BIBLIOTHÈQUE
CÉGEP DE L'ABITIBI-TÉMISCAMINGUE
UNIVERSITÉ DU QUÉBEC EN ABITIBI-TÉMISCAMINGUE

Mise en garde

La bibliothèque du Cégep de l'Abitibi-Témiscamingue et de l'Université du Québec en Abitibi-Témiscamingue (UQAT) a obtenu l'autorisation de la personne autrice de ce document afin de diffuser, dans un but non lucratif, une copie de son œuvre dans Depositum, site d'archives numériques, gratuit et accessible à tous. L'autrice ou l'auteur conserve néanmoins ses droits de propriété intellectuelle, dont son droit d'auteur, sur cette œuvre.

Warning

The library of the Cégep de l'Abitibi-Témiscamingue and the Université du Québec en Abitibi-Témiscamingue (UQAT) obtained the permission of the author to use a copy of this document for nonprofit purposes in order to put it in the open archives Depositum, which is free and accessible to all. The author retains ownership of the copyright on this document.

Université du Québec en Abitibi-Témiscamingue

SYSTÈMES INTELLIGENTS POUR LA PRÉVENTION ET LA GESTION DES DÉFAUTS DANS LES PARCS ÉOLIENS

Mémoire
présenté
comme exigence partielle
de la Maîtrise en ingénierie

Par
Aicha Ibtissem Fadel

Août 2026

REMERCIEMENTS

Ces quelques lignes témoignent de ma profonde reconnaissance envers toutes celles et ceux qui ont contribué, de près ou de loin, à l'aboutissement de ce mémoire.

Je tiens d'abord à remercier chaleureusement mon directeur de recherche, le **Pr Fouad Slaoui Hasnaoui**, pour sa confiance, sa disponibilité et la rigueur scientifique qu'il m'a transmise. Ses conseils avisés et son enthousiasme constant ont été déterminants pour la réussite de ce travail.

Ma gratitude s'adresse également à l'ensemble du **corps professoral** et au **personnel administratif de l'UQAT** : merci pour votre accueil, votre accompagnement attentif et la qualité des enseignements dispensés dans le programme de maîtrise en ingénierie, concentration génie électrique.

Je remercie enfin les **membres du jury** d'avoir accepté d'évaluer ce mémoire. J'espère que vous y trouverez la clarté et la rigueur attendues.

Une pensée toute particulière va à **ma famille**, surtout à **mon père** qui, s'il était encore là, serait très fier de moi. Leur soutien moral et économique m'a permis de mener à bien ce parcours.

Je souhaite exprimer mon affection et ma gratitude à **mon mari**, qui, par ses relectures méticuleuses, ses conseils techniques et son engagement constant, a joué le rôle d'un véritable co-encadrant officieux.

Sans votre appui indéfectible, ce projet n'aurait pu aboutir. À chacune et chacun d'entre vous, recevez l'expression de ma plus sincère reconnaissance.

DÉDICACE

Je dédie ce travail à ceux qui ont été ma source de force, de courage et de motivation tout au long de ce parcours.

À la mémoire de mon père, dont l'amour, les valeurs et les enseignements continuent de m'accompagner malgré son absence. Papa, j'aurais tant aimé que tu sois présent pour partager avec moi l'aboutissement de ce parcours. Ta confiance en moi et tout ce que tu m'as transmis resteront à jamais une source d'inspiration.

À ma mère, pour son amour inconditionnel, sa patience, sa bienveillance et son soutien sans faille. Maman, tes encouragements et ta présence m'ont permis de continuer à avancer, même dans les moments les plus difficiles.

À mon mari, Michel, pour sa patience, sa compréhension, son soutien et ses encouragements tout au long de cette aventure. Ta présence à mes côtés a été précieuse dans l'accomplissement de ce travail.

À ma fille, Fatima, qui est arrivée dans ma vie au moment des dernières étapes de ce parcours. Sa naissance a repoussé ma soutenance, mais elle l'a aussi rendue encore plus spéciale puisqu'elle a finalement pu y assister à mes côtés. Ta présence a donné à cet accomplissement une signification encore plus profonde.

Enfin, à toute ma famille et à mes proches, en particulier à mes oncles pour leur présence, leur soutien et leurs encouragements constants. Vous avez toujours été un appui précieux pour moi tout au long de ce parcours.

TABLE DES MATIÈRES

REMERCIEMENTS II
REMERCIEMENTS ..... **ERROR! BOOKMARK NOT DEFINED.**
TABLE DES MATIÈRES IV
LISTE DES FIGURES VII
LISTE DES TABLEAUX IX
LISTE DES SIGLES ET DES ABRÉVIATIONS X
RÉSUMÉ XI
ABSTRACT XII
CHAPITRE 1. INTRODUCTION 1
1.1 Contexte général 1
1.2 Problématique 2
1.3 Objectifs 3
1.3.1 Objectif général 3
1.3.2 Objectifs spécifiques 3
1.4 Hypothèses de recherche 3
1.5 Méthodologie de recherche 4
1.5.1 Approche scientifique 4
1.5.2 Démarche suivie 5
1.5.2.1. Présentation de la base de données utilisée 5
1.5.2.2. Le prétraitement des données (preprocessing) 6
1.6 Conclusion 6
CHAPITRE 2. REVUE DE LA LITTÉRATURE 7
2.1 Présentation des éoliennes modernes 7
2.1.1 Rotor et pales 7
2.1.2 Transmission et générateur 8
2.1.3 Systèmes de contrôle et de protection 8
2.1.4 Courbe de puissance théorique 9
2.1.5 Coefficient de puissance et efficacité 10
2.1.6 Types principaux d'éoliennes 11
2.1.6.1. Éoliennes terrestres 11
2.1.6.2. Éoliennes offshores 11
2.2 Surveillance des pannes et maintenance des éoliennes 11
2.2.1 Introduction au système SCADA dans les parcs éoliens 11

2.2.2 Paramètres SCADA fondamentaux 12
2.2.3 Défis liés à l'utilisation des données SCADA 13
2.2.4 Potentiel et avenir des données SCADA 13
2.2.5 Définitions 14
2.2.6 Techniques de détection des défauts 14
2.2.7 Les différents types de maintenance 15
2.2.7.1. Maintenance Corrective 15
2.2.7.2. Maintenance Préventive 16
2.2.7.3. Maintenance Prédictive 16
CHAPITRE 3. CADRE THÉORIQUE 18
3.1 Méthodes Basées sur l'Intelligence Artificielle et Prédiction des Défauts 18
3.1.1 Machine Learning (Apprentissage Automatique) 18
3.1.1.1. L'Apprentissage Supervisé 18
3.1.1.2. L'Apprentissage Non Supervisé 20
3.1.2 Ensemble Learning 21
3.1.2.1. Random Forest 23
3.1.2.2. XGBoost 25
3.1.2.3. LightGBM 26
3.1.2.4. Logistic regression 27
3.1.2.5. CatBoost 28
3.1.3 Le Méta-Modèle en Apprentissage Automatique 29
CHAPITRE 4. MÉTHODOLOGIE 31
4.1 Stratégie d'apprentissage et processus 31
4.1.1 Présentation et Prétraitement des données 32
4.2 Stratégie d'apprentissage et entraînement du modèle 33
4.2.1 Intégration et validation du système intelligent 35
4.2.2 Méthodologie en Trois Couches : Fonctionnement détaillé 36
4.2.2.1. Couche 1 : Détection préliminaire des anomalies 36
4.2.2.2. Couche 2 : Classification multi-classe des anomalies 37
4.2.2.3. Couche 3 : Affinage et validation des anomalies rares 37
4.2.3 Évaluation des modèles 39
4.2.3.1. Définition des composantes de la matrice de confusion 40
4.2.3.2. Définition des métriques de performance 42
4.2.3.3. Courbe ROC et AUC (Receiver Operating Characteristic) 43

4.2.3.4. Log-Loss (Fonction de perte logarithmique) 43
CHAPITRE 5. RÉSULTATS ET DISCUSSION 45
5.1 Rappel de l'organisation du dispositif 45
5.1.1 Première couche : détection initiale des signaux suspects 45
5.1.2 Deuxième couche : combinaison des prédictions 46
5.1.3 Troisième couche : traitement ciblé des cas ambigus 46
5.2 Évaluation de l'apprentissage du modèle 47
5.2.1 Couche 1 : Classification initiale 47
5.2.1.1. Taille de l'échantillon analysé 47
5.2.1.2. Comparaison des performances globales des modèles supervisés 48
5.2.1.3. Analyse des erreurs à partir des matrices de confusion 50
5.2.1.4. Évaluation de la capacité de généralisation 52
5.2.1.5. Validation sur un jeu de données indépendant 55
5.2.1.6. Résultats de la détection non supervisée 57
5.2.2 Couche 2 : Classification des anomalies détectées 58
5.2.2.1. Performances globales des modèles 59
5.2.2.2. Analyse des erreurs à partir des matrices de confusion 61
5.2.2.3. Évaluation de la performance : Test et validation croisée 66
5.2.3 Couche 3 : spécialisation par cas ambigus 67
5.2.3.1. Comparaison des performances des modèles sur le sous-groupe communication 68
5.2.3.2. Analyse des erreurs à partir des matrices de confusion 69
5.2.3.3. Comparaison des performances des modèles sur le sous-groupe Informational 71
5.2.3.4. Comparaison des performances des modèles sur le sous-groupe Warning 74
5.2.3.5. Comparaison des performances des modèles sur le sous-groupe Stop 76
5.3 Évaluation fonctionnelle du système déployé 78
5.4 Analyse critique du travail réalisé 80
CONCLUSION 83
ANNEXE A – EXTRAITS DES DONNÉES SCADA POUR LA TURBINE 1 85
BIBLIOGRAPHIE 86

LISTE DES FIGURES

Figure 1 Illustration des composants des éoliennes [4]. 7
Figure 2 Illustration schématique d'une turbine avec boîte de vitesse et une turbine direct-drive [7]. 8
Figure 3 Courbe de puissance d'une éolienne en fonction de la vitesse du vent [9] 10
Figure 4 Courbe représentant la relation entre la réduction de vitesse du vent incident (a) et le coefficient de puissance (c_p) [9]. 10
Figure 5 Schéma d'un apprentissage supervisé [25]. 19
Figure 6 Schéma d'un apprentissage non-supervisé [25]. 20
Figure 7 Représentations schématiques des principales techniques d'ensemble learning [30, 31]. 22
Figure 8 Représentation schématique du processus d'agrégation de l'algorithme Random Forest [39]. 24
Figure 9 Schéma de fonctionnement de l'algorithme XGBoost [40]. 25
Figure 10 Schéma de fonctionnement de l'algorithme LightGBM [41]. 26
Figure 11 Schéma de fonctionnement de l'algorithme CatBoost. 29
Figure 12 Système intelligent pour la prévention 32
Figure 13 Processus de prétraitement des données. 33
Figure 14 Processus de développement du système. 34
Figure 15 Schéma global du processus d'intégration, validation et déploiement du système. 35
Figure 16 Caractéristiques les plus importantes pour la division des sous-groupes. 39
Figure 17 Matrice de confusion pour un problème à trois classes. 41
Figure 18 Taille et répartition des données utilisées. 48
Figure 19 Matrice de confusion du méta-modèle (régression logistique) 52
Figure 20 Comparaison des taux d'accuracy en entraînement et en test pour chaque modèle. 53
Figure 21 Comparaison des taux d'accuracy en test pour chaque modèle et pour le méta-modèle 54
Figure 22 Matrice de confusion obtenue avec les données de validation 55
Figure 23 Rapport de classification obtenu avec les données de validation. 56
Figure 24 Rapport de classification du Meta-Modèle basé sur Isolation Forest. 57
Figure 25 Visualisation de la détection des anomalies par Isolation Forest dans l'espace réduit PCA. 58
Figure 26 Encodage des types de messages SCADA pour la classification. 59
Figure 27 Matrices de confusion des modèles de classification de la deuxième couche. 61

Figure 28 Matrice de confusion du méta-modèle de la deuxième couche. 62
Figure 29 Rapports de classification des modèles de la deuxième couche 63
Figure 30 Rapport de classification du méta-modèle pour la deuxième couche. 65
Figure 31 Comparaison des performances globales : Entraînement, test et validation croisée. . 66
Figure 32 Comparaison des performances des différents modèles appliqués au sous-groupe "Communication". 68
Figure 33 Matrices de confusion comparées pour LightGBM, Random Forest, XGBoost, CatBoost, Gradient Boosting et Régression Logistique. 70
Figure 34 Comparaison des performances des différents modèles appliqués au sous-groupe "Informational". 71
Figure 35 Comparaison des taux d'accuracy en test pour chaque modèle, pour le sous-groupe "Informational". 73
Figure 36 Comparaison des performances des différents modèles appliqués au sous-groupe "Warning". 75
Figure 37 Comparaison des performances des différents modèles appliqués au sous-groupe "Stop". 77
Figure 38 Interface utilisateur du système de surveillance intelligent. 79
Figure 39 Extrait de l'historique des enregistrements SCADA pour la turbine 1. 85
Figure 40 Extrait de données SCADA pour les événements de la turbine 1. 85

LISTE DES TABLEAUX

Tableau 1 Résumé comparatif des caractéristiques des éoliennes terrestres et *offshore* 11
Tableau 2 Paramètres de base des éoliennes dans le système SCADA [5]. 12
Tableau 3 Types de défauts dans les turbines des éoliennes [14]. 14
Tableau 4 Comparaison des différents types de maintenance..... 17
Tableau 5 Liste des messages d'alertes ainsi que le niveau de gravité correspondant. 38
Tableau 6 Résumé comparatif des performances des modèles dans la première couche de détection 49
Tableau 7 Matrices de confusion des modèles supervisés dans la première couche de détection 51
Tableau 8 Résumé comparatif des performances de classification dans la deuxième couche... 60

LISTE DES SIGLES ET DES ABRÉVIATIONS

IA	Intelligence Artificielle
ML	Machine Learning
SCADA	Supervisory Control and Data Acquisition
XGB	Extreme Gradient Boosting (XGBoost)
LGBM	Light Gradient Boosting Machine (LightGBM)
CB	CatBoost
RF	Random Forest
GB	Gradient Boosting
LR	Logistic Regression (Régression Logistique)
IF	Isolation Forest
KDE	Kernel Density Estimation
MC	Méthode de Monte Carlo
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
CV	Cross-Validation (Validation croisée)
CM	Confusion Matrix (Matrice de confusion)
SD	Synthetic Data (Données synthétiques)
F1	F1-Score
Prec.	Précision
Rec.	Rappel
PCA	Principal Component Analysis

RÉSUMÉ

La fiabilité des parcs éoliens repose en grande partie sur la capacité à détecter rapidement les anomalies dans les données SCADA, qui sont à la fois volumineuses et fortement déséquilibrées. Ce mémoire porte sur le développement d'une approche de classification hiérarchique visant à identifier les états anormaux et à préciser la nature des défauts afin de soutenir la maintenance prédictive.

L'approche proposée repose sur une architecture en trois couches. La première permet de distinguer les observations normales des observations anormales. La deuxième classe les anomalies selon quatre catégories opérationnelles : Communication, Informational, Warning et Stop. La troisième couche est consacrée aux cas rares ou ambigus afin d'affiner le diagnostic lorsque la classification devient plus difficile.

La méthodologie combine plusieurs modèles d'apprentissage supervisé, notamment XGBoost, LightGBM, CatBoost et Random Forest, intégrés dans une approche de stacking, ainsi qu'un modèle non supervisé, Isolation Forest. Des données synthétiques générées par Monte-Carlo et KDE sont également utilisées afin d'améliorer la représentation des classes minoritaires. Les résultats obtenus montrent une exactitude de 99,8 % avec un AUC-ROC proche de 1,00 pour la détection binaire, un F1-score moyen supérieur à 0,97 pour la classification multi-classes et une précision pouvant atteindre 96,7 % pour certains sous-groupes complexes.

Enfin, un prototype d'interface a été développé afin de simuler le fonctionnement du système sur un parc de six éoliennes. Les données historiques sont rejouées sous forme de flux en quasi-temps réel et les diagnostics sont affichés en continu. Cette validation permet d'évaluer le comportement du système dans un contexte proche d'une utilisation opérationnelle. Les travaux futurs pourront notamment porter sur l'explicabilité des modèles, l'optimisation des ressources informatiques et la validation de l'approche sur plusieurs parcs éoliens.

Mots-clés : maintenance prédictive, éolien, données SCADA, classification hiérarchique, modèles ensemblistes, détection d'anomalies, données déséquilibrées, génération synthétique, isolation forest, interface temps réel

ABSTRACT

The reliability of wind farms largely depends on the ability to detect anomalies in SCADA data quickly and accurately. These datasets are typically large and highly imbalanced. This thesis focuses on the development of a hierarchical classification approach designed to identify abnormal operating conditions and determine the nature of detected faults in order to support predictive maintenance.

The proposed approach is based on a three-layer architecture. The first layer distinguishes normal observations from abnormal ones. The second layer classifies detected anomalies into four operational categories: Communication, Informational, Warning, and Stop. The third layer focuses on rare or ambiguous cases to refine the diagnosis when classification becomes more difficult.

The methodology combines several supervised learning models, including XGBoost, LightGBM, CatBoost, and Random Forest, integrated through a stacking approach, together with the unsupervised Isolation Forest algorithm. Synthetic data generated using Monte Carlo simulation and Kernel Density Estimation (KDE) are also used to improve the representation of minority classes. The results show an accuracy of 99.8% with an AUC-ROC close to 1.00 for binary anomaly detection, an average F1-score above 0.97 for multi-class classification, and precision reaching up to 96.7% for some complex subgroups.

Finally, a prototype interface was developed to simulate the operation of the system on a six-turbine wind farm. Historical SCADA data are replayed as a quasi-real-time stream, while diagnostic results are displayed continuously. This validation makes it possible to assess the behavior of the proposed system under conditions close to operational use. Future work may focus on improving model explainability, optimizing computational resources, and validating the approach across multiple wind farms.

Keywords: predictive maintenance, wind energy, SCADA data, hierarchical classification, ensemble models, anomaly detection, imbalanced data, synthetic data generation, isolation forest, real-time interface

CHAPITRE 1. INTRODUCTION

1.1 Contexte général

L'énergie éolienne, source d'énergie renouvelable issue du vent, occupe une place de plus en plus importante dans le paysage énergétique mondial. Grâce à ses caractéristiques propres, durables et respectueuses de l'environnement, elle constitue une alternative crédible aux sources d'énergie conventionnelles, en particulier dans le contexte de la lutte contre les émissions de gaz à effet de serre.

Ces dernières décennies, les avancées en ingénierie, en matériaux et en aérodynamique ont permis de développer des éoliennes plus performantes, fiables et économiquement viables. Cette évolution technologique s'est accompagnée d'une forte croissance des capacités installées à l'échelle mondiale [1]. Le Canada illustre cette dynamique : en 2022, la puissance installée en énergie éolienne y a atteint 17 132 MW, positionnant le pays au 9e rang mondial. Ce développement est soutenu par de nombreux projets ambitieux, à l'image du méga-parc éolien d'Hydro-Québec au Saguenay–Lac-Saint-Jean, qui s'étend sur plus de 5 000 km² avec une capacité totale de 3 000 MW [2].

Cependant, cette expansion rapide s'accompagne de nouveaux défis, notamment en matière de surveillance et de maintenance. Souvent installés dans des zones isolées ou difficiles d'accès, les parcs éoliens sont soumis à des conditions environnementales changeantes et à des contraintes techniques élevées. La détection précoce des anomalies de fonctionnement est donc devenue une priorité pour garantir la continuité du service, la sécurité des installations et la rentabilité des investissements.

Dans ce contexte, les systèmes intelligents basés sur l'analyse des données opérationnelles apparaissent comme une voie prometteuse. En particulier, les données issues des systèmes SCADA (Supervisory Control And Data Acquisition) fournissent des informations essentielles pour le suivi de l'état des turbines. Couplées à des algorithmes d'apprentissage automatique, ces données permettent de détecter les dérives, de classer les défauts et de contribuer à une maintenance proactive et optimisée.

Ce mémoire s'inscrit dans cette dynamique d'innovation. Il a pour objectif de développer une architecture intelligente pour la prévention et la gestion des défauts dans les parcs éoliens. L'approche proposée s'appuie sur l'analyse des données SCADA et l'utilisation combinée de techniques d'apprentissage supervisé, non supervisé et de méta-modélisation. Elle sera évaluée

à l'aide de données réelles ou simulées, afin de valider sa robustesse, sa précision et sa capacité à être mise en œuvre sur le terrain.

1.2 Problématique

Bien qu'elle constitue une alternative durable aux énergies fossiles, l'énergie éolienne est confrontée à des défis majeurs en matière de surveillance, de maintenance et de fiabilité de ses infrastructures. Les composants critiques des turbines, tels que les boîtes de vitesses, les roulements ou les génératrices, sont particulièrement exposés à des défaillances coûteuses, en raison des charges mécaniques dynamiques et des conditions environnementales variables [3]. Ces défaillances peuvent entraîner des pertes économiques significatives, notamment sous forme d'arrêts de production prolongés, d'interventions d'urgence et d'une réduction de la disponibilité opérationnelle des turbines.

Pour surveiller l'état de fonctionnement des équipements, les exploitants s'appuient généralement sur les systèmes SCADA (Supervisory Control and Data Acquisition), qui enregistrent en continu des données opérationnelles. Toutefois, ces systèmes présentent des limitations importantes : ils ne permettent pas de capter efficacement les signaux faibles ou transitoires associés à l'apparition précoce de certains défauts et nécessitent encore une forte intervention humaine pour l'analyse.

Par ailleurs, les contraintes géographiques propres aux sites éoliens rendent les opérations de maintenance plus complexes. Qu'il s'agisse de parcs situés en mer (offshore), dans des zones montagneuses ou dans des régions isolées, l'accessibilité difficile exige une logistique complexe, une planification rigoureuse et le recours à des ressources spécialisées. Ces conditions augmentent les délais d'intervention et les coûts d'exploitation ainsi que les risques pour les équipes techniques.

En outre, la complexité croissante des systèmes éoliens modernes rend le diagnostic plus difficile: les pannes peuvent être multiples, interdépendantes et de nature non linéaire. Les événements rares mais critiques sont sous-représentés dans les historiques, ce qui rend leur détection particulièrement délicate. À cela s'ajoute la problématique de la qualité des données : bruit, valeurs manquantes ou anomalies de mesure limitent la fiabilité des analyses traditionnelles.

Dans ce contexte, il devient nécessaire d'explorer des solutions innovantes, capables de tirer parti des données SCADA disponibles pour automatiser la détection, la classification et la localisation des défauts. C'est dans cette dynamique que s'inscrit le présent travail, qui propose une approche basée sur l'intelligence artificielle, structurée en plusieurs couches, et appliquée à des données

accessibles issues de sources ouvertes. L'objectif est d'améliorer la précision, la robustesse et l'adaptabilité des systèmes de diagnostic pour les parcs éoliens modernes.

1.3 Objectifs

1.3.1 Objectif général

Développer un modèle avancé basé sur les méthodes d'intelligence artificielle afin d'interagir de manière plus précise et efficace avec les anomalies des éoliennes.

1.3.2 Objectifs spécifiques

- Analyser les données SCADA afin d'en extraire les variables les plus représentatives.
- Développer un module intelligent, capable de détecter automatiquement les anomalies dans les données SCADA, s'appuyant à la fois sur des approches supervisées et non supervisées.
- Implémenter une architecture hiérarchique de classification des défauts prenant en compte l'hétérogénéité et le déséquilibre des classes, permettant de distinguer différents types d'événements (arrêts, avertissements, communications, etc.) en fonction de leur criticité.
- Développer un module de traitement ciblé pour les cas ambigus, en s'appuyant sur des approches multi-modèles et des stratégies de spécialisation par sous-groupes, dans le but d'améliorer la sensibilité du système aux situations complexes ou rares.
- Évaluer les performances globales du système intelligent à l'aide de données réelles ou synthétiques, en mobilisant des métriques telles que la précision, le rappel, le F1-score ou encore le log-loss, afin de valider la robustesse et la généralisabilité du modèle.
- Déployer une interface de surveillance interactive permettant une visualisation claire des diagnostics en temps réel et la génération de rapports exploitables pour les équipes de maintenance.

1.4 Hypothèses de recherche

- Les variables extraites des données SCADA devraient permettre de modéliser de façon fiable le comportement normal et anormal des éoliennes, et de constituer une base pertinente pour l'apprentissage automatique.
- Un système combinant des approches supervisées et non supervisées devrait pouvoir détecter efficacement les anomalies dans les données SCADA, même en l'absence d'un étiquetage exhaustif.

- Une architecture hiérarchique de classification devrait améliorer la capacité du système à distinguer les différents types de défauts en tenant compte de leur criticité et du déséquilibre des classes.
- L'utilisation de modèles spécialisés pour traiter les cas ambigus devrait améliorer la sensibilité du système face aux anomalies rares ou difficiles à classer.
- L'utilisation de techniques statistiques comme la simulation Monte Carlo pour générer des données synthétiques devrait permettre d'améliorer la diversité des scénarios d'apprentissage et, par conséquent, la performance globale des modèles développés.
- Le système intelligent devrait maintenir des performances robustes et stables lorsqu'il est appliqué à des jeux de données réelles et synthétiques.

1.5 Méthodologie de recherche

1.5.1 Approche scientifique

Dans le cadre de la présente étude, l'approche méthodologique adoptée s'articule autour d'une structure hiérarchique à trois échelons, élaborée dans le but de traiter les défaillances affectant un parc éolien de manière séquentielle, à partir des données recueillies par le système SCADA. Cette méthodologie a été élaborée pour pallier les principaux défis associés à la variabilité des signaux, à la rareté des anomalies critiques, ainsi qu'à l'hétérogénéité des types de défaillances observés sur le terrain.

Le premier niveau, dédié à la détection initiale des anomalies, vise une sensibilité optimale. Il permet d'isoler les observations potentiellement déviantes tout en conservant les cas incertains, afin de réduire au minimum le risque de faux négatifs. Parallèlement, les observations initialement considérées comme normales sont soumises à une analyse approfondie, suivant une démarche non supervisée, afin d'identifier d'éventuels comportements atypiques non représentés dans les données d'apprentissage.

Les données jugées anormales sont par la suite transmises au deuxième niveau, dédié à leur classification fonctionnelle. Cette étape permet de regrouper les anomalies selon leur nature opérationnelle, de distinguer les événements critiques des alertes secondaires, et de structurer les diagnostics selon leur degré de gravité.

Le troisième niveau, quant à lui, prend en charge les cas complexes, ambigus ou faiblement représentés. Les données y sont analysées par sous-ensembles homogènes, créés de manière à introduire une variabilité contrôlée. Cette étape a pour objectif de perfectionner les diagnostics

en tenant compte des particularités contextuelles et de pallier l'asymétrie entre les classes par une intervention ciblée et spécialisée.

Cette approche graduelle vise à accroître la robustesse, la précision et la pertinence du diagnostic global, tout en préservant une structure modulaire, extensible à d'autres parcs ou contextes industriels. Les méthodologies spécifiques à chaque niveau seront examinées ultérieurement.

1.5.2 Démarche suivie

1.5.2.1. Présentation de la base de données utilisée

Les données utilisées dans le cadre de cette étude sont issues d'un parc éolien situé au Royaume-Uni, mises à disposition via une source libre. L'infrastructure se compose de six turbines, et les données disponibles couvrent la période de 2016 à 2021. Ces données sont réparties en deux catégories :

- Données statiques, comprenant les coordonnées géographiques, les caractéristiques techniques des éoliennes et d'autres paramètres fixes ;
- Données dynamiques, principalement constituées d'un jeu SCADA (Supervisory Control and Data Acquisition) enregistré à une fréquence de 10 minutes, représentant l'évolution temporelle du comportement opérationnel des turbines.

Le fichier SCADA contient 299 variables couvrant un large éventail de mesures, telles que la température, la vitesse du vent, la puissance produite, les vibrations, ou encore les états de fonctionnement (des captures d’écran peuvent être consultées, en annexe A, pour un aperçu des données utilisées). Cependant, en raison de l'importance du volume de données, seules les variables jugées les plus pertinentes ont été sélectionnées pour l'analyse. En outre, l'analyse révèle la présence d'un taux significatif de valeurs manquantes, ce qui requiert un prétraitement rigoureux en amont de l'apprentissage automatique.

Une analyse exploratoire approfondie a été conduite sur l'ensemble des turbines et des périodes disponibles afin d'évaluer la densité, la qualité et la richesse des événements enregistrés. À l’issue de cette étape, le choix s’est porté sur les données enregistrées en 2020, pour l’ensemble du parc, en raison de leur densité événementielle plus élevée, de leur structure temporelle plus complète, et de leur représentativité par rapport aux défauts observés. Cette sélection constitue ainsi un échantillon riche, à la fois en termes de diversité des anomalies et de volume exploitable pour l’apprentissage automatique.

1.5.2.2. Le prétraitement des données (preprocessing)

Le prétraitement des données a consisté à fusionner deux ensembles dynamiques distincts en un seul cadre temporel cohérent. Cette unification repose sur l'utilisation d'un index chronologique commun, permettant d'aligner les événements techniques enregistrés dans le journal des attributs (représentant les anomalies ou états particuliers) avec les mesures opérationnelles archivées toutes les 10 minutes dans le fichier SCADA principal. Chaque événement est ainsi associé à l'enregistrement le plus proche dans le temps, garantissant une correspondance pertinente entre le contexte de fonctionnement et l'occurrence d'un incident.

Par ailleurs, afin de rendre les données compatibles avec les algorithmes d'apprentissage automatique, la variable cible (indiquant le type d'événement) a été transformée à l'aide d'un encodage par labels (label encoding), permettant une représentation numérique exploitable par les modèles de classification. Enfin, une étape essentielle dans cette phase consiste à préparer les données pour leur intégration dans l'architecture intelligente proposée. Les opérations de nettoyage, d'imputation, de sélection des variables pertinentes et de traitement du déséquilibre des classes seront détaillées dans un chapitre subséquent.

1.6 Conclusion

Ce chapitre a établi le cadre général de la recherche en présentant la problématique étudiée, en définissant les objectifs poursuivis, et en exposant de manière synthétique l'approche méthodologique retenue. Il pose ainsi les bases nécessaires à la compréhension du dispositif proposé, dont les aspects techniques seront développés dans les chapitres suivants.

CHAPITRE 2. REVUE DE LA LITTÉRATURE

2.1 Présentation des éoliennes modernes

Les éoliennes modernes sont des systèmes sophistiqués intégrant plusieurs composantes et technologies avancées pour maximiser la production d'énergie et maintenir une efficacité optimale. La figure 1 ci-dessous illustre les principaux composants d'une éolienne, mettant en évidence la structure de la nacelle, du rotor et des pales, ainsi que les systèmes de transmission et de contrôle.

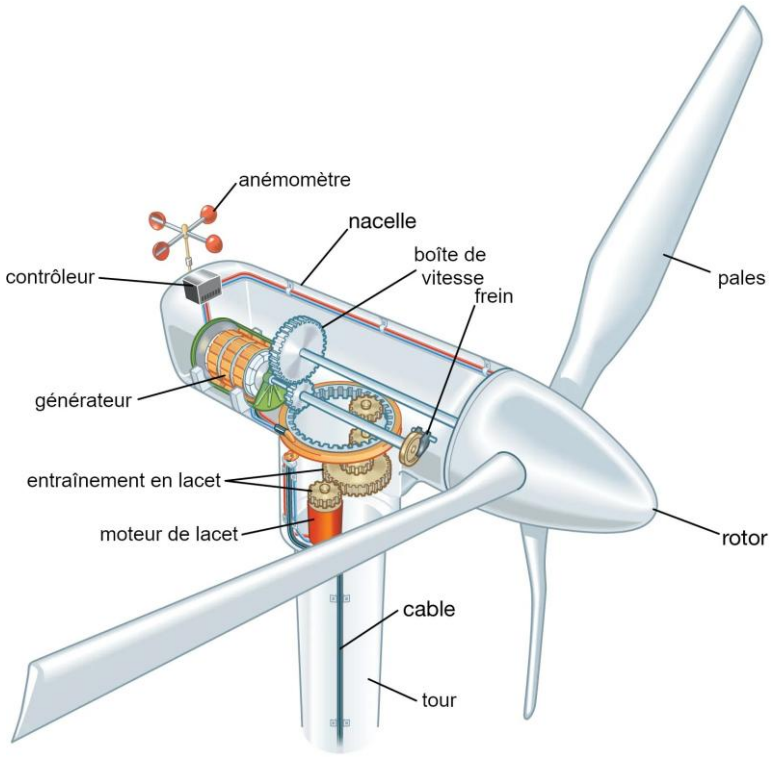


Figure 1
Illustration des composants des éoliennes [4]

La figure montre que, au centre de l'éolienne, se trouve la nacelle, un compartiment abritant les éléments essentiels, tels que le générateur et le système de transmission. La nacelle est fixée au sommet d'une tour suffisamment haute pour capter des vents plus réguliers et puissants, ce qui optimise la rentabilité énergétique de l'installation [5].

2.1.1 Rotor et pales

Le rotor est constitué de trois pales au design aérodynamique qui captent l'énergie cinétique du vent et la convertissent en mouvement de rotation. Ces pales sont fabriquées à partir de matériaux

légers et robustes comme la fibre de verre ou la fibre de carbone. Leur forme aérodynamique, conçue pour générer une force de portance due à la différence de pression de l'air sur chaque côté des pales, fait tourner l'arbre principal de l'éolienne, créant le couple nécessaire à sa rotation [6].

2.1.2 Transmission et générateur

L'énergie de rotation des pales est transmise à l'arbre lent, puis une boîte de vitesses accélère l'arbre rapide pour adapter la vitesse de rotation à celle requise par le générateur. Cependant, certaines éoliennes modernes, comme le montre la figure 2, utilisent un système direct-drive, qui élimine le besoin d'une boîte de vitesses en connectant directement un générateur à faible vitesse mais de grande puissance. Ce système réduit la complexité mécanique et peut améliorer la fiabilité et la longévité des composants.

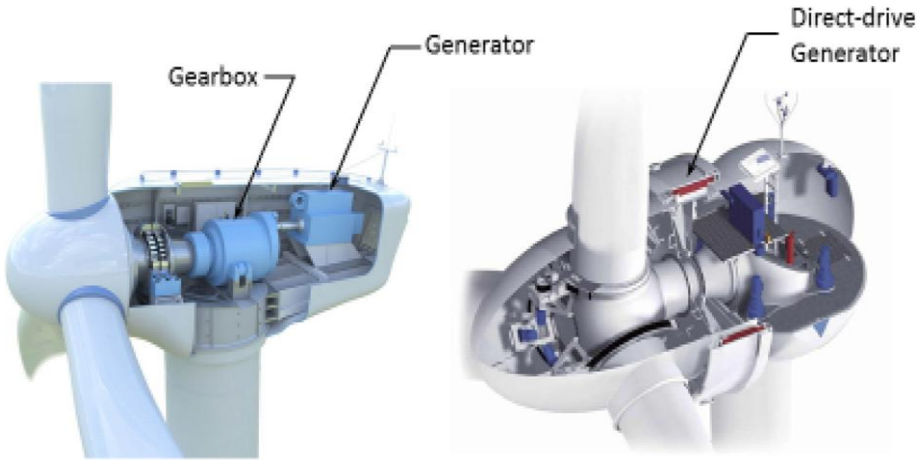


Figure 2
Illustration schématique d'une turbine avec boîte de vitesse et une turbine direct-drive [7].

2.1.3 Systèmes de contrôle et de protection

L'éolienne est équipée de systèmes de contrôle sophistiqués pour ajuster sa position et son fonctionnement en fonction des conditions de vent. Le système de lacet permet à l'éolienne de s'orienter automatiquement face au vent pour optimiser la captation de l'énergie, tandis que le contrôle du pas ajuste l'angle des pales pour réguler la portance et la traînée, maintenant ainsi une vitesse de rotation idéale.

En cas de vents trop forts, des systèmes de freinage (aérodynamiques et mécaniques) qui protègent l'éolienne contre les surcharges. En complément, un système hydraulique assure la

lubrification et le refroidissement de la boîte de vitesses, tandis que le générateur est généralement refroidi par air, garantissant ainsi la durabilité de ces équipements. Ces dispositifs de contrôle et de protection sont essentiels pour optimiser la performance de l'éolienne, maximiser la production et protéger les équipements, même dans des conditions extrêmes [8].

2.1.4 Courbe de puissance théorique

La figure 3 montre la courbe de puissance d'une éolienne en fonction de la vitesse du vent. Elle est divisée en quatre régions distinctes :

- **Région I** : Lorsque la vitesse du vent est inférieure à la vitesse de démarrage, l'éolienne ne produit aucune énergie, car le vent est trop faible pour générer le couple nécessaire à la mise en rotation du rotor.

- **Région II** : Entre la vitesse de démarrage et la vitesse nominale, la puissance produite par l'éolienne augmente presque proportionnellement au cube de la vitesse du vent. Cette relation est décrite par la formule suivante:

$$P = \frac{1}{2} c_p \rho v^3 A$$

où P est la puissance produite, ρ la densité de l'air, A la surface balayée par les pales, c_p le coefficient de puissance, et v la vitesse du vent.

- Région III : Dans cette région, entre la vitesse nominale et la vitesse maximale, l'éolienne atteint sa puissance maximale et un système de limitation de puissance ajuste l'angle des pales pour stabiliser la production, empêchant toute surcharge du générateur.

- Région IV : Lorsque la vitesse du vent dépasse la vitesse maximale, l'éolienne se met en sécurité pour éviter les dommages mécaniques. Les pales sont orientées pour réduire la portance, ou un frein est appliqué au rotor, stoppant la production d'énergie.

La vitesse du vent est un facteur déterminant dans la quantité d'énergie produite par une éolienne. À des vitesses modérées (environ 12 à 14 m/s), l'éolienne atteint son rendement optimal. Au-delà de 25 m/s, elle est généralement arrêtée pour éviter les risques de défaillance [9].

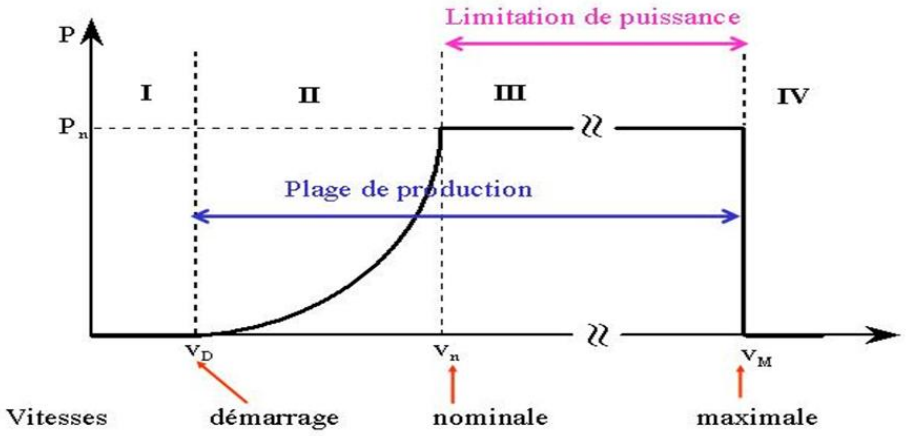


Figure 3
Courbe de puissance d'une éolienne en fonction de la vitesse du vent [9]

2.1.5 Coefficient de puissance et efficacité

Le rendement d'une éolienne est mesuré par son coefficient de puissance (c_p), qui représente la proportion de l'énergie du vent captée par l'éolienne. Ce coefficient a une limite théorique de 59,3 %, appelée limite de Betz, ce qui signifie qu'aucune éolienne ne peut convertir plus de cette proportion d'énergie cinétique en électricité, même dans des conditions idéales. Maximiser ce coefficient est essentiel pour augmenter la production d'électricité éolienne [10].

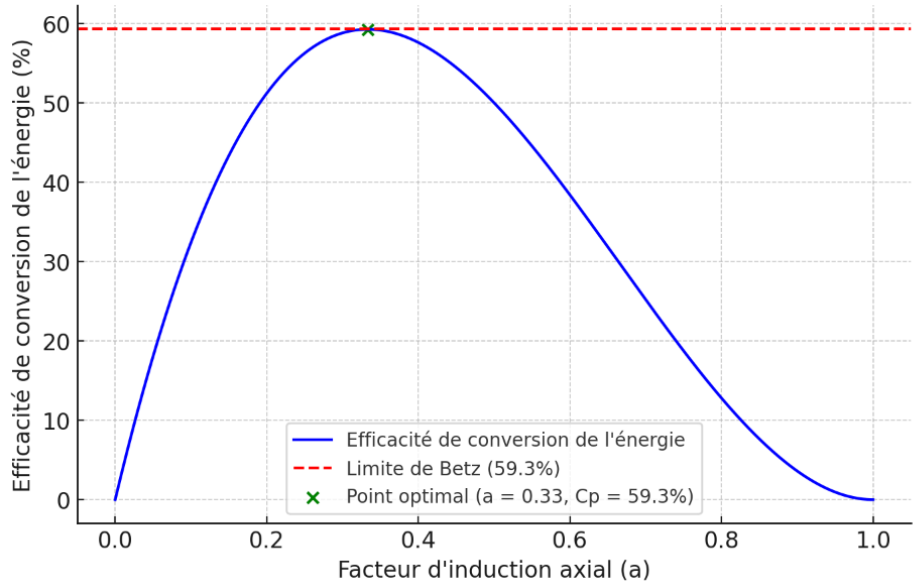


Figure 4
Courbe représentant la relation entre la réduction de vitesse du vent incident (a) et le coefficient de puissance (c_p) [9].

2.1.6 Types principaux d'éoliennes

Les éoliennes modernes se divisent en deux types principaux : **éoliennes terrestres** et **éoliennes offshore**, chacun présentant des caractéristiques distinctes.

2.1.6.1. Éoliennes terrestres

Installées sur la terre ferme, les éoliennes terrestres sont moins coûteuses à installer et à entretenir, car elles bénéficient d'un accès facile aux infrastructures existantes. Cependant, elles sont exposées à des vents souvent moins réguliers et moins puissants, limitant parfois leur rendement énergétique. Leur proximité avec les zones habitées peut également poser des problèmes d'impact visuel et sonore.

2.1.6.2. Éoliennes offshores

Les éoliennes *offshores*, quant à elles, sont situées en mer, souvent loin des côtes, où elles profitent de vents plus constants et puissants. Cela leur confère un rendement énergétique plus stable et élevé. En revanche, les coûts d'installation et de maintenance sont plus élevés, car elles nécessitent des infrastructures adaptées aux conditions marines exigeantes, comme la résistance à la corrosion et un accès complexe pour l'entretien.

Tableau 1
Résumé comparatif des caractéristiques des éoliennes terrestres et *offshore*

Critère	**Éoliennes terrestres**	**Éoliennes *offshore***
Coûts d'installation	Moindres	Élevés
Maintenance	Plus simple	Plus complexe
Rendement énergétique	Variable	Plus stable et élevé
Impact environnemental	Visuel et sonore	Moindre

2.2 Surveillance des pannes et maintenance des éoliennes

2.2.1 Introduction au système SCADA dans les parcs éoliens

SCADA est un système de collecte et d'acquisition de données conçu pour recueillir en temps réel les informations nécessaires à la surveillance des éoliennes [1]. Grâce à un réseau de capteurs installés sur les différents composants de la turbine, SCADA collecte des données variées et cruciales pour la gestion opérationnelle et la maintenance conditionnelle. Ces données sont classées en deux catégories principales : les signaux discrets et les signaux continus [11].

Les signaux discrets indiquent généralement des états binaires, tels que les alarmes ou l'activation de certains dispositifs, tandis que les signaux continus mesurent des variables physiques comme la température, la vitesse de rotation, ou la puissance.

2.2.2 Paramètres SCADA fondamentaux

Les données SCADA se répartissent en plusieurs catégories, comme indiqué dans le Tableau 2. Ces catégories reflètent les divers aspects surveillés pour assurer un fonctionnement optimal et prévenir les défaillances potentielles [12].

Tableau 2
Paramètres de base des éoliennes dans le système SCADA [5].

Environmental	Electrical characteristics	Part temperatures	Control variables
Wind speed	Active power output	Gearbox bearing	Pitch angle
Wind direction	Power factor	Gearbox lubricant oil	Yaw angle
Ambient temperature	Reactive power	Generator winding	Rotor shaft speed
Nacelle temperature	Generator voltages	Generator bearing	Generator speed
	Generator phase current	Main bearing	Fan speed / status
	Voltage frequency	Rotor shaft	Cooling pump status
		Generator shaft	Number of yaw movements
		Generator slip ring	Set pitch angle / deviation
		Inverter phase	Number of starts / stops
		Converter cooling water	Operational status code
		Transformer phase	
		Hub controller	
		Top controller	
		Converter controller	
		Grid busbar	

- **Paramètres environnementaux** : Ils incluent la vitesse du vent, la direction du vent, la température ambiante et la température de la nacelle. Ces données sont essentielles pour comprendre l'influence des conditions météorologiques sur le fonctionnement de la turbine et ajuster son orientation et son angle de pales en conséquence [7].
- **Caractéristiques électriques** : Ces paramètres mesurent la performance électrique de la turbine, notamment la puissance active produite, le facteur de puissance, la puissance réactive, ainsi que les tensions et courants du générateur. Ces mesures permettent de détecter les anomalies de production et d'assurer la conformité des performances électriques de la turbine.
- **Températures des composants** : La surveillance des températures des roulements, de l'huile de lubrification, des enroulements et d'autres composants critiques est cruciale pour anticiper les défaillances dues à la surchauffe ou à l'usure.
- **Variables de contrôle** : Ces paramètres incluent l'angle de pas des pales, l'angle de lacet, la vitesse de l'arbre du rotor et le statut de différents sous-systèmes (pompes, ventilateurs,

etc.). Ils permettent de gérer et de contrôler la turbine en fonction des variations des conditions de vent et de charge.

2.2.3 Défis liés à l'utilisation des données SCADA

Bien que les données SCADA soient un atout précieux pour la surveillance de l'état des éoliennes, leur utilisation dans les modèles de maintenance conditionnelle présente plusieurs défis :

- **Faible résolution temporelle** : Les données SCADA sont typiquement collectées toutes les 10 minutes. Bien que cette résolution réduise la charge de stockage et de transmission des données, elle limite l'application de techniques avancées de surveillance, telles que l'analyse spectrale, qui nécessitent des données à haute fréquence pour capturer les changements rapides dans les signaux de vibration ou d'autres paramètres dynamiques.
- **Non-stationnarité des variables** : En raison des conditions de fonctionnement dynamiques et changeantes des éoliennes, de nombreuses variables SCADA, comme la température et la puissance, présentent une nature non stationnaire. Ces variations constantes rendent difficile l'extraction de caractéristiques stables et la construction de modèles fiables de comportement normal. Cette non-stationnarité complique également la détection des anomalies [5].
- **Confidentialité et manque de données historiques** : Les contraintes de confidentialité et l'accès limité aux données historiques de pannes (comme les exemples de défaillances, les enregistrements et les étiquettes) constituent un autre obstacle. Le manque de données de défaillances rend difficile la création de modèles prédictifs précis et robustes pour détecter les pannes potentielles, limitant ainsi l'efficacité de la maintenance prédictive.

2.2.4 Potentiel et avenir des données SCADA

Malgré ces défis, les données SCADA représentent une opportunité prometteuse pour la maintenance des éoliennes. Elles permettent de surveiller en continu l'état des composants critiques et d'identifier des anomalies avant qu'elles n'entraînent des pannes majeures. Des techniques d'apprentissage automatique, telles que les réseaux de neurones et les modèles de régression multivariée, sont de plus en plus intégrées pour l'analyse des données. Ces approches peuvent exploiter les variations non-stationnaires pour établir des modèles prédictifs capables de signaler les défaillances imminentes. En combinant les systèmes conventionnels avec des analyses avancées, les opérateurs de parcs éoliens peuvent améliorer la fiabilité des turbines et réduire les coûts associés à la maintenance et aux temps d'arrêt imprévus.

2.2.5 Définitions

En termes pertinents pour la surveillance des conditions, Isermann [13] a défini les termes suivants :

- **Défaillance** : Une déviation non autorisée d'au moins une caractéristique propre (fonction) du système par rapport à la condition normale acceptable et habituelle
- **Panne** : Une interruption permanente de la capacité d'un système à effectuer une fonction requise dans des conditions d'exploitation spécifiées.
- **Dysfonctionnement** : Une irrégularité intermittente dans la réalisation de la fonction souhaitée d'un système.

Plusieurs études récentes ont évalué les taux de défaillance et les temps d'arrêt associés aux éoliennes. Les systèmes de commande électrique et électronique ont été identifiés comme étant les plus sujets aux pannes, mais les pannes de boîte de vitesses et de générateur ont causé les temps d'arrêt les plus longs [14]. Le tableau 3 présente les principaux types de défaillance retrouvées dans ces turbines.

Tableau 3
Types de défauts dans les turbines des éoliennes [14].

Blade-Related Faults: 1. Blade Misalignment 2. Blade Erosion and Wear	Generator Faults: 1. Electrical Winding Failures 2. Power Output Irregularities
Gearbox Anomalies: 1. Gear Tooth Wear 2. Bearing Failures	Pitch Mechanism Irregularities: 1. Pitch Actuator Malfunctions 2. Blade Angle Inconsistencies
Other Faults: 1. Icing and Cold Weather 2. Lightning Strikes & Harsh Wind Conditions	Yaw System Deviations: 1. Yaw Drive Failures 2. Yaw Control Inaccuracies

2.2.6 Techniques de détection des défauts

La détection des défauts dans les éoliennes repose sur plusieurs techniques essentielles, chacune visant à anticiper les pannes et à optimiser la maintenance. Parmi ces techniques, l'analyse des vibrations est largement utilisée, notamment grâce à la transformée de Fourier

rapide (FFT), qui convertit les signaux temporels en spectres de fréquence, permettant ainsi de détecter les variations anormales de certaines fréquences liées aux défauts [15].

La surveillance de la santé structurelle (SHM) est une autre approche cruciale, s'appuyant sur des mesures de déformations [13], de vibrations, et parfois sur des technologies avancées comme la thermographie et les ultrasons pour identifier les défaillances structurelles dans des composants tels que les pales et la tour.

De plus, l'analyse de la signature électrique surveille les variations de certains signaux électriques, indiquant de possibles défaillances dans les systèmes électriques, comme le générateur et le convertisseur.

Enfin, les techniques d'apprentissage automatique jouent un rôle de plus en plus important dans la maintenance prédictive [13]. En particulier, les réseaux de neurones et les machines à vecteurs de support (SVM) permettent de classer et de prédire les défauts à partir de grandes quantités de données. Ces méthodes, en analysant les données de vibration et autres indicateurs, offrent une précision accrue dans la détection précoce des anomalies et contribuent à une maintenance plus proactive des éoliennes.

2.2.7 Les différents types de maintenance

La maintenance est définie comme l'ensemble des techniques et actions mises en œuvre afin de maintenir ou de rétablir un équipement dans un état lui permettant d'assurer son bon fonctionnement [16]. Elle joue un rôle essentiel dans la gestion des infrastructures, en permettant d'optimiser la durée de vie des équipements et de réduire les coûts liés aux interruptions de service et aux réparations non planifiées [17].

Différentes stratégies de maintenance sont appliquées, leur choix dépendant de plusieurs facteurs tels que le type d'industrie, la criticité des équipements et les contraintes budgétaires [18]. Voici quelques-unes des stratégies de maintenance les plus couramment adoptées et largement reconnues pour leur efficacité dans divers secteurs industriels:

2.2.7.1. Maintenance Corrective

La maintenance corrective, également appelée maintenance réactive, consiste à intervenir sur un équipement uniquement lorsqu'une panne survient. Cette approche est historiquement la plus utilisée, en raison de sa simplicité et de l'absence de coûts liés à des interventions préventives [19]. Toutefois, elle présente de nombreux inconvénients, notamment des arrêts de production

imprévus, des coûts élevés de réparation et une dégradation accélérée des équipements en raison du manque d'entretien régulier [6].

2.2.7.2. Maintenance Préventive

Contrairement à la maintenance corrective, la maintenance préventive repose sur un entretien régulier et planifié des équipements, indépendamment de leur état réel de fonctionnement [20]. Cette approche vise à réduire la probabilité de défaillance en effectuant des inspections périodiques, des remplacements de pièces et d'autres actions d'entretien programmées [18].

L'un des principaux avantages de la maintenance préventive est qu'elle permet de limiter les arrêts imprévus et d'améliorer la fiabilité des équipements [21]. Toutefois, elle peut également engendrer des coûts inutiles si des interventions sont réalisées sur des équipements encore en bon état, ce qui constitue un défi pour les entreprises cherchant à optimiser leurs ressources.

2.2.7.3. Maintenance Prédictive

La maintenance prédictive représente une évolution majeure par rapport aux approches précédentes, en exploitant des technologies avancées pour surveiller en temps réel l'état des équipements et anticiper les pannes avant qu'elles ne surviennent [16]. Contrairement à la maintenance préventive, qui repose sur un calendrier fixe, la maintenance prédictive s'appuie sur des capteurs intelligents et des algorithmes de Machine Learning pour analyser les données de fonctionnement et identifier les anomalies pouvant indiquer une future défaillance [6].

L'un des principaux atouts de cette approche est sa capacité à optimiser l'allocation des ressources en limitant les interventions inutiles et en ciblant précisément les équipements nécessitant une maintenance [22].

Les progrès réalisés dans le domaine de l'intelligence artificielle et du Big Data ont considérablement amélioré l'efficacité des modèles prédictifs, permettant une détection plus précise des pannes potentielles [18].

Le tableau 4 résume les avantages et inconvénients de ces différents types de maintenance.

Tableau 4
Comparaison des différents types de maintenance.

Stratégie de maintenance	Principe	Avantages	Inconvénients
Corrective	Intervention après panne	Pas de coûts préventifs, simple à mettre en œuvre	Coûts élevés, interruptions imprévues, impact sur la productivité
Préventive	Entretien régulier planifié	Réduction des pannes, amélioration de la fiabilité	Risque d'interventions inutiles, coûts de maintenance élevés
Prédictive	Surveillance continue et analyse des données	Optimisation des coûts, réduction des temps d'arrêt, intervention ciblée	Investissement initial important, besoin d'une infrastructure IoT et IA

CHAPITRE 3. CADRE THÉORIQUE

3.1 Méthodes Basées sur l'Intelligence Artificielle et Prédiction des Défauts

L'intelligence artificielle (IA) s'impose de plus en plus comme un élément clé dans divers secteurs industriels, y compris la maintenance prédictive, en dépassant le cadre des laboratoires de recherche pour être intégrée aux processus de production et de gestion des infrastructures [23]. Grâce à ses avancées rapides, l'IA facilite l'identification des défaillances potentielles, optimise la gestion des opérations et améliore l'efficacité des interventions en exploitant de vastes quantités de données issues de capteurs connectés.

L'IA est définie comme un ensemble de technologies informatiques capables de simuler des fonctions cognitives humaines, telles que la prise de décision, la classification, l'interprétation et l'inférence, afin de résoudre des problèmes complexes. Elle repose sur une combinaison de techniques de traitement et d'analyse des données, notamment le Machine Learning (ML) et le Deep Learning, permettant d'identifier des modèles et effectuer des prédictions.

3.1.1 Machine Learning (Apprentissage Automatique)

L'apprentissage automatique (ML), un sous-ensemble de l'intelligence artificielle, comprend une large gamme d'algorithmes adaptés à diverses tâches incluant notamment la régression, qui prédit des résultats continus, et la classification, où l'objectif est d'attribuer la classe correcte à un échantillon donné dans un système binaire ou multi-classes [12]. C'est essentiellement une méthode évoluée d'analyse des données qui permet d'automatiser la construction de modèles analytiques. Il repose sur le principe que les systèmes peuvent apprendre à partir de données, identifier des modèles et prendre des décisions avec un minimum d'intervention humaine [20]. Il existe quatre principaux types d'apprentissage automatique : l'apprentissage supervisé, non supervisé, semi-supervisé et par renforcement [15].

3.1.1.1. L'Apprentissage Supervisé

C'est une technique du ML qui vise à identifier la meilleure fonction ou modèle reliant les variables d'entrée aux variables de sortie étiquetées [23]. En d'autres termes, cette méthode repose sur un processus d'apprentissage guidé, où un algorithme est entraîné à partir d'un ensemble de données préalablement annotées, permettant ainsi d'optimiser sa capacité à généraliser et à faire des prédictions sur de nouvelles données. La figure 5 présente le processus de l'apprentissage supervisé [7, 24].

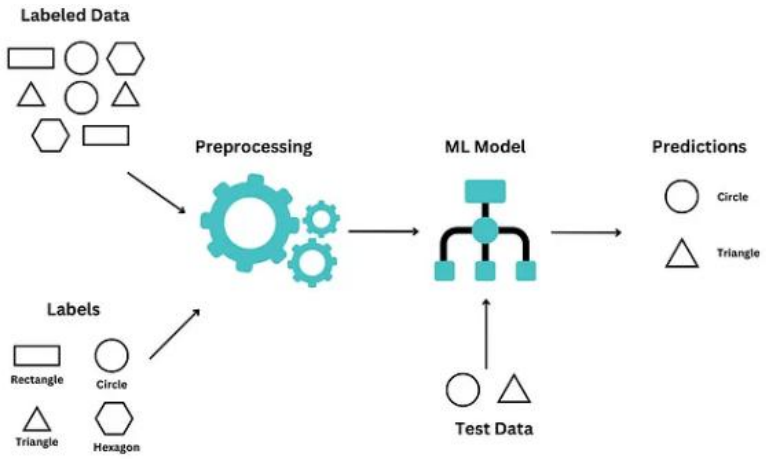


Figure 5
Schéma d'un apprentissage supervisé [25].

L'apprentissage supervisé permet d'effectuer deux types de tâches principales :

- La classification, dont l'objectif est d'assigner une catégorie à chaque observation en fonction de ses caractéristiques. Par exemple, dans un contexte industriel, un modèle peut classer l'état d'un composant comme "fonctionnel" ou "défectueux", ou bien "à surveiller" ou "normal", en se basant sur les données opérationnelles issues de capteurs SCADA. D'autres applications incluent la classification de cycles de fonctionnement comme "stable", "instable" ou "en arrêt programmé" dans les systèmes de production ou les turbines éoliennes [12].
- La régression, qui consiste à prédire des valeurs continues à partir de variables d'entrée. Dans un contexte industriel, cela peut inclure la prévision de la température du roulement d'une turbine, de la vibration d'un arbre moteur, ou encore de la puissance générée par une éolienne en fonction des conditions environnementales (vent, humidité, pression) mesurées par le système SCADA [26].

Les ensembles de données utilisés pour l'apprentissage supervisé sont divisés en deux parties distinctes :

- L'ensemble d'entraînement, qui sert à construire le modèle en ajustant les paramètres et en optimisant les poids des connexions neuronales si l'on utilise un réseau de neurones.
- L'ensemble de test, utilisé pour évaluer la capacité généralisante du modèle et mesurer son efficacité en le confrontant à des données inédites.

La création d'ensembles de données étiquetées est cependant un processus fastidieux, nécessitant une annotation manuelle des exemples et une compréhension claire des relations entre les données d'entrée et de sortie. Cette contrainte peut être un frein à l'utilisation massive de l'apprentissage supervisé, bien que cette approche offre un taux de réussite élevé avec une marge d'erreur minimale, à condition que l'étiquetage soit précis.

3.1.1.2. L'Apprentissage Non Supervisé

L'apprentissage non supervisé est une approche du Machine Learning qui repose sur l'analyse de données non étiquetées, c'est-à-dire sans correspondance explicite entre les entrées et les sorties. Contrairement à l'apprentissage supervisé, où un modèle apprend à partir d'exemples annotés par des experts, l'apprentissage non supervisé fonctionne sans indications précises sur ce que le modèle doit rechercher [23]. L'un des principaux désavantages de l'apprentissage non-supervisé est la difficulté d'interprétation des classes que le modèle définit. Il est particulièrement utile pour détecter des structures cachées, regrouper des données similaires et identifier des anomalies. La figure 6 présente le processus de l'apprentissage non-supervisé.

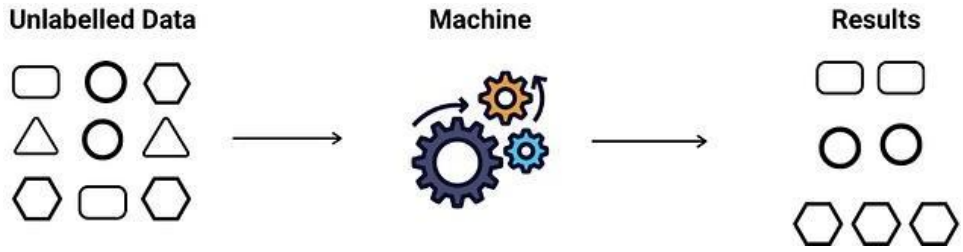


Figure 6
Schéma d'un apprentissage non-supervisé [25].

L'apprentissage non-supervisé regroupe plusieurs techniques analytiques avancées permettant d'exploiter efficacement de vastes ensembles de données. Parmi les méthodes les plus courantes figurent :

A. Le clustering (classification non supervisée) : Cette approche consiste à regrouper les données en clusters (groupes) partageant des caractéristiques similaires. Les algorithmes de clustering les plus utilisés incluent:
   - K-Means : Segmente les données en un nombre défini de groupes basés sur leur proximité dans l'espace des variables [27].

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) : Utilisé pour détecter des anomalies et des comportements inhabituels, notamment dans les données issues de capteurs intelligents [20].

B. La détection d'anomalies : L'apprentissage non supervisé est souvent employé pour identifier des comportements inhabituels dans les séries temporelles et les capteurs industriels. Les modèles comme l'autoencodeur et l'analyse en composantes indépendantes (PCA) permettent d'extraire des motifs cachés dans les données de surveillance des équipements [28].

3.1.2 Ensemble Learning

L'apprentissage par ensemble (Ensemble Learning), est une approche avancée du Machine Learning qui vise à améliorer la robustesse et la précision des modèles en combinant plusieurs algorithmes d'apprentissage. Plutôt que de s'appuyer sur un modèle unique, cette méthode exploite la diversité des prédictions fournies par plusieurs modèles pour obtenir un résultat global plus performant. Le principe fondamental repose sur le fait qu'un ensemble de modèles diversifiés permet de réduire les erreurs de prédiction et d'améliorer la généralisation des performances, en minimisant le risque de sur-apprentissage [20, 23, 29].

Parmi les principales techniques d'Ensemble Learning, on distingue le Bagging, le Boosting, le Voting et le Stacking, chacune ayant des caractéristiques spécifiques adaptées à différents types de problèmes. Une représentation schématique de chaque technique se trouve à la figure 7.

Le Bagging (Figure 7b) repose sur l'entraînement de plusieurs modèles indépendants sur des sous-échantillons du jeu de données, améliorant la robustesse des prédictions par rapport aux variations des données. L'un des algorithmes les plus représentatifs du Bagging est Random Forest, qui combine plusieurs arbres de décision pour produire une prédiction plus fiable et robuste. Cette approche est particulièrement efficace sur des jeux de données de grande taille où les erreurs aléatoires peuvent être atténuées grâce à l'agrégation des prédictions. Toutefois, bien que le Bagging permette de réduire le sur-apprentissage, il peut être coûteux en temps de calcul, notamment lorsque l'ensemble de modèles est volumineux et nécessite des ressources computationnelles élevées.

À l'inverse, le Boosting (Figure 7a) est une technique qui repose sur un apprentissage séquentiel, où chaque modèle est construit en corrigeant les erreurs du modèle précédent. Contrairement au Bagging, qui entraîne les modèles indépendamment, le Boosting attribue un poids plus important

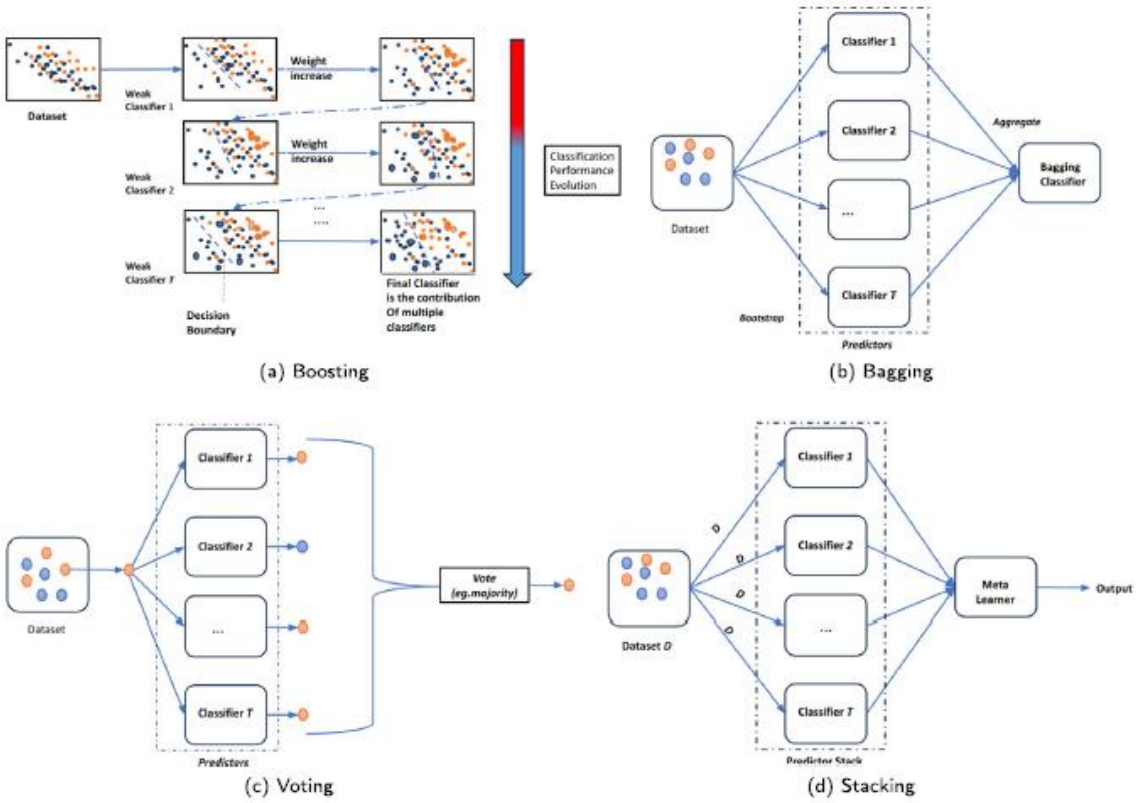


Figure 7
Représentations schématiques des principales techniques d'ensemble learning [30, 31].

aux exemples mal classés, permettant ainsi d'améliorer progressivement les performances du modèle global. Parmi les algorithmes les plus performants du Boosting, on retrouve XGBoost (Extreme Gradient Boosting), LightGBM (Light Gradient Boosting Machine) et CatBoost (Categorical Boosting), chacun étant optimisé pour des types de données spécifiques et des volumes massifs d'informations. Ces techniques sont particulièrement adaptées aux problèmes où la précision des prédictions est essentielle, notamment dans les systèmes de maintenance prédictive où la détection précoce des anomalies est un enjeu majeur. Cependant, l'inconvénient majeur du Boosting réside dans le risque de sur-apprentissage, car les modèles successifs peuvent devenir trop spécialisés sur l'ensemble d'entraînement, ce qui réduit leur capacité de généralisation sur de nouvelles données [7].

Le Voting (Figure 7c), quant à lui, consiste à entraîner plusieurs modèles en parallèle, puis à combiner leurs prédictions à l'aide d'une règle d'agrégation simple, telle que le vote majoritaire pour la classification, ou la moyenne arithmétique pour la régression. Contrairement au Boosting ou au Stacking, le Voting ne cherche pas à corriger les erreurs précédentes ni à apprendre une

pondération optimale. Il est néanmoins apprécié pour sa simplicité, sa facilité de mise en œuvre et sa bonne robustesse lorsqu'il est appliqué à des modèles suffisamment diversifiés.

Enfin, le Stacking (Figure 7d), ou empilement de modèles, constitue une approche plus avancée qui combine les prédictions de plusieurs algorithmes hétérogènes à l'aide d'un méta-modèle. Cette technique repose sur une architecture à plusieurs niveaux, où les modèles de base produisent des prédictions qui sont ensuite utilisées comme nouvelles caractéristiques pour entraîner un modèle final, souvent une régression logistique ou un réseau de neurones. Le Stacking permet ainsi d'exploiter les forces de chaque modèle en pondérant leurs contributions respectives pour obtenir une décision plus robuste. Toutefois, l'un des défis du Stacking est de trouver la bonne combinaison de modèles, car une mauvaise sélection peut entraîner une augmentation du bruit dans les prédictions et une complexité accrue des calculs [29, 31-33].

Dans cette étude, plusieurs algorithmes supervisés sont combinés pour former le modèle de base du Stacking. Les algorithmes choisis sont : Random Forest, XGBoost, LightGBM et CatBoost. Chacun de ces modèles est sélectionné pour ses caractéristiques spécifiques, permettant d'améliorer la robustesse et la précision du modèle final.

3.1.2.1. Random Forest

L'algorithme Random Forest, proposé par Breiman [34], s'inscrit dans le cadre des méthodes d'apprentissage supervisé par ensemble, en combinant une multitude d'arbres de décision construits de manière indépendante pour produire une prédiction agrégée plus stable, robuste et généralisable. Ce modèle repose sur deux mécanismes clés qui introduisent une double forme d'aléa : d'une part, l'échantillonnage aléatoire avec remise (bootstrap sampling) appliqué aux observations ; d'autre part, la sélection aléatoire de variables explicatives à chaque nœud de l'arbre.

Dans un premier temps, chaque arbre est entraîné sur un sous-échantillon bootstrap du jeu de données d'apprentissage. Ce sous-échantillon est obtenu par tirage avec remise, ce qui signifie que certaines observations peuvent apparaître plusieurs fois, tandis que d'autres ne figurent pas dans l'échantillon. Ce mécanisme contribue à la réduction de la variance et permet d'atténuer le sur-apprentissage [35]. Statistiquement, environ 63 % des données originales sont utilisées pour entraîner l'arbre, et les 37 % restantes, appelées données out-of-bag (OOB), peuvent servir à estimer la performance du modèle sans recourir à une validation croisée explicite.

Dans un second temps, lors de la construction de chaque nœud d'un arbre, l'algorithme ne considère qu'un sous-ensemble aléatoire de variables (souvent de taille $\sqrt{p}$ pour la classification, où p est le nombre total de variables) pour déterminer la meilleure scission. Cette stratégie, appelée feature bagging, favorise la diversité structurelle entre les arbres de la forêt et diminue la corrélation entre leurs erreurs, ce qui permet de réduire efficacement la variance globale du modèle. À la différence d'un arbre isolé, potentiellement très instable, la Random Forest exploite la diversité de ses composants pour stabiliser la prédiction, sans augmenter significativement le biais. Le compromis biais–variance, central en apprentissage statistique est ainsi mieux maîtrisé.

Une fois la forêt d'arbres construite, les prédictions sont agrégées selon la nature de la tâche : en classification, on applique un vote majoritaire, où la classe prédite le plus fréquemment est retenue comme sortie du modèle ; en régression, on calcule la moyenne arithmétique des prédictions numériques émises par chaque arbre. Ce processus est illustré dans la figure 8 ci-dessous, qui schématise les principales étapes : les arbres (entraînés sur des échantillons bootstrap) reçoivent les mêmes données d'entrée, émettent chacun une prédiction individuelle (Result-1 à Result-N), puis leurs sorties sont combinées pour former une prédiction finale, plus fiable et moins sensible au bruit ou aux fluctuations des données d'apprentissage [3, 36-38].

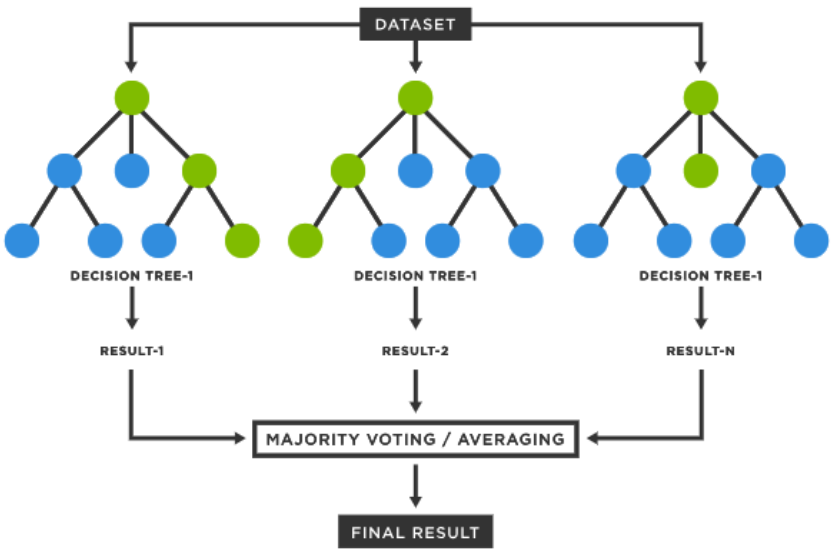


Figure 8
Représentation schématique du processus d'agrégation de l'algorithme Random Forest [39].

3.1.2.2. XGBoost

L'algorithme XGBoost (eXtreme Gradient Boosting) repose sur une construction séquentielle d'arbres de décision, où chaque nouvel arbre est entraîné à corriger les erreurs (résidus) de prédiction générée par les arbres précédents. Comme illustré dans la figure 9, le processus débute par un jeu de données divisé en échantillons, à partir desquels un premier arbre est construit pour générer une prédiction initiale. L'écart entre cette prédiction et la valeur réelle est ensuite modélisé sous forme de résidus, que les arbres suivants apprennent à corriger.

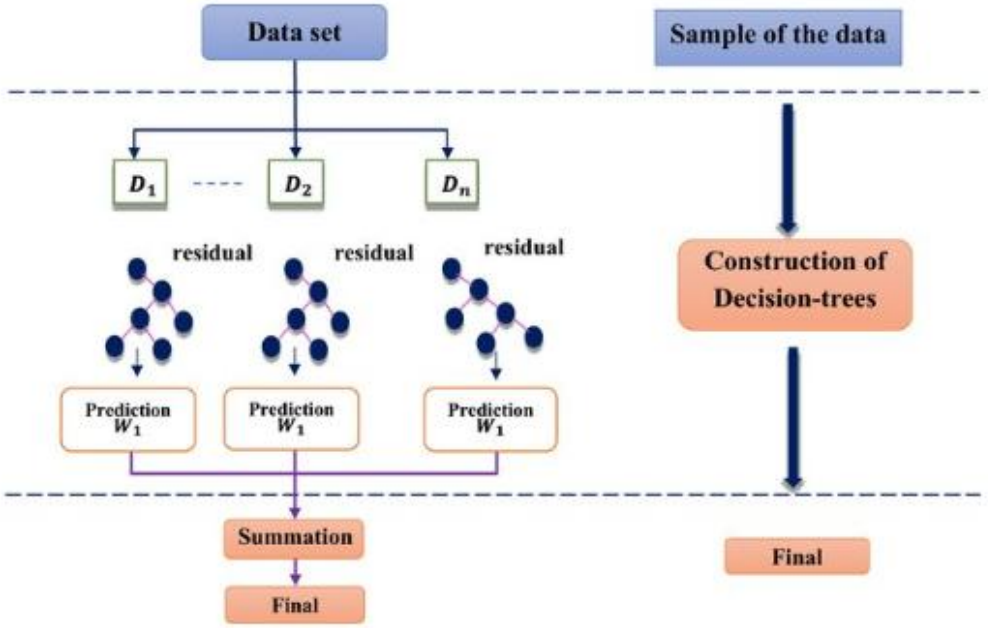


Figure 9
Schéma de fonctionnement de l'algorithme XGBoost [40].

Cette approche est encadrée par une fonction de perte, typiquement la log-loss dans les tâches de classification, qui mesure l'écart entre les prédictions et les vraies étiquettes, en attribuant une pénalité plus importante aux erreurs de classification mal calibrées. XGBoost se distingue par l'utilisation d'une approximation de second ordre de cette fonction de perte, via un développement de Taylor, exploitant à la fois les gradients (erreurs directes) et les hessiennes (information sur la courbure), permettant une optimisation plus fine à chaque itération.

À chaque étape, l'algorithme sélectionne les divisions des nœuds d'arbre qui maximisent le gain de réduction de la fonction de perte, tout en pénalisant les modèles trop complexes à l'aide d'un terme de régularisation. Ce dernier agit sur le nombre de feuilles et les poids attribués à celles-ci,

pour limiter le surajustement. Chaque nouvel arbre produit une prédiction intermédiaire, pondérée par un taux d'apprentissage, et la prédiction finale résulte de la somme pondérée de toutes les contributions des arbres. Grâce à cette combinaison de boosting séquentiel, optimisation précise de la log-loss, régularisation intégrée et agrégation contrôlée, XGBoost offre un modèle puissant, à la fois précis, robuste et particulièrement adapté à des données complexes et bruitées [41-45].

3.1.2.3. LightGBM

LightGBM (Light Gradient Boosting Machine) est un algorithme d'apprentissage supervisé de type ensemble, conçu pour optimiser à la fois la vitesse d'entraînement, l'usage mémoire et la précision prédictive. Comme l'illustre la figure 10 ci-dessous, le principe fondamental de LightGBM repose sur la combinaison de plusieurs arbres de décision faibles (appelés aussi arbres de régression ou de classification) pour construire un modèle robuste. Chaque arbre reçoit en entrée un vecteur de caractéristiques issu d'un échantillon du jeu de données. À chaque itération, un nouvel arbre est entraîné pour corriger les erreurs des prédictions des arbres précédents, en minimisant une fonction de perte. Ce processus se poursuit jusqu'à ce que le modèle atteigne un niveau de performance prédéfini. À la fin, les prédictions des différents arbres sont combinées, souvent à l'aide d'un vote majoritaire (dans le cas de la classification, comme indiqué par «Majority-Voting» dans le schéma), afin de produire la classe finale.

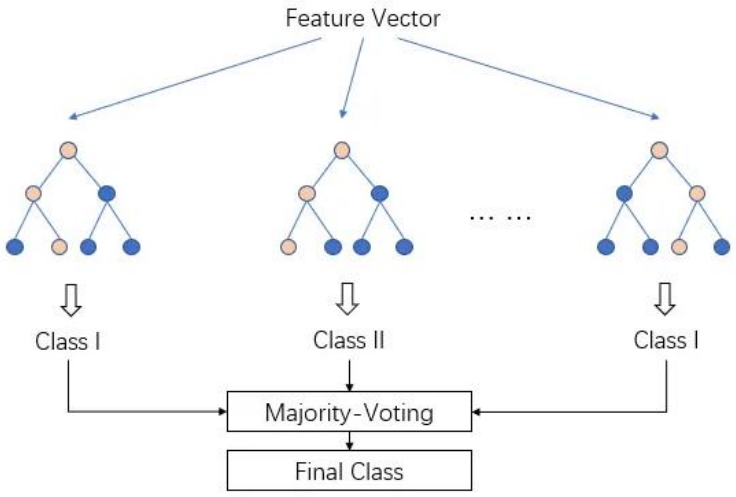


Figure 10
Schéma de fonctionnement de l'algorithme LightGBM [46].

Contrairement aux approches traditionnelles de Gradient Boosting qui développent les arbres niveau par niveau (depth-wise), LightGBM adopte une stratégie leaf-wise, c'est-à-dire qu'il fait

croître les arbres à partir des feuilles les plus prometteuses, c'est-à-dire celles qui offrent le gain de perte le plus élevé. Cette stratégie permet au modèle d'atteindre une meilleure précision, au prix toutefois d'un risque accru de surapprentissage si elle n'est pas régularisée. LightGBM introduit également deux optimisations fondamentales pour accélérer l'entraînement sur de grands ensembles de données. La première, GOSS (Gradient-based One-Side Sampling), consiste à ne sélectionner que les exemples ayant les plus grands gradients (c'est-à-dire ceux sur lesquels le modèle fait le plus d'erreurs) tout en en échantillonnant aléatoirement un petit pourcentage des autres. Cela permet de réduire la taille du sous-ensemble utilisé pour l'apprentissage sans perdre d'information importante. La deuxième, EFB (Exclusive Feature Bundling), regroupe les caractéristiques qui ne sont jamais non nulles simultanément, réduisant ainsi la dimensionnalité effective du jeu de données.

Par ailleurs, LightGBM repose sur un algorithme de type histogramme pour accélérer la recherche de seuils de division. Plutôt que d'évaluer toutes les valeurs possibles d'une caractéristique, les valeurs sont d'abord discrétisées en bacs (bins), ce qui réduit considérablement le coût de calcul. En classification, l'arbre produit à chaque nœud une prédiction de classe (comme « Class I » ou « Class II » dans l'image), et l'ensemble du modèle final repose sur la décision issue de la majorité des arbres (majority voting). Ce mécanisme ensembliste permet de mieux gérer les données bruitées, les interactions non linéaires et les structures complexes, tout en gardant une forte capacité de généralisation. Grâce à ces propriétés, LightGBM est devenu un outil incontournable pour les tâches de classification ou de régression dans des contextes industriels exigeants, notamment en maintenance prédictive, détection d'anomalies et systèmes intelligents [46-51].

3.1.2.4. Logistic regression

La régression logistique [26, 52-56] est une méthode d'apprentissage supervisé utilisée pour résoudre des problèmes de classification binaire ou multiclasse. Contrairement à la régression linéaire qui prédit une valeur continue, la régression logistique modélise la probabilité qu'un échantillon appartienne à une classe donnée. Son fonctionnement repose sur l'application d'une fonction logistique sigmoïde à une combinaison linéaire pondérée des variables d'entrée. Cette fonction transforme toute valeur réelle en une probabilité comprise entre 0 et 1. Le modèle apprend les poids associés à chaque variable en maximisant la vraisemblance des observations dans l'ensemble d'entraînement, en utilisant généralement un algorithme d'optimisation comme la descente de gradient.

Lors de la phase d'entraînement, le modèle ajuste ses paramètres pour minimiser une fonction de perte logistique, qui pénalise fortement les prédictions éloignées des valeurs cibles. Une fois entraîné, le modèle peut prédire la probabilité d'appartenance d'un nouvel échantillon à une classe, et cette probabilité est ensuite comparée à un seuil (souvent fixé à 0,5) pour attribuer une classe. La régression logistique est également appréciée pour sa simplicité, son efficacité sur des jeux de données linéairement séparables, et sa capacité à fournir une interprétation directe de l'importance des variables via les coefficients estimés. Elle est largement utilisée comme référence dans de nombreux systèmes intelligents, notamment dans les phases de détection préliminaire ou de classification linéaire dans des architectures plus complexes [26, 52-56].

3.1.2.5. CatBoost

CatBoost est un algorithme d'apprentissage supervisé basé sur le principe du gradient boosting, conçu pour améliorer la précision, la robustesse et la gestion des variables catégorielles dans les jeux de données tabulaires. Le processus commence par un ensemble de données contenant n échantillons et m caractéristiques, comme illustré dans la figure 11 ci-dessous. L'algorithme génère plusieurs sous-ensembles d'entraînement à l'aide de la technique de bootstrap (échantillonnage aléatoire avec remise). Sur chaque sous-échantillon, un arbre de décision est construit séquentiellement, et non en parallèle, chaque nouvel arbre visant à corriger les erreurs (résidus) commises par l'ensemble des arbres précédents. À chaque itération, les exemples mal prédits reçoivent un poids plus élevé, ce qui oriente la construction des arbres suivants vers les cas les plus difficiles. Ce processus est appelé expansion de poids et constitue le cœur de la dynamique adaptative du boosting.

CatBoost introduit deux innovations majeures qui le distinguent des autres algorithmes de boosting comme XGBoost ou LightGBM. Premièrement, il applique un encodage statistique régularisé des variables catégorielles, appelé Target-Based Statistics, qui permet de convertir les catégories en valeurs numériques tout en évitant la fuite d'information (le "target leakage") en utilisant uniquement des informations disponibles avant l'observation courante. Deuxièmement, il implémente le boosting ordonné (Ordered Boosting), une stratégie qui prévient le décalage de prédiction (prediction shift) causé par l'utilisation des vraies cibles pour calculer les gradients dans les premiers arbres. Cela se fait en construisant un ordre aléatoire des observations et en garantissant que chaque prédiction est faite à partir d'un modèle entraîné uniquement sur des données antérieures.

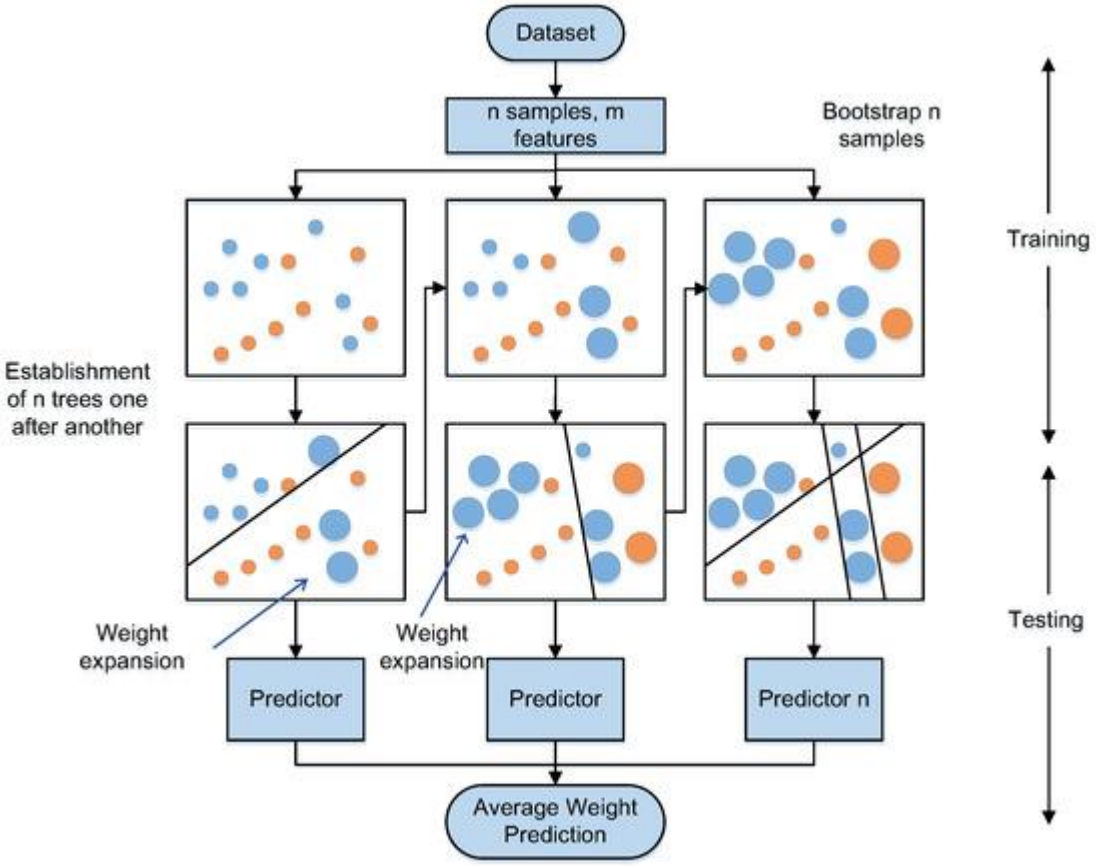


Figure 11
Schéma de fonctionnement de l'algorithme CatBoost.

Enfin, après avoir entraîné l'ensemble des arbres, CatBoost effectue une agrégation pondérée des prédictions issues de chaque arbre, comme montré en bas du schéma. Ce mécanisme permet d'obtenir une prédiction finale plus stable, en minimisant le risque de surapprentissage. Grâce à ces mécanismes intégrés, CatBoost offre une grande précision prédictive tout en assurant une bonne généralisation, en particulier sur les données hétérogènes et non linéaires rencontrées en pratique industrielle ou environnementale [26, 57-60]..

3.1.3 Le Méta-Modèle en Apprentissage Automatique

Contrairement aux méthodes conventionnelles où un modèle unique est utilisé, un méta-modèle permet d'exploiter la complémentarité de plusieurs algorithmes afin de corriger leurs erreurs respectives et renforcer la fiabilité des prévisions [61]. Un méta-modèle est une structure en plusieurs niveaux qui permet d'agréger les prédictions de plusieurs modèles de base pour améliorer la précision globale des prévisions.

Cette technique repose sur le principe de Stacking, où les résultats intermédiaires de plusieurs algorithmes sont réutilisés comme entrées pour un modèle de niveau supérieur chargé d'optimiser la décision finale [36]. L'architecture d'un méta-modèle se compose généralement de deux niveaux :

- Niveau de base : Plusieurs modèles sont entraînés indépendamment sur le même ensemble de données, en exploitant différentes méthodes telles que les arbres de décision, les algorithmes de boosting (XGBoost, LightGBM, CatBoost) ou encore les modèles neuronaux [62].
- Niveau méta : Un second modèle, souvent une régression logistique ou un réseau de neurones, apprend à pondérer et à combiner les prédictions des modèles de base afin de produire une sortie finale optimisée [61].

CHAPITRE 4. MÉTHODOLOGIE

4.1 Stratégie d'apprentissage et processus

Dans cette étude, l'apprentissage automatique est mobilisé pour la prévention et la gestion intelligente des défauts au sein d'un parc éolien composé de six turbines et situé au Royaume-Uni. Les données exploitées proviennent d'une source publique [63] et font l'objet d'un prétraitement rigoureux visant à garantir leur qualité, leur cohérence et leur intégration dans un cadre unifié. Cette étape consiste notamment à fusionner les données dynamiques provenant de sources différentes afin de les structurer pour l'apprentissage automatique.

Une fois les données préparées, elles sont injectées dans une architecture d'apprentissage en trois couches. La première couche assure une détection binaire des anomalies : elle identifie la présence d'une anomalie dans le fonctionnement des turbines. En l'absence d'anomalie, un mécanisme de détection des cas inconnus est déclenché afin de renforcer le système par apprentissage adaptatif. En cas d'anomalie détectée, les données sont transmises à la couche 2, qui effectue une classification multi-classes des défauts en tenant compte de leur nature, de leur gravité et de leur impact opérationnel.

Enfin, la couche 3 est dédiée à l'affinage des diagnostics dans les cas ambigus ou rares. Un module spécialisé est intégré pour subdiviser ces cas complexes en sous-groupes homogènes, permettant ainsi une détection plus précise. Cette couche fournit également un message de diagnostic explicite, accompagné d'une localisation précise du défaut détecté. Par ailleurs, une boucle de mise à jour continue est prévue pour permettre l'adaptation progressive des modèles aux nouvelles données collectées sur le terrain.

Le système développé est conçu pour être intégré à une interface intelligente en temps réel, permettant la visualisation interactive des alertes, des diagnostics, des localisations et des historiques de défauts. Cette approche modulaire et évolutive est illustrée dans la figure 12, qui résume les principales étapes du processus.

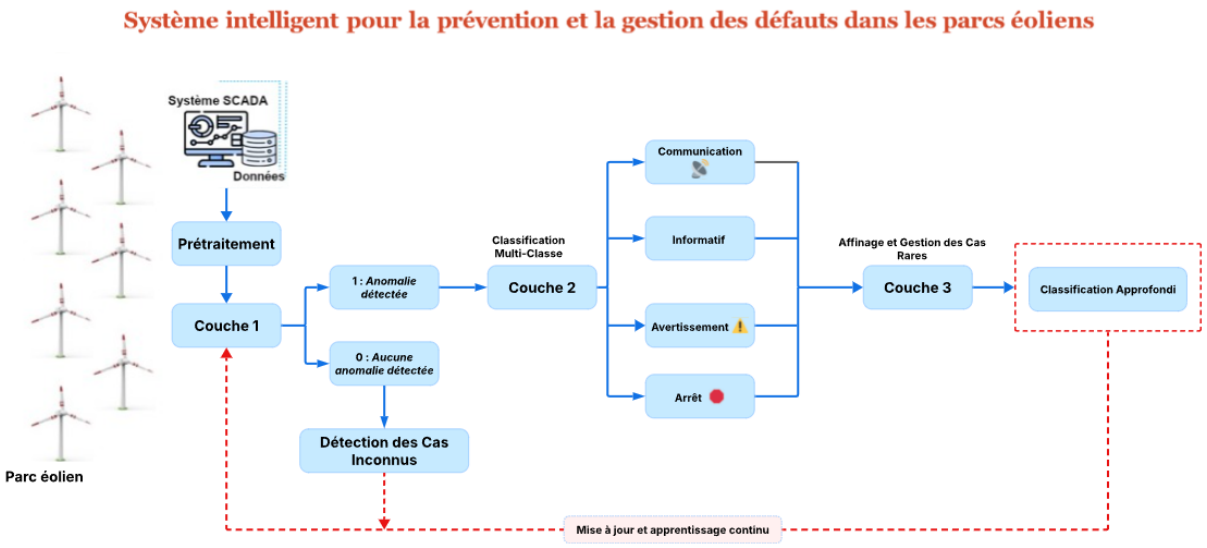


Figure 12
Système intelligent pour la prévention

4.1.1 Présentation et Prétraitement des données

Tel que décrit précédemment, les données utilisées pour l'apprentissage proviennent d'un parc éolien situé au Royaume-Uni, composé de six turbines. Issues du système SCADA (Supervisory Control and Data Acquisition), elles sont enregistrées à des intervalles réguliers de 10 minutes et se déclinent en deux ensembles principaux : les mesures continues et les événements techniques. Les mesures incluent des paramètres physiques tels que la vitesse du vent, la température des composants ou encore la puissance générée, tandis que les événements correspondent à des alertes ou erreurs signalées par le système de contrôle, englobant des messages de type avertissement, arrêt, communication ou information.

Afin d'exploiter efficacement ces données, un pipeline de traitement structuré a été mis en œuvre, tel qu'illustré à la Figure 13. Il débute par une étape de nettoyage visant à éliminer les doublons, corriger les valeurs aberrantes et gérer les données manquantes. Ensuite, un encodage des variables catégorielles a été appliqué aux événements par Label Encoding, permettant leur traitement par les modèles d'apprentissage automatique. Les deux ensembles de données sont ensuite fusionnés à l'aide d'un index chronologique commun, alignant chaque événement avec les mesures SCADA les plus proches dans le temps. Cette fusion temporelle assure une contextualisation précise des anomalies par rapport aux conditions de fonctionnement des turbines.

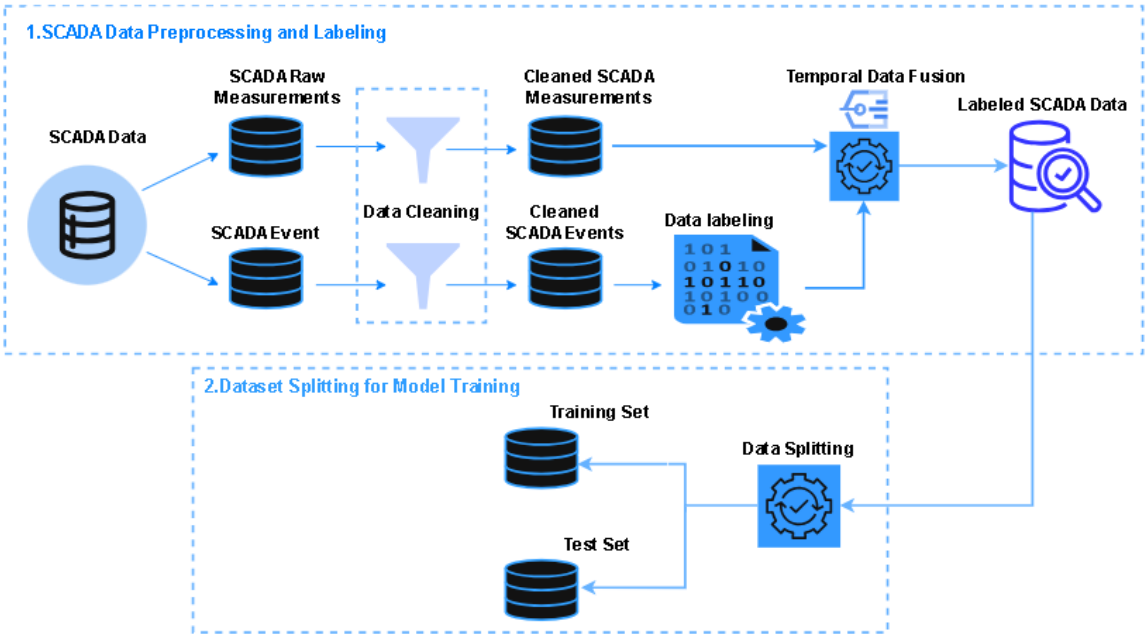


Figure 13
Processus de prétraitement des données.

Une phase de structuration avancée permet de raffiner les classes d'anomalies, en les segmentant selon leur nature et fréquence. Toutefois, certaines anomalies rares demeurent sous-représentées, induisant un fort déséquilibre de classes. Pour atténuer ce biais, une méthode de génération Monte Carlo a été utilisée afin de synthétiser des exemples supplémentaires tout en conservant les caractéristiques statistiques du jeu initial.

Enfin, l'ensemble des données traitées est divisé en deux sous-ensembles : un ensemble d'entraînement (70 %) utilisé pour l'apprentissage des modèles, et un ensemble de test (30 %) dédié à l'évaluation des performances. Ce processus de prétraitement, d'équilibrage et de partition des données est répété pour chaque couche de l'architecture hiérarchique, afin de garantir une cohérence dans l'apprentissage et l'évaluation à tous les niveaux du système. Ce dernier est ensuite intégré dans une interface SCADA temps réel, dédiée à la détection, la classification et la localisation intelligente des défauts.

4.2 Stratégie d'apprentissage et entraînement du modèle

Dans le cadre de cette méthodologie, chaque composante fonctionnelle du système (détection initiale, classification multi-classes, spécialisation des cas ambigus) est considérée comme un

module autonome d'apprentissage supervisé ou non supervisé. Chacun de ces modules suit rigoureusement un processus de développement standardisé, illustré dans la figure 14, incluant les étapes de séparation des données, d'entraînement, de test, d'évaluation, puis d'optimisation.

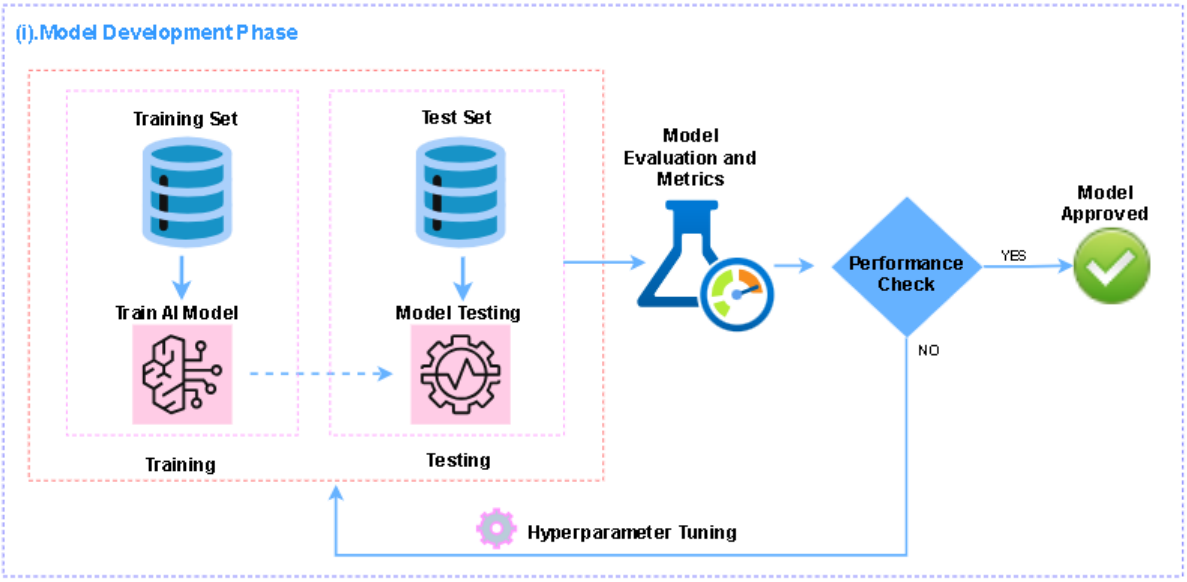


Figure 14
Processus de développement du système.

Plus précisément, pour chaque couche, les données SCADA sont divisées en sous-ensembles distincts (training set, test set) afin de garantir une évaluation impartiale des performances. Les modèles sont d'abord entraînés sur le jeu d'apprentissage à partir des caractéristiques spécifiques à chaque tâche (par exemple : variables brutes pour la détection binaire, étiquettes SCADA pour la classification, ou pour l'étape de spécialisation). Ensuite, les prédictions générées sur le jeu de test sont confrontées aux valeurs réelles et analysées à l'aide de métriques adaptées : précision, rappel, F1-score, AUC et log-loss, selon les objectifs de la couche.

Dans le cas où les performances ne répondent pas aux seuils de qualité prédéfinis, une phase de réglage des hyperparamètres est automatiquement déclenchée, visant à optimiser les modèles par validation croisée. Ce processus est itératif : il est répété jusqu'à atteindre un équilibre satisfaisant entre la sensibilité du système aux anomalies et la maîtrise des fausses alertes.

L'uniformisation de ce processus à travers toutes les couches garantit une cohérence méthodologique du pipeline, tout en assurant que chaque module soit optimisé localement selon ses objectifs spécifiques. Ainsi, l'intelligence du système ne repose pas uniquement sur

l'empilement de modèles, mais sur une stratégie d'apprentissage progressive, synchronisée et spécialisée, assurant une prise de décision fiable dans le contexte complexe des parcs éoliens.

4.2.1 Intégration et validation du système intelligent

À l'issue du développement individuel de chaque composant fonctionnel (détection initiale, classification multi-classe, traitement spécialisé), une phase structurée d'intégration est entreprise afin d'assurer la cohérence opérationnelle de l'ensemble du système. Comme illustré dans la Figure 15, chaque module est soumis à une validation indépendante selon un protocole uniforme, reposant sur des seuils de performance définis (F1-score, AUC, précision, rappel, etc.).

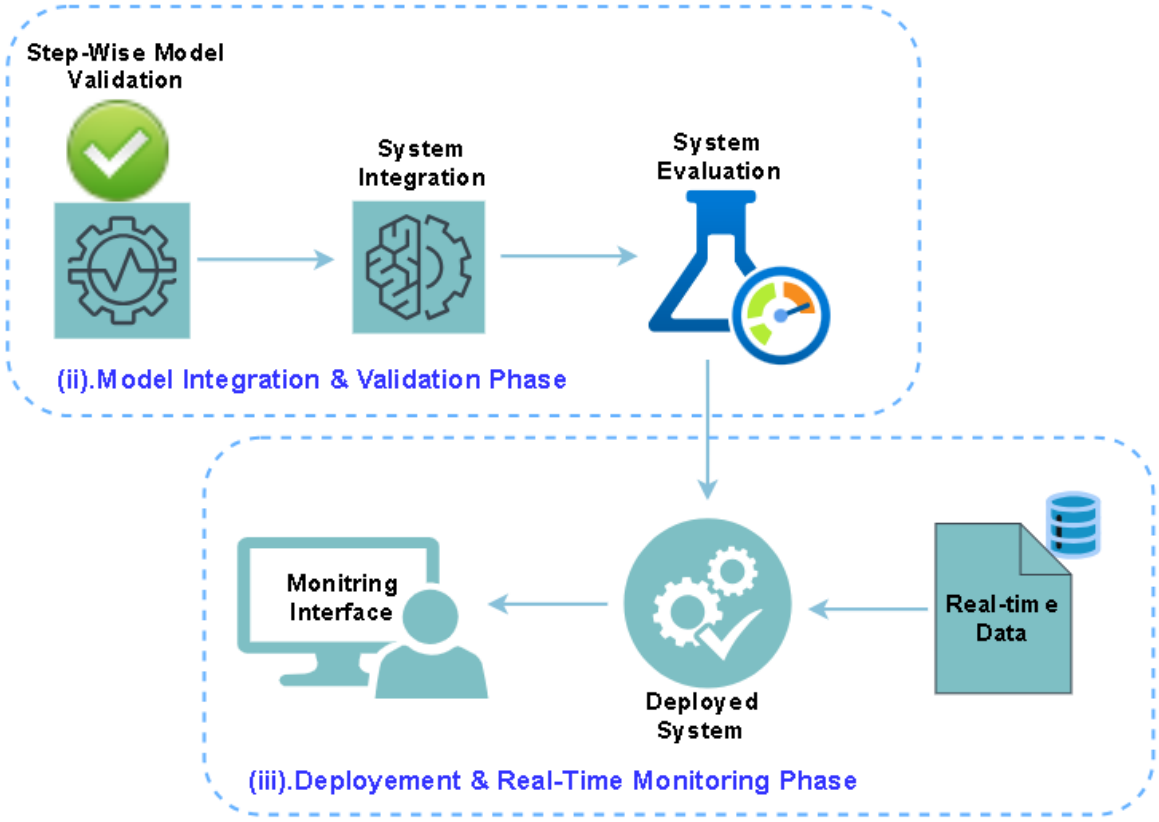


Figure 15
Schéma global du processus d'intégration, validation et déploiement du système.

Une fois validés individuellement, les modèles sont progressivement intégrés à une architecture globale hiérarchique, reproduisant le flux de décision envisagé. Cette intégration modulaire est suivie d'une évaluation sur un jeu de données transversal, simulant les enchaînements réels de

décisions entre les couches. Ce test global permet de vérifier l'empilement des modules, en assurant que le système fonctionne comme une entité cohérente et fiable.

4.2.2 Méthodologie en Trois Couches : Fonctionnement détaillé

Tel qu'exposé précédemment, la méthode proposée repose sur une architecture hiérarchique en trois étapes, chacune remplissant un rôle distinct et complémentaire dans le processus de détection, de classification et d'interprétation des anomalies SCADA. Ce découpage méthodologique permet une analyse progressive, structurée et adaptée à la complexité des données issues d'un parc éolien. Les sections suivantes présentent en détail le fonctionnement de chaque phase du système.

4.2.2.1. Couche 1 : Détection préliminaire des anomalies

La première étape du système agit comme un filtre intelligent, chargé d'effectuer une première distinction entre les comportements normaux et anormaux à partir des données SCADA. Cette détection repose sur une combinaison synergique d'approches supervisées et non supervisées, afin de garantir à la fois précision, adaptabilité et robustesse face à la diversité des scénarios opérationnels.

Le module supervisé s'appuie sur quatre classifieurs puissants (Random Forest, XGBoost, LightGBM et CatBoost) entraînés sur un ensemble de données étiquetées, structurés autour de la détection binaire (normal vs anomalie). Ces modèles sont intégrés au sein d'une architecture de type *Stacking*, où une régression logistique agit en tant que méta-modèle. Cette combinaison permet d'agréger les prédictions de manière pondérée, en exploitant les forces de chaque algorithme tout en compensant leurs biais respectifs.

Une fois cette première classification effectuée, deux flux de traitement sont déclenchés :

- Les observations classées comme anormales sont transmises à l'étape suivante pour une analyse multi-classe visant à identifier le type spécifique d'anomalie (Stop, Warning, etc.).
- Les observations jugées normales sont réexaminées par un module non supervisé, notamment basé sur l'algorithme Isolation Forest, afin de détecter d'éventuelles dérives, comportements inhabituels ou cas inédits non rencontrés pendant la phase d'entraînement. Cette vérification complémentaire permet de repérer les failles potentielles du modèle supervisé, en particulier dans les cas d'anomalies émergentes ou subtiles.

Ce double mécanisme assure une couverture maximale du spectre des anomalies, en combinant la puissance prédictive des modèles entraînés avec la capacité adaptative d'un détecteur non supervisé. Il constitue ainsi la première ligne de défense d'une architecture hiérarchique conçue pour une gestion intelligente, évolutive et contextualisée des défaillances.

4.2.2.2. Couche 2 : Classification multi-classe des anomalies

Les observations identifiées comme anormales lors de l'étape précédente sont ensuite classées selon leur gravité fonctionnelle. Cette phase permet de distinguer les types d'anomalies en quatre catégories :

- Informative, pour les événements sans impact immédiat,
- Avertissement (Warning), indiquant une surveillance nécessaire,
- Arrêt (Stop), représentant des pannes critiques,
- Communication, signalant des interruptions ou défaillances dans les échanges de données.

L'analyse statistique préalable des données a révélé un déséquilibre significatif entre ces catégories. La classe Informative est largement représentée, tandis que les anomalies de type Stop, Warning ou Communication sont beaucoup plus rares. Pour pallier ce déséquilibre et améliorer la capacité du modèle à reconnaître les anomalies critiques, une méthode de Monte Carlo a été utilisée pour générer des exemples synthétiques réalistes. Cette approche repose sur la reproduction probabiliste des distributions statistiques des entrées d'origine, garantissant la cohérence des données générées avec les scénarios réels observés.

Tout comme dans la première couche, la classification repose sur un méta-modèle basé sur le Stacking, où XGBoost, LightGBM, CatBoost et Random Forest sont combinés pour maximiser la performance du modèle. L'agrégation des prédictions est effectuée via une régression logistique, jouant le rôle de méta-classificateur afin d'améliorer la gestion des classes minoritaires et de réduire les erreurs de classification. Des ajustements dynamiques des seuils de décision complètent cette phase pour renforcer la robustesse du système dans un contexte de classification déséquilibrée.

4.2.2.3. Couche 3 : Affinage et validation des anomalies rares

Afin de traiter les cas ambigus ou à faible représentativité, cette dernière étape s'appuie sur une stratégie de spécialisation par sous-groupes au sein des principales catégories d'événements

SCADA. Sur les 83 messages d'alerte identifiés dans le jeu de données (voir tableau 5), les classes *Warning*, *Stop* et *Communication* sont nettement sous-représentées par rapport aux messages *Informational*, ce qui engendre un déséquilibre important. Ce débalancement limite la capacité des modèles globaux à identifier les anomalies critiques, souvent diluées dans des flux d'événements plus fréquents.

Tableau 5
Liste des messages d'alertes ainsi que le niveau de gravité correspondant.

Communication	Informational	Warning	Stop
Data communication unavailable	Absence of wind during run-up	4-20 mA vane 2	4-20mA conv. generator speed
	Automatic start-up	4-20mA anemometer 1	Battery test
	Brake program 170	4-20mA anemometer 2	Battery voltage axis 1
	Brake program 180	Battery charge cycle axis 1 error	Cable autounwind
	Brake program 190	Battery charge cycle axis 2 error	Deviation of wind direction>45°
	Brake program 200	Battery charge cycle axis 3 error	Deviation winddirection > 60°
	Brake program 50	Brake accumulator defect	Drive train monitor level 2
	Brake program 52	Brake pads worn	Error pitch converter 2
	Brake program 60	Breakdown obstacle light	Frequency converter error
	Brake program 75	Comm. failure FPM	Frequency converter not ready
	Bypass limit switches	Drivetrain oscillations	Grid loss
	Cable panel breaker open	Easy yaw	Hydraulic oil flushing operation
	Fallback setpoints active	Error brake resistor CHP	Icing (anemometer)
	Gear heating enabled	Heating/fan top box faulty	Icing (stop)
	Gearbox warm-up stage	High temp. gear bearing 1	Low hydraulic pressure
	Mains connection	High yaw motor current	Maximum grid frequency
	Mains operation	Icing (dev. electr. power)	Min. voltage cut in
	Mains run-up	Limit switch error 95° axis 1	Park master stop
	Open disc brake	Limit switch error 95° axis 2	Stop battery test
	Reset of icing st. cd. possible	Limit switch error 95° axis 3	Tower oscillation X level 1
	Run-up	Max. acceleration	Tower oscillation X level 2
	Semi-automatic operation	Overload generator fan 1	Tower oscillation Y level 1
	System OK	Overload generator fan 2	Tower oscillation Y level 2
	System test 1	Overload generator fan 3	Tower resonance
	System test 2	Overload generator heating	Uncontrolled yaw
	System test 3	Overload transformer fan outlet air	
	Wind < start wind	Parameter outside limits	
		Pitch batteries charging cycle	
		Pitch measuring system 1><2	
		Timeout brake closed	

Par ailleurs, il est difficile pour un classifieur unique d'apprendre efficacement à reconnaître un grand nombre de cas hétérogènes simultanément. Pour pallier cette limitation, une stratégie de sous-division a été mise en place : chaque grande classe a été divisée en sous-groupes homogènes, construits en fonction de la similarité des variables explicatives (features), identifiée à l'aide de la sélection de caractéristiques (cf. figure 16). Cette étape de segmentation vise à réduire la complexité intra-classe et à améliorer la capacité des modèles à détecter des signaux faibles ou spécifiques.

Chaque sous-groupe est ensuite pris en charge par un modèle dédié, choisi parmi les algorithmes suivants : LightGBM, XGBoost, Random Forest, Gradient Boosting, Régression logistique et

CatBoost. Ces modèles sont entraînés de manière indépendante pour maximiser leur performance locale, en tenant compte des dynamiques propres à chaque sous-ensemble.

Enfin, les décisions issues des différents modèles sont consolidées au sein d'un système de vote pondéré, dans lequel les prédictions sont fusionnées selon des poids déterminés par validation croisée. Cette stratégie permet non seulement d'améliorer la robustesse de la détection sur les cas complexes, mais aussi de garantir une meilleure précision globale, sans compromettre la sensibilité du système aux anomalies rares.

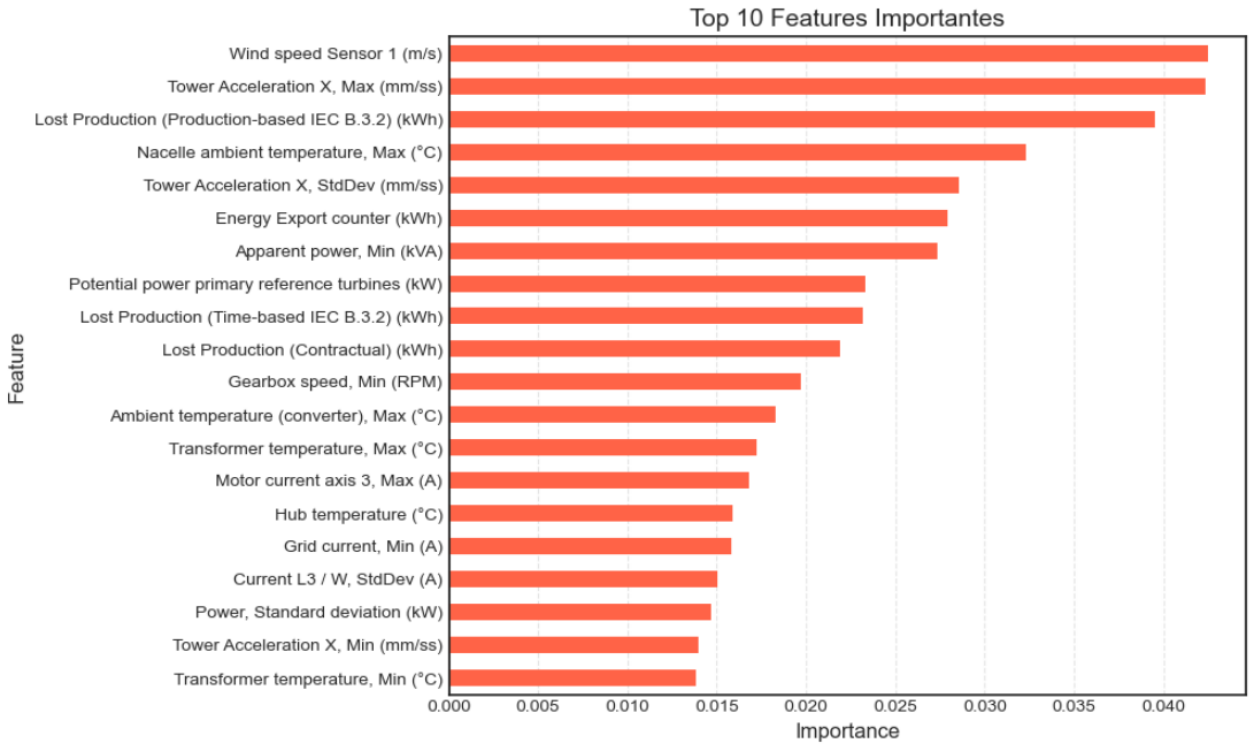


Figure 16
Caractéristiques les plus importantes pour la division des sous-groupes.

4.2.3 Évaluation des modèles

L'évaluation d'un modèle d'apprentissage automatique constitue une étape déterminante dans le processus de développement, car elle permet de mesurer sa capacité à produire des prédictions fiables face à des données inédites. Afin de garantir la pertinence de cette évaluation, une pratique courante consiste à réserver une fraction des données (typiquement 30 %) à un jeu de test indépendant. Cette séparation permet de simuler un contexte opérationnel réaliste, proche des conditions de déploiement en production.

L'analyse de performance repose sur un ensemble de métriques statistiques, largement accessibles via la bibliothèque *Scikit-learn*, notamment à travers son module *metrics*. Les indicateurs couramment utilisés en classification comprennent la précision, le rappel, le score F1, l'exactitude (accuracy), la log-loss, l'aire sous la courbe ROC (AUC-ROC), ainsi que la matrice de confusion, chacun jouant un rôle complémentaire dans l'évaluation du modèle. Avant d'en détailler les formules et interprétations, quelques éléments théoriques doivent être rappelés [20].

La matrice de confusion constitue un outil central dans l'évaluation des performances d'un classifieur. Elle distingue les prédictions correctes (vrais positifs, vrais négatifs) des erreurs (faux positifs, faux négatifs), offrant une vue synthétique des décisions du modèle. Cette structure permet d'identifier les biais de classification et d'orienter les ajustements [7].

Afin de consolider la robustesse de cette évaluation, la matrice est généralement couplée à une méthode de validation croisée, dans laquelle les données sont réparties en sous-ensembles (folds), avec alternance entre entraînement et test. Cela limite le surapprentissage (overfitting) et améliore la généralisation.

En complément, des métriques probabilistes sont mobilisées :

- La log-loss (ou logarithmic loss) évalue la qualité des probabilités prédites par le modèle. Elle pénalise fortement les prédictions incorrectes très confiantes. Plus sa valeur est faible, meilleure est la calibration du modèle.
- La courbe ROC (Receiver Operating Characteristic) et son aire sous la courbe (AUC) mesurent la capacité du modèle à distinguer les classes, indépendamment du seuil de décision choisi. Un AUC proche de 1 indique une excellente séparabilité, tandis qu'un score proche de 0,5 correspond à une performance aléatoire.

L'intérêt de cette approche globale est de fournir une vision complète du comportement du modèle dans des conditions réalistes, tout en garantissant l'interprétabilité et la reproductibilité des résultats.

4.2.3.1. Définition des composantes de la matrice de confusion

Pour interpréter correctement les résultats d'un modèle de classification, il est essentiel de comprendre les composantes de base de la matrice de confusion. Celle-ci permet de comparer les prédictions du modèle avec les classes réelles, en distinguant les cas correctement ou

incorrectement classés [31]. Un exemple de matrice de confusion pour un problème à trois classes est montré à la figure 17.

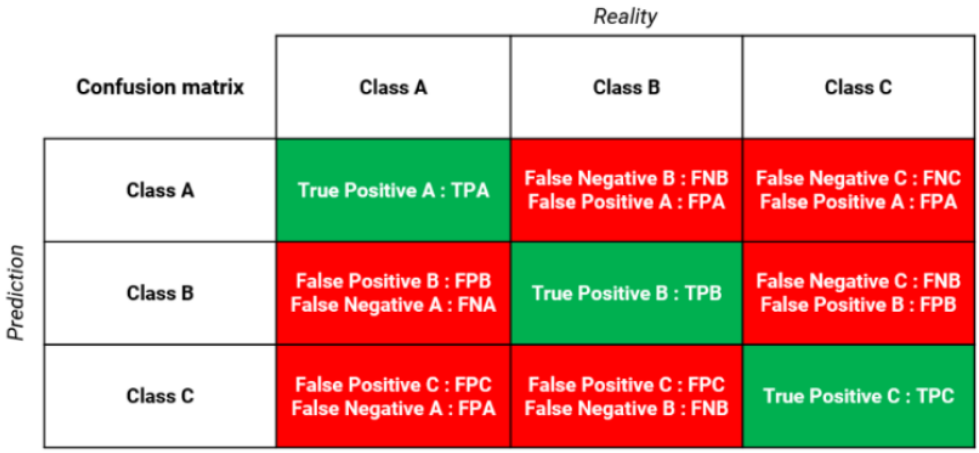


Figure 17
Matrice de confusion pour un problème à trois classes.

Chaque case indique le nombre de cas classés correctement ou incorrectement par le modèle selon les définitions suivantes :

- Vrai Positif (TP) : le modèle prédit correctement qu'une observation appartient à une classe donnée.
- Faux Négatif (FN) : le modèle échoue à détecter une classe présente (erreur d'omission).
- Faux Positif (FP) : le modèle attribue à tort une classe à une observation (erreur d'attribution).
- Vrai Négatif (TN) : le modèle identifie correctement qu'une observation n'appartient pas à la classe cible.

Dans le cadre d'une classification multi-classe, ces notions s'appliquent à chaque classe en considérant celle-ci comme "positive" et toutes les autres comme "négatives". La figure ci-dessus illustre cette décomposition pour trois classes (A, B, C), avec les vrais positifs (TP) sur la diagonale, et les erreurs de classification en dehors.

4.2.3.2. Définition des métriques de performance

Les métriques suivantes, essentielles en apprentissage automatique, sont dérivées des quatre valeurs fondamentales mentionnées ci-dessus. Elles permettent une évaluation quantitative précise de la performance d'un modèle de classification.

Précision (Precision) : elle indique la capacité du modèle à ne pas classer à tort une observation comme appartenant à une classe donnée. Elle se définit comme la proportion de prédictions positives correctes parmi l'ensemble des prédictions positives.

$$Precision = \frac{TP}{TP + FP}$$

Une précision élevée signifie que peu de faux positifs ont été générés, ce qui est crucial lorsque le coût d'une fausse alerte est élevé.

Rappel (Recall ou Sensibilité) : cette métrique évalue la capacité du modèle à identifier toutes les observations pertinentes (positives). Elle se calcule comme le rapport entre les vrais positifs et l'ensemble des cas réellement positifs.

$$Recall = \frac{TP}{TP + FN}$$

Un rappel élevé est essentiel dans des contextes où manquer une détection (faux négatif) est coûteux, comme en médecine ou en détection de défauts critiques.

F-mesure (F1-score) : il s'agit de la moyenne harmonique entre la précision et le rappel. Cette mesure équilibre les deux métriques et est particulièrement utile lorsque les classes sont déséquilibrées.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Une F-mesure élevée reflète un bon compromis entre précision et rappel.

Exactitude (Accuracy) : elle représente le pourcentage global de bonnes prédictions, positives et négatives confondues. C'est l'indicateur le plus intuitif mais qui peut être trompeur en cas de déséquilibre des classes.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Dans des jeux de données fortement déséquilibrés, une accuracy élevée peut masquer une performance médiocre sur la classe minoritaire.

4.2.3.3. Courbe ROC et AUC (Receiver Operating Characteristic)

La courbe ROC est un outil graphique permettant d'évaluer la performance d'un modèle de classification binaire. Elle trace le taux de vrais positifs (sensibilité) en fonction du taux de faux positifs pour différents seuils de classification. L'objectif est de visualiser la capacité du modèle à distinguer les classes, indépendamment d'un seuil particulier.

La surface sous la courbe ROC, appelée AUC (Area Under the Curve), fournit un score unique compris entre 0 et 1 :

- Un AUC = 0.5 indique une performance aléatoire (absence de capacité de discrimination),
- Un AUC = 1.0 traduit une séparation parfaite des classes.

Cette métrique est particulièrement adaptée en contexte de déséquilibre des classes, car elle prend en compte la performance sur les deux classes (positive et négative), quelle que soit leur fréquence d'apparition.

4.2.3.4. Log-Loss (Fonction de perte logarithmique)

La logarithmic loss (ou log-loss) constitue une métrique largement utilisée pour évaluer les performances d'un modèle de classification probabiliste. Contrairement à l'accuracy, qui ne prend en compte que la classe prédite, la log-loss mesure la qualité de la probabilité associée à chaque prédiction. Elle permet ainsi d'évaluer non seulement la justesse de la classification, mais aussi la confiance du modèle dans sa décision

Pour un échantillon i, supposons que le modèle prédit une probabilité p_i que l'étiquette soit la classe correcte (par exemple, « défaut présent »). La contribution à la log-loss est :

- Si la classe réelle est 1 (positive) : $-\log(p_i)$
- Si la classe réelle est 0 (négative) : $-\log(1 - p_i)$

Ainsi, sur l'ensemble des N échantillons, la log-loss est donnée par :

$$\text{Log-Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log p_i + (1 - y_i) \log (1 - p_i)$$

Dans le cas multi-classe, la somme est étendue sur chaque classe j, en utilisant les probabilités p_{ij} attribuées à la classe correcte et la log-loss devient :

$$\text{Log-Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} y_{ij} \log p_i$$

Quelques observations concernant l'interprétation de cette métrique :

- Une log-loss proche de zéro traduit un modèle à la fois précis et bien calibré.
- À l'inverse, une-log loss élevée indique soit une mauvaise classification, soit une mauvaise estimation probabiliste (sur-confiance dans une mauvaise classe, ou incertitude excessive).
- Contrairement à l'accuracy, la log-loss est sensible aux erreurs de calibration et constitue un indicateur précieux dans les contextes où les probabilités sont exploitées pour la prise de décision (comme en maintenance prédictive).

CHAPITRE 5. RÉSULTATS ET DISCUSSION

Ce chapitre présente les résultats issus de l'approche mise en œuvre pour repérer et anticiper les défauts dans les systèmes d'éoliennes. L'objectif est d'évaluer concrètement l'efficacité du dispositif proposé, en mettant en lumière les performances observées à chaque étape du traitement. Les données exploitées, collectées à partir du système SCADA, intègrent différentes mesures (température, vibration, intensité, vitesse du vent…) reflétant l'état de santé des composants techniques.

5.1 Rappel de l'organisation du dispositif

L'analyse repose sur une architecture composée de trois niveaux successifs, chacun jouant un rôle précis dans le processus de détection et d'interprétation. Sans revenir sur les choix méthodologiques détaillés dans le chapitre précédent, on rappellera brièvement ici le rôle et l'intérêt des outils utilisés, avant de présenter les résultats associés.

5.1.1 Première couche : détection initiale des signaux suspects

Cette étape constitue le premier niveau du système de détection d'anomalies appliqué aux données SCADA du parc éolien. Elle repose sur l'entraînement en parallèle de quatre modèles de classification de type arbres ensemblistes. Ces derniers ont été sélectionnés pour leurs caractéristiques complémentaires et leur capacité à faire ressortir les premiers signaux d'anomalie. Ensuite, leurs prédictions ont été combinées par une méthode de stacking : un méta-modèle de régression logistique est entraîné sur les sorties de ces quatre classifieurs de base afin de produire la décision finale. Ainsi les méthodes d'ensemble suivantes ont été choisies pour leur efficacité sur les données tabulaires, leur capacité à capturer des relations non linéaires et leur robustesse vis-à-vis du bruit.

- **XGBoost** s'est distingué par sa faculté à repérer les variables les plus pertinentes, même en présence de bruit ou de redondance dans les données. Il a montré de bons résultats pour isoler les cas évidents de dérive.

- **LightGBM**, de son côté, a bien répondu aux exigences des données chronologiques, notamment celles comportant des mesures fréquentes. Il a permis de détecter des changements progressifs sur certaines variables, utiles pour anticiper les dégradations lentes.

- **CatBoost** a démontré sa pertinence pour traiter des jeux de données variés, en particulier ceux où des capteurs de nature différente étaient combinés. Son aptitude à intégrer automatiquement les données mixtes (numériques et symboliques) a facilité l'exploitation des signaux provenant d'alarmes internes.

- **Random Forest**, utilisé ici comme point de comparaison, a permis de valider les tendances détectées par les autres modèles, en particulier dans les cas où plusieurs paramètres variaient simultanément.

Afin de compléter cette approche supervisée, une phase de détection non supervisée a été ajoutée, reposant sur l'algorithme Isolation Forest. Ce dernier a été entraîné sur les observations classées comme normales par la première étape, avec pour objectif d'identifier des comportements inconnus ou non représentés dans les données d'apprentissage. L'Isolation Forest a été configuré de manière à conserver une haute sensibilité aux comportements atypiques, tout en maintenant un taux de fausses alertes bas. Cette deuxième analyse permet ainsi de renforcer la détection, en captant des signaux faibles ou émergents, souvent absents de l'historique supervisé.

Les résultats issus de cette couche montrent une capacité encourageante à détecter les premiers signes de dysfonctionnement, avec des performances équilibrées sur les différentes classes malgré la rareté de certains types d'événements.

5.1.2 Deuxième couche : combinaison des prédictions

Les événements classés comme anomalie à la sortie de la première couche alimentent, en entrée, une deuxième couche dont l'architecture est composée d'un stacking des mêmes quatre algorithmes employés dans la première couche, mais cette fois-ci, pour une classification multi-classes à quatre catégories.

5.1.3 Troisième couche : traitement ciblé des cas ambigus

La dernière étape s'attache aux situations les plus complexes, où les signaux sont faibles, progressifs, ou difficiles à interpréter. C'est le cas par exemple de la détérioration lente d'un roulement ou d'une variation subtile dans un comportement électrique.

Ici, l'architecture se spécialise selon plusieurs sous-catégories. Chaque type de situation est traité indépendamment, afin de tenir compte de ses spécificités. Pour cela, les données ont été

regroupées par profil de fonctionnement, puis subdivisées en sous-groupes, de manière à mieux explorer la diversité interne et corriger les déséquilibres entre classes.

Chaque sous-groupe a été abordé comme un petit problème de classification spécifique, avec des modèles adaptés. Les outils mobilisés dans cette couche sont :

- **LightGBM** et **XGBoost**, appliqués sur des segments temporels pour suivre l'évolution des anomalies potentielles,
- **Random Forest** et **Gradient Boosting**, choisis pour leur sensibilité différente aux variations rapides ou aux tendances lentes,
- un modèle de **régression logistique** qui agit comme filtre linéaire, pour éliminer les signaux douteux,
- et enfin **CatBoost**, utilisé en fin de parcours pour trancher, grâce à un vote pondéré basé sur la validation croisée.

Cette structure permet d'affiner le diagnostic et de repérer des situations qui, autrement, auraient pu passer inaperçues. Les résultats montrent une amélioration nette des scores sur les cas difficiles, avec une précision accrue sans pour autant augmenter les faux positifs.

5.2 Évaluation de l'apprentissage du modèle

5.2.1 Couche 1 : Classification initiale

Les résultats présentés dans cette section permettent d'évaluer la performance globale de cette couche dans la détection précoce, ainsi que la complémentarité entre les méthodes supervisées et non supervisées pour isoler les données potentiellement à risque.

5.2.1.1. Taille de l'échantillon analysé

L'analyse présentée dans cette étude repose sur des données collectées par le système SCADA d'un parc éolien. Ces données, initialement brutes, ont fait l'objet d'un prétraitement. À l'issue de ce prétraitement, l'échantillon retenu pour l'analyse comprend 359 024 observations, réparties sur 271 variables issues de différents capteurs (vibrations, température, courant, vitesse du vent, etc.).

```
Type de X: <class 'pandas.core.frame.DataFrame'>
Type de y: <class 'pandas.core.series.Series'>
Forme de X: (359024, 271)
Forme de y: (359024,)
Classes uniques: [0 1]
Distribution des classes:
Classe 0: 301509 instances
Classe 1: 57515 instances
Forme après split - X_train: (229775, 271), X_test: (71805, 271)
```

Figure 18
Taille et répartition des données utilisées.

Comme le montre la figure 18, l'échantillon a été divisé en deux sous-ensembles :

- Un ensemble d'entraînement de 229 775 observations,
- Et un ensemble de test de 71 805 observations, réservé à l'évaluation.

La distribution des classes au sein des données est déséquilibrée, ce qui reflète la réalité du terrain dans les systèmes de production :

- 301 509 observations normales,
- Contre 57 515 observations anormales.

Ce déséquilibre, bien que fréquent dans le domaine de la maintenance prédictive, représente un défi important pour les modèles de classification.

Dès lors, l'analyse de la performance des modèles doit aller au-delà de la simple accuracy. Des métriques comme le rappel, le F1-score, ou encore la surface sous la courbe ROC (AUC-ROC) deviennent essentielles pour évaluer la capacité du système à capturer efficacement les anomalies, même lorsqu'elles sont peu fréquentes.

5.2.1.2. Comparaison des performances globales des modèles supervisés

Dans cette section, nous présentons les performances obtenues par les modèles de classification utilisés dans cette couche du système de détection. Les résultats sont exposés selon une logique progressive, incluant une comparaison des métriques globales, une analyse des erreurs via les matrices de confusion, l'étude de la généralisation sur des données temporelles distinctes, ainsi qu'un examen de la stabilité de l'apprentissage.

Le Tableau 6 présente une synthèse des résultats obtenus par les cinq modèles de classification employés dans cette couche du système de détection : les quatre modèles de base (Random Forest, XGBoost, CatBoost, LightGBM) ainsi que le méta-modèle issu de leur combinaison par empilement (stacking). Cinq métriques standard ont été retenues pour évaluer les performances de ces classifieurs : l'exactitude (accuracy), la précision, le rappel, le F1-score, et l'AUC-ROC. Chacune de ces mesures apporte un éclairage distinct sur la qualité de la détection des anomalies.

Tableau 6
Résumé comparatif des performances des modèles dans la première couche de détection

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Random Forest	99,75%	99,88%	98,59%	99,23%	99,92%
XGBoost	99,72%	99,41%	98,83%	99,12%	99,95%
CatBoost	99,67%	99,03%	98,89%	98,96%	99,94%
LightGBM	99,48%	98,09%	98,65%	98,37%	99,92%
Méta-modèle (Stacking)	99,75%	99,66%	98,75%	99,20%	99,96%

De manière générale, l'ensemble des modèles atteint un niveau de performance très élevé. L'accuracy dépasse 99,4 % pour tous les algorithmes, ce qui traduit une capacité globale à discriminer efficacement les situations normales des cas anormaux au sein des données SCADA. Toutefois, une lecture plus fine à travers les autres métriques met en évidence des différences notables dans le comportement de chaque modèle.

Le méta-modèle en stacking, basé sur une régression logistique entraînée sur les sorties des modèles de base, se distingue par sa performance équilibrée : avec une accuracy de 99,75 %, un F1-score de 99,20 % et un AUC de 99,96 %, il offre un compromis optimal entre sensibilité et spécificité. Il parvient à capitaliser sur les points forts des autres classifieurs, tout en en atténuant certaines limites.

Le Random Forest présente la précision la plus élevée (99,88 %), traduisant une excellente maîtrise des faux positifs. Toutefois, son rappel (98,59 %) reste légèrement inférieur, ce qui indique une tendance à manquer certains cas d'anomalie.

Le modèle XGBoost affiche des performances très homogènes. Il combine une bonne précision (99,41 %) à un rappel élevé (98,83 %), ce qui se traduit par un F1-score solide de 99,12 %. Sa capacité à capturer des interactions complexes dans des données déséquilibrées est bien illustrée ici, et son AUC de 99,95 % confirme sa robustesse.

De son côté, CatBoost obtient un F1-score de 98,96 % grâce à un bon équilibre entre précision (99,03 %) et rappel (98,89 %). Sa conception le rend particulièrement efficace sur des jeux de données hétérogènes mêlant variables continues et discrètes, comme c'est le cas dans les systèmes SCADA.

Enfin, LightGBM, tout en demeurant performant, présente des résultats légèrement inférieurs par rapport aux autres modèles. Avec une précision de 98,09 % et un F1-score de 98,37 %, il apparaît comme plus sensible aux variations contextuelles des données, ce qui pourrait expliquer une plus forte production de fausses alertes.

Dans l'ensemble, ces résultats suggèrent que la stratégie d'empilement améliore la fiabilité du système en exploitant les complémentarités des modèles. Le méta-modèle, agissant comme un filtre décisionnel de second niveau, parvient à maintenir un haut niveau de détection tout en minimisant les erreurs critiques, renforçant ainsi la robustesse opérationnelle de cette première couche du pipeline de surveillance.

5.2.1.3. Analyse des erreurs à partir des matrices de confusion

Afin de compléter l'évaluation des performances observées, il est essentiel d'examiner plus en détail la répartition des erreurs de classification à l'aide des matrices de confusion. Celles-ci offrent une lecture plus fine que les indicateurs globaux, en distinguant les cas correctement identifiés des erreurs de type faux positifs (FP) et faux négatifs (FN) — respectivement des cas normaux faussement signalés comme anomalies, et des anomalies réellement présentes mais non détectées. Le tableau 7 regroupe les matrices de confusion des quatre modèles de base afin d'illustrer les tendances spécifiques de chacun en matière d'erreurs de classification.

Tableau 7
Matrices de confusion des modèles supervisés dans la première couche de détection

	Modèle	TP	TN	FP	FN
0	XGBoost	11371	60256	46	132
1	LightGBM	11331	60109	193	172
2	CatBoost	11362	60198	104	141
3	Random Forest	11335	60288	14	168

Quelques observations se dégagent :

- **Random Forest** se distingue par un nombre minimal de faux positifs (14), ce qui en fait le modèle le plus conservateur. Toutefois, il laisse passer 168 anomalies (faux négatifs), traduisant une sensibilité plus faible.

- **XGBoost** affiche un profil plus équilibré, avec 46 faux positifs et 132 faux négatifs, ce qui reflète une meilleure capacité de détection des cas anormaux sans engendrer trop de fausses alertes.

- **CatBoost** adopte une posture intermédiaire, commettant 104 fausses alertes pour 141 anomalies manquées. Ce profil traduit une tolérance modérée à l'ambiguïté des cas limites.

- **LightGBM**, enfin, enregistre les performances les plus hétérogènes, avec 193 faux positifs et 172 faux négatifs, suggérant une plus grande sensibilité au bruit et à la variabilité des données.

L'analyse croisée de ces matrices confirme l'intérêt de l'approche en stacking. La Figure 19 présente la matrice de confusion du méta-modèle basé sur la régression logistique, entraîné sur les prédictions des quatre modèles de base.

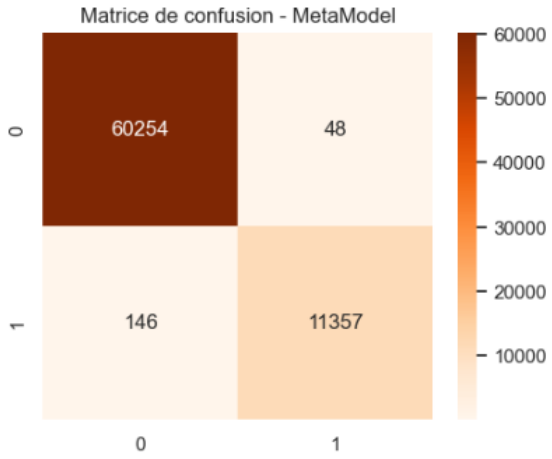


Figure 19
Matrice de confusion du méta-modèle (régression logistique)

Ce modèle combiné a correctement identifié 11 357 anomalies (vrais positifs) et 60 254 cas normaux (vrais négatifs). En revanche, 48 cas normaux ont été signalés à tort comme anomalies (faux positifs), tandis que 146 anomalies ont été manquées (faux négatifs).

Ces résultats traduisent une excellente capacité de détection, avec un compromis très équilibré entre sensibilité et spécificité. Le taux de faux positifs est extrêmement bas, ce qui limite les alertes inutiles, tandis que le nombre d'anomalies non détectées reste modéré et inférieur à celui de plusieurs modèles de base. Le méta-modèle parvient à corriger certaines faiblesses des classifieurs individuels, en particulier :

- Il réduit les faux négatifs par rapport au Random Forest (146 vs. 168),
- Et limite les faux positifs par rapport à LightGBM et CatBoost (48 vs. 193 et 104 respectivement).

Il agit ainsi comme un filtre de second niveau, capable d'ajuster finement les seuils de décision selon les cas, en combinant les tendances détectées par les modèles de base.

5.2.1.4. Évaluation de la capacité de généralisation

L'évaluation comparative des performances en entraînement et en test constitue une étape cruciale pour juger de la capacité de généralisation d'un modèle. Un écart trop important entre les deux phases peut révéler un phénomène de surajustement (*overfitting*), tandis que des résultats faibles dans les deux cas traduisent généralement un apprentissage insuffisant ou un modèle sous-entraîné.

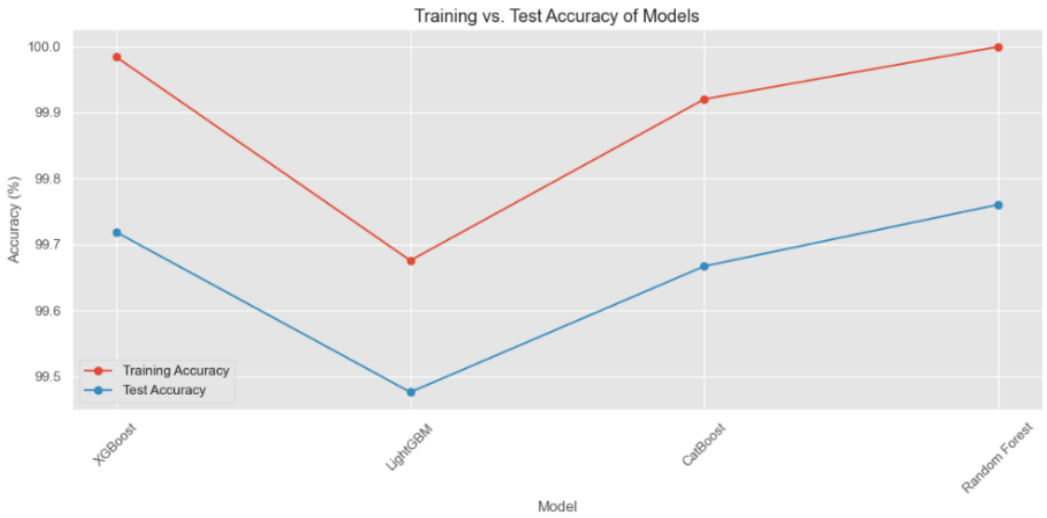


Figure 20
Comparaison des taux d'accuracy en entraînement et en test pour chaque modèle

Comme illustré dans la Figure 20, tous les modèles de cette couche présentent une consistance élevée entre leurs performances en apprentissage et en test, ce qui témoigne d'un entraînement stable et d'une faible variance :

- Random Forest atteint 100 % de précision à l'entraînement et conserve un très bon score en test (~99,75 %), preuve qu'il reste fiable sur des données nouvelles.
- XGBoost suit une tendance similaire avec une précision très élevée sur les deux phases (~99,72 %), sans signe de surajustement.
- CatBoost présente une légère baisse en test (~99,67 %), tout en maintenant des performances solides.
- LightGBM obtient une précision plus modeste dès l'entraînement (~99,5 %) et reproduit ce niveau en test. Cela peut indiquer une plus grande sensibilité aux variations des données, mais aussi un risque moindre de surajustement.

Dans l'ensemble, les modèles sont bien équilibrés : ils apprennent des schémas représentatifs des phénomènes à surveiller, sans se contenter de mémoriser les exemples. Enfin, même s'il n'est pas illustré dans la figure, le méta-modèle hérite de cette stabilité. Il s'appuie sur les prédictions fiables des modèles de base, ce qui renforce sa capacité à bien fonctionner sur des données jamais vues.

En complément des tableaux de métriques, une visualisation graphique a été réalisée afin de comparer plus directement les taux de précision (*accuracy*) de chaque modèle, y compris celui issu du stacking. Cela permet d'illustrer rapidement les écarts de performance sur ce critère spécifique.

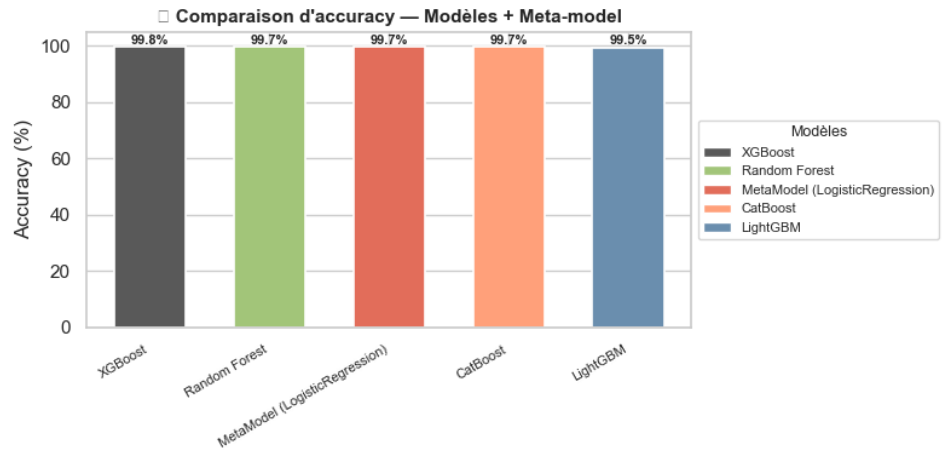


Figure 21
Comparaison des taux d'accuracy en test pour chaque modèle et pour le méta-modèle

La figure 21 montre clairement que le modèle XGBoost se démarque avec la meilleure précision, atteignant 99,8 %, suivi de très près par Random Forest et le méta-modèle, tous deux à 99,7 %. CatBoost affiche un résultat quasiment équivalent, tandis que LightGBM obtient une performance légèrement inférieure, mais toujours élevée, avec 99,5 %.

Ces résultats confirment les métriques chiffrées qui laissaient déjà entrevoir la capacité de l'ensemble des modèles testés à bien distinguer les comportements normaux des anomalies. Toutefois, il faut rester prudent dans l'interprétation de la précision, surtout dans un contexte de déséquilibre important entre les classes, comme c'est le cas ici. Un modèle qui prédirait uniquement la classe majoritaire pourrait afficher un excellent taux de précision, mais ne serait pas utile pour détecter les anomalies réellement présentes.

Enfin, même si tous les modèles dépassent les 99,5 % de précision, cela ne suffit pas à les départager. Ce sont plutôt des indicateurs plus sensibles, comme le F1-score, le rappel ou la capacité à éviter les fausses alertes, qui feront la différence, surtout dans un contexte réel où chaque décision peut avoir un impact opérationnel. Même s'il n'est pas celui qui atteint la précision brute la plus élevée, le méta-modèle reste le plus équilibré et le plus fiable à l'échelle du système.

5.2.1.5. Validation sur un jeu de données indépendant

Afin de tester la capacité de généralisation du système au fil du temps, une phase de validation a été réalisée en appliquant les modèles sur des données issues d'une autre année d'exploitation du même parc. Ce choix permet d'observer si les performances restent stables malgré les possibles variations dues à l'usure des composants, aux changements saisonniers ou à l'évolution des conditions opérationnelles. Le modèle sélectionné pour cette validation est Random Forest, reconnu pour sa stabilité et ses performances constantes observées lors des phases précédentes.

La Figure 22 montre que sur cet échantillon indépendant, le modèle a identifié correctement 1 737 anomalies sur 1 771 (vrais positifs), et 9 989 observations normales sur 10 069 (vrais négatifs). On enregistre 34 anomalies manquées (faux négatifs) et 80 fausses alertes (faux positifs).

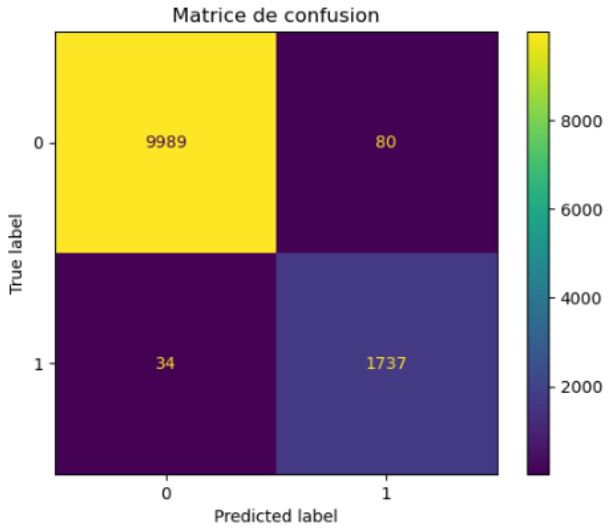


Figure 22
Matrice de confusion obtenue avec les données de validation

Afin d'évaluer la robustesse du modèle sur le long terme, une validation a été effectuée à partir d'un échantillon indépendant, correspondant à une période temporelle différente du même parc éolien. Cette démarche permet de simuler un contexte réaliste dans lequel le modèle est confronté à des variations naturelles liées aux saisons, à l'usure des composants ou aux changements dans les conditions d'exploitation. Le rapport de classification issu de cette évaluation est présenté en Figure 23.

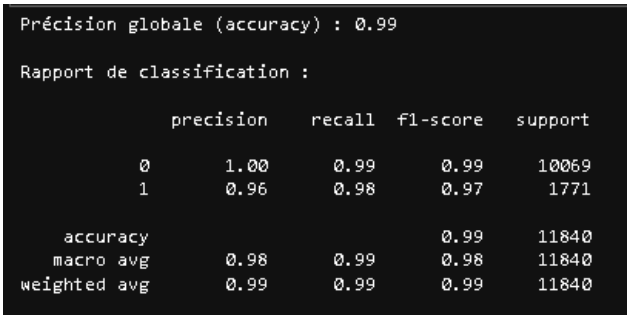


Figure 23
Rapport de classification obtenu avec les données de validation.

Les résultats indiquent une accuracy globale de 99 %, ce qui confirme une performance élevée malgré le décalage temporel. La classe normale est parfaitement identifiée, avec une précision de 100 % et un rappel de 99 %, traduisant la stabilité du modèle dans la détection des états de fonctionnement standards.

Pour la classe des anomalies, on observe une précision de 96 % et un rappel de 98 %, ce qui indique que, même si quelques fausses alertes ont été générées, la grande majorité des anomalies ont été correctement détectées. Le F1-score atteint un excellent niveau de 0,97, confirmant un bon équilibre entre la sensibilité et la spécificité du modèle. De plus, les moyennes macro et pondérée respectivement 0,98 et 0,99 soulignent que le modèle reste performant même dans un contexte de déséquilibre marqué entre les classes.

Le nombre de fausses alertes reste limité, ce qui demeure acceptable dans le cadre d'une application industrielle, où l'objectif est de minimiser les interventions inutiles. Par ailleurs, le rappel élevé (98,1 %) met en évidence la capacité du modèle à détecter efficacement la quasi-totalité des anomalies présentes dans l'échantillon de validation, malgré un décalage temporel par rapport aux données d'entraînement.

Cette cohérence des résultats dans le temps suggère que le modèle n'a pas été appris de manière superficielle ou spécifique à une période donnée, mais qu'il a su identifier des motifs structurels et génériques caractéristiques des états de fonctionnement anormaux. Cela témoigne d'un apprentissage robuste, capable de généraliser au-delà du domaine de référence initial, ce qui est une condition essentielle pour garantir la fiabilité du système en conditions réelles d'utilisation.

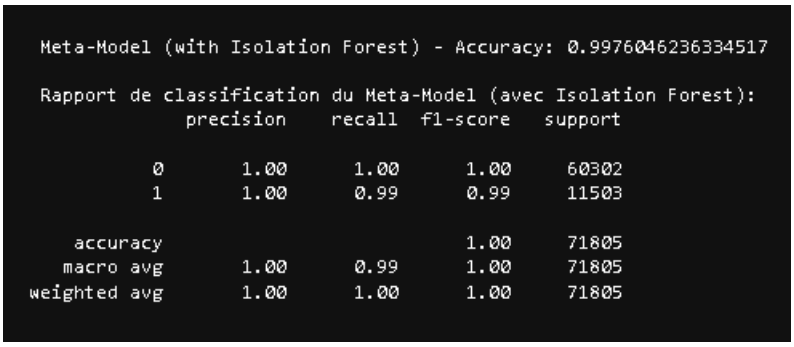
En conclusion, les résultats obtenus valident l'efficacité de la composante supervisée de cette dernière. Qu'il s'agisse de performances sur le jeu de test initial ou sur des données issues d'un

autre horizon temporel, les modèles déployés démontrent leur capacité à identifier de manière précise, stable et cohérente les anomalies. La stratégie d'ensemble utilisée, qui repose sur le stacking de modèles complémentaires, permet d'atteindre un bon compromis entre sensibilité et robustesse, tout en limitant les erreurs de détection et les alertes injustifiées. Cette première couche constitue ainsi un filtre fiable et performant sur lequel les couches supérieures peuvent s'appuyer pour affiner le diagnostic et la prise de décision.

5.2.1.6. Résultats de la détection non supervisée

L'algorithme Isolation Forest, appliqué ici comme composant non supervisé au sein d'un système de détection hiérarchique, a démontré une remarquable capacité à identifier les observations anormales dans les données SCADA. Son efficacité s'observe tant sur les métriques de performance classiques que sur la visualisation de la séparation entre *inliers* et *outliers*.

Le rapport de classification du méta-modèle incluant l'Isolation Forest (figure 24) indique une accuracy globale de 99,76 %, avec des scores F1 très élevés pour les deux classes (1.00 pour les données normales, 0.99 pour les anomalies). Ces résultats traduisent une excellente capacité de détection, sans sur-ajustement manifeste, malgré le déséquilibre important des classes (60 302 instances normales contre 11 503 anomalies).



```
Meta-Model (with Isolation Forest) - Accuracy: 0.9976046236334517

Rapport de classification du Meta-Model (avec Isolation Forest):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     60302
           1       1.00      0.99      0.99     11503

    accuracy                           1.00     71805
   macro avg       1.00      0.99      1.00     71805
weighted avg       1.00      1.00      1.00     71805
```

Figure 24
Rapport de classification du Meta-Modèle basé sur Isolation Forest.

La projection PCA vient renforcer cette évaluation en fournissant une interprétation visuelle convaincante, telle qu'illustrée à la figure 25. Le nuage de points montre une séparation claire entre les *inliers* (bleu) et les *outliers* (rouge), ces derniers se regroupant en marge de la distribution principale. Cette dissociation géométrique confirme que l'algorithme a appris une structure cohérente de la normalité à partir des attributs disponibles.

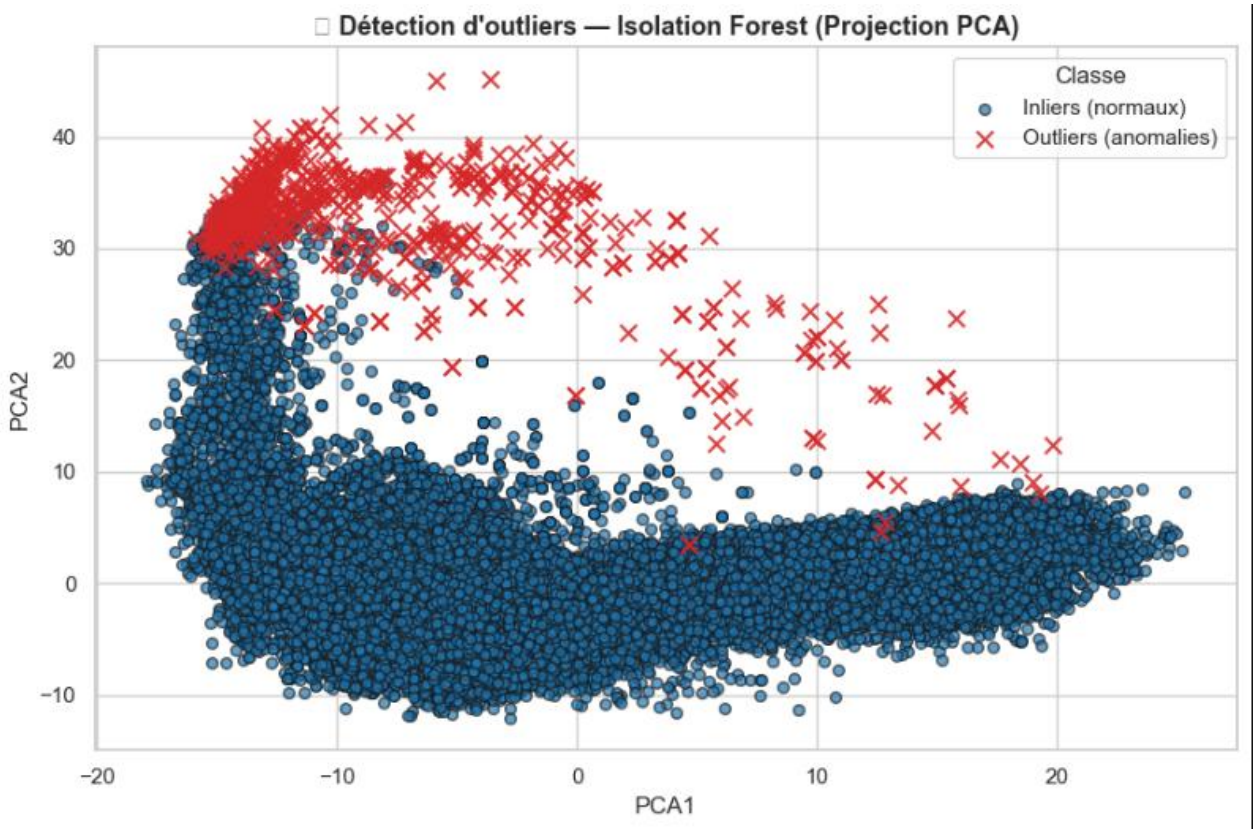


Figure 25
Visualisation de la détection des anomalies par Isolation Forest dans l'espace réduit PCA.

Notons que, malgré un environnement non supervisé, l'Isolation Forest s'est avéré très robuste, capturant les motifs atypiques tout en conservant une faible sensibilité au bruit. Cette robustesse est essentielle dans un contexte de maintenance prédictive, où les anomalies peuvent être rares mais critiques.

Enfin, l'intégration de ce modèle comme module d'appoint dans une architecture multi-couches renforce la pertinence de son usage, en assurant une détection complémentaire aux modèles supervisés. Son efficacité est donc non seulement justifiée par les scores, mais aussi par sa contribution à la détection d'anomalies subtiles non repérées par les classifieurs standards.

5.2.2 Couche 2 : Classification des anomalies détectées

À la suite du premier niveau de détection, la deuxième couche du système prend le relais avec un objectif plus spécifique : classer les anomalies identifiées selon leur type. Cette étape est essentielle pour permettre une intervention ciblée en fonction de la gravité et de la nature du

message. Dans le contexte des systèmes SCADA, les messages d'alerte ont été regroupés en quatre catégories principales tel que présenté dans la figure suivante.

	Message_Original	Message_Encoded
0	Communication	0
1	Informational	1
2	Stop	2
3	Warning	3

Figure 26
Encodage des types de messages SCADA pour la classification.

Les anomalies détectées précédemment sont désormais analysées de manière plus fine. L'objectif ici n'est plus simplement de repérer une anomalie, mais d'en comprendre la nature. Chaque événement est donc classé dans l'une des quatre catégories fonctionnelles : *Communication*, *Informational*, *Stop* ou *Warning*.

Cependant, comme cela est souvent le cas dans ce type de données industrielles, la répartition entre les classes est très inégale. La majorité des messages relèvent de la catégorie *Informational*, tandis que les classes *Stop* et *Warning*, bien que critiques, sont beaucoup plus rares. Ce déséquilibre complique naturellement l'apprentissage des modèles, qui risquent de négliger les cas peu fréquents.

Pour faire face à cette difficulté, tel qu'expliqué dans le chapitre méthodologique, des données synthétiques ont été générées à l'aide d'une simulation Monte Carlo, dans le respect des distributions d'origine. L'idée était de fournir aux modèles un aperçu plus complet des différentes situations possibles, en particulier celles qui sont rares mais sérieuses.

5.2.2.1. Performances globales des modèles

Pour évaluer la capacité des modèles à distinguer les différents types de messages SCADA anormaux (Communication, Informational, Stop, Warning), plusieurs algorithmes ont été testés : Random Forest, XGBoost, CatBoost, LightGBM, ainsi qu'un méta-modèle de stacking. Les performances obtenues sur le jeu de test sont synthétisées dans le tableau 8 présenté ci-dessous.

Tableau 8
Résumé comparatif des performances de classification dans la deuxième couche.

Model	Train Accuracy	Test Accuracy	Recall	Precision	F1-Score	Cross-Val
lightgbm	98,15	97,37	97,37	97,35	97,35	97,39
xgboost	98,15	97,37	97,37	97,36	97,34	97,33
catboost	98,10	97,37	97,37	97,36	97,34	97,42
random_forest	98,15	97,07	97,07	97,06	97,05	97,22
Meta-Model	98,15	97,01	97,01	97,00	96,99	97,42

L'analyse comparative des performances des cinq modèles supervisés (LightGBM, XGBoost, CatBoost, Random Forest et Meta-Model) révèle des résultats globalement élevés, reflétant la robustesse de l'approche sur les données SCADA. Tous les modèles obtiennent des scores supérieurs à 97 % en précision, F1-score et rappel, tant en entraînement qu'en validation croisée. Toutefois, des différences significatives émergent selon la nature du modèle.

- Le Meta-Model (stacking) montre un excellent compromis entre biais et variance, avec un F1-score pondéré de 96,99 % et une précision de 97 %. Sa capacité à agréger les décisions des autres modèles en fait un choix pertinent pour les contextes à forte hétérogénéité.
- XGBoost et LightGBM offrent les meilleurs taux de précision (97,35 % et 97,36 %), et atteignent un F1-score de 97,34–97,35 %, indiquant une bonne robustesse sur les classes fréquentes.
- CatBoost se distingue avec un léger avantage en validation croisée (97,42 %), démontrant une meilleure capacité de généralisation, notamment sur les classes minoritaires grâce à sa gestion native des variables catégorielles.
- Random Forest, bien que performant, montre une légère baisse de F1-score (97,05 %) et de précision (97,06 %) par rapport aux modèles boosting, ce qui suggère une sensibilité accrue au bruit ou aux déséquilibres.

Cette analyse justifie l'utilisation d'un stacking en tant que méta-apprentissage, particulièrement adapté aux problèmes complexes, où chaque modèle apporte une vision partielle mais complémentaire. Elle met également en lumière l'importance de la validation croisée pour estimer

la capacité de généralisation dans des environnements SCADA sujets à des fluctuations multivariées.

5.2.2.2. Analyse des erreurs à partir des matrices de confusion

La figure 27 illustre les matrices de confusion obtenues pour chacun des modèles de classification testés dans cette deuxième couche. L'analyse conjointe de ces matrices permet d'appréhender les comportements spécifiques de chaque modèle face à une tâche de classification déséquilibrée et à forte variabilité.

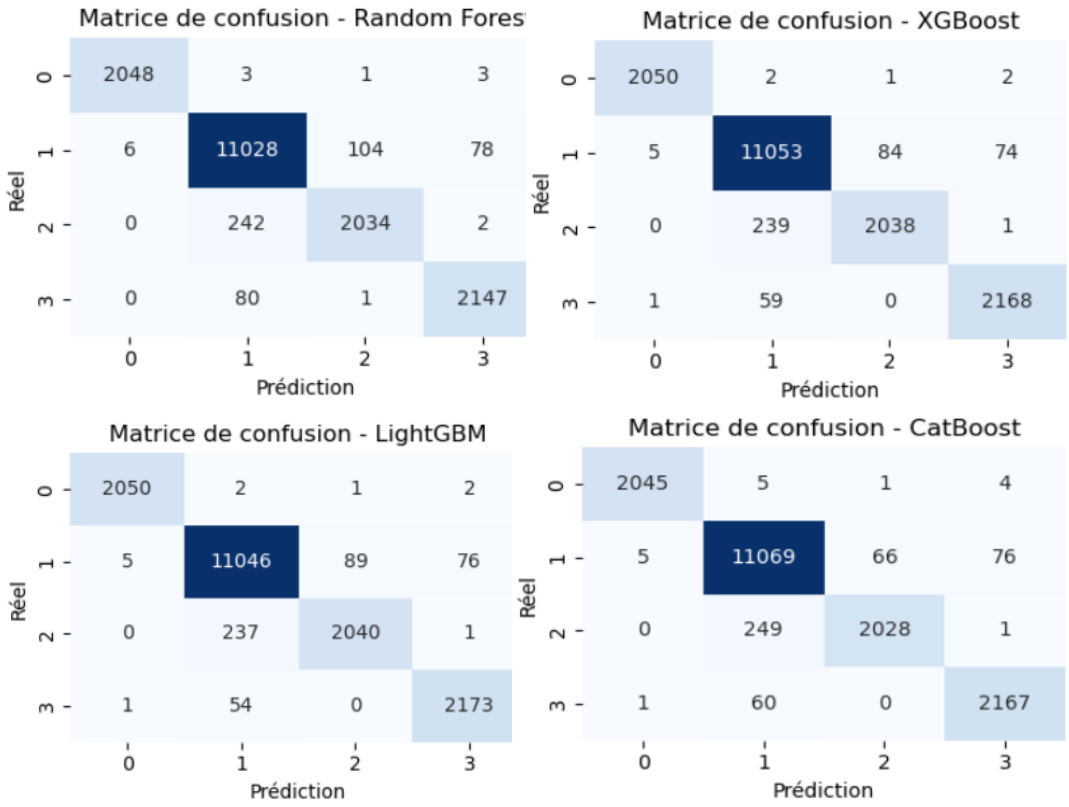


Figure 27
Matrices de confusion des modèles de classification de la deuxième couche.

De manière générale, tous les modèles présentent une excellente capacité à identifier la classe majoritaire Informational (classe 1), avec un taux de reconnaissance supérieur à 11 000 cas correctement prédits. Ce résultat traduit une assimilation robuste des structures dominantes du jeu de données. Cependant, des difficultés apparaissent systématiquement dans la distinction des classes minoritaires, notamment Stop (classe 2) et Warning (classe 3), souvent confondues avec la classe dominante en raison de leur fréquence plus faible et de leur proximité sémantique.

XGBoost, par exemple, confond 239 instances de la classe Stop avec la classe Informational, et 59 exemples Warning sont également mal attribués. Ce schéma d'erreur, bien que modéré, indique une propension du modèle à se replier sur la classe dominante lorsqu'une incertitude est présente. LightGBM montre un comportement similaire, avec des confusions notables sur les classes minoritaires mais une prédiction stable sur les classes 0 et 1.

CatBoost se distingue légèrement par une meilleure gestion de la classe Stop, avec seulement 66 confusions, soit la meilleure performance sur ce point parmi les modèles testés. Toutefois, les erreurs sur la classe Warning restent comparables aux autres méthodes. Le Random Forest, quant à lui, bien qu'efficace sur les classes majoritaires, souffre d'un taux d'erreur plus élevé sur les classes critiques, avec 104 confusions pour la classe Stop.

Enfin, le méta-modèle issu du stacking synthétise les prédictions des quatre modèles précédents. Les résultats ainsi obtenus sont illustrés dans la figure 28.

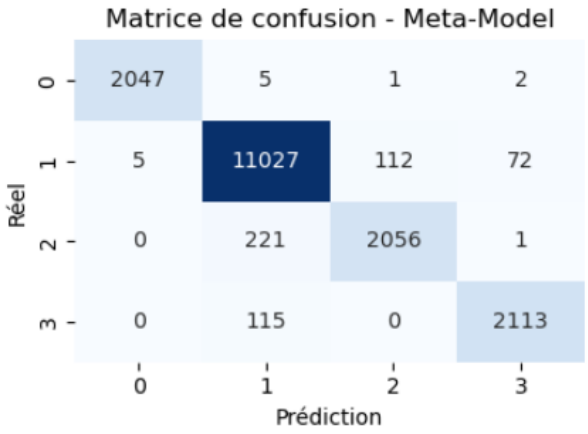


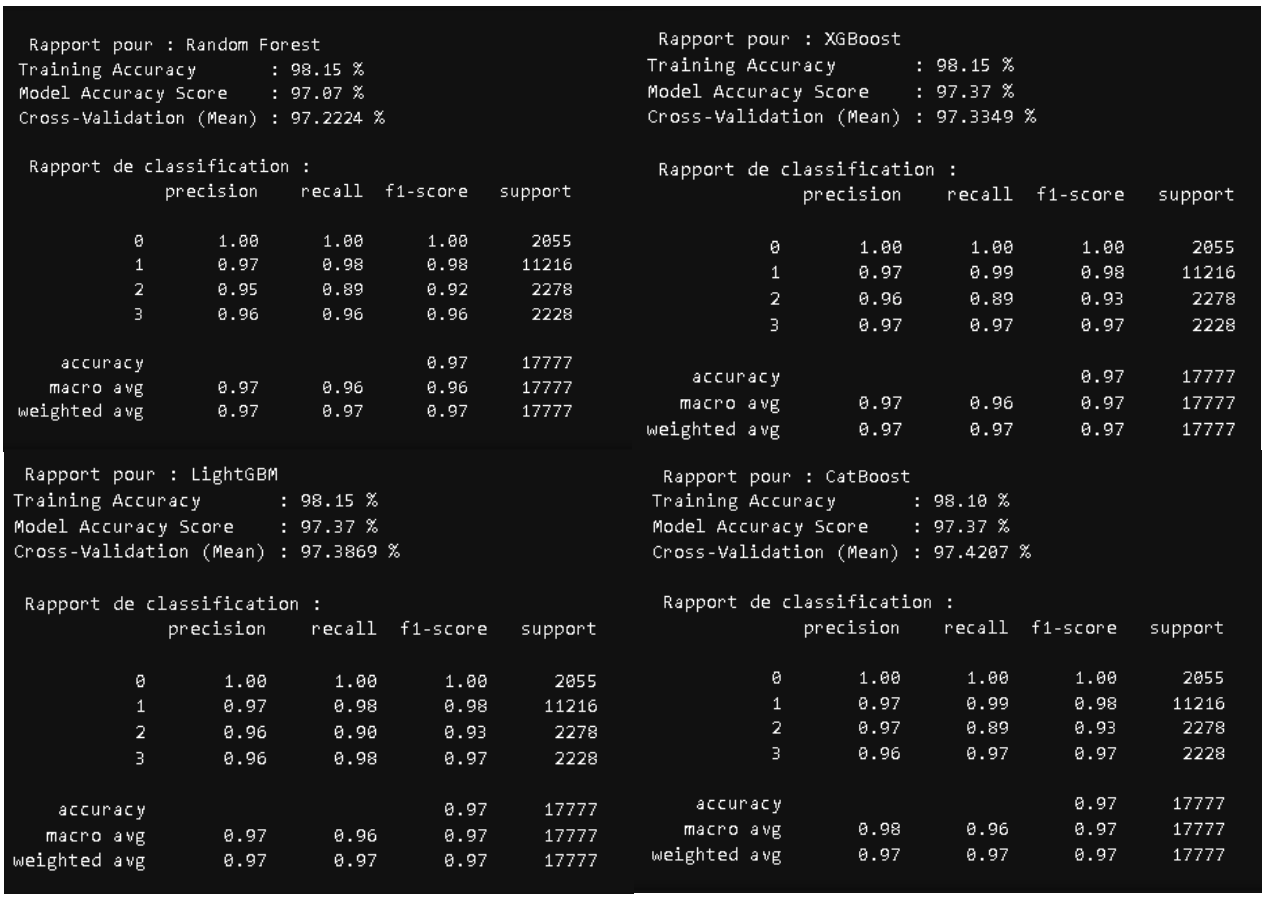
Figure 28
Matrice de confusion du méta-modèle de la deuxième couche.

On remarque une capacité robuste à identifier les classes majoritaires, avec 11 027 messages correctement classés comme *Informational* et 2 047 comme *Communication*, illustrant une bonne stabilité sur les comportements les plus fréquents. Pour les classes minoritaires, il parvient à détecter 2056 messages de type *Stop* et 2113 de type *Warning*.

Le nombre d'erreurs reste modéré sur la classe *Stop* (221 cas mal classés), tandis que la classe *Warning* demeure plus délicate, avec 115 erreurs. Ces résultats traduisent une performance équilibrée, même si des marges d'amélioration subsistent sur les catégories les plus rares. Le stacking joue ici un rôle de stabilisation des prédictions, en limitant à la fois les faux positifs et les

faux négatifs, ce qui est essentiel dans une logique industrielle de maintenance prédictive où la fiabilité des alertes est primordiale.

La figure 29 présente les rapports de classification détaillés des modèles testés pour la classification multi-classes des messages SCADA : Random Forest, XGBoost, LightGBM, CatBoost. Ces performances sont mesurées sur les quatre classes cibles (Communication, Informational, Stop, Warning) à l'aide d'indicateurs clés : précision, rappel, F1-score, ainsi que les moyennes pondérées (weighted average) et non pondérées (macro average).



Rapport pour : Random Forest
Training Accuracy : 98.15 %
Model Accuracy Score : 97.07 %
Cross-Validation (Mean) : 97.2224 %

Rapport de classification :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2055
1	0.97	0.98	0.98	11216
2	0.95	0.89	0.92	2278
3	0.96	0.96	0.96	2228
accuracy			0.97	17777
macro avg	0.97	0.96	0.96	17777
weighted avg	0.97	0.97	0.97	17777

Rapport pour : XGBoost
Training Accuracy : 98.15 %
Model Accuracy Score : 97.37 %
Cross-Validation (Mean) : 97.3349 %

Rapport de classification :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2055
1	0.97	0.99	0.98	11216
2	0.96	0.89	0.93	2278
3	0.97	0.97	0.97	2228
accuracy			0.97	17777
macro avg	0.97	0.96	0.97	17777
weighted avg	0.97	0.97	0.97	17777

Rapport pour : LightGBM
Training Accuracy : 98.15 %
Model Accuracy Score : 97.37 %
Cross-Validation (Mean) : 97.3869 %

Rapport de classification :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2055
1	0.97	0.98	0.98	11216
2	0.96	0.90	0.93	2278
3	0.96	0.98	0.97	2228
accuracy			0.97	17777
macro avg	0.97	0.96	0.97	17777
weighted avg	0.97	0.97	0.97	17777

Rapport pour : CatBoost
Training Accuracy : 98.10 %
Model Accuracy Score : 97.37 %
Cross-Validation (Mean) : 97.4207 %

Rapport de classification :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2055
1	0.97	0.99	0.98	11216
2	0.97	0.89	0.93	2278
3	0.96	0.97	0.97	2228
accuracy			0.97	17777
macro avg	0.98	0.96	0.97	17777
weighted avg	0.97	0.97	0.97	17777

Figure 29
Rapports de classification des modèles de la deuxième couche.

Sur la classe Communication (0), tous les modèles affichent une précision et un rappel de 1,00, traduisant une identification parfaite des messages de cette catégorie. La classe Informational (1), bien qu'étant la plus représentée, montre un très léger écart : les rappels oscillent entre 0,98 (Random Forest) et 0,99 (XGBoost, CatBoost), tandis que la précision varie entre 0,97 et 0,98. Cela reflète une stabilité forte sur les classes dominantes.

Les différences les plus notables apparaissent sur les classes Stop (2) et Warning (3). Pour la classe Stop (2), le rappel est de 0,89 pour CatBoost et XGBoost, de 0,88 pour LightGBM, et descend à 0,87 pour Random Forest. La précision, quant à elle, varie de 0,93 (Random Forest) à 0,94 (CatBoost), indiquant un taux d'identification acceptable mais encore perfectible sur cette classe critique.

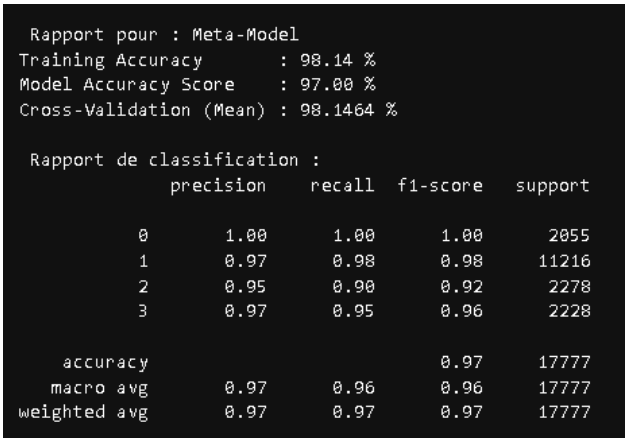
Concernant la classe Warning, qui reste la plus difficile à prédire, les modèles peinent davantage. En effet, le rappel descend à 0,85 pour LightGBM, et se maintient entre 0,86 et 0,88 pour les autres modèles. La précision varie entre 0,88 et 0,91, avec CatBoost qui obtient les meilleurs résultats sur cette classe sensible.

En termes de F1-score global (macro average), CatBoost et XGBoost atteignent tous deux 0,97, traduisant un bon équilibre entre précision et rappel sur l'ensemble des classes. LightGBM est légèrement en retrait (0,96), notamment à cause de ses performances plus modestes sur Warning. Le méta-modèle, quant à lui, affiche un F1 macro de 0,96, mais compense par une meilleure stabilité inter-classes.

Les scores weighted average, qui tiennent compte du déséquilibre entre classes, se situent tous entre 0,97 et 0,98, confirmant une performance solide sur les classes majoritaires. Enfin, l'accuracy globale des modèles varie peu : 97 % pour Random Forest et LightGBM, et 98 % pour CatBoost et XGBoost.

En résumé, les résultats montrent que tous les modèles sont très performants sur les classes fréquentes, mais que CatBoost et XGBoost offrent une meilleure couverture sur les classes minoritaires. Le méta-modèle en stacking parvient à équilibrer ces performances, assurant une classification robuste et fiable même dans un contexte de forte hétérogénéité des données SCADA.

La figure 30 ci-dessous présente le rapport de classification du méta-modèle obtenu par empilement des prédictions des quatre modèles de base (XGBoost, LightGBM, CatBoost et Random Forest). Il s'agit d'une approche visant à tirer parti des forces respectives de chaque classifieur afin d'améliorer la robustesse globale de la prédiction.



```
 Rapport pour : Meta-Model
Training Accuracy       : 98.14 %
Model Accuracy Score    : 97.00 %
Cross-Validation (Mean) : 98.1464 %

 Rapport de classification :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2055
           1       0.97      0.98      0.98     11216
           2       0.95      0.90      0.92      2278
           3       0.97      0.95      0.96      2228

    accuracy                           0.97     17777
   macro avg       0.97      0.96      0.96     17777
weighted avg       0.97      0.97      0.97     17777
```

Figure 30
Rapport de classification du méta-modèle pour la deuxième couche.

Ce modèle atteint une accuracy globale de 97,00 %, indiquant une capacité de généralisation satisfaisante sur l'ensemble de test. D'un point de vue plus granulaire, les performances par classe se déclinent comme suit :

- Classe 0 (Communication) : le modèle atteint ici une précision et un rappel parfaits de 1,00, traduisant une détection sans erreur de cette catégorie bien représentée.
- Classe 1 (Informational) : avec une précision de 0,97 et un rappel de 0,98, le modèle parvient à bien capter cette classe majoritaire tout en limitant les fausses alertes. Le F1-score s'établit à 0,98.
- Classe 2 (Stop) : cette classe plus rare affiche une précision de 0,95 et un rappel de 0,90, ce qui implique que certains cas sont encore manqués. Le F1-score de 0,92 reste toutefois acceptable.
- Classe 3 (Warning) : avec une précision de 0,97 et un rappel de 0,95, le modèle montre une bonne sensibilité malgré la rareté des événements. Le F1-score atteint 0,96.

Les moyennes globales confirment la régularité du modèle :

- Macro average : 0,96 (précision, rappel, F1-score)
- Weighted average : 0,97

Ces résultats illustrent que le méta-modèle, en combinant les prédictions des modèles de base, offre un compromis efficace entre détection des cas fréquents et gestion des classes minoritaires. Il constitue ainsi un levier pertinent dans le cadre d'une stratégie de classification multi-classes appliquée à un environnement industriel déséquilibré.

5.2.2.3. Évaluation de la performance : Test et validation croisée

La figure 31 ci-dessous présente une synthèse visuelle des performances globales des modèles supervisés utilisés dans cette couche, selon trois indicateurs clés : l'accuracy sur le jeu d'entraînement, l'accuracy sur le jeu de test, et la moyenne des scores de validation croisée. Cette triple évaluation permet d'apprécier à la fois la capacité d'apprentissage, la généralisation et la stabilité de chaque modèle.

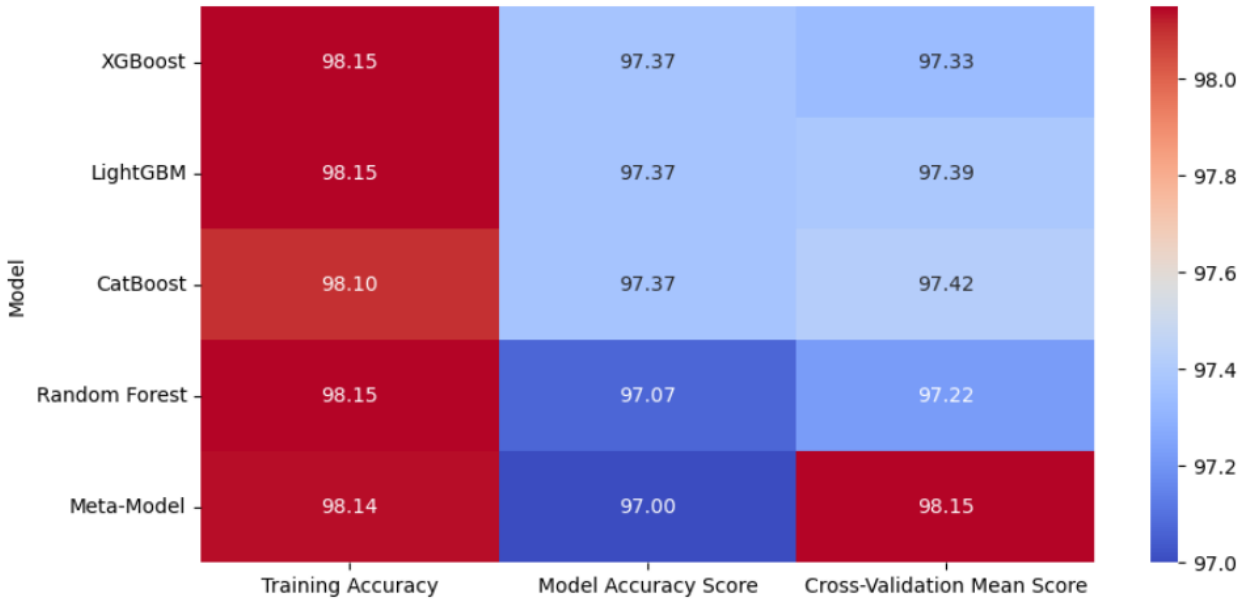


Figure 31
Comparaison des performances globales : Entraînement, test et validation croisée.

On observe que tous les modèles atteignent un niveau d'accuracy élevé en entraînement, avoisinant 98,10 à 98,15 %. Le modèle CatBoost présente la valeur d'apprentissage la plus faible (98,10 %), ce qui peut refléter une régularisation plus marquée. À l'inverse, XGBoost, LightGBM, Random Forest et le méta-modèle enregistrent des performances similaires autour de 98,14–98,15 %, traduisant une bonne capacité d'apprentissage sur les données d'origine.

En phase de test, les différences se creusent légèrement. CatBoost, XGBoost et LightGBM se maintiennent à 97,37 % d'accuracy, suggérant une excellente robustesse sur des données non vues. Random Forest décroît légèrement à 97,07 %, tandis que le méta-modèle présente la précision en test la plus faible (97,00 %), indiquant que la combinaison des prédictions n'améliore pas systématiquement l'accuracy brute.

Cependant, lorsqu'on s'intéresse à la validation croisée, le comportement du méta-modèle se distingue. Il atteint le score moyen le plus élevé (98,15 %), surpassant les autres modèles, dont les scores varient entre 97,22 % (Random Forest) et 97,42 % (CatBoost). Cette performance suggère que le stacking confère au modèle une stabilité supérieure, le rendant moins sensible aux fluctuations du jeu de données.

En résumé, bien que l'accuracy test du méta-modèle ne soit pas la plus élevée, sa régularité en validation croisée en fait un candidat fiable et robuste pour la tâche de classification multi-classes, notamment dans ce contexte de déséquilibre. Ces résultats confirment l'intérêt du stacking pour consolider les forces des modèles de base et améliorer la généralisation du système.

L'ensemble des résultats obtenus à travers cette deuxième couche met en évidence la pertinence des approches supervisées pour la classification fine des anomalies SCADA. Malgré une forte asymétrie dans la distribution des classes, les modèles testés ont montré une capacité remarquable à généraliser.

Ainsi, cette couche remplit pleinement son rôle de catégorisation détaillée des anomalies, en fournissant non seulement une détection, mais aussi une interprétation plus fine et contextualisée des alertes. Elle constitue une brique essentielle du système global, en contribuant directement à la fiabilité des décisions d'intervention sur le terrain.

5.2.3 Couche 3 : spécialisation par cas ambigus

La troisième couche du système vise à affiner la détection pour les cas ambigus, difficiles à discriminer, souvent caractérisés par des signaux faibles, flous ou peu fréquents. Comme évoqué dans la méthodologie, certains événements critiques se répètent rarement (une ou deux fois seulement), ce qui complique leur apprentissage. Pour pallier cette limitation, des techniques avancées de génération de données synthétiques, telles que les simulations Monte Carlo et l'approche KDE (Kernel Density Estimation), ont été mobilisées afin de mieux représenter ces cas rares tout en respectant les distributions initiales.

Sur le plan structurel, la couche est organisée autour de quatre groupes principaux correspondant aux classes de la couche précédente : Warning, Stop, Informational et Communication. Chaque groupe (à l'exception de la classe Communication) a été subdivisé en sous-groupes internes, de façon à créer de la variabilité intra-classe et capturer la diversité comportementale présente dans les données. Cette stratégie permet de renforcer la granularité de la détection et d'éviter l'apprentissage biaisé.

Le traitement de la classe Communication a suivi une approche spécifique. En l'absence de sous-variabilité dans les cas observés, une classification binaire dédiée a été construite. Un échantillon représentatif du comportement normal a été extrait, puis confronté à la classe Communication dans le but de construire un modèle spécialisé, chargé de discriminer ces cas sans interférence avec les autres classes.

C'est à partir de cette structuration que les résultats de la troisième couche sont présentés dans les sections suivantes, avec une attention particulière portée à la capacité des modèles à détecter correctement les cas ambigus et rares.

5.2.3.1. Comparaison des performances des modèles sur le sous-groupe communication

La classe *Communication* a été traitée comme un ensemble homogène, ne nécessitant pas de division en sous-catégories. Pour améliorer sa détection, un modèle binaire a été entraîné à distinguer ces messages d'un échantillon de données représentatives du comportement normal. Cette approche ciblée visait à isoler efficacement les messages non critiques tout en évitant leur confusion avec d'éventuelles anomalies. La figure 32 ci-dessous présente une synthèse comparative des performances des différents modèles appliqués au sous-groupe "Communication", à travers plusieurs métriques classiques : accuracy, recall, precision, F1-score, log-loss et validation croisée.

	Model	Train Accuracy	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	LightGBM	0.9998	0.9921	1.0000	0.9789	0.9893	0.0254	0.9883
1	RandomForest	1.0000	0.9909	0.9983	0.9773	0.9877	0.0272	0.9869
2	XGBoost	1.0000	0.9945	1.0000	0.9853	0.9926	0.0185	0.9907
3	CatBoost	0.9985	0.9903	0.9967	0.9772	0.9869	0.0238	0.9878
4	GradientBoosting	1.0000	0.9951	1.0000	0.9869	0.9934	0.0127	0.9921
5	LogisticRegression	0.7790	0.7574	0.8391	0.6262	0.7172	0.4698	0.7874

Figure 32
Comparaison des performances des différents modèles appliqués au sous-groupe "Communication".

Les résultats révèlent une excellente capacité des modèles d'ensemble à capturer la dynamique de ce sous-groupe. Les modèles XGBoost, Gradient Boosting, et LightGBM se distinguent particulièrement, atteignant tous une précision en test supérieure à 99 %, un rappel parfait (1,00),

ainsi qu'un F1-score élevé (> 0,99). Ces performances suggèrent une capacité remarquable à détecter correctement les cas positifs tout en minimisant les fausses alertes.

Le Gradient Boosting en particulier se démarque avec le meilleur F1-score (0,9934) et le log-loss le plus faible (0,0127), ce qui confirme sa capacité à modéliser avec finesse les cas ambigus, tout en maintenant une forte confiance dans ses prédictions. Son score moyen de validation croisée (0,9921) renforce sa robustesse sur des échantillons variés.

Le modèle CatBoost affiche également des performances solides, avec un rappel de 0,9967 et un F1-score de 0,9869, soulignant sa bonne capacité à généraliser sans sur-ajuster les données d'apprentissage.

À l'inverse, la régression logistique présente des résultats nettement en retrait, avec une précision de 0,6262, un F1-score de 0,7172, et un log-loss élevé (0,4698). Ces chiffres traduisent une difficulté notable à capturer la frontière de décision dans ce sous-groupe, probablement en raison de la nature non linéaire des interactions entre les variables.

En résumé, les modèles basés sur les arbres de décision montrent une forte adéquation avec le sous-groupe "Communication", tant sur les cas fréquents que sur les cas plus ambigus. Leur capacité à conserver un équilibre entre précision et rappel, tout en minimisant l'incertitude prédictive (log-loss), confirme leur pertinence dans le cadre d'une détection intelligente des messages de type "communication".

5.2.3.2. Analyse des erreurs à partir des matrices de confusion

Afin de visualiser concrètement les performances de chaque modèle sur la classification binaire du groupe *Communication*, la figure 33 rassemble les matrices de confusion correspondantes aux six algorithmes évalués. L'objectif principal est de garantir une détection fiable des cas positifs tout en minimisant les fausses alertes.

	Modèle	TP	TN	FP	FN
0	LightGBM	603	1028	14	0
1	RandomForest	601	1027	15	2
2	LogisticRegression	507	721	321	96
3	XGBoost	603	1033	9	0
4	GradientBoosting	603	1035	7	0
5	CatBoost	601	1029	13	2

Figure 33
Matrices de confusion comparées pour LightGBM, Random Forest, XGBoost, CatBoost, Gradient Boosting et Régression Logistique.

Le modèle Gradient Boosting s'impose nettement avec une excellente détection, ne manquant aucun cas réel (FN = 0) et ne produisant qu'un faible nombre d'alertes erronées (FP = 7). Cette performance témoigne d'une très bonne adéquation à la problématique industrielle posée.

XGBoost offre également des résultats satisfaisants, avec une absence complète de faux négatifs, mais un nombre légèrement supérieur de faux positifs (FP = 9). Bien que proche du Gradient Boosting, il induirait potentiellement une légère hausse des vérifications inutiles sur le terrain.

Concernant LightGBM, bien que ne manquant aucun cas réel, le nombre plus élevé de fausses alertes (FP = 14) représente une faiblesse relative qui nécessiterait une analyse approfondie avant son adoption dans un contexte opérationnel.

Les modèles Random Forest et CatBoost montrent une légère diminution de sensibilité avec deux cas réels non détectés chacun. Leur taux modéré de fausses alertes (FP autour de 13-15) reste toutefois acceptable selon le contexte opérationnel envisagé.

Enfin, la Régression Logistique affiche des résultats manifestement insuffisants avec un nombre significatif d'erreurs dans les deux catégories (FP = 321, FN = 96), rendant ce modèle clairement inapproprié pour cette tâche.

En conclusion, les modèles ensemblistes, particulièrement le Gradient Boosting, se révèlent être les plus adaptés au traitement précis et efficace des messages de type *Communication*, permettant une gestion fiable et pragmatique en milieu industriel

5.2.3.3. Comparaison des performances des modèles sur le sous-groupe Informational

Les événements du groupe Informational dominent en nombre dans le jeu de données analysé, ce qui indique qu'il s'agit principalement de messages de routine et non d'anomalies. Cette surreprésentation couplée à une faible variabilité entre ces événements rend leur classification plus subtile, comme le montrent les résultats, dans la figure 34 suivante. Nous examinons brièvement comment ces caractéristiques influencent les performances obtenues.

Résultats pour le modèle : GradientBoosting

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	87.24	87.24	86.80	87.01	0.3820	87.29
1	Groupe_2	96.31	96.31	96.34	96.32	0.1678	96.61
2	Groupe_3	86.24	86.24	86.46	86.30	0.8024	87.78
3	Groupe_4	18.59	18.59	18.31	18.45	1.1200	21.46
4	Groupe_5	33.95	33.95	33.57	33.74	0.9478	37.67

Résultats pour le modèle : CatBoost

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	89.08	89.08	88.88	88.93	0.2118	88.04
1	Groupe_2	96.69	96.69	96.80	96.74	0.0799	96.68
2	Groupe_3	85.32	85.32	85.80	85.34	0.3790	88.31
3	Groupe_4	33.52	33.52	32.59	33.00	0.9556	35.08
4	Groupe_5	49.03	49.03	48.06	48.48	0.7572	50.31

Résultats pour le modèle : LightGBM

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	87.89	87.89	90.88	89.16	0.4060	89.18
1	Groupe_2	96.18	96.18	96.37	96.25	0.1281	96.76
2	Groupe_3	82.11	82.11	82.29	82.17	0.5735	86.30
3	Groupe_4	20.23	20.23	26.41	22.41	1.0433	23.46
4	Groupe_5	42.73	42.73	42.55	41.07	0.9209	42.16

Résultats pour le modèle : RandomForest

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	86.84	86.84	88.60	87.62	0.5066	88.18
1	Groupe_2	96.11	96.11	96.18	96.15	0.1683	96.49
2	Groupe_3	83.03	83.03	83.89	83.31	0.4318	87.46
3	Groupe_4	17.01	17.01	17.73	17.35	2.2221	20.10
4	Groupe_5	31.79	31.79	32.10	31.89	1.6871	36.54

Résultats pour le modèle : LogisticRegression

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	73.49	73.49	87.01	78.35	1.1453	81.44
1	Groupe_2	92.09	92.09	92.39	91.83	0.4744	90.05
2	Groupe_3	82.57	82.57	83.20	82.28	0.4473	78.31
3	Groupe_4	42.19	42.19	41.20	41.25	1.0602	41.03
4	Groupe_5	44.21	44.21	46.81	44.51	1.3760	43.28

Résultats pour le modèle : XGBoost

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	87.83	87.83	87.76	87.76	0.2737	87.83
1	Groupe_2	96.24	96.24	96.17	96.20	0.1182	96.66
2	Groupe_3	85.78	85.78	86.18	85.82	0.5404	88.13
3	Groupe_4	19.98	19.98	19.88	19.93	1.0551	21.95
4	Groupe_5	36.87	36.87	36.48	36.65	0.8317	39.73

Figure 34
Comparaison des performances des différents modèles appliqués au sous-groupe "Informational".

Les résultats présentés témoignent d'une disparité notable entre les groupes et soulignent la capacité différente des modèles à les distinguer. Pour les Groupes 1 et 2, les algorithmes à base d'arbres (GradientBoosting, CatBoost, LightGBM, RandomForest, XGBoost) atteignent des niveaux de précision et de rappel supérieurs à 85 %, voire souvent au-delà de 90 %. Ces performances se reflètent dans des F1-scores élevés (généralement supérieurs à 85 %) et des log-loss modérés, traduisant une bonne calibration probabiliste. On observe que la Régression Logistique reste en retrait, notamment pour le Groupe 1, avec une précision de 73,49 % contre plus de 87 % chez les méthodes d'ensemble, mais elle se défend un peu mieux sur le Groupe 2 (92,09 %). Cette différence suggère que la nature des variables et leurs interactions bénéficient davantage d'une approche non linéaire, comme le proposent les forêts ou les modèles de boosting.

