REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديموقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE SUPERIEURE EN SCIENCES APPLIQUEES
TLEMCEN



المدرