d'autres aspects de la performance des algorithmes, ce qui nous permet de prendre des décisions éclairées dans le choix de l'algorithme le plus performant. Voici un tableau 3.6 récapitulant les métriques fréquemment utilisées, accompagnées de leurs définitions et de leurs formules correspondantes :

En complément de l'analyse basée sur la matrice de confusion, nous avons également pris en compte d'autres mesures telles que la taille de la mémoire requise et la vitesse d'apprentissage. Ces métriques nous ont permis de comparer les performances des algorithmes non seulement en termes de précision et de rappel, mais aussi en termes d'efficacité et de consommation de ressources.

En conclusion, la matrice de confusion nous a fourni une visualisation précise des prédictions de chaque algorithme. En prenant également en compte d'autres métriques, nous avons obtenu une évaluation globale de la performance des algorithmes, ce qui nous a permis de choisir le modèle le plus approprié pour notre tâche de maintenance prédictive.

Métrique	Définition	Formule
Exactitude	La mesure qui indique la proportion d'observations correctement prédites parmi toutes les observations.	$\frac{TP+TN}{TP+TN+FP+FN}$
Précision	La mesure qui montre la proportion d'observations positives correctement prédites parmi toutes les observations prédites comme positives.	$\frac{TP}{TP+FP}$
Rappel	La mesure qui indique la proportion d'observations positives correctement prédites parmi toutes les observations réellement positives.	$\frac{TP}{TP+FN}$
Score F1	La mesure qui combine la précision et le rappel en une seule valeur, offrant ainsi une mesure globale de la performance du modèle.	$\frac{2(PrecisionRappel)}{Precision+Rappel}$

TAB. 3.6 : Métriques d'évaluation des modèles

3.4 Méthodologie expérimentale

Dans cette section, nous décrirons en détail les quatre ensembles de données sélectionnés pour notre évaluation des algorithmes RF, DT et KNN. Chaque ensemble de données représente un scénario spécifique et est pertinent pour la maintenance prédictive. Nous fournirons des informations sur la taille de chaque ensemble, le nombre de caractéristiques incluses, ainsi que leur source et leur contexte d'application dans le domaine de la maintenance prédictive.

Notre méthodologie pour évaluer et comparer les performances des trois algorithmes comprend quatre étapes clés :

1. La préparation des données
2. La visualisation des données
3. L'entraînement du modèle
4. L'évaluation, ainsi que la visualisation et l'analyse des résultats

3.4.1 Préparation de données

La figure 3.1 illustre les différentes étapes de la phase de préparation des données.

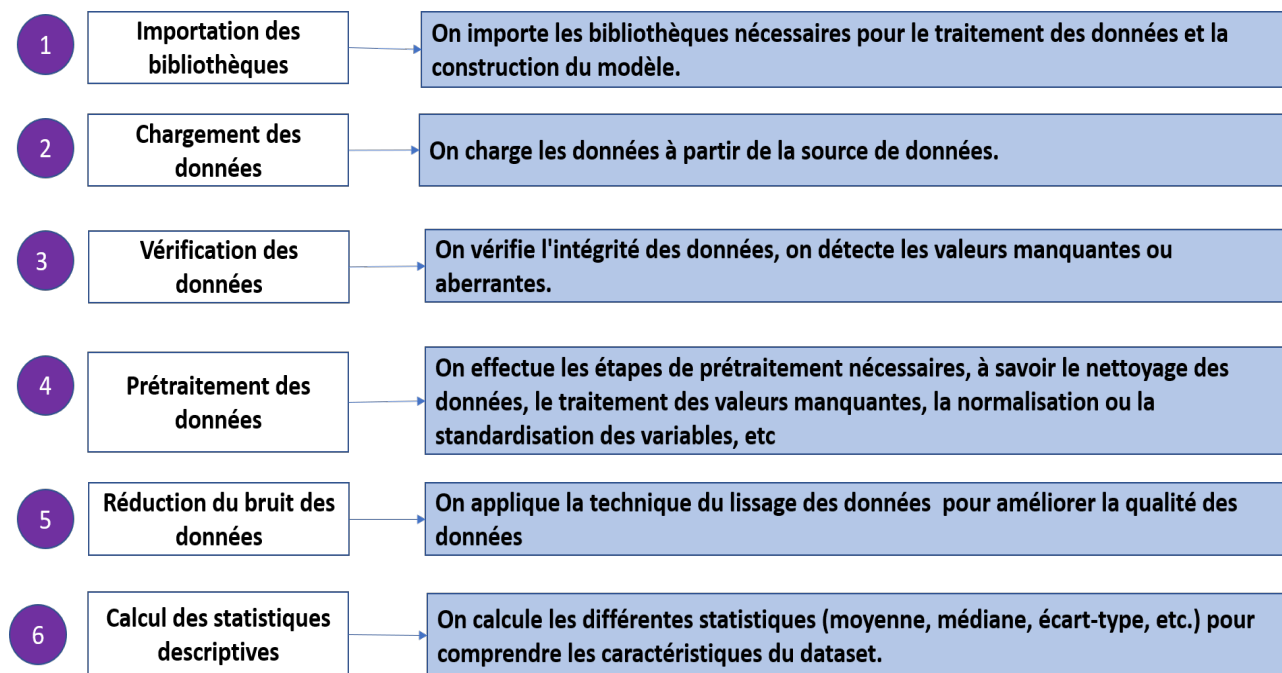


FIG. 3.1 : Etapes de préparation de données

3.4.2 Visualisation des données

La figure 3.2 représente les diverses étapes de la phase de visualisation des données de manière détaillée.

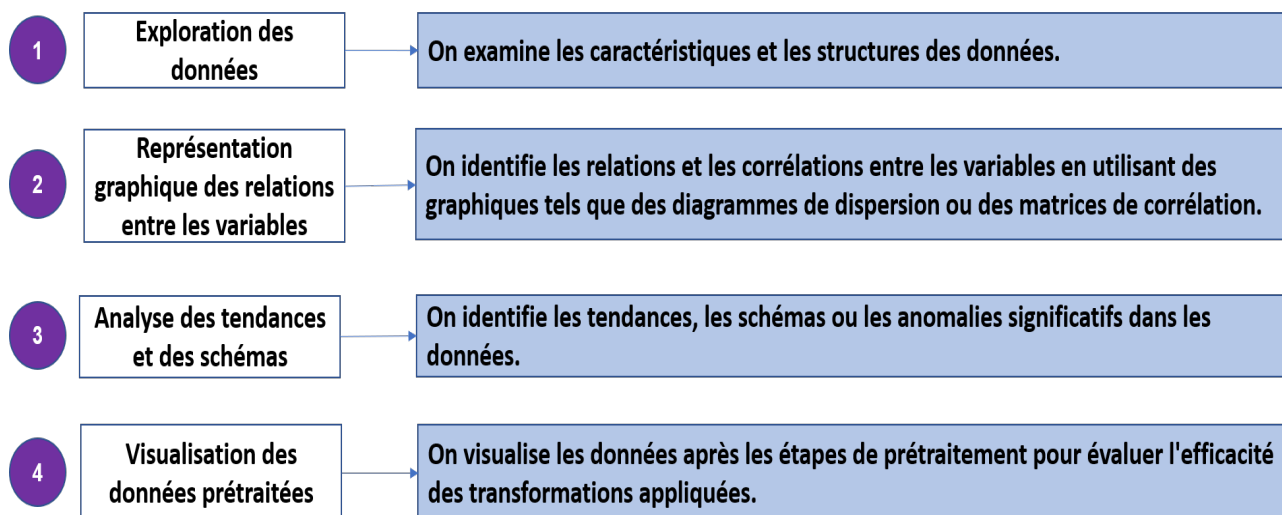


FIG. 3.2 : Visualisation des données

3.4.3 Entraînement et évaluation des modèles

Une fois les données prétraitées dans le format approprié, nous passons à la construction des différents modèles.

3.4.4 Random Forest

L'entraînement du modèle se déroule de la manière suivante : les données sont d'abord divisées en ensembles d'entraînement et de test à l'aide de la fonction `train_test_split()`, ce qui permet d'obtenir les variables `X_train`, `X_test`, `y_train` et `y_test`. Ensuite, un classifieur Random Forest est initialisé à l'aide de la classe `RandomForestClassifier()`. Le modèle Random Forest est ensuite entraîné sur l'ensemble d'entraînement en utilisant la méthode `fit(X_train, y_train)`. Les prédictions sont effectuées sur l'ensemble de test à l'aide de la méthode `predict(X_test)`, donnant ainsi les prédictions stockées dans la variable `rf_pred`. Pour évaluer les performances du modèle, des mesures telles que l'exactitude, la précision, le rappel et le score F1 sont calculées à l'aide des fonctions `accuracy_score()`, `precision_score()`, `recall_score()` et `f1_score()` respectivement. Une validation croisée est également effectuée en utilisant la fonction `cross_val_score()` avec le modèle Random Forest, les données complètes (`x` et `y`) et 5 folds. Le temps d'exécution de l'apprentissage et de la classification est mesuré en enregistrant le temps de début avant la validation croisée et le temps de fin après celle-ci. Enfin, la taille mémoire du modèle est calculée en utilisant la fonction `sys.getsizeof()` pour obtenir sa taille en octets, qui est ensuite convertie en gigaoctets.

```
X_train, X_test, y_train, y_test = train_test_split(x, y, random_state=1, test_size=0.3)
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
rf_pred = rf.predict(X_test)
accuracy1 = accuracy_score(y_test, rf_pred)
precision1 = precision_score(y_test, rf_pred)
recall1 = recall_score(y_test, rf_pred)
f1_score1 = f1_score(y_test, rf_pred)
start_time1 = time()
cv_scores = cross_val_score(rf, x, y, cv=5)
end_time1 = time()
speed1 = end_time1 - start_time1
memory1 = sys.getsizeof(rf) / 1024 # Taille mémoire en GB
```

FIG. 3.3 : Code du RF

3.4.5 DT

L'entraînement du modèle se déroule de la manière suivante : les données sont d'abord divisées en ensembles d'entraînement et de test à l'aide de la fonction `train_test_split(x, y, random_state=1, test_size=0.3)`, ce qui permet d'obtenir les variables `X_train`, `X_test`, `y_train` et `y_test`. Ensuite, un classifieur de l'arbre de décision est initialisé à l'aide de la classe `DecisionTreeClassifier()`. Le modèle de l'arbre de décision est entraîné sur l'ensemble d'entraînement en utilisant la méthode `fit(X_train, y_train)`. Les prédictions sont effectuées sur l'ensemble de test à l'aide de la méthode `predict(X_test)`, donnant ainsi les prédictions stockées dans la variable `dtr_pred`. Les étapes suivantes, c'est-à-dire les mesures de performance, la validation croisée, la mesure du temps d'exécution et la taille mémoire du modèle, sont identiques à celles du modèle Random Forest.

```
X_train, X_test, y_train, y_test = train_test_split(x, y, random_state=1, test_size=0.3)
dtr = DecisionTreeClassifier()
dtr.fit(X_train, y_train)
dtr_pred = dtr.predict(X_test)
accuracy2 = accuracy_score(y_test, dtr_pred)
precision2 = precision_score(y_test, dtr_pred)
recall2 = recall_score(y_test, dtr_pred)
f1_score2 = f1_score(y_test, dtr_pred)
start_time2 = time()
cv_scores = cross_val_score(dtr, x, y, cv=5)
end_time2 = time()
speed2 = end_time2 - start_time2
memory2 = sys.getsizeof(dtr) / 1024
```

FIG. 3.4 : Code de DT

3.4.6 KNN

L'entraînement du modèle se déroule de la manière suivante : les données sont d'abord divisées en ensembles d'entraînement et de test à l'aide de la fonction `train_test_split(x, y, random_state=1, test_size=0.3)`, ce qui permet d'obtenir les variables `X_train`, `X_test`, `y_train` et `y_test`. Ensuite, un classifieur KNN est initialisé à l'aide de la classe `KNeighborsClassifier()`. Le modèle KNN est ensuite entraîné sur l'ensemble d'entraînement en utilisant la méthode `fit(X_train, y_train)`. Les prédictions sont effectuées sur l'ensemble de test à l'aide de la méthode `predict(X_test)`, donnant ainsi les prédictions stockées dans la variable `knn_pred`. Les étapes qui suit sont identiques à celles du modèle Random Forest.

```
X_train, X_test, y_train, y_test = train_test_split(x, y, random_state=1, test_size=0.3)
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
knn_pred = knn.predict(X_test)
accuracy3 = accuracy_score(y_test, knn_pred)
precision3 = precision_score(y_test, knn_pred)
recall3 = recall_score(y_test, knn_pred)
f1_score3 = f1_score(y_test, knn_pred)
start_time3 = time()
cv_scores = cross_val_score(knn, x, y, cv=5)
end_time3 = time()
speed3 = end_time3 - start_time3
memory3 = sys.getsizeof(knn) / 1024 # Taille mémoire en GB
```

FIG. 3.5 : Code de KNN

3.4.7 Visualisation et Analyse des résultats

Dans cette section, nous présenterons les résultats obtenus pour chaque algorithme et ensemble de données, en mettant l'accent sur une analyse approfondie des performances. Nous comparerons les métriques d'évaluation et discuterons des forces et des faiblesses de chaque algorithme dans le contexte de la maintenance prédictive.

En résumé, notre méthodologie expérimentale comprend la description des ensembles de données, la préparation des données, la visualisation des données, l'entraînement et l'évaluation des modèles, ainsi que la visualisation et l'analyse des résultats. Cela nous permettra d'obtenir une évaluation complète des performances des algorithmes RF, DT et KNN pour la maintenance prédictive.

3.5 Tests et résultats

3.5.1 Résultats et discussions de la première base de données

Les figures 3.6, 3.9, 3.12, 3.15, représentent les matrices de confusion des trois algorithmes utilisés pour les quatre bases de données respectivement.

Les histogrammes illustrés dans les figures 3.7, 3.10, 3.13, 3.16 présentent les valeurs des différentes métriques pour les trois algorithmes utilisés sur les quatre bases de données respectivement.



FIG. 3.6 : Matrices de confusion des trois algorithmes pour évaluer les prédictions réalisées sur la première base de données.

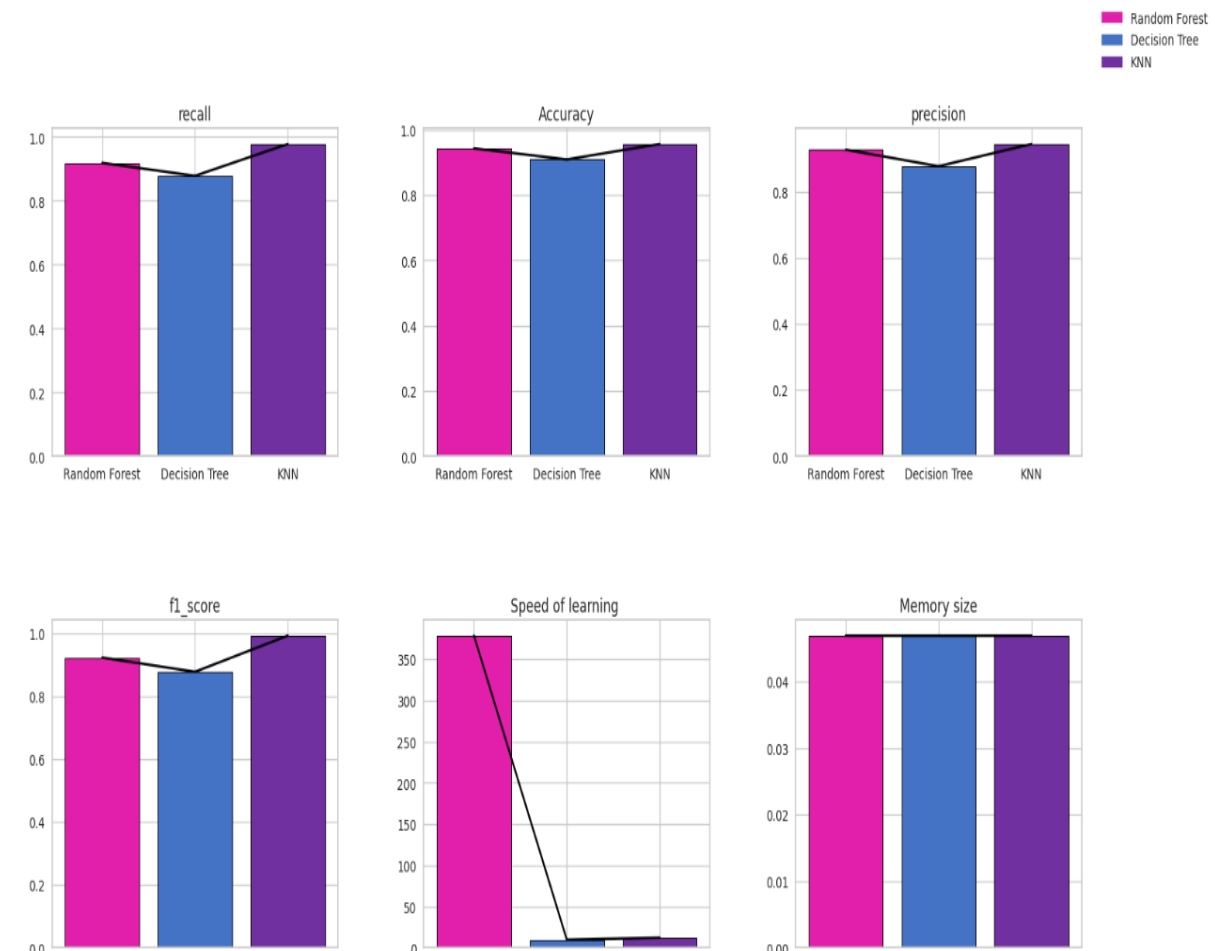


FIG. 3.7 : Comparaison des métriques des trois algorithmes pour évaluer les prédictions réalisées sur la première base de données.

Le tableau suivant 3.8 présente un récapitulatif des valeurs des différentes métriques pour les trois algorithmes utilisés sur la première base de données. Ces valeurs ont été récupérées à partir du code et permettent d'évaluer les performances relatives de chaque algorithme. En examinant ces valeurs, on peut analyser et comparer les performances des algorithmes, ce qui est crucial pour prendre des décisions éclairées lors du choix de l'algorithme le plus approprié pour une tâche donnée.

	Algorithm	Recall	accuracy	precision	f1_score	Speed of learning	Memory size
0	Random Forest Classifier	0.919034	0.943076	0.927123	0.923061	378.663514	0.046875
1	Decision Tree Classifier	0.878460	0.909045	0.876951	0.877705	9.713382	0.046875
2	KNN Classifier	0.938204	0.956446	0.944218	0.941201	12.220388	0.046875

FIG. 3.8 : Récapitulatif des métriques pour les trois algorithmes de la première base de données.

Discussion

Lorsque nous examinons les performances des trois algorithmes RF, DT et KNN sur la première base de données, nous constatons que le KNN Classifier s'est révélé être le plus performant parmi les trois.

En termes de rappel, le KNN Classifier a obtenu un score élevé de 0.938204, indiquant qu'il a pu détecter et classer correctement un pourcentage élevé des instances positives de l'ensemble de données. Le RF Classifier a obtenu un rappel de 0.919034, tandis que le DT Classifier a obtenu un rappel légèrement inférieur de 0.878460.

En ce qui concerne la précision, le KNN Classifier a également obtenu le score le plus élevé, avec une précision de 0.944218. Cela signifie que le KNN Classifier a réussi à classifier correctement un pourcentage élevé des instances positives sans introduire trop de faux positifs. Le RF Classifier a obtenu une précision de 0.927123, tandis que le DT Classifier a obtenu une précision légèrement inférieure de 0.876951.

Si nous considérons le score F1, qui combine à la fois la précision et le rappel, le KNN Classifier a montré une performance globale légèrement supérieure. Son score F1 de 0.941201 est supérieur à celui du RF Classifier 0.923061 et du DT Classifier 0.877705.

Enfin, en termes de vitesse d'apprentissage, le modèle de l'arbre de décision a la vitesse d'apprentissage la plus rapide parmi les trois algorithmes testés, avec un score de 9.713382s. Cela signifie que l'entraînement avec l'arbre de décision est rapide par rapport aux autres modèles, suivi par le KNN qui a une vitesse d'apprentissage légèrement plus élevée que l'arbre de décision avec un score de 12.220388s, et enfin le RF qui a la vitesse d'apprentissage la plus lente de 378.663514s, ce qui signifie que la prédiction avec ce modèle prend plus de temps par rapport aux autres modèles.

En conclusion, sur la première base de données, le KNN Classifier s'est avéré être l'algorithme le plus performant en termes de rappel, de précision et de score F1. Il a démontré une meilleure capacité à détecter et à classer correctement les instances positives par rapport aux autres algorithmes.

3.5.2 Résultats et discussions de la deuxième base de données

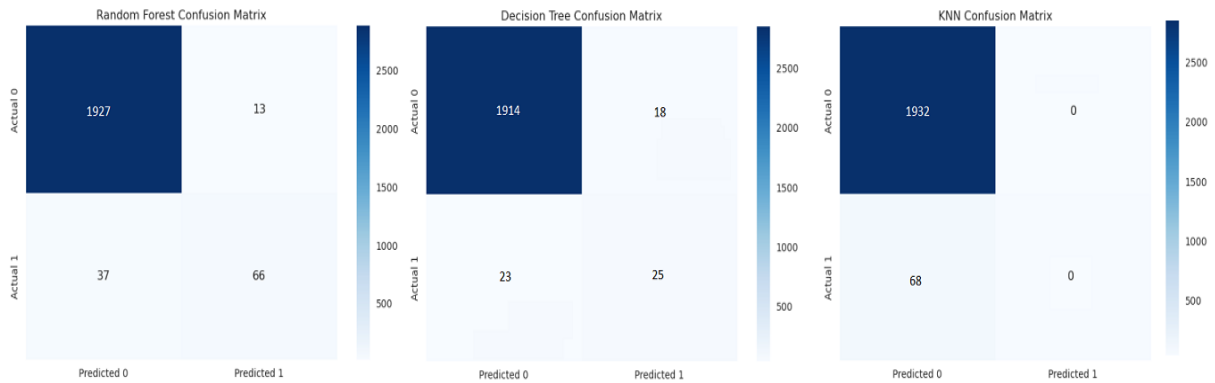


FIG. 3.9 : Matrices de confusion des trois algorithmes pour évaluer les prédictions réalisées sur la deuxième base de données.

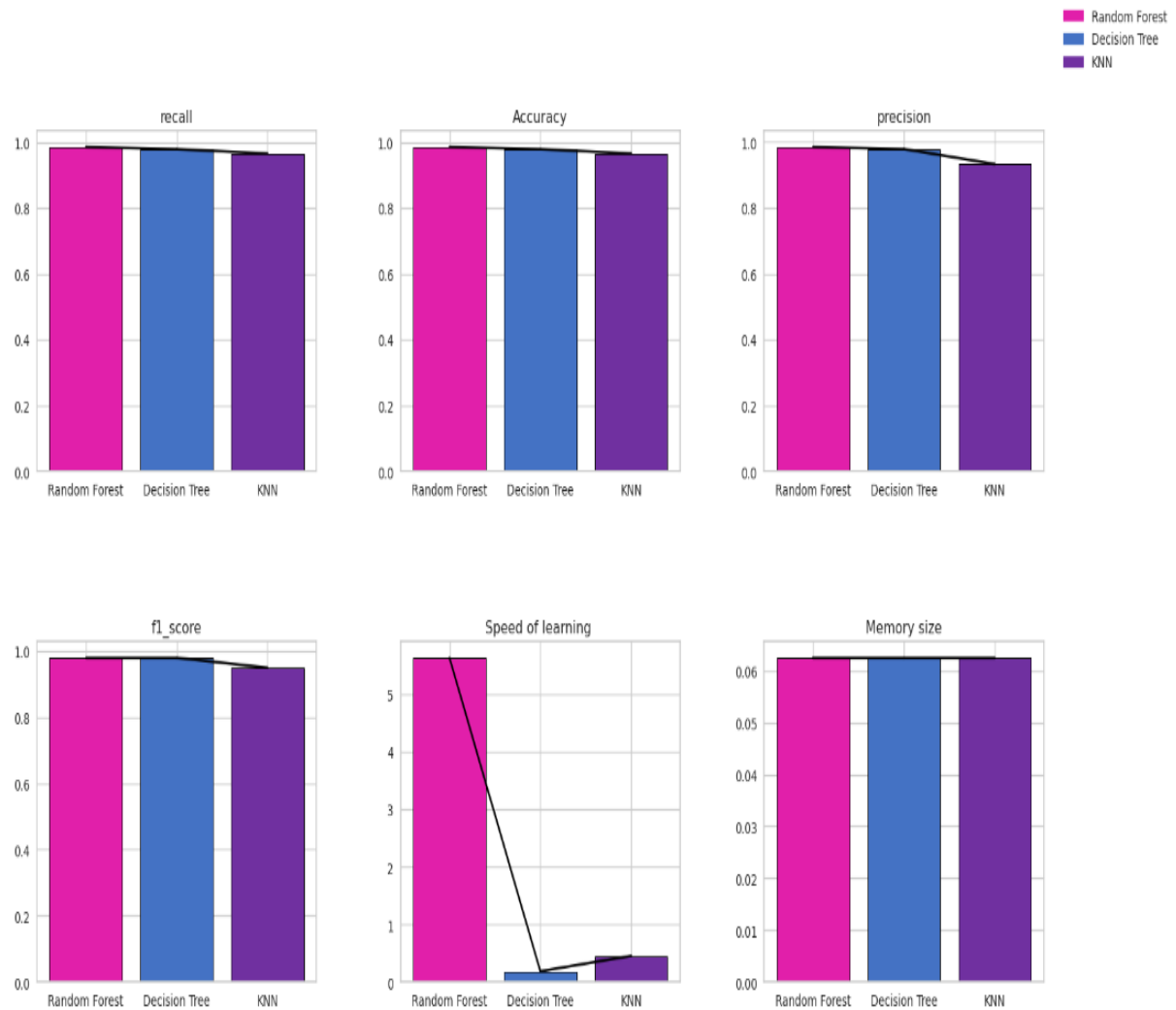


FIG. 3.10 : Comparaison des métriques pour trois algorithmes pour évaluer les prédictions réalisées sur la deuxième base de données.

Le tableau suivant 3.11 présente un récapitulatif des valeurs des différentes métriques (recall, accuracy, precision, f1-score, memory size et speed of learning) pour les trois algorithmes utilisés sur la deuxième base de données.

	Algorithm	Recall	accuracy	precision	f1_score	Speed of learning	Memory size
0	Random Forest Classifier	0.985500	0.985500	0.984600	0.979100	5.499048	0.062500
1	Decision Tree Classifier	0.979500	0.979500	0.978800	0.979100	0.184607	0.062500
2	KNN Classifier	0.966000	0.966000	0.933200	0.949300	0.436225	0.062500

FIG. 3.11 : Récapitulatif des métriques pour les trois algorithmes de la deuxième base de données.

Discussion

Après avoir analysé les performances des trois algorithmes sur la deuxième base de données, nous pouvons discuter de leur ordre de performance en nous basant sur les différentes métriques évaluées.

Le modèle RF Classifier se positionne comme le plus performant parmi les trois algorithmes étudiés. Il affiche un rappel de 0.985500, une précision de 0.984600 et un score F1 de 0.979100, ce qui indique une capacité élevée à classer correctement les instances des deux classes. L'exactitude du modèle est également élevée, avec une valeur de 0.985500. En termes de vitesse d'apprentissage, le modèle nécessite un temps de 5.499048 secondes, ce qui est raisonnable compte tenu des performances obtenues. La consommation de mémoire reste la même que celle des autres modèles, soit 0.062500 GB.

Ensuite, nous avons le modèle DT Classifier qui présente des performances légèrement inférieures par rapport au modèle RF Classifier sur la deuxième base de données. Il présente un rappel de 0.979500, une précision de 0.978800 et un score F1 de 0.979100. L'exactitude du modèle est également élevée, avec une valeur de 0.979500. Bien que le modèle DT Classifier soit légèrement moins performant que le modèle RF Classifier, il maintient des performances globalement élevées. En termes de vitesse d'apprentissage, le modèle est le plus rapide parmi les trois, avec un temps de 0.184607 seconde. La consommation de mémoire reste la même que celle des autres modèles, soit 0.062500 GB.

Finalement, le modèle KNN Classifier présente des performances légèrement inférieures par rapport aux deux autres modèles sur la deuxième base de données. Il affiche un rappel de 0.966000, une précision de 0.933200 et un score F1 de 0.949300. L'exactitude du modèle est également de 0.966000. Bien que le modèle KNN Classifier soit moins performant que les autres modèles, il est capable de classer les instances des deux classes avec une précision satisfaisante. En termes de vitesse d'apprentissage, le modèle KNN Classifier nécessite un temps de 0.436225 seconde, ce qui est plus élevé que celui du modèle DT Classifier. La consommation de mémoire reste la même que celle des autres modèles, soit 0.062500 GB.

3.5.3 Résultats et discussions de la troisième base de données

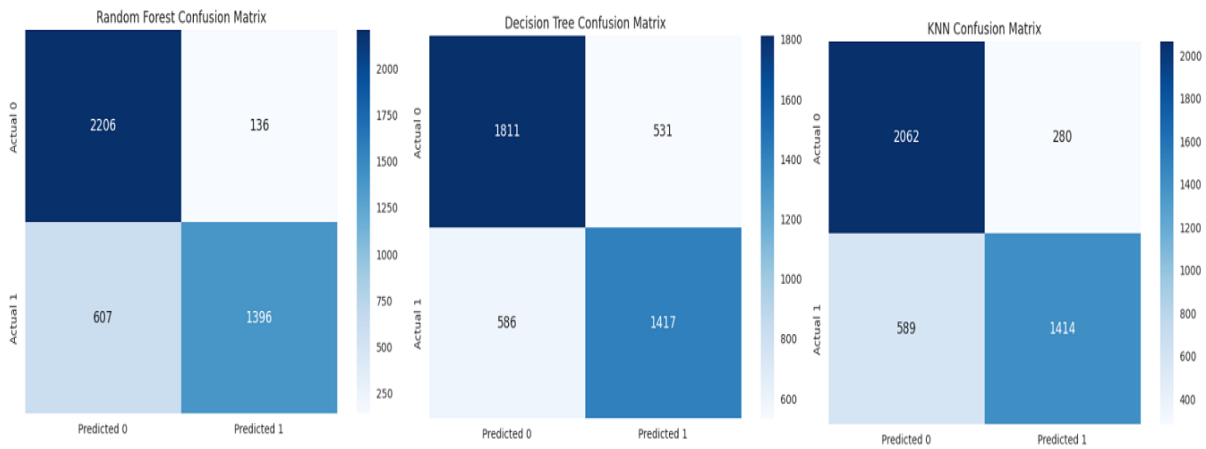


FIG. 3.12 : Matrices de confusion des trois algorithmes pour évaluer les prédictions réalisées sur la troisième base de données.

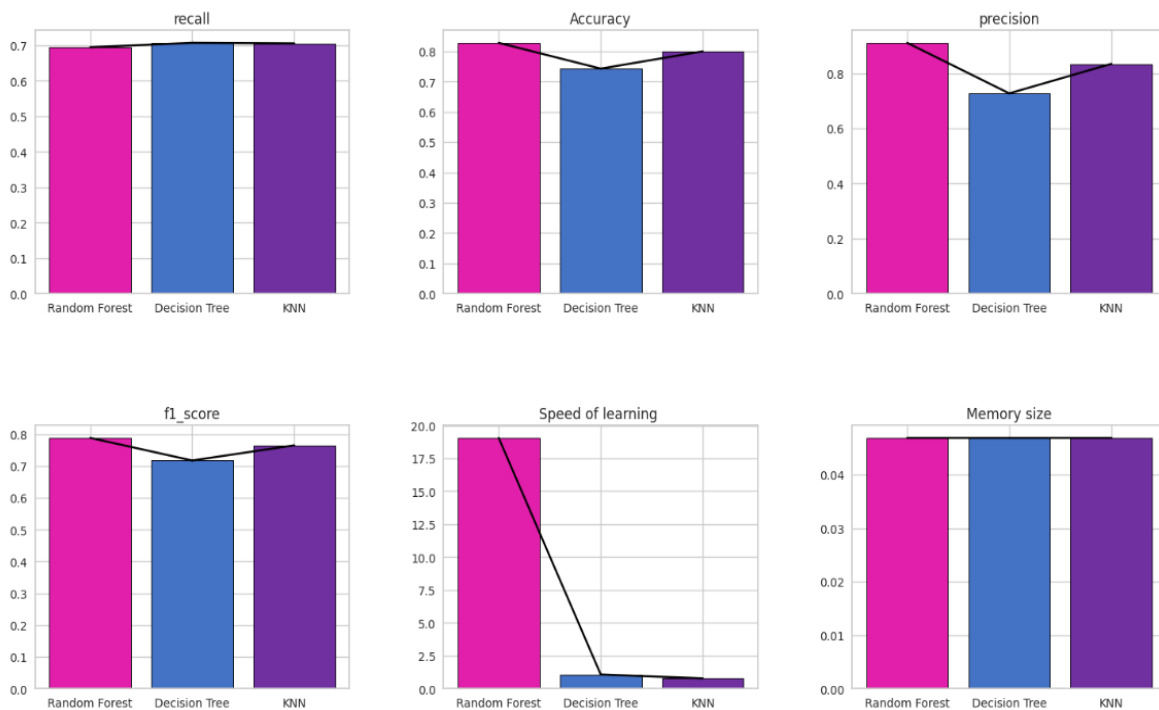


FIG. 3.13 : Comparaison des métriques pour les trois algorithmes pour évaluer les prédictions réalisées sur la troisième base de données.

	Algorithm	Recall	accuracy	precision	f1_score	Speed of learning	Memory size
0	Random Forest Classifier	0.696455	0.827388	0.907612	0.788136	0.160286	0.046875
1	Decision Tree Classifier	0.701448	0.741772	0.728357	0.714649	0.003638	0.046875
2	KNN Classifier	0.701448	0.805524	0.846707	0.714649	0.600522	0.046875

FIG. 3.14 : Récapitulatif des métriques pour les trois algorithmes de la troisième base de données.

Le tableau 3.14 présente un récapitulatif des valeurs des différentes métriques (recall, accuracy, precision, f1-score, memory size et speed of learning) pour les trois algorithmes utilisés sur la troisième base de données.

Discussion

Dans la troisième base de données, en se basant sur les résultats des métriques et des matrices de confusion, nous pouvons discuter de l'ordre de performance des algorithmes.

Le modèle RF Classifier affiche de bonnes performances sur la troisième base de données. La matrice de confusion révèle qu'il prédit correctement la plupart des instances, avec un nombre relativement faible de faux positifs et de faux négatifs. Les métriques montrent un rappel de 0.696455, une exactitude de 0.827388, une précision de 0.907612 et un score F1 de 0.788136. Ces résultats indiquent que le modèle est capable de classifier correctement environ 82,7% des instances de test, avec un bon équilibre entre la précision et le rappel.

Le modèle KNN Classifier affiche également de bonnes performances, bien qu'un peu inférieures à celles du modèle RF. La matrice de confusion révèle une certaine précision dans la prédiction des instances, mais avec un nombre légèrement plus élevé de faux positifs et de faux négatifs par rapport au modèle RF. Les métriques montrent un rappel de 0.701448, une exactitude de 0.805524, une précision de 0.846707 et un score F1 de 0.714649. Ces résultats indiquent que le modèle parvient à classifier correctement environ 80,6% des instances de test, avec une précision élevée.

Le modèle DT Classifier présente des performances légèrement inférieures aux deux autres modèles sur la troisième base de données. La matrice de confusion montre une prédiction correcte des instances, mais avec un nombre plus élevé de faux positifs et de faux négatifs par rapport aux autres modèles. Les métriques affichent un rappel de 0.701448, une exactitude de 0.741772, une précision de 0.728357 et un score F1 de 0.714649. Ces résultats indiquent que le modèle est capable de classifier correctement environ 74,2% des instances de test, avec une précision raisonnable, et en terme de vitesse d'apprentissage Le modèle de DT a la vitesse d'apprentissage la plus rapide parmi les trois algorithmes testés.

3.5.4 Résultats et discussions de la quatrième base de données

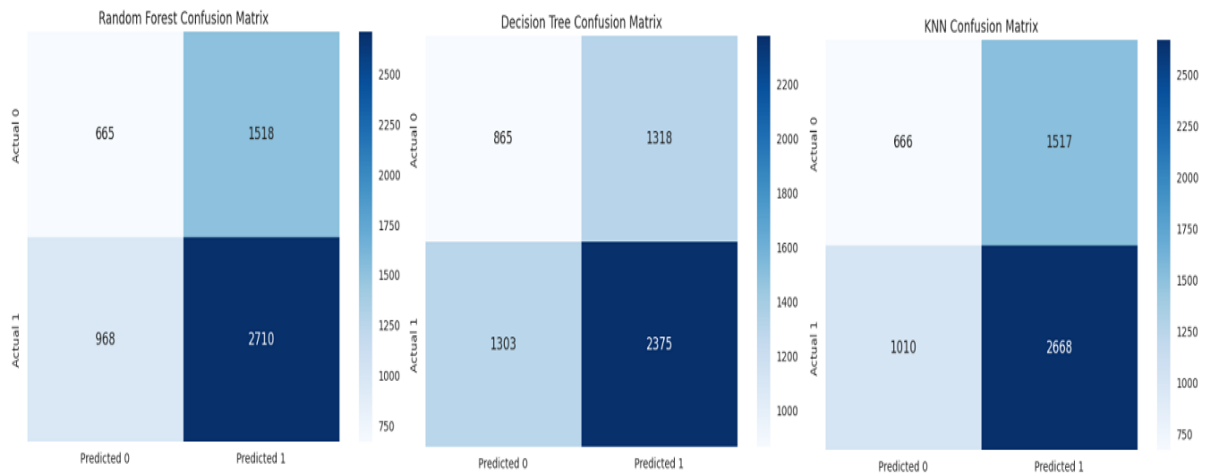


FIG. 3.15 : Matrices de confusion des trois algorithmes pour évaluer les prédictions réalisées sur la quatrième base de données.

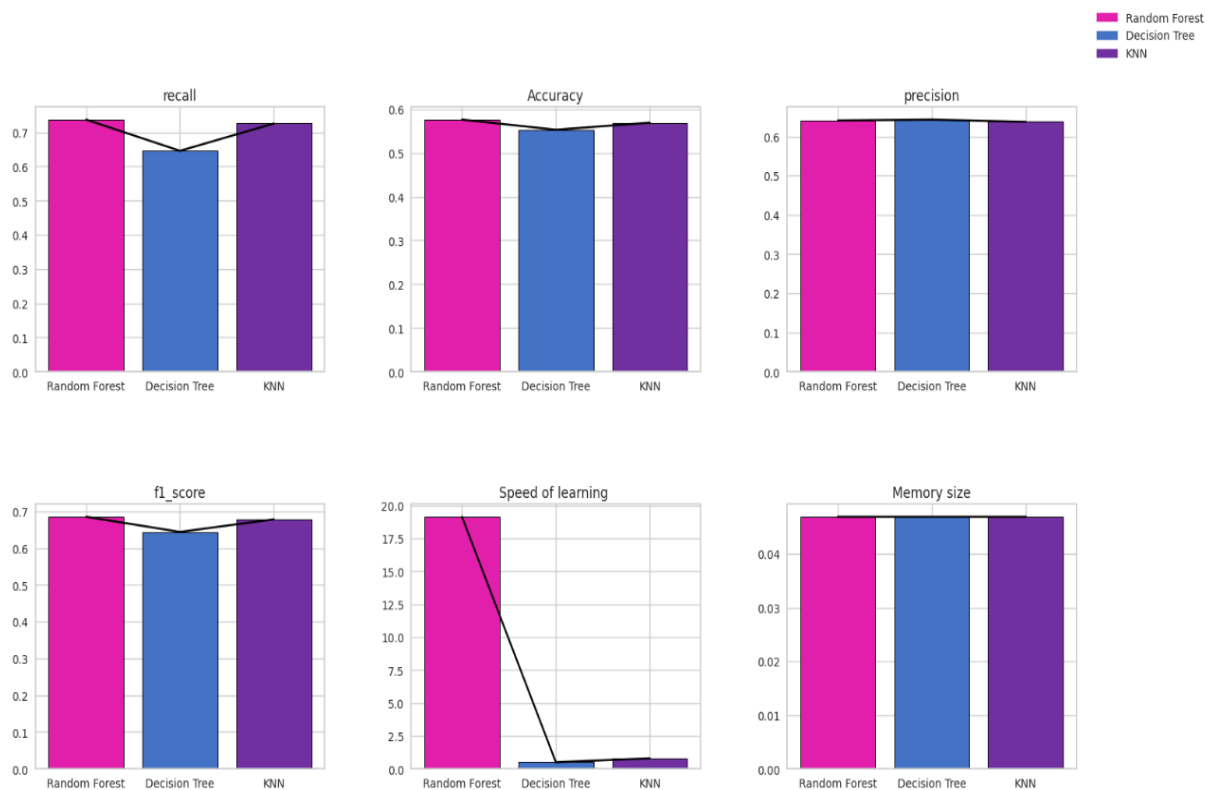


FIG. 3.16 : Comparaison des métriques pour les trois algorithmes pour évaluer les prédictions réalisées sur la quatrième base de données.

Le tableau suivant 3.17 présente un récapitulatif des valeurs des différentes métriques (recall, accuracy, precision, f1-score, memory size et speed of learning) pour les trois algorithmes utilisés sur la quatrième base de données.

	Algorithm	Recall	accuracy	precision	f1_score	Speed of learning	Memory size
0	Random Forest Classifier	0.730560	0.572940	0.639914	0.682239	18.556964	0.046875
1	Decision Tree Classifier	0.646003	0.552465	0.642683	0.644339	0.502911	0.046875
2	KNN Classifier	0.717510	0.561850	0.633157	0.672699	0.586431	0.046875

FIG. 3.17 : Récapitulatif des métriques pour les trois algorithmes de la quatrième base de données.

Discussion

Lorsque nous examinons les performances des trois algorithmes sur la quatrième base de données, nous pouvons évaluer leur ordre de performance en se basant sur les différentes métriques mesurées.

Le modèle RF Classifier affiche une précision globale modérée, avec un rappel de 0.730560, une précision de 0.639914 et un score F1 de 0.682239. Cela indique que le modèle parvient à bien détecter les vrais positifs et à minimiser les faux positifs. Cependant, l'exactitude du modèle est relativement basse à 0.572940, ce qui signifie qu'il peut encore y avoir un pourcentage élevé d'instances mal classées. En termes de vitesse d'apprentissage, le modèle RF Classifier prend plus de temps que les autres modèles, avec une valeur de 18.556964 et sa consommation de mémoire est de 0.046875 GB.

Le modèle DT Classifier présente des performances similaires au modèle RF Classifier. Il affiche un rappel de 0.646003, une précision de 0.642683 et un score F1 de 0.644339. Cependant, son exactitude est encore plus basse à 0.552465, indiquant une plus grande proportion d'instances mal classées. En revanche, le modèle DT Classifier est plus rapide en termes de temps d'apprentissage, avec une valeur de 0.502911, et sa consommation de mémoire est également de 0.046875 GB.

Le modèle KNN Classifier présente des performances similaires au RF en termes de rappel, de précision et de score F1, avec des valeurs de 0.717510, 0.633157 et 0.672699 respectivement. Cependant, il affiche une exactitude de 0.561850, indiquant une meilleure capacité à classer correctement les instances. En revanche, le modèle KNN Classifier prend plus de temps en termes de vitesse d'apprentissage, avec une valeur de 0.586431, et sa consommation de mémoire est également de 0.046875 GB.

3.5.5 Comparaison entre les résultats des 4 ensembles de données

Dans cette section, nous effectuons une comparaison approfondie des résultats obtenus pour les quatre ensembles de données en utilisant trois algorithmes différents. Pour faciliter cette comparaison, nous avons créé des histogrammes qui regroupent les résultats de toutes les métriques évaluées pour chaque algorithme dans chaque base de données.

Les histogrammes représentent visuellement les performances de chaque algorithme en termes de métriques clés telles que le rappel (recall), la précision (precision), le score F1 (f1-score), l'exactitude (accuracy). Chaque barre de l'histogramme représente une métrique spécifique, tandis que la hauteur de la barre indique la valeur associée à cette métrique.

Comparaison des performances de Random Forest par métrique pour chaque base de données

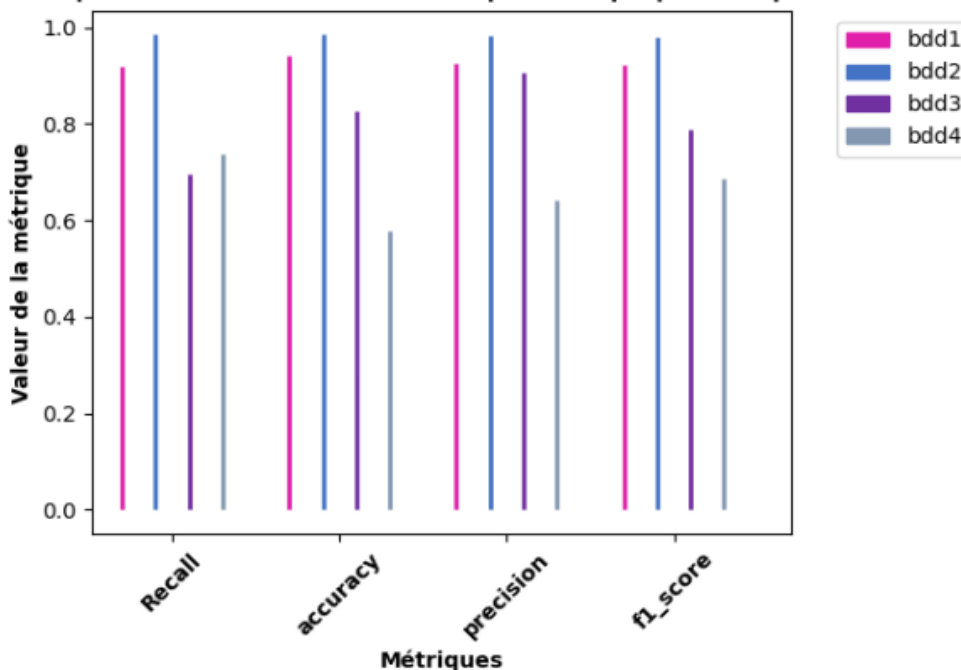


FIG. 3.18 : Comparaison des performances du RF dans les 4 ensembles de données pour les différentes métriques

Comparaison des performances de Decision Tree par métrique pour chaque base de données

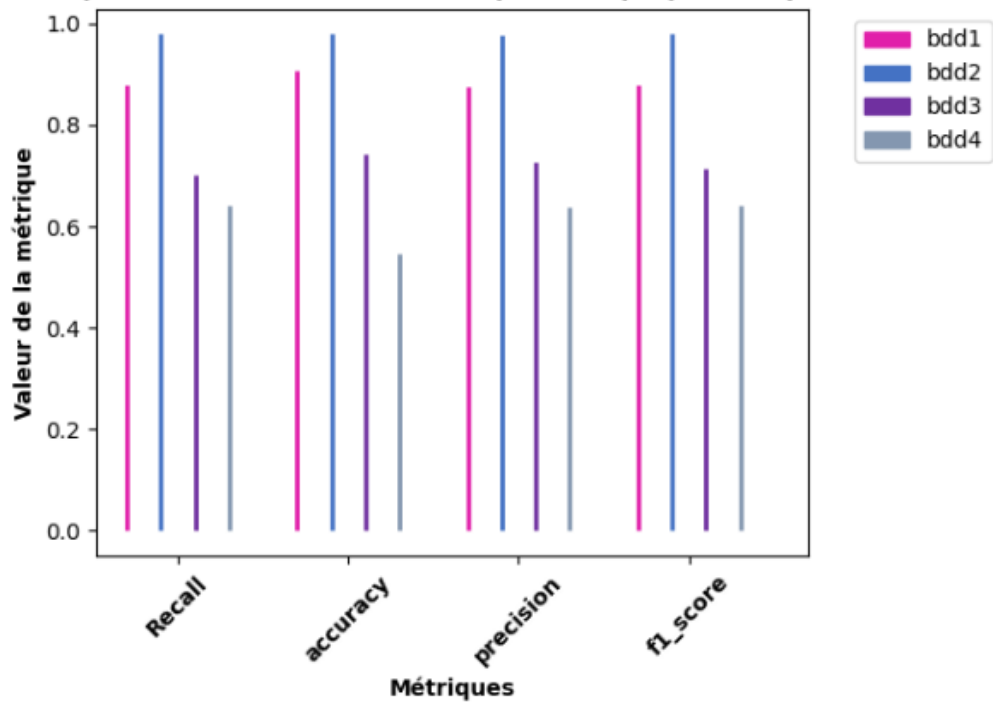


FIG. 3.19 : Comparaison des performances de DT dans les 4 ensembles de données pour les différentes métriques

Comparaison des performances du KNN par métrique pour chaque base de données

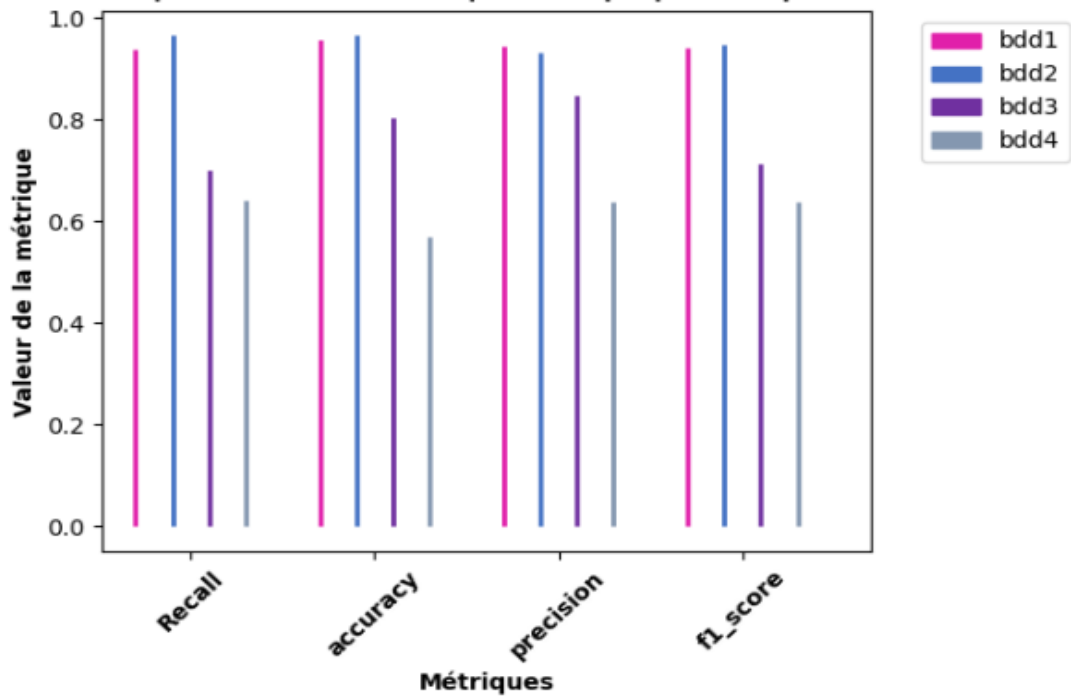


FIG. 3.20 : Comparaison des performances du KNN dans les 4 ensembles de données pour les différentes métriques

Discussion

Après avoir minutieusement analysé les performances des trois algorithmes sur les quatre ensembles de données, des conclusions significatives peuvent être tirées. La deuxième base de données a nettement surpassé les autres ensembles de données en termes de résultats, mettant en évidence l'importance cruciale de l'abondance de données et du nombre de caractéristiques pour améliorer les performances des algorithmes.

De manière similaire, la première base de données a affiché des performances très satisfaisantes, légèrement inférieures à celles de la deuxième base de données. Cette première base de données était également caractérisée par une taille de données importante et un nombre moyen de caractéristiques .

La troisième base de données a également donné de bons résultats, avec une taille de données importante et une présence d'un nombre élevé de caractéristiques . Cette combinaison d'un volume de données conséquent et d'un nombre de caractéristiques adéquat semble avoir été bénéfique pour les performances des algorithmes, leur permettant d'extraire des informations utiles et de produire de bons résultats.

En revanche, la quatrième base de données a montré des résultats relativement faibles en comparaison aux autres. Malgré sa taille de données importante, cette base de données était limitée à seulement trois caractéristiques. Cette restriction en termes de nombre de caractéristiques a probablement eu un impact négatif sur les performances des algorithmes, même avec une quantité de données conséquente.

Il est également essentiel de noter que la deuxième base de données se distinguait par une corrélation significative entre les caractéristiques et la variable cible, tandis que la première base de données présentait une corrélation de niveau moyen. En revanche, les autres ensembles de données présentaient une corrélation faible.

Du fait de cette corrélation modérée, les performances des algorithmes étaient nettement supérieures dans la deuxième et la première base de données. Cela suggère que les algorithmes ont réussi à exploiter efficacement cette corrélation pour obtenir de meilleurs résultats. Dans les autres ensembles de données, où la corrélation était faible, les algorithmes ont peut-être rencontré des difficultés à identifier des modèles significatifs et à obtenir des performances similaires.

3.5.6 Comparaison avec d'autres travaux

Dans cette section, notre objectif est de réaliser une comparaison approfondie entre les résultats obtenus à partir de notre étude sur la maintenance prédictive et ceux présentés dans deux articles scientifiques pertinents. Pour assurer une comparabilité maximale, nous avons spécifiquement choisi la deuxième base de données utilisée dans la partie précédente, car les deux articles sélectionnés ont également effectué leurs études sur cette même base de données.

Aperçu des deux articles sélectionnés

Article A : "Explainable Artificial Intelligence for Predictive Maintenance Applications using a Local Surrogate Model (2021) "

Dans cet article, intitulé "Intelligence Artificielle Explicable pour les Applications de Maintenance Prédictive à l'aide d'un Modèle de Surrogate Local", les auteurs se sont penchés sur l'application de l'algorithme LIME (Local Surrogate Model) pour fournir une interface explicative supplémentaire sur un ensemble de données de maintenance prédictive. L'objectif était de comparer les modèles présentés dans l'article [22] avec un modèle basé sur LIME, afin de déterminer si une qualité d'explication globalement supérieure pouvait être atteinte.[41].

Article B : "Explainable Artificial Intelligence for Predictive Maintenance Applications (2021) "

Dans cet article intitulé "Intelligence Artificielle Explicable pour les Applications de Maintenance Prédictive", les auteurs ont évalué deux méthodes visant à expliquer les résultats de classification d'un classifieur complexe. L'évaluation a été réalisée sur un ensemble de données synthétique de maintenance prédictive. Les deux méthodes présentent des avantages et des limitations inhérentes, mais les arbres de décision se révèlent plus informatifs.[22].

3.5.7 Méthodologie de comparaison

Nous allons nous concentrer sur les métriques d'évaluation couramment utilisées pour évaluer les performances des algorithmes, notamment le rappel, la précision et le F1-score, afin de comparer nos résultats. Dans un premier temps, nous avons collecté les matrices de confusion pour nos deux algorithmes, ainsi que celles des deux articles sélectionnés. Ensuite, à partir de ces matrices de confusion, nous avons calculé les mesures de performance clés. En comparant ces métriques de performance obtenues à partir de nos résultats avec celles des articles, nous serons en mesure d'évaluer les performances de nos algorithmes par rapport à celles présentées dans la littérature.

Résultats

Dans cette section, nous résumons les résultats des différentes métriques d'évaluation dans le tableau ci-dessous 3.7, mettant en évidence les performances des différents algorithmes proposés par les différentes approches.

Approche	Notre approche		Article A	Article B
Technique de ML	RF	DT	LIME	ENSEMBLE D'ARBRES BAGGÉS
Précision	98%	97%	30.7%	86.7%
Rappel	98%	97%	90.9%	70.8%
F1 score	97%	97%	45.9%	80%

TAB. 3.7 : Performances des trois approches selon les différentes métriques d'évaluation

Comparaison et Discussion

En comparant nos résultats avec ceux des articles sélectionnés, nous constatons que notre algorithme RF présente des performances solides, avec un taux de 98%, légèrement supérieur à la précision de 97% obtenue avec Decision Tree. En comparaison, l'approche de l'Article A a montré une précision beaucoup plus faible, seulement 30,7%. Toutefois, l'Article B a obtenu une précision plus élevée de 86,7%.

En ce qui concerne le rappel, nous avons observé des performances similaires entre notre approche avec Random Forest et Decision Tree, avec un taux de rappel de 98%. L'Article A a obtenu un rappel plus élevé de 90,9%, tandis que l'Article B a montré un rappel de 70,8%.

En ce qui concerne le score F1, notre approche avec Random Forest et Decision Tree a obtenu un score F1 de 97%, reflétant une performance équilibrée entre précision et rappel. L'Article A a montré un score F1 plus faible de 45,9%, tandis que l'Article B a obtenu un score F1 plus élevé de 80%.

On constate que notre approche avec RF a généralement affiché les meilleures performances. Elle a surpassé l'approche basée sur DT ainsi que les approches présentées dans l'Article A et l'Article B. Cependant, il convient de noter que l'approche de l'Article B a obtenu des performances compétitives. Ces résultats mettent en évidence l'importance d'explorer différentes techniques du ML pour trouver la méthode la plus adaptée à un scénario de maintenance spécifique. De plus, une préparation minutieuse des données garantit la fiabilité et la précision des résultats.

3.6 Conclusion

Dans ce chapitre consacré à l'implémentation, nos résultats démontrent que les RF et les arbres de décision fonctionnent selon le même principe de partitionnement de l'espace des caractéristiques, mais présentent des performances différentes selon les caractéristiques spécifiques du problème. Pour de petits ensembles de données, il n'y a pas de grande différence de performances entre ces deux algorithmes, bien que le RF se révèle légèrement meilleur que le KNN dans ces cas. Toutefois, avec l'augmentation de la taille des données et l'existence de relations non linéaires entre les variables, le KNN se positionne comme un choix plus efficace.

Dans l'ensemble, nous avons constaté que les algorithmes de machine learning donnent de meilleurs résultats lorsque la taille des données est grande, le nombre de caractéristiques est élevé et lorsque la corrélation entre les caractéristiques et la variable cible est importante. Les RF se démarquent particulièrement dans les problèmes de grande dimension, où ils peuvent gérer efficacement un grand nombre de caractéristiques, réduisant ainsi les risques de surajustement. D'autre part, le KNN excelle dans la capture de relations complexes et non linéaires, surpassant même le RF dans ces situations spécifiques.

Cependant, il est essentiel de noter que chaque algorithme présente ses propres limites et que le choix approprié dépend des caractéristiques spécifiques des ensembles de données. Par exemple, bien que l'arbre de décision ait été le plus performant en termes de taille mémoire grâce à sa structure hiérarchique, il peut souffrir de surajustement lorsqu'il est exposé à des ensembles de données complexes ou bruités. De plus, le KNN, en raison de sa nature basée sur la distance, peut être sensible à l'échelle des variables et nécessiter un prétraitement soigneux des données. Le RF, bien qu'il offre de bonnes performances dans de nombreux cas, peut être plus lent en termes d'apprentissage et nécessiter une plus grande puissance de calcul.

Enfin, il est recommandé de prendre des décisions basées sur le machine learning avec précaution, en tenant compte des spécificités des ensembles de données et en réalisant des tests statistiques. Cela permet de garantir des résultats fiables et d'éviter de prendre des décisions coûteuses basées sur des prédictions peu précises du modèle.