En revanche, pour les Groupes 4 et 5, tous les modèles peinent à surpasser les 50 % de précision. Les algorithmes d'ensemble, habituellement performants, n'atteignent souvent que 20 % à 40 % de taux de réussite, et la Régression Logistique ne fait guère mieux (42,19 % pour le Groupe 4, 44,21 % pour le Groupe 5). Les rappels et précisions très bas dans ces groupes mènent à des F1-scores faibles (souvent < 35 %), tandis que le log-loss élevé (> 1,0 dans certains cas) traduit la difficulté à calibrer correctement les probabilités. Ce constat indique que la distribution et la complexité des données dans ces deux groupes (peut-être peu d'observations, un fort déséquilibre des classes ou des patterns trop hétérogènes) compliquent la tâche des algorithmes, qu'ils soient linéaires ou non linéaires.

Enfin, les scores de validation croisée (Cross-Val) confirment la robustesse relative des modèles de boosting et de forêts sur les Groupes 1, 2 et 3, où ils affichent régulièrement des moyennes > 85 %. Pour les Groupes 4 et 5, même si les scores restent faibles, la Cross-Val indique la même tendance : aucune méthode ne parvient à extraire efficacement des règles de décision, suggérant un besoin d'analyse plus poussée (sélection de variables, ingénierie de caractéristiques, gestion du déséquilibre ou éventuels problèmes de qualité de données). En somme, ces observations soulignent que les méthodes d'ensemble sont plus à même de modéliser des groupes "faciles" ou modérément complexes (1, 2, 3), tandis que de nouveaux ajustements ou de nouvelles approches seraient requis pour améliorer substantiellement les résultats sur les Groupes 4 et 5.

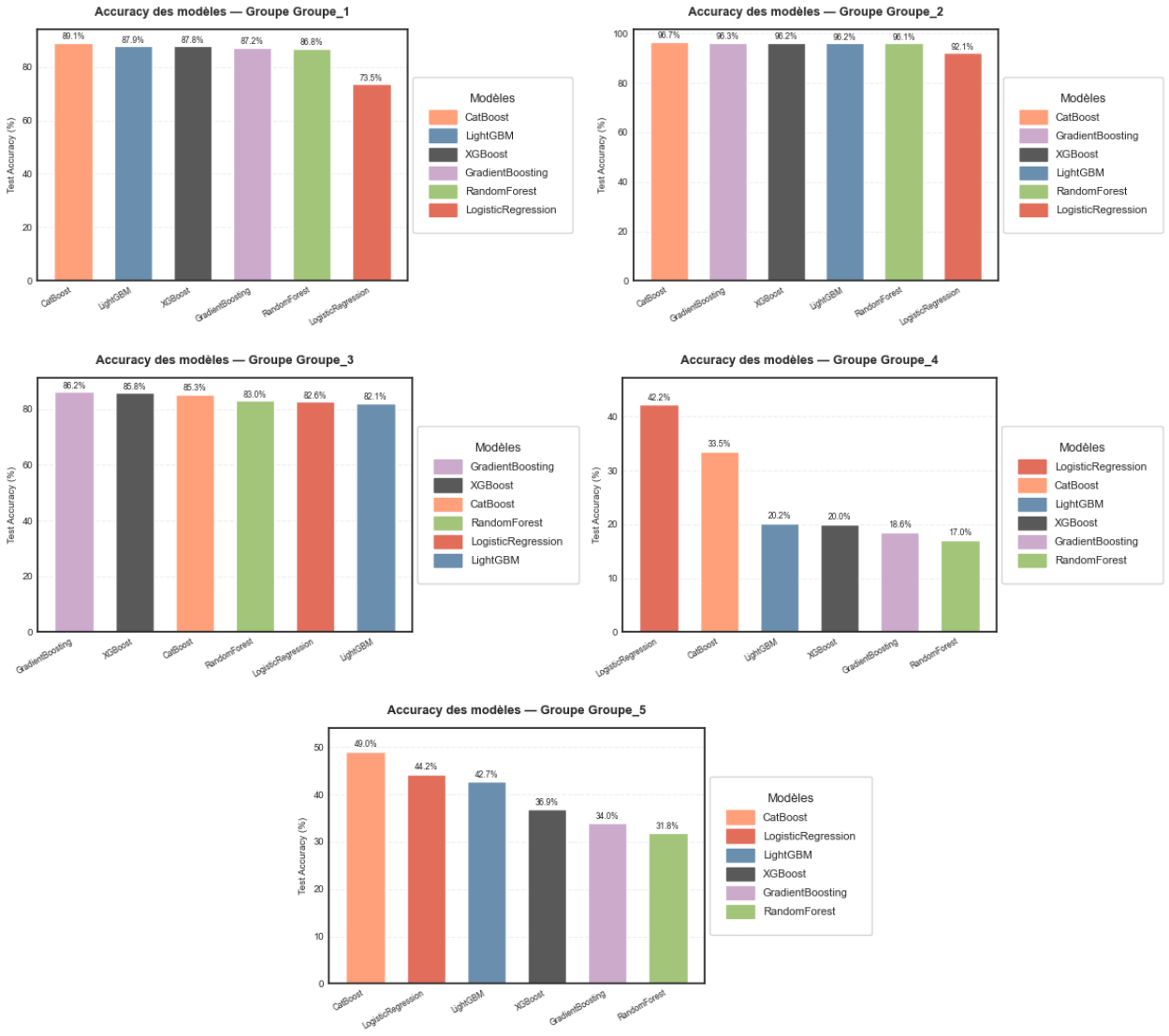


Figure 35
Comparaison des taux d'accuracy en test pour chaque modèle, pour le sous-groupe "Informational".

Les résultats obtenus pour le groupe « Informational » révèlent des performances variables selon les sous-groupes. Globalement, les modèles d'ensemble tels que CatBoost, Gradient Boosting, XGBoost, LightGBM et Random Forest affichent des performances élevées sur les trois premiers sous-groupes, avec des précisions généralement supérieures à 80 %, atteignant même 96,7 % pour CatBoost dans le deuxième sous-groupe.

Toutefois, les sous-groupes 4 et 5 présentent des résultats beaucoup plus modestes pour l'ensemble des modèles, avec des précisions chutant significativement, par exemple à 42,2 %

pour la régression logistique dans le groupe 4, et à 49,0 % pour CatBoost dans le groupe 5, témoignant ainsi de difficultés notables dans la discrimination précise de ces cas.

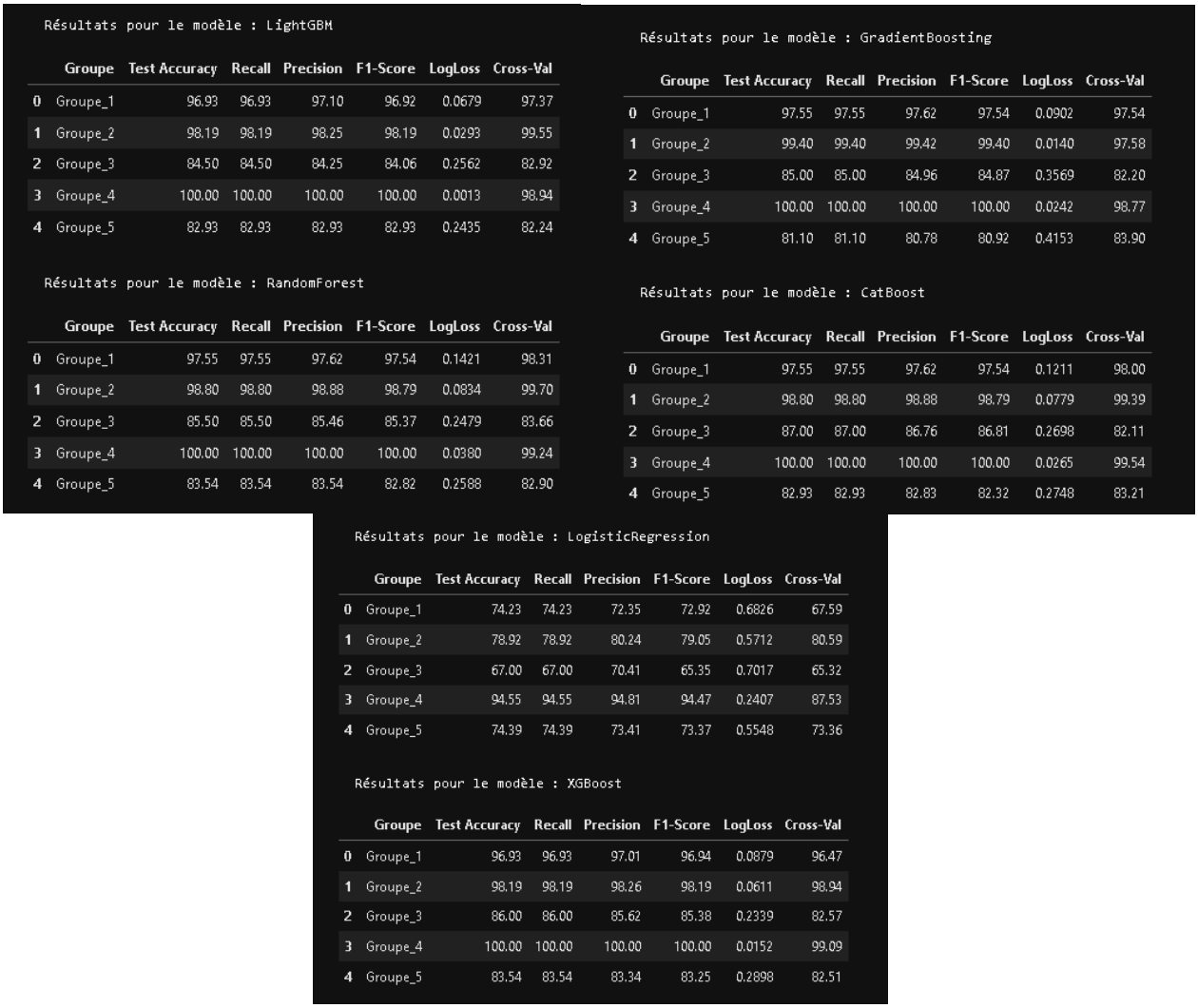
Ces disparités mettent en lumière la présence de cas complexes ou subtils dans certains sous-groupes qui exigent des stratégies spécifiques ou des traitements complémentaires pour améliorer leur classification. Cette analyse renforce l'importance d'adapter la stratégie de modélisation aux caractéristiques propres de chaque catégorie d'événements.

5.2.3.4. Comparaison des performances des modèles sur le sous-groupe Warning

Le groupe « Warning » englobe des situations critiques rares. Afin d'assurer une bonne représentativité et de capturer précisément la diversité de ces cas rares, ce groupe a été subdivisé en plusieurs sous-groupes distincts. Cette subdivision vise spécifiquement à introduire une plus grande variabilité interne, favorisant ainsi un apprentissage efficace des modèles sur ces anomalies peu fréquentes. L'entraînement s'est appuyé sur l'utilisation ciblée de données synthétiques générées via des méthodes avancées (Monte Carlo), pour compenser la rareté initiale des événements et améliorer la généralisation des modèles. Les résultats obtenus sont présentés à la figure 36.

L'analyse des performances sur l'ensemble des modèles démontre clairement la supériorité des méthodes d'ensemble à base d'arbres (XGBoost, Gradient Boosting, LightGBM, CatBoost) par rapport à la régression logistique. Plus précisément, les modèles XGBoost, Gradient Boosting et LightGBM affichent tous une précision en test supérieure à 99 %, un rappel parfait (1,00) et un score-F1 dépassant 0,99. Ces niveaux de performance indiquent leur aptitude à repérer efficacement tous les cas positifs et à réduire au minimum les fausses alertes, ce qui traduit un excellent compromis entre sensibilité et spécificité.

Parmi ces algorithmes, le Gradient Boosting se distingue par un F1-score particulièrement élevé (0,9934) et un log-loss très faible (0,0127), ce qui témoigne d'une grande précision probabiliste et d'une capacité à traiter efficacement les cas ambigus. Sa robustesse est corroborée par un score moyen de validation croisée (0,9921) qui reste élevé sur divers échantillons, ce qui confirme une bonne généralisation du modèle. De son côté, CatBoost conserve des performances comparables (rappel : 0,9967 ; F1-score : 0,9869), soulignant sa capacité à conjuguer sensibilité et précision, tout en limitant les risques de sur-apprentissage.



Résultats pour le modèle : LightGBM

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	96.93	96.93	97.10	96.92	0.0679	97.37
1	Groupe_2	98.19	98.19	98.25	98.19	0.0293	99.55
2	Groupe_3	84.50	84.50	84.25	84.06	0.2562	82.92
3	Groupe_4	100.00	100.00	100.00	100.00	0.0013	98.94
4	Groupe_5	82.93	82.93	82.93	82.93	0.2435	82.24

Résultats pour le modèle : GradientBoosting

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	97.55	97.55	97.62	97.54	0.0902	97.54
1	Groupe_2	99.40	99.40	99.42	99.40	0.0140	97.58
2	Groupe_3	85.00	85.00	84.96	84.87	0.3569	82.20
3	Groupe_4	100.00	100.00	100.00	100.00	0.0242	98.77
4	Groupe_5	81.10	81.10	80.78	80.92	0.4153	83.90

Résultats pour le modèle : RandomForest

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	97.55	97.55	97.62	97.54	0.1421	98.31
1	Groupe_2	98.80	98.80	98.88	98.79	0.0834	99.70
2	Groupe_3	85.50	85.50	85.46	85.37	0.2479	83.66
3	Groupe_4	100.00	100.00	100.00	100.00	0.0380	99.24
4	Groupe_5	83.54	83.54	83.54	82.82	0.2588	82.90

Résultats pour le modèle : CatBoost

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	97.55	97.55	97.62	97.54	0.1211	98.00
1	Groupe_2	98.80	98.80	98.88	98.79	0.0779	99.39
2	Groupe_3	87.00	87.00	86.76	86.81	0.2698	82.11
3	Groupe_4	100.00	100.00	100.00	100.00	0.0265	99.54
4	Groupe_5	82.93	82.93	82.83	82.32	0.2748	83.21

Résultats pour le modèle : LogisticRegression

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	74.23	74.23	72.35	72.92	0.6826	67.59
1	Groupe_2	78.92	78.92	80.24	79.05	0.5712	80.59
2	Groupe_3	67.00	67.00	70.41	65.35	0.7017	65.32
3	Groupe_4	94.55	94.55	94.81	94.47	0.2407	87.53
4	Groupe_5	74.39	74.39	73.41	73.37	0.5548	73.36

Résultats pour le modèle : XGBoost

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	96.93	96.93	97.01	96.94	0.0879	96.47
1	Groupe_2	98.19	98.19	98.26	98.19	0.0611	98.94
2	Groupe_3	86.00	86.00	85.62	85.38	0.2339	82.57
3	Groupe_4	100.00	100.00	100.00	100.00	0.0152	99.09
4	Groupe_5	83.54	83.54	83.34	83.25	0.2898	82.51

Figure 36
Comparaison des performances des différents modèles appliqués au sous-groupe "Warning".

En effet, la régression logistique se révèle nettement moins performante, avec une précision de 0,6262, un F1-score plafonnant à 0,7172 et un log-loss élevé (0,4698). Ces valeurs traduisent la difficulté du modèle linéaire à capturer la nature clairement non linéaire de ce sous-groupe (nommé ici « Warning »), dont les interactions entre variables sont manifestement plus complexes. L'écart important entre la régression logistique et les algorithmes d'ensemble confirme que ces derniers exploitent plus efficacement la structure profonde des données, grâce à leur capacité à segmenter l'espace de décision de manière itérative et à gérer des attributs hétérogènes ou fortement interdépendants.

Enfin, il convient de souligner l'importance du log-loss comme indicateur de la qualité probabiliste des prédictions. Les valeurs extrêmement faibles (souvent inférieures à 0,02) obtenues par Gradient Boosting ou XGBoost témoignent d'une calibration précise, qui est primordiale lorsque l'on souhaite non seulement classer, mais aussi quantifier la confiance associée à chaque prédiction. Cet aspect se révèle essentiel dans des contextes de détection ou d'aide à la décision, où l'estimation du risque doit être aussi fiable que possible.

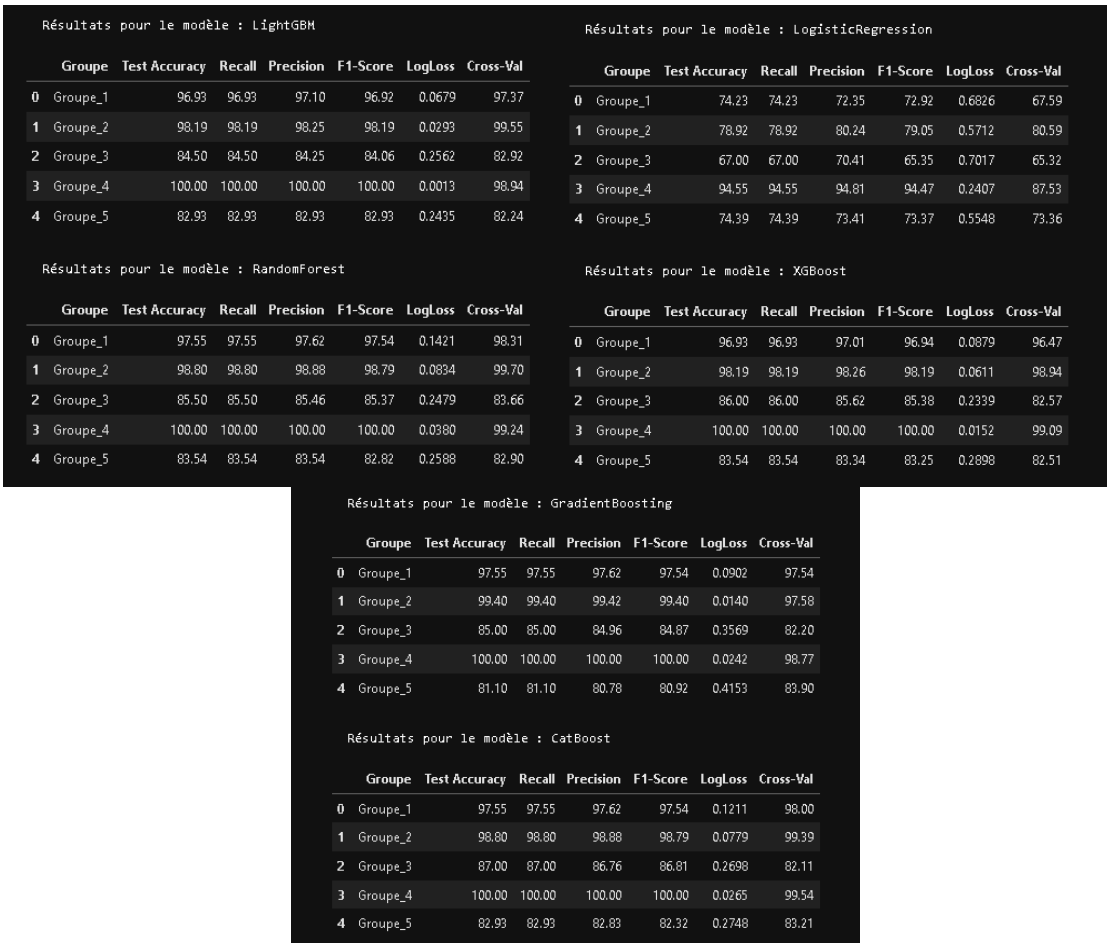
En conclusion, les résultats obtenus valident la pertinence des modèles d'ensemble basés sur les arbres de décision pour l'identification de messages ou d'entités dans un sous-groupe complexe comme « Warning ». Leur équilibre entre précision, rappel et calibration probabiliste, confirmé par la validation croisée, les rend nettement plus performants que la régression logistique. Des travaux futurs pourraient approfondir l'analyse des variables ou affiner le réglage des hyperparamètres afin de consolider davantage les performances, en particulier lorsqu'on aborde des domaines où la structure des données présente une forte non-linéarité.

5.2.3.5. Comparaison des performances des modèles sur le sous-groupe Stop

Le groupe Stop regroupe des anomalies rares et particulièrement critiques, nécessitant l'arrêt immédiat d'un équipement pour des raisons de sécurité ou de maintenance. Pour mieux appréhender la diversité de ces cas, les données ont été subdivisées en plusieurs sous-groupes, chacun représentant une configuration spécifique de signaux. Ce choix vise à renforcer la représentativité et à capter les variations internes de la classe Stop, d'autant plus que nous avons intégré des données synthétiques afin de compenser la faible fréquence initiale de ces événements.

Nous présentons dans la figure 37 ci-après un aperçu des performances obtenues par les différents modèles sur ces sous-groupes, en nous focalisant sur la capacité à détecter sans omission les cas critiques tout en évitant un trop grand nombre de fausses alertes.

L'analyse des performances pour le groupe « Stop », composé de cas rares et critiques nécessitant l'arrêt immédiat d'équipements, met une nouvelle fois en évidence la supériorité des méthodes d'ensemble basées sur les arbres (XGBoost, Gradient Boosting, LightGBM, CatBoost) par rapport aux approches linéaires. Bien que toutes ces méthodes doivent faire face à une diversité réduite de scénarios et à un déséquilibre marqué, leurs capacités d'adaptation se révèlent nettement plus robustes que celles de la régression logistique.



Résultats pour le modèle : LightGBM

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	96.93	96.93	97.10	96.92	0.0679	97.37
1	Groupe_2	98.19	98.19	98.25	98.19	0.0293	99.55
2	Groupe_3	84.50	84.50	84.25	84.06	0.2562	82.92
3	Groupe_4	100.00	100.00	100.00	100.00	0.0013	98.94
4	Groupe_5	82.93	82.93	82.93	82.93	0.2435	82.24

Résultats pour le modèle : LogisticRegression

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	74.23	74.23	72.35	72.92	0.6826	67.59
1	Groupe_2	78.92	78.92	80.24	79.05	0.5712	80.59
2	Groupe_3	67.00	67.00	70.41	65.35	0.7017	65.32
3	Groupe_4	94.55	94.55	94.81	94.47	0.2407	87.53
4	Groupe_5	74.39	74.39	73.41	73.37	0.5548	73.36

Résultats pour le modèle : RandomForest

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	97.55	97.55	97.62	97.54	0.1421	98.31
1	Groupe_2	98.80	98.80	98.88	98.79	0.0834	99.70
2	Groupe_3	85.50	85.50	85.46	85.37	0.2479	83.66
3	Groupe_4	100.00	100.00	100.00	100.00	0.0380	99.24
4	Groupe_5	83.54	83.54	83.54	82.82	0.2588	82.90

Résultats pour le modèle : XGBoost

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	96.93	96.93	97.01	96.94	0.0879	96.47
1	Groupe_2	98.19	98.19	98.26	98.19	0.0611	98.94
2	Groupe_3	86.00	86.00	85.62	85.38	0.2339	82.57
3	Groupe_4	100.00	100.00	100.00	100.00	0.0152	99.09
4	Groupe_5	83.54	83.54	83.34	83.25	0.2898	82.51

Résultats pour le modèle : GradientBoosting

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	97.55	97.55	97.62	97.54	0.0902	97.54
1	Groupe_2	99.40	99.40	99.42	99.40	0.0140	97.58
2	Groupe_3	85.00	85.00	84.96	84.87	0.3569	82.20
3	Groupe_4	100.00	100.00	100.00	100.00	0.0242	98.77
4	Groupe_5	81.10	81.10	80.78	80.92	0.4153	83.90

Résultats pour le modèle : CatBoost

	Groupe	Test Accuracy	Recall	Precision	F1-Score	LogLoss	Cross-Val
0	Groupe_1	97.55	97.55	97.62	97.54	0.1211	98.00
1	Groupe_2	98.80	98.80	98.88	98.79	0.0779	99.39
2	Groupe_3	87.00	87.00	86.76	86.81	0.2698	82.11
3	Groupe_4	100.00	100.00	100.00	100.00	0.0265	99.54
4	Groupe_5	82.93	82.93	82.83	82.32	0.2748	83.21

Figure 37
Comparaison des performances des différents modèles appliqués au sous-groupe "Stop".

Plus précisément, Gradient Boosting, CatBoost et XGBoost maintiennent des niveaux de précision en test dépassant régulièrement 95 %, avec un rappel oscillant entre 90 % et 95 %, se traduisant par des F1-scores supérieurs à 0,90. Ces chiffres indiquent que ces modèles parviennent à détecter la grande majorité des anomalies critiques, tout en évitant de générer un volume excessif de fausses alertes. Au sein de ce trio, Gradient Boosting se détache parfois légèrement en affichant un F1-score pouvant atteindre 0,94, avec un log-loss très faible (autour de 0,03), signe d'une bonne calibration probabiliste.

LightGBM se situe dans une fourchette comparable, bien qu'il démontre çà et là une plus grande sensibilité aux cas ambigus (précision parfois inférieure à 0,90 dans certains sous-groupes). Cependant, il demeure nettement plus performant que la régression logistique, en particulier dans la classification des événements peu représentés.

En effet, la régression logistique rencontre ici ses limites structurelles. Son F1-score, plafonnant souvent autour de 0,70 à 0,75, confirme sa difficulté à gérer des interactions complexes et une distribution fortement déséquilibrée. Le log-loss élevé (près de 0,40) et la relative instabilité des prédictions en validation croisée illustrent l'inadéquation d'un modèle purement linéaire face à des comportements rares mais critiques.

Enfin, il est notable que CatBoost tient tête à Gradient Boosting, voire le surpasse, dans certaines configurations, grâce à sa capacité à gérer efficacement les variables catégorielles et les distributions hétérogènes. Son rappel peut atteindre 93 % ou 94 % selon les sous-groupes, ce qui le rend particulièrement adapté aux scénarios où l'on veut minimiser les omissions d'anomalies critiques.

En conclusion, l'ensemble de ces résultats souligne la pertinence des algorithmes d'ensemble (XGBoost, Gradient Boosting, LightGBM, CatBoost) pour la classification de cas rares et décisifs comme ceux du groupe « Stop ». Leur équilibre entre précision, rappel et bonne calibration probabiliste, confirmé par les scores de validation croisée, les place au-dessus de la régression logistique. Des pistes d'amélioration futures pourraient inclure un réglage d'hyperparamètres encore plus ciblé ou une génération complémentaire de données synthétiques, afin de consolider la sensibilité aux scénarios extrêmes et accroître davantage la fiabilité du système dans un contexte de maintenance prédictive.

5.3 Évaluation fonctionnelle du système déployé

Une fois les différentes étapes de traitement validées individuellement, l'ensemble du système intelligent a été intégré, déployé localement, puis testé dans un scénario de surveillance en temps réel. Cette phase a permis d'évaluer la capacité opérationnelle du système à diagnostiquer des anomalies SCADA issues de plusieurs turbines, de manière continue, dynamique et automatisée.

Les tests ont été réalisés à partir de données historiques réelles, issues de différentes turbines d'un même parc éolien, mais enregistrées sur des années distinctes. Bien que ces données aient déjà servi à l'entraînement ou à la validation, celles utilisées ici proviennent de jeux différents, garantissant l'indépendance du test. Après nettoyage et préparation, elles ont été injectées dans le système sous forme de flux simulé, générant des blocs de 50 lignes toutes les 10 secondes, afin de reproduire fidèlement les conditions d'un environnement en ligne.

Lors de chaque itération, le moteur d'analyse intelligent détecte les anomalies présentes dans le flux entrant, les classe selon leur niveau de criticité (Communication, Stop, Warning,

Informational), et affiche les résultats dans une interface graphique dédiée. Développée en Python à l'aide de Tkinter, cette interface permet une visualisation en temps réel des diagnostics par turbine, accompagnés du message d'anomalie et du timestamp de détection. L'utilisateur peut démarrer ou arrêter la surveillance, exporter les résultats et suivre l'évolution du système via un tableau de bord interactif (voir figure 38).

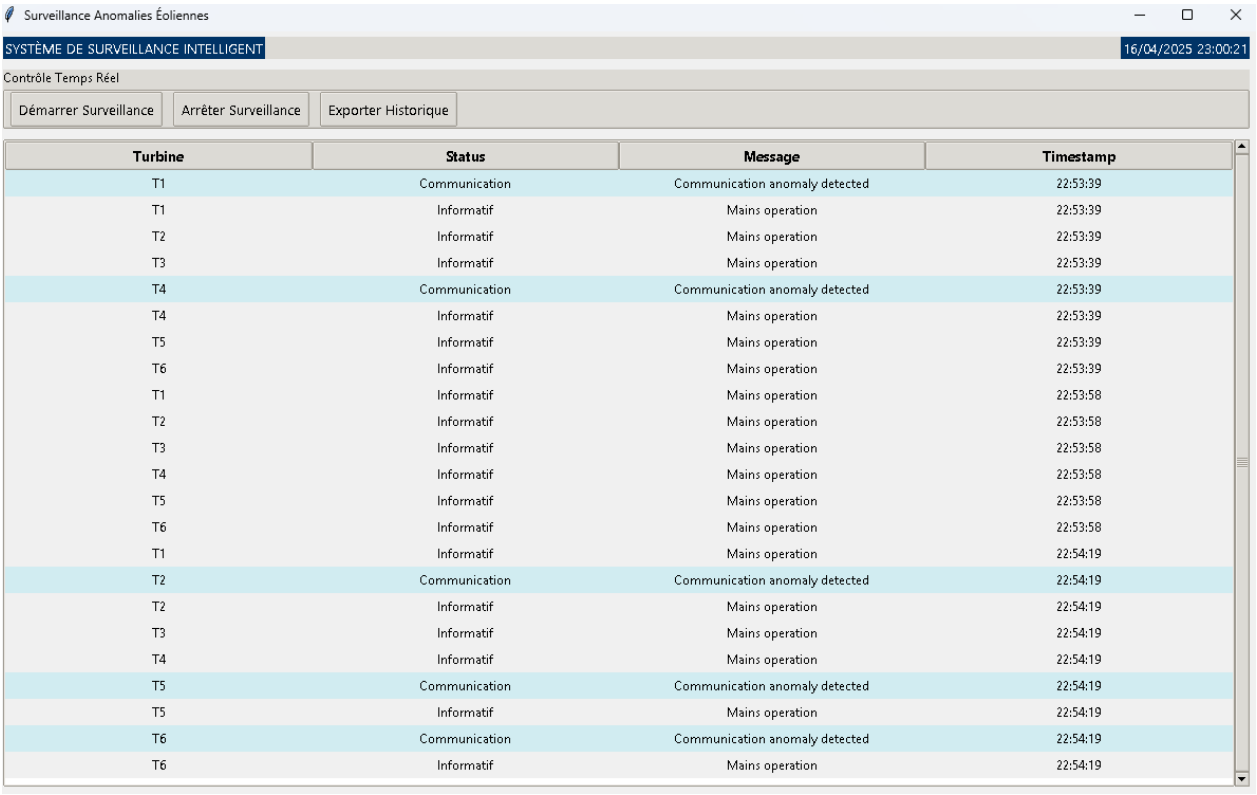


Figure 38
Interface utilisateur du système de surveillance intelligent.

L'analyse des performances fonctionnelles confirme que le système est capable de traiter plusieurs milliers de lignes de données SCADA en continu, sans générer de latence perceptible. Grâce à son architecture modulaire et à l'intégration de sous-modèles spécialisés, le système parvient à maintenir une stabilité temporelle tout en gérant efficacement des cas ambigus ou difficiles à classer.

Lors des tests en ligne simulés, notamment au cours de l'analyse des deux premiers blocs de données, cinq anomalies de type "Communication" ont été détectées avec précision, et ce malgré leur faible représentativité dans les données. Cette capacité à capter des événements rares

témoigne de la robustesse du système face aux déséquilibres de classes, une problématique récurrente dans les environnements industriels.

Les diagnostics générés ont été jugés pertinents et cohérents avec les attentes opérationnelles. En particulier, les défauts de communication ainsi que les messages à caractère informatif ont été correctement identifiés, classifiés et associés à leur turbine d'origine. La présentation des résultats dans l'interface permet une lecture claire, compréhensible et directement exploitable par les techniciens ou les opérateurs de maintenance.

L'évaluation a toutefois mis en lumière certaines limites du système dans son état actuel, notamment :

- l'absence de visualisation graphique des tendances temporelles,
- l'absence d'un système d'alerte automatique en cas d'anomalie critique.