Conclusion et perspectives

Conclusion générale

Le projet de fin d'études s'est focalisé sur l'application de la maintenance prédictive basée sur l'apprentissage automatique. Notre étude approfondie de l'état de l'art de la maintenance prédictive nous a permis d'acquérir une compréhension approfondie des concepts fondamentaux liés à la maintenance, ainsi que d'explorer les différentes formes de maintenance, telles que la maintenance corrective, préventive et prévisionnelle.

Nous avons consacré une attention particulière à l'examen minutieux des outils, des étapes et des techniques de la maintenance prédictive, mettant en évidence son rôle croissant dans une gestion efficace des actifs industriels. De plus, nous avons réalisé une étude comparative des différents types de maintenance, mettant en évidence les avantages et les défis spécifiques inhérents à la maintenance prédictive.

Une étape essentielle de notre projet a été la recherche approfondie de la littérature scientifique relative à la maintenance prédictive et aux travaux antérieurs. Cette exploration nous a permis de découvrir les différentes approches utilisées dans ce domaine, tout en identifiant les lacunes et les critiques formulées à l'égard des études précédentes. Cette analyse critique nous a ainsi permis de renforcer notre propre étude et d'adopter une approche rigoureuse dans la sélection des algorithmes d'apprentissage automatique.

Grâce à cette démarche réflexive, nous avons pu effectuer une sélection minutieuse des algorithmes les plus adaptés à notre étude de maintenance prédictive. L'évaluation des performances de ces algorithmes sur nos jeux de données nous a permis d'obtenir des résultats fiables et pertinents en vue d'optimiser la gestion des actifs industriels.

Au cours de notre étude, nous avons observé que les performances des algorithmes varient en fonction des caractéristiques spécifiques du problème, telles que la taille des données, le nombre de variables et les corrélations entre celles-ci. Les Random Forest se sont révélés efficaces pour les problèmes de grande dimension, tandis que le KNN excelle dans la capture de relations complexes et non linéaires.

Il est important de souligner que chaque algorithme présente ses propres limites, et le choix approprié dépend des particularités propres aux ensembles de données étudiés. En outre, il convient de noter que le machine learning n'est pas exempt de limitations. Il peut générer des résultats imprécis, notamment en présence de données mal étiquetées. Ainsi, une vérification minutieuse des métriques et une validation croisée rigoureuse sont essentielles pour évaluer la capacité de généralisation des modèles.

Perspectives

Pour les perspectives futures,

Explorer d'avantage les techniques d'apprentissage automatique avancées, telles que les réseaux de neurones profonds, pour améliorer les performances de la maintenance prédictive.

Il serait également bénéfique d'approfondir l'analyse des résultats et de proposer des méthodologies plus robustes pour évaluer la performance des modèles de maintenance prédictive.

Notre travail envisage également de proposer une solution hybride, offrant aux futurs chercheurs la possibilité de choisir l'algorithme le plus performant parmi une liste prédéfinie, en comparant leurs performances selon des critères tels que la précision, le rappel, le F1 score, tout en tenant compte de la corrélation entre les features et d'autres paramètres pertinents.

Bibliographie

- [1] Kabouche Abdallah. *Techniques de Maintenance Prédictive pour l'Amélioration de la disponibilité des Installations*. PhD thesis, Université de Annaba-Badji Mokhtar, 2007.
- [2] Mounia Achouch, Mariya Dimitrova, Khaled Ziane, Sasan Sattarpanah Karganroudi, Rizck Dhoub, Hussein Ibrahim, and Mehdi Adda. On predictive maintenance in industry 4.0: Overview, models, and challenges. *Applied Sciences*, 12(16) :8081, 2022.
- [3] Alhamza Alalousi, Rozmie Razif, Mosleh AbuAlhaj, Mohammed Anbar, and Shahrul Nizam. A preliminary performance evaluation of k-means, knn and em unsupervised machine learning methods for network flow classification. *International Journal of Electrical and Computer Engineering*, 6(2) :778, 2016.
- [4] Md Asfi-Ar-Raihan Asif, Mirza Muntasir Nishat, Fahim Faisal, Rezuhanur Rahman Dip, Mahmudul Hasan Udoy, Md Fahim Shikder, and Ragib Ahsan. Performance evaluation and comparative analysis of different machine learning algorithms in predicting cardiovascular disease. *Engineering Letters*, 29(2), 2021.
- [5] Salah Eddine BENCHETTOUH. *Elaboration d'un système de prédiction des pannes et de planification des maintenances*. PhD thesis, UNIVERSITE MOHAMED BOUDIAF-M'SILA-FACULTE MATHEMATIQUES ET DE L'INFORMATIQUE, 2019.
- [6] Messaoud BENZOUAI. Gestion de la maintenance industrielle.
- [7] Hans-Hermann Bock. Clustering methods : a history of k-means algorithms. *Selected contributions in data analysis and classification*, pages 161–172, 2007.
- [8] Rishabh Choudhary and Hemant Kumar Gianey. Comprehensive review on supervised machine learning algorithms. In *2017 International Conference on Machine Learning and Data Science (MLDS)*, pages 37–43. IEEE, 2017.
- [9] Zeki Murat Çınar, Abubakar Abdussalam Nuhu, Qasim Zeeshan, Orhan Korhan, Mohammed Asmael, and Babak Safaei. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, 12(19) :8211, 2020.
- [10] Balbir S Dhillon. *Engineering maintenance : a modern approach*. cRc press, 2002.

- [11] Houda El Aoufir and Driss Bouami. Maintenance des équipements de production : les enjeux de la maîtrise des coûts. *Revue Française de Gestion Industrielle*, 23(3) :71–86, 2004.
- [12] Ashkan Entezari, Alireza Aslani, Rahim Zahedi, and Younes Noorollahi. Artificial intelligence and machine learning in energy systems : A bibliographic perspective. *Energy Strategy Reviews*, 45:101017, 2023.
- [13] Bhumika Gupta, Aditya Rawat, Akshay Jain, Arpit Arora, and Naresh Dhama. Analysis of various decision tree algorithms for classification in data mining. *International Journal of Computer Applications*, 163(8) :15–19, 2017.
- [14] Lakhdar HAMDAOUI, Hocine KHABBAR, et al. *Etude analytique de la maintenance préventive D'un compresseur à vis-ATLAS COPCO GA15*. PhD thesis, 2018.
- [15] Basna Mohammed Salih Hasan and Adnan Mohsin Abdulazeez. A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*, 2(1) :20–30, 2021.
- [16] Sayali D Jadhav and HP Channe. Comparative study of k-nn, naive bayes and decision tree classification techniques. *International Journal of Science and Research (IJSR)*, 5(1) :1842–1845, 2016.
- [17] Archit P Kane, Ashutosh S Kore, Advait N Khandale, Sarish S Nigade, and Pranjali P Joshi. Predictive maintenance using machine learning. *arXiv preprint arXiv :2205.09402*, 2022.
- [18] Khairy AH Kobbacy, DN Prabhakar Murthy, Gabriella Budai, Rommert Dekker, and Robin P Nicolai. Maintenance and production : a review of planning models. *Complex system maintenance handbook*, pages 321–344, 2008.
- [19] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. Supervised machine learning : A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1) :3–24, 2007.
- [20] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industrie 4.0. *Wirtschaftsinformatik*, 56:261–264, 2014.
- [21] Lingjun Li, Shigang Liu, Yali Peng, and Zengguo Sun. Overview of principal component analysis algorithm. *Optik*, 127(9) :3935–3944, 2016.
- [22] Stephan Matzka. Explainable artificial intelligence for predictive maintenance applications. In *2020 third international conference on artificial intelligence for industries (ai4i)*, pages 69–74. IEEE, 2020.
- [23] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.
- [24] George Moraru, Daniel Brun-Picard, Mustapha Ouladsine, and Sébastien Mas. Diagnostic et maintenance prédictive des électrobroches ugv. *Mécanique & industries*, 6(4) :425–430, 2005.

- [25] FY Osisanwo, JET Akinsola, O Awodele, JO Hinmikaiye, O Olakanmi, J Akinjobi, et al. Supervised machine learning algorithms : classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 48(3) :128–138, 2017.
- [26] Marina Paolanti, Luca Romeo, Andrea Felicetti, Adriano Mancini, Emanuele Frontoni, and Jelena Loncarski. Machine learning approach for predictive maintenance in industry 4.0. In *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6, 2018.
- [27] Dominique Pastre. L’intelligence artificielle definition-generalites-historique-domaines. 2000.
- [28] V Mohan Patro and Manas Ranjan Patra. Augmenting weighted average with confusion matrix to enhance classification accuracy. *Transactions on Machine Learning and Artificial Intelligence*, 2(4) :77–91, 2014.
- [29] Peter Poór, David Ženíšek, and Josef Basl. Historical overview of maintenance management strategies : Development from breakdown maintenance to predictive maintenance in accordance with four industrial revolutions. In *Proceedings of the international conference on industrial engineering and operations management, Pilsen, Czech Republic*, pages 23–26, 2019.
- [30] Susmita Ray. A quick review of machine learning algorithms. In *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 35–39. IEEE, 2019.
- [31] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.
- [32] Ramalingam Saravanan and Pothula Sujatha. A state of art techniques on machine learning algorithms : a perspective of supervised learning approaches in data classification. In *2018 Second international conference on intelligent computing and control systems (ICICCS)*, pages 945–949. IEEE, 2018.
- [33] Iqbal H Sarker. Machine learning : Algorithms, real-world applications and research directions. *SN computer science*, 2(3) :160, 2021.
- [34] Sebastian Schwendemann, Zubair Amjad, and Axel Sikora. A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. *Computers in Industry*, 125:103380, 2021.
- [35] Kehar Singh, Dimple Malik, Naveen Sharma, et al. Evolving limitations in k-means algorithm in data mining and their removal. *International Journal of Computational Engineering & Management*, 12(1) :105–109, 2011.
- [36] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance : A multiple classifier approach. *IEEE transactions on industrial informatics*, 11(3) :812–820, 2014.