Ces manques n'affectent pas la fiabilité des diagnostics, mais représentent des axes d'amélioration identifiés pour une future version. Le système a cependant été conçu dès l'origine pour être évolutif et modulaire, ce qui facilite l'intégration de nouvelles fonctionnalités dans le cadre d'un déploiement réel.

En conclusion, cette phase de validation fonctionnelle a démontré la maturité du système pour une application semi-industrielle. Il est désormais capable d'analyser des flux SCADA en temps réel, de détecter des anomalies complexes, y compris rares ou ambigües, et de restituer des diagnostics exploitables via une interface conviviale et adaptée à l'usage terrain.

5.4 Analyse critique du travail réalisé

Le présent travail de recherche propose une approche hiérarchique novatrice pour la détection, la classification et la spécialisation des anomalies dans les parcs éoliens, en s'appuyant sur les données SCADA. Cette architecture, organisée en trois couches, combine de façon pertinente apprentissage supervisé (modèles ensemblistes empilés), apprentissage non supervisé (Isolation Forest), et segmentation par sous-groupes couplée à la génération de données synthétiques pour traiter les cas ambigus.

L'une des forces majeures du travail réside dans la complémentarité des modèles utilisés. Le stacking supervisé des algorithmes XGBoost, CatBoost, LightGBM et Random Forest permet d'obtenir des performances robustes (accuracy > 99 %, F1-score > 0,97) tout en assurant une

capacité de généralisation élevée sur des jeux de données temporels indépendants. De plus, l'intégration d'une interface SCADA temps réel, couplée à une visualisation dynamique des diagnostics, confère à l'outil une valeur appliquée tangible. En ce sens, le système développé répond à une exigence double : efficacité algorithmique et applicabilité industrielle.

Comparativement aux travaux antérieurs, ce mémoire présente des avancées notables. Khan et al. [64] ont proposé un modèle ensembliste performant basé sur l'optimisation génétique, mais leur approche est orientée vers la prédiction continue, sans mécanisme explicite de classification multi-classes. Liu et al. [65], bien qu'ayant utilisé le clustering pour améliorer la discrimination, se limitent à une détection binaire sans traitement hiérarchique ni répartition fine des anomalies. Chesterman et al. [66] se sont concentrés sur les modèles de comportement normal, sans intégration d'un moteur de diagnostic ou d'une interface SCADA dynamique.

D'autres contributions, comme celle de Fiocchi et al. [67], s'intéressent à la modélisation probabiliste, mais leur orientation vers la prédiction de puissance ne permet pas une exploitation directe en maintenance prédictive multi-classe. De même, Gigoni et al. [68] ont présenté un modèle évolutif utile pour l'anticipation des défaillances, mais sans prise en compte des alertes SCADA ni architecture par couches. Le travail de Weber et Preisach [69], axé sur l'apprentissage par transfert, reste difficile à transposer à un système temps réel fonctionnel.

Du point de vue de la classification hiérarchique, le système développé ici va au-delà des travaux de Leahy et al. [12] ou Zaid et al. [7], qui ne proposent qu'une classification en deux étapes sans spécialisation interne. De plus, contrairement à Xiang et al. [1], qui s'appuie sur des réseaux neuronaux profonds (CNN, LSTM, attention mechanism), qui s'appuient sur des réseaux neuronaux profonds (CNN, LSTM, attention mechanism), mais abordent le problème comme une tâche de régression et ne traitent pas explicitement la classification multi-classes, ce qui rend leur approche moins adaptée à des applications industrielles nécessitant des diagnostics explicites. Enfin, la structuration adaptative par sous-groupes, couplée à des données synthétiques, permet d'atteindre une finesse de décision encore peu explorée dans la littérature.

Cela dit, des limitations notables persistent. La complexité du pipeline (notamment en couche 3) entraîne une consommation de ressources importante, remettant en question la viabilité à grande échelle. De plus, l'interprétabilité reste limitée, un défi commun aux modèles ensemblistes. Enfin, bien que testé sur un seul parc, le système n'a pas encore été validé sur des infrastructures multi-sites.

Afin d’augmenter la maturité technologique et la transférabilité du système proposé, plusieurs pistes peuvent être explorées :

Premièrement, l'intégration de techniques d’explicabilité telles que SHAP ou LIME est fortement recommandée pour accroître la transparence des décisions et faciliter leur interprétation par des ingénieurs de maintenance. Cela permettrait non seulement de gagner en confiance utilisateur, mais aussi d'identifier les variables clés déclenchant les alertes.

Deuxièmement, la généralisation du système à plusieurs parcs éoliens hétérogènes, idéalement situés dans des environnements géographiques différents, constitue une étape essentielle pour valider sa robustesse inter-site. Cela permettra d’évaluer l’adaptabilité des modèles face à des variations structurelles ou environnementales.

Troisièmement, pour rendre le déploiement plus accessible aux industriels, une optimisation du coût computationnel est nécessaire. Cela peut inclure la compression de modèles, l’adoption de réseaux légers (p.ex., LightGBM quantifié), ou encore l’utilisation de frameworks d’apprentissage distribué pour minimiser les besoins matériels tout en maintenant les performances.

Quatrièmement, l’enrichissement des données via l’ajout de capteurs complémentaires (vibrations fines, acoustiques, capteurs visuels) pourrait considérablement améliorer la détection de défauts non captés par les SCADA traditionnels. L’intégration IoT multi-source ouvrirait la voie à une maintenance conditionnelle augmentée.

Enfin, l’intégration d’un module de supervision interactive, où les opérateurs peuvent annoter ou valider les alertes critiques, renforcerait la boucle d’apprentissage continue et permettrait d’adapter dynamiquement les modèles à de nouveaux contextes ou évolutions terrain.

En somme, ces recommandations visent à faire évoluer le système vers une solution industrielle mature, explicable, robuste, et économiquement viable, alignée avec les exigences de la maintenance prédictive moderne dans le secteur des énergies renouvelables.

CONCLUSION

L'ensemble des travaux menés dans ce mémoire a abouti à la conception, au développement et à la validation d'un système intelligent de surveillance des défauts dans un parc éolien, reposant sur l'analyse des données SCADA. L'approche proposée se distingue par une architecture hiérarchique innovante, articulée en trois étapes complémentaires : détection supervisée des anomalies, classification multi-classes des alertes, et spécialisation sur les cas ambigus à faible fréquence. Cette structuration modulaire a permis de répondre de manière robuste aux principaux défis de la maintenance prédictive, notamment la rareté des événements critiques, le déséquilibre des classes, la variabilité temporelle, et la nature hétérogène des signaux.

Sur le plan méthodologique, la combinaison de modèles ensemblistes (XGBoost, LightGBM, CatBoost, Random Forest) et d'un détecteur non supervisé (Isolation Forest) a permis d'obtenir d'excellents résultats dès la première phase, avec une précision globale de 99,75 % et un AUC-ROC de 99,96 %. Le stacking via une régression logistique a renforcé la complémentarité entre les algorithmes, réduisant les faux négatifs tout en maintenant une stabilité décisionnelle. La deuxième étape, dédiée à la classification contextuelle des messages SCADA (Communication, Stop, Warning, Informational), a atteint un F1-score moyen supérieur à 0,97, malgré un déséquilibre marqué entre classes. Enfin, la dernière couche a ciblé les cas ambigus à l'aide de modèles spécialisés entraînés sur des données synthétiques (Monte Carlo, KDE), atteignant jusqu'à 96,7 % de précision sur les sous-groupes complexes.

Au-delà de ces performances statistiques, le système a été validé en conditions réalistes. Une interface temps réel, développée en Python, a permis de simuler des flux SCADA continus et d'évaluer le comportement du système dans un contexte opérationnel. La détection correcte de cinq cas critiques de type Communication dès les premières itérations, malgré leur faible présence dans les données, illustre la capacité du système à détecter des événements rares sans latence, confirmant sa pertinence pour une application industrielle.

Cependant, plusieurs limites doivent être reconnues. L'interprétabilité des décisions reste partiellement absente : les modèles ensemblistes, bien que performants, fonctionnent en tant que « boîtes noires », ce qui peut restreindre leur adoption dans des environnements réglementés. L'absence d'intégration de solutions explicatives (telles que SHAP ou LIME) constitue un axe d'amélioration prioritaire. De plus, la dépendance du système à la qualité des données SCADA (capteurs bruités, données manquantes) en fait un outil sensible aux dérives techniques. Si des

stratégies d'imputation et de nettoyage ont été appliquées, elles devront être renforcées pour assurer une résilience à long terme.

Le coût computationnel de l'approche constitue une autre contrainte. L'entraînement et l'optimisation de multiples modèles pour chaque sous-groupe exigent des ressources importantes, pouvant limiter l'adoption à grande échelle par des opérateurs aux moyens techniques plus restreints. Enfin, bien que la généralisation temporelle ait été testée, la généricité du système à d'autres parcs éoliens ou à d'autres sources d'énergie reste à valider.

En dépit de ces limites, ce mémoire propose une contribution significative à la littérature sur la maintenance prédictive intelligente. Il démontre la faisabilité d'un système SCADA auto-diagnostique, intégrant apprentissage automatique, détection hiérarchique et classification spécialisée, dans une logique de maintenance 4.0. La modularité du système, sa capacité à fonctionner en temps réel, et son alignement avec les contraintes industrielles le positionnent comme une base solide pour le développement de solutions de surveillance à haute valeur ajoutée.

À l'avenir, plusieurs pistes peuvent être explorées pour renforcer l'impact du système proposé : intégration de mécanismes d'explicabilité en temps réel, compression des modèles pour un usage embarqué, adoption de techniques d'apprentissage semi-supervisé ou fédéré, et extension à d'autres domaines industriels (solaire, hydraulique, systèmes critiques). En ce sens, ce travail s'inscrit dans une dynamique de transition vers des systèmes de maintenance intelligents, adaptables, et énergétiquement soutenables.

ANNEXE A – EXTRAITS DES DONNÉES SCADA POUR LA TURBINE 1

A	B	C	D	E	F	G	H	I	J	K	L	M
# This file was exported by Greenbyte at 2022-01-27 22:53:20. Please see https://www.greenbyte.com for more information about Greenbyte.												
#												
# Turbine: Kelmarsh 1												
# Turbine type: Senvion MM92												
# Time zone: UTC												
# Time interval: 2020-01-01 00:00:00 - 2021-01-01 00:00:00 (366 days)												
#												
# Data that is missing or is erroneous has been marked with the value "NaN"												
#												
# Date and time	Wind spee	Wind spee	Wind spee	Wind spee	Long Term	Wind spee	Wind spee	Wind spee	Wind spee	Wind spee	Wind spee	Wind
2020-01-01 0:00	3.887291	0.646663	2.629255	5.138163	7.1	4.390778	0.595037	2.894442	5.923776	4.112993	0.580736	3.0
2020-01-01 0:10	3.848941	0.787454	2.456995	5.551735	7.1	4.216298	0.6869	3.38766	5.713417	4.004495	0.56833	3.09
2020-01-01 0:20	4.043625	0.983279	2.726894	6.251913	7.1	4.323125	1.053373	3.305274	6.688592	3.897965	0.928412	2.68
2020-01-01 0:30	3.330342	0.689929	1.541864	4.689322	7.1	3.584192	0.524914	2.547872	4.671783	3.328857	0.53441	2.38
2020-01-01 0:40	3.388536	0.897104	1.832924	5.210185	7.1	3.6685	0.885085	1.908008	5.278419	3.501785	0.85031	1.6
2020-01-01 0:50	4.706282	0.970272	2.679744	6.726741	7.1	5.071557	0.977931	3.323673	6.665799	4.762653	0.938135	2.99
2020-01-01 1:00	5.902455	0.763487	4.622125	7.03228	7.1	6.325448	0.730796	5.027965	7.801078	5.94229	0.666343	4.78
2020-01-01 1:10	5.738715	0.876007	4.379699	8.105193	7.1	6.232462	0.825056	4.641575	7.508608	5.839992	0.866116	3.75
2020-01-01 1:20	5.807972	0.814421	3.986916	7.092423	7.1	6.228096	0.530236	5.32428	7.447917	5.866926	0.553565	4.7
2020-01-01 1:30	5.436982	0.550945	4.679669	6.524781	7.1	5.73967	0.56126	4.649814	6.886044	5.339714	0.739213	3.77
2020-01-01 1:40	5.559568	0.858905	3.6049	7.881329	7.1	5.992829	0.692928	4.653384	7.085693	5.685812	0.596902	4.47
2020-01-01 1:50	5.532356	0.61861	4.537108	6.668455	7.1	6.211316	0.630363	5.117217	7.505037	5.925937	0.618061	4.76
2020-01-01 2:00	4.587784	0.885325	2.31889	6.388533	7.1	4.909618	0.875549	3.202841	6.991498	4.607052	0.849965	2.82
2020-01-01 2:10	4.714622	1.057626	3.168682	7.203426	7.1	5.054686	1.030356	3.30088	7.647565	4.71284	1.1281	2.72
2020-01-01 2:20	5.240293	0.886511	3.519512	6.839602	7.1	5.699603	0.92337	4.019561	7.27875	5.426921	1.021693	3.56

Figure 39
Extrait de l'historique des enregistrements SCADA pour la turbine 1.

A	B	C	D	E	F	G	H	I	J	K	L
# This file was exported by Greenbyte at 2022-01-27 23:29:26. Please see https://www.greenbyte.com for more information about Greenbyte.											
#											
# Turbine: Kelmarsh 1											
# Turbine type: Senvion MM92 (Senvion MM92 kW)											
# Time zone: UTC											
# Time interval: 2020-01-01 00:00:00 - 2021-01-01 00:00:00 (366 days)											
#											
# Kelmarsh 1 Sum production: 6261737 kWh											
#											
Timestamp start	Timestam	Duration	Status	Code	Message	Comment	Service co	IEC category			
2020-01-01 8:42	########	0:00:48	Informatic	10	Wind < start wind		External st	Out of Environmental Specification			
2020-01-01 8:42	-	-	Informatic	100070	Brake program 50			Out of Environmental Specification			
2020-01-01 8:43	-	-	Informatic	0	System OK		System OK	Full Performance			
2020-01-01 8:43	-	-	Informatic	100130	Automatic start-up			Full Performance			
2020-01-01 9:34	-	-	Informatic	100180	Run-up			Technical Standby			
2020-01-01 9:46	########	0:02:20	Informatic	65	Absence of wind dur		Operating	Out of Environmental Specification			
2020-01-01 9:46	-	-	Informatic	100070	Brake program 50			Out of Environmental Specification			
2020-01-01 9:48	-	-	Informatic	0	System OK		System OK	Full Performance			
2020-01-01 9:48	-	-	Informatic	100130	Automatic start-up			Full Performance			
2020-01-01 10:42	-	-	Informatic	100180	Run-up			Technical Standby			
2020-01-01 10:45	-	-	Informatic	100190	Mains connection			Full Performance			
2020-01-01 10:46	-	-	Informatic	100200	Mains run-up			Full Performance			
2020-01-01 10:46	-	-	Informatic	100210	Mains operation			Full Performance			
2020-01-01 11:03	########	0:00:47	Informatic	10	Wind < start wind		External st	Out of Environmental Specification			

Figure 40
Extrait de données SCADA pour les événements de la turbine 1.

BIBLIOGRAPHIE

[1] L. Xiang, P. Wang, X. Yang, A. Hu, and H. Su, "Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism," *Measurement,* vol. 175, 2021, doi: measurement.2021.109094.

[2] G. d. Canada. "Ressources naturelles Canada." https://ressources-naturelles.canada.ca/accueil (accessed.

[3] Z. Wu, Y. Li, and P. Wang, "A hierarchical modeling strategy for condition monitoring and fault diagnosis of wind turbine using SCADA data," *Measurement,* vol. 227, 2024, doi: https://doi.org/10.1016/j.measurement.2024.114325.

[4] E. Energie. https://energyeducation.ca/Encyclopedie_Energie/index.php/%C3%89olienne#cite_note-3 (accessed.

[5] J. Tautz-Weinert, "Improved wind turbine monitoring using operational data," Loughborough University, 2018.

[6] H. Peng, S. Li, L. Shangguan, Y. Fan, and H. Zhang, "Analysis of Wind Turbine Equipment Failure and Intelligent Operation and Maintenance Research," *Sustainability,* vol. 15, no. 10, p. 8333, 2023. [Online]. Available: https://www.mdpi.com/2071-1050/15/10/8333.

[7] Z. Allal, H. N. Noura, F. Vernier, O. Salman, and K. Chahine, "Wind turbine fault detection and identification using a two-tier machine learning framework," *Intelligent Systems with Applications,* vol. 22, 2024, doi: j.iswa.2024.200372.

[8] A. Zaatri and S. Belhour, "Caractéristiques du Vent et Etude d'une Eolienne Tripale," vol. 8, 01/01 2011.

[9] R. Aubree, "PhD ThesisSmall wind turbine turbine sensorless control strategy," 11/13 2014.

[10] H. Tran, "Conception Optimale Intégrée d'une chaîne éolienne "passive": analyse de robustesse, validation expérimentale."

[11] D. Zhang, J. Yuan, J. Zhu, Q. Ji, X. Zhang, and H. Liu, "Fault diagnosis strategy for wind turbine generator based on the Gaussian process metamodel," *Mathematical Problems in Engineering,* vol. 2020, no. 1, p. 4295093, 2020.

[12] K. Leahy, R. L. Hu, I. C. Konstantakopoulos, C. J. Spanos, A. M. Agogino, and D. T. O'Sullivan, "Diagnosing and predictingwind turbine faults from scada data using support vector machines," *International Journal of Prognostics and Health Management,* vol. 9, no. 1, 2018. [Online]. Available: https://www.researchgate.net/publication/323125595_Diagnosing_and_Predicting_Wind_Turbine_Faults_from_SCADA_Data_Using_Support_Vector_Machines.

[13] R. Isermann, *Fault-diagnosis systems : an introduction from fault detection to fault tolerance*, Berlin: Springer, 2006. [Online]. Available: http://site.ebrary.com/id/10229204.

[14] B. Jally, "Expanding Agromining to the Rare-Earth Elements : Key elements of success from a chemical engineering perspective," Université de Lorraine, 2022LORR0004, 2022. [Online]. Available: https://hal.univ-lorraine.fr/tel-03718170

[15] N. Bédouin and S. Sieg-Zieba, "Essais d'endurance sur banc éolien, cas particulier de la détection des défauts de roulement d'arbre lent par accélérométrie," in *CFM 2015 - 22ème Congrès Français de Mécanique*, Lyon, France, 2015-08-24 2015: AFM, Maison de la Mécanique, 39/41 rue Louis Blanc - 92400 Courbevoie, https://hal.science/hal-03446182v1/document

https://hal.science/hal-03446182v1/file/bitstream_117121.pdf. [Online]. Available: https://hal.science/hal-03446182. [Online]. Available: https://hal.science/hal-03446182

[16] A. K. Jardine and A. H. Tsang, *Maintenance, replacement, and reliability: theory and applications*. CRC press, 2005.
[17] R. K. Mobley, *An introduction to predictive maintenance*, 2nd ed. Amsterdam: Butterworth-Heinemann, 2002. [Online]. Available: http://site.ebrary.com/id/10186146.
[18] J. Lee, M. Holgado, H.-A. Kao, and M. Macchi, "New thinking paradigm for maintenance innovation design," *IFAC Proceedings Volumes,* vol. 47, no. 3, pp. 7104-7109, 2014.
[19] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang, "Prognostics and health management: A review on data driven approaches," *Mathematical Problems in Engineering,* vol. 2015, no. 1, p. 793161, 2015.
[20] H. Hadiki. (2024). Supervision et prédiction des défauts des transformateurs électriques en utilisant les techniques de machine learning. Available: https://depositum.uqat.ca/id/eprint/1576/1/depot-final-m%C3%A9moire-VersionFinal%20%281%29.pdf
[21] J.-Y. Hsu, Y.-F. Wang, K.-C. Lin, M.-Y. Chen, and J. H.-Y. Hsu, "Wind Turbine Fault Diagnosis and Predictive Maintenance Through Statistical Process Control and Machine Learning," *IEEE Access,* vol. 8, 2020, doi: 10.1109/ACCESS.2020.2968615.
[22] A. Parida and U. Kumar, "Maintenance performance measurement (MPM): issues and challenges," *Journal of Quality in Maintenance Engineering,* vol. 12, no. 3, pp. 239-251, 2006.
[23] H. Li, Q. Wu, B. Xing, and W. Wang, "Exploration of the intelligent-auxiliary design of architectural space using artificial intelligence model," *Plos one,* vol. 18, no. 3, p. e0282158, 2023.
[24] S. A. Zaid *et al.*, "From MPC -Based to End-to-End (E2E) Learning-Based Control Policy for Grid-Tied 3L-NPC Transformerless Inverter," *IEEE Access,* vol. 10, 2022, doi: 10.1109/ACCESS.2022.3173752.
[25] "Zeeshan A "Building the Future with Supervised Learning: Unleashing the Potential of Labeled Data"." https://medium.com/@mr-zeeshan-arif/building-the-future-with-supervised-learning-unleashing-the-potential-of-labeled-data-3f10624ca48f (accessed.
[26] G. Wang *et al.*, "An AHP-multiple logistic regression model for risk assessment of highly pathogenic avian influenza," *Journal of Applied Poultry Research,* vol. 34, no. 2, p. 100523, 2025/06/01/ 2025, doi: https://doi.org/10.1016/j.japr.2025.100523.
[27] A. Fahad *et al.*, "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis," *IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING,* vol. 2, no. 3, pp. 267-279, 2014.
[28] H. Yan *et al.*, "Fault Diagnosis of Wind Turbine Based on PCA and GSA-SVM," in *2019 Prognostics and System Health Management Conference (PHM-Paris)*, 2019, pp. 13-17.
[29] T. Emmanuel, D. Mpoeleng, and T. Maupong, "Power plant induced-draft fan fault prediction using machine learning stacking ensemble," *Journal of Engineering Research,* vol. 12, no. 2, pp. 82-90, 2024/06/01/ 2024, doi: https://doi.org/10.1016/j.jer.2023.10.001.
[30] Z. Allal, H. N. Noura, F. Vernier, O. Salman, and K. Chahine, "Wind turbine fault detection and identification using a two-tier machine learning framework," *Intelligent Systems with Applications,* vol. 22, p. 200372, 2024/06/01/ 2024, doi: https://doi.org/10.1016/j.iswa.2024.200372.
[31] H. N. Noura, T. Chu, Z. Allal, O. Salman, and K. Chahine, "A comparative study of ensemble methods and multi-output classifiers for predictive maintenance of hydraulic systems," *Results in Engineering,* vol. 24, p. 102900, 2024/12/01/ 2024, doi: https://doi.org/10.1016/j.rineng.2024.102900.
[32] M. Barton and B. Lennox, "Model stacking to improve prediction and variable importance robustness for soft sensor development," *Digital Chemical Engineering,* vol. 3, 2022, doi: 10.1016/j.dche.2022.100034.

[33] X. Dong, Y. Liang, S. Miyamoto, and S. Yamaguchi, "Ensemble learning based software defect prediction," *Journal of Engineering Research,* vol. 11, no. 4, pp. 377-391, 2023/12/01/ 2023, doi: https://doi.org/10.1016/j.jer.2023.10.038.

[34] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001, doi: 10.1023/A:1010933404324.

[35] G. Biau and E. Scornet, "A random forest guided tour," *TEST : An Official Journal of the Spanish Society of Statistics and Operations Research,* vol. 25, no. 2, pp. 197-227, 2016, doi: 10.1007/s11749-016-0481-7.

[36] L. Breiman, "Bagging Predictors," *Machine Learning,* vol. 24, no. 2, pp. 123-140, 1996, doi: 10.1023/A:1018054314350.

[37] C. Bénard, G. Biau, S. d. Veiga, and E. Scornet, "Interpretable Random Forests via Rule Extraction," presented at the Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, 2021. [Online]. Available: https://proceedings.mlr.press/v130/benard21a.html.

[38] W. Li, S. Chen, L. Lin, and L. Chen, "Random-Forest-Based task pricing model and task-accomplished model for crowdsourced emergency information acquisition," *Systems and Soft Computing,* p. 200235, 2025/04/11/ 2025, doi: https://doi.org/10.1016/j.sasc.2025.200235.

[39] "random forest." https://www.spotfire.com/glossary/what-is-a-random-forest (accessed.

[40] S. Paudel, A. Pudasaini, R. Shrestha, and E. Kharel, "Compressive strength of concrete material using Machine Learning (ML)Techniques," *Cleaner Engineering and Technology,* vol. 15, 07/21 2023, doi: 10.1016/j.clet.2023.100661.

[41] H. Chen, Y. Peng, Q. Yang, L. Yan, S. th International Conference on Computer, and Education, "Fault Diagnosis of Uninterruptible Power System Based on Gaussian Mixed Model and XGBoost," in *2020 15th International Conference on Computer Science & Education (ICCSE)*, 2020, pp. 627-634.

[42] S. Zhou, "Gwo-ga-xgboost-based model for Radio-Frequency power amplifier under different temperatures," *Expert Systems with Applications,* vol. 278, p. 127439, 2025/06/10/ 2025, doi: https://doi.org/10.1016/j.eswa.2025.127439.

[43] E. T. S. Caramês *et al.*, "Identification of Fusarium sambucinum species complex by surface-enhanced Raman spectroscopy and XGBoost algorithm," *Food Chemistry,* vol. 480, p. 143848, 2025/07/15/ 2025, doi: https://doi.org/10.1016/j.foodchem.2025.143848.

[44] A. Wan, Z. Gong, T. Chen, and K. Al-Bukhaiti, "Mass flow characteristics prediction of refrigerants through electronic expansion valve based on XGBoost," *International Journal of Refrigeration,* vol. 158, pp. 345-352, 2024/02/01/ 2024, doi: https://doi.org/10.1016/j.ijrefrig.2023.12.011.

[45] P. Das, A. Kashem, J. U. Rahat, and R. Karim, "A comparative study of ensemble machine learning models for compressive strength prediction in recycled aggregate concrete and parametric analysis," *Multiscale and Multidisciplinary Modeling, Experiments and Design,* vol. 7, 04/07 2024, doi: 10.1007/s41939-024-00409-3.

[46] İ. Kılıç. "Light GBM: A Powerful Gradient Boosting Algorithm." https://medium.com/@ilyurek/light-gbm-a-powerful-gradient-boosting-algorithm-fe145a1cd8a6 (accessed.

[47] H. E. Sadig *et al.*, "Advanced time complexity analysis for real-time COVID-19 prediction in Saudi Arabia using LightGBM and XGBoost," *Journal of Radiation Research and Applied Sciences,* vol. 18, no. 2, p. 101364, 2025/06/01/ 2025, doi: https://doi.org/10.1016/j.jrras.2025.101364.

[48] Y. Zhou, S. Wang, Z. Li, R. Feng, and C. Fernandez, "Battery pack capacity estimation based on improved cooperative co-evolutionary strategy and LightGBM hybrid models using indirect health features," *Journal of Energy Storage,* vol. 114, p. 115914, 2025/04/10/ 2025, doi: https://doi.org/10.1016/j.est.2025.115914.

[49] P. Jain, R. Gupta, A. Joshi, and A. Kuzmin, "Enhanced cardiovascular diagnostics using wearable ECG and bioimpedance monitoring with LightGBM classifier," *Biosensors and Bioelectronics: X,* vol. 24, p. 100617, 2025/08/01/ 2025, doi: https://doi.org/10.1016/j.biosx.2025.100617.

[50] M. Li, H. Tao, M. Liu, and T. He, "Study on enhanced fault diagnosis of chiller units in HVAC systems under the imbalanced data environment using GA-Optimized LightGBM," *Energy and Buildings,* vol. 330, p. 115360, 2025/03/01/ 2025, doi: https://doi.org/10.1016/j.enbuild.2025.115360.

[51] B. Niu and Z. Huang, "Multi-step probabilistic forecasting for sunspot numbers based on LightGBM," *Advances in Space Research,* 2025/03/20/ 2025, doi: https://doi.org/10.1016/j.asr.2025.03.041.

[52] Y. Lu *et al.*, "Comparison of machine learning and logistic regression models for predicting emergence delirium in elderly patients: A prospective study," *International Journal of Medical Informatics,* vol. 199, p. 105888, 2025/07/01/ 2025, doi: https://doi.org/10.1016/j.ijmedinf.2025.105888.

[53] Z. Zhang, M. Tantai, H. Ma, S. Yu, B. Chen, and Z. Lu, "Analysis of Risk Factors for Lumbar Spondylolisthesis: A Logistic Regression Study," *World Neurosurgery,* vol. 197, p. 123931, 2025/05/01/ 2025, doi: https://doi.org/10.1016/j.wneu.2025.123931.

[54] S. Baroumand, A. R. Abbasi, and M. Mahmoudi, "Integrative fault diagnostic analytics in transformer windings: Leveraging logistic regression, discrete wavelet transform, and neural networks," *Heliyon,* vol. 11, no. 4, p. e42872, 2025/02/28/ 2025, doi: https://doi.org/10.1016/j.heliyon.2025.e42872.

[55] A. Vázquez-Veloso, A. Toraño Caicoya, F. Bravo, P. Biber, E. Uhl, and H. Pretzsch, "Does machine learning outperform logistic regression in predicting individual tree mortality?," *Ecological Informatics,* p. 103140, 2025/04/12/ 2025, doi: https://doi.org/10.1016/j.ecoinf.2025.103140.

[56] M. K. Pawar and P. Patil, "Logistic Regression for Enhancing Scalability of Blockchain System," *Procedia Computer Science,* vol. 252, pp. 146-153, 2025/01/01/ 2025, doi: https://doi.org/10.1016/j.procs.2024.12.016.

[57] C. Han, X. Lu, and S. Chen, "Modeling rater judgments of interpreting quality: Ordinal logistic regression using neural-based evaluation metrics, acoustic fluency measures, and computational linguistic indices," *Research Methods in Applied Linguistics,* vol. 4, no. 1, p. 100194, 2025/04/01/ 2025, doi: https://doi.org/10.1016/j.rmal.2025.100194.

[58] D. T. Tran, J. Shiau, S. Keawsawasvong, and P. Jamsawang, "VHM failure envelopes of spudcan foundations buried in natural clay: FELA and FPA-CatBoost optimization algorithms," *Ocean Engineering,* vol. 325, p. 120779, 2025/05/01/ 2025, doi: https://doi.org/10.1016/j.oceaneng.2025.120779.

[59] Z. Fan, J. Gou, and S. Weng, "Complementary CatBoost based on residual error for student performance prediction," *Pattern Recognition,* vol. 161, p. 111265, 2025/05/01/ 2025, doi: https://doi.org/10.1016/j.patcog.2024.111265.

[60] H. Yang, Z. Liu, Y. Li, H. Wei, and N. Huang, "CatBoost–Bayesian Hybrid Model Adaptively Coupled with Modified Theoretical Equations for Estimating the Undrained Shear Strength of Clay," *Applied Sciences,* vol. 13, no. 9, p. 5418, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/9/5418.

[61] D. H. Wolpert, "Stacked generalization," *Neural Networks,* vol. 5, no. 2, pp. 241-259, 1992/01/01/ 1992, doi: https://doi.org/10.1016/S0893-6080(05)80023-1.

[62] T. G. Dietterich, *Ensemble methods in machine learning*. 2000, pp. 1-15.

[63] C. Plumley. "Kelmarsh wind farm data." https://zenodo.org/records/5841834 (accessed.

[64] P. W. Khan, C. Yeun, and Y. Byun, "Fault detection of wind turbines using SCADA data and genetic algorithm-based ensemble learning," *Engineering Failure Analysis,* vol. 148, p. 107209, 03/01 2023, doi: 10.1016/j.engfailanal.2023.107209.

[65] X. Liu, S. Lu, Y. Ren, and Z. Wu, "Wind turbine anomaly detection based on SCADA data mining," *Electronics,* vol. 9, p. 751, 05/02 2020, doi: 10.3390/electronics9050751.

[66] X. Chesterman, P.-J. Daems, A. Nowe, and J. Helsen, "Overview of normal behavior modeling approaches for SCADA-based wind turbine condition monitoring demonstrated on data from operational wind farms," *Wind Energy Science,* vol. 8, pp. 893-924, 06/05 2023, doi: 10.5194/wes-8-893-2023.

[67] F. Fiocchi, D. Ladopoulou, and P. Dellaportas, "Probabilistic Multilayer Perceptrons for Wind Farm Condition Monitoring," *Wind Energy,* vol. 28, 03/13 2025, doi: 10.1002/we.70012.

[68] L. Gigoni, A. Betti, M. Tucci, and E. Crisostomi, *A Scalable Predictive Maintenance Model for Detecting Wind Turbine Component Failures Based on SCADA Data*. 2019, pp. 1-5.

[69] K. Weber and C. Preisach, *Supervised Transfer Learning Framework for Fault Diagnosis in Wind Turbines*. 2024.