- [37] Richard S Sutton and Andrew G Barto. *Reinforcement learning : An introduction*. MIT press, 2018.
- [38] E Burton Swanson. The dimensions of maintenance. In *Proceedings of the 2nd international conference on Software engineering*, pages 492–497, 1976.
- [39] Mouna Tarik and Khalid Jebari. Maintenance prediction by machine learning : Study review of some supervised learning algorithms. In *Proceedings of the 2nd African International Conference on Industrial Engineering and Operations Management. Harare, Zimbabwe : IEOM Society International*, 2020.
- [40] Tony Thomas, Athira P Vijayaraghavan, and Sabu Emmanuel. *Machine learning approaches in cyber security analytics*. Springer, 2020.
- [41] Andrea Torcianti and Stephan Matzka. Explainable artificial intelligence for predictive maintenance applications using a local surrogate model. In *2021 4th International Conference on Artificial Intelligence for Industries (AI4I)*, pages 86–88. IEEE, 2021.
- [42] Ngoc Trung Tran, Hung Truong Trieu, Vu Tung Tran, Huu Hai Ngo, and Quang Khoa Dao. An overview of the application of machine learning in predictive maintenance. *Petrovietnam Journal*, 10:47–61, 2021.
- [43] Sergii Voronov. *Machine learning models for predictive maintenance*. PhD thesis, Linköping University Electronic Press, 2020.

Annexes

Annexe A

A.1 Préparation de données

```
# DataFrame
from collections import Counter
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

# Visualization
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib as mpl
import plotly.express as px
import plotly.figure_factory as ff
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import missingno as msno
from sklearn.metrics import confusion_matrix, recall_score, precision_score

# Styling
%matplotlib inline
from termcolor import colored, cprint
mpl.rcParams['axes.unicode_minus'] = False
plt.rcParams["font.family"] = "cursive"

# Models
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier

# Preprocessing
from sklearn.model_selection import cross_val_score, KFold
from sklearn.preprocessing import StandardScaler

# Model Evaluation
from sklearn import metrics
from sklearn.metrics import accuracy_score, recall_score, f1_score, roc_curve
```

FIG. A.1

```
data = pd.read_csv('/kaggle/input/premier-data-set-.csv')
```

FIG. A.2

```
#Suppression des valeurs non nécessaires  
data.drop(["ID", "Datetime"], axis=1, inplace=True)
```

FIG. A.3

```
# Vérifier les informations générales sur les données  
print("Informations sur les données :")  
print(data.info())  
  
# Vérifier s'il y a des valeurs manquantes dans les données  
print("\nValeurs manquantes :")  
print(data.isnull().sum())  
  
# Vérifier s'il y a des valeurs aberrantes dans les données  
print("\nValeurs aberrantes :")  
# Supposons que nous voulons vérifier la colonne 'valeur' pour des valeurs aberrantes  
aberrant_threshold = 3 # Définir le seuil pour déterminer les valeurs aberrantes  
aberrant_values = data[data['valeur'] > aberrant_threshold]  
print(aberrant_values)
```

FIG. A.4

```
# Créer un objet imputeur avec la stratégie de remplacement par la médiane
imputer = SimpleImputer(strategy='median')

# Remplacer les valeurs manquantes par la médiane
data_imputed = pd.DataFrame(imputer.fit_transform(data), columns=data.columns)

# Vérifier s'il y a encore des valeurs manquantes dans les données imputées
print("Valeurs manquantes après l'imputation :")
print(data_imputed.isnull().sum())

# Définir la limite supérieure et inférieure pour détecter les valeurs aberrantes
upper_limit = data['valeur'].mean() + 3 * data['valeur'].std()
lower_limit = data['valeur'].mean() - 3 * data['valeur'].std()

# Remplacer les valeurs aberrantes par des valeurs interpolées
data['valeur'] = np.where(data['valeur'] > upper_limit, np.nan, data['valeur'])
data['valeur'] = np.where(data['valeur'] < lower_limit, np.nan, data['valeur'])
data['valeur'].interpolate(inplace=True)

# Vérifier s'il y a encore des valeurs aberrantes dans les données après l'interpolation
print("Valeurs aberrantes après l'interpolation :")
aberrant_values = data[(data['valeur'] > upper_limit) | (data['valeur'] < lower_limit)]
print(aberrant_values)
```

FIG. A.5

```
# Normalisation des données (plage 0-1)
scaler = MinMaxScaler()
data_normalized = pd.DataFrame(scaler.fit_transform(data), columns=data.columns)

# Standardisation des variables (moyenne 0, écart-type 1)
scaler = StandardScaler()
data_standardized = pd.DataFrame(scaler.fit_transform(data), columns=data.columns)

# Réduction de bruit des données avec PCA
pca = PCA(n_components=2)
data_pca = pd.DataFrame(pca.fit_transform(data_normalized), columns=['PC1', 'PC2'])

# Vérifier les statistiques descriptives après les transformations
print("Statistiques descriptives après les transformations :")
print(data_pca.describe())
```

FIG. A.6

```
import pandas as pd
#AFFICHER LES DIFFERENTES CARACTERISTIQUES DU DATASET

# Taille de la base de données
taille_base_de_donnees = data.shape[0] # Nombre d'instances
nombre_variables = data.shape[1] # Nombre de variables

# Nature des variables
types_variables = data.dtypes # Types de données des variables dans la base de données

# Qualité des données
donnees_manquantes = data.isnull().sum() # Nombre de valeurs manquantes par variable
valeurs_aberrantes = data.describe() # Résumé statistique pour détecter des valeurs aberrantes

# Distribution des classes (pour chaque colonne catégorielle)
distribution_classes = {}
for colonne in data.select_dtypes(include='object'): # Sélectionne uniquement les colonnes catégorielles
    distribution_classes[colonne] = data[colonne].value_counts()

# Bruit dans les données (pour chaque colonne numérique)
sparsite_donnees = (data == 0).sum() / data.shape[0] # Pourcentage de valeurs nulles par variable
# Afficher les résultats
print("Taille de la base de données :", taille_base_de_donnees)
print("Nombre de variables :", nombre_variables)
print("Types de variables :", types_variables)
print("Données manquantes par variable :\n", donnees_manquantes)
print("Distribution des classes :")
for colonne, distribution in distribution_classes.items():
    print(colonne, ":\n", distribution)
print("Sparsité des données :\n", sparsite_donnees)
```

FIG. A.7

A.2 Visualisation des données

```
# Calculer la matrice de corrélation  
correlation_matrix = data.corr()  
  
# Obtenir les valeurs de corrélation sous forme de texte  
correlation_text = correlation_matrix.to_string()  
  
# Afficher les valeurs de corrélation  
print(correlation_text)
```

FIG. A.8

```
# Dessiner graphiquement chaque paire de features dans un diagramme de dispersion  
sns.pairplot(data)  
plt.show()
```

FIG. A.9

```
# Visualiser les tendances avec un graphique linéaire  
sns.lineplot(data=data)  
plt.show()  
  
# Identifier les anomalies avec un boxplot  
sns.boxplot(data=data)  
plt.show()
```

FIG. A.10

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Supposons que vous ayez un DataFrame appelé "data" avec des valeurs aberrantes

# Identifiez les valeurs aberrantes pour chaque colonne
outliers = (data - data.mean()) > 3 * data.std()

# Remplacez les valeurs aberrantes par la médiane de chaque colonne
data_cleaned = data.copy()
for column in data.columns:
    median = data[column].median()
    data_cleaned.loc[outliers[column], column] = median

# Dessinez les diagrammes de boîte pour chaque colonne après le traitement des valeurs aberrantes
for column in data_cleaned.columns:
    plt.boxplot(data_cleaned[column])
    plt.title(f"Diagramme de boîte de la colonne {column} après le traitement des valeurs aberrantes")
plt.show()
```

FIG. A.11

ملخص

يركز مشروع التخرج هذا على تطبيق التعلم الآلي في الصيانة التنبؤية. في الجزء الأول من المشروع، يتم إجراء مراجعة شاملة لأحدث اتجاهات الصيانة التنبؤية. يستكشف الجزء الثاني من المشروع المفاهيم الأساسية للتعلم الآلي وتطبيقها في سياق الصيانة التنبؤية. وأخيراً، يتم تقديم النتائج التي تم الحصول عليها في الجزء الثالث من المشروع بعد تطبيق خوارزميات التعلم الآلي على مجموعات البيانات الخاصة بالصيانة التنبؤية. يقدم هذا المشروع إسهاماً مهماً في تحسين ممارسات الصيانة التنبؤية باستخدام التعلم الآلي. وتشير النتائج المحصل عليها إلى فعالية الخوارزميات المختارة في توقع الأعطال وتقييم أداء النظم.

أقرب كي القرار، شجرة العشوائية، الغابات بالإشراف، اليتم التعلم التنبؤية، الصيانة الآلة، تعلم مفتاحية: كلمات الجيران

Résumé

Ce projet de fin d'études se concentre sur l'application de l'apprentissage automatique à la Maintenance Prédictive. Dans un premier temps, une revue approfondie de l'état de l'art de la Maintenance Prédictive est effectuée. La deuxième partie du projet explore les concepts clés de l'apprentissage automatique et leur mise en pratique dans le contexte de la Maintenance Prédictive. Enfin, la troisième partie du projet présente les résultats obtenus lors de l'application des algorithmes d'apprentissage automatique à des ensembles de données spécifiques de la Maintenance Prédictive. Ce projet apporte une contribution significative à l'amélioration des pratiques de Maintenance Prédictive en utilisant l'apprentissage automatique. Les résultats obtenus démontrent l'efficacité des algorithmes sélectionnés pour prédire les pannes et évaluer les performances des systèmes.

Mots clés : Apprentissage automatique ,Maintenance prédictive, Apprentissage supervisé , Forêts aléatoires ,Arbre de décision ,KNN

Abstract

This final-year project focuses on the application of machine learning to Predictive Maintenance. It begins with an in-depth review of the state of the art in Predictive Maintenance. The second part of the project explores the key concepts of machine learning and their practical application in the context of Predictive Maintenance. Finally, the third part of the project presents the results obtained when applying machine learning algorithms to specific Predictive Maintenance datasets. This project makes a significant contribution to the improvement of Predictive Maintenance practices using machine learning. The results obtained demonstrate the effectiveness of the selected algorithms in predicting failures and assessing system performance.

Keywords : Machine learning , Predictive maintenance , Supervised learning , Random forest , Decision Tree ,KNN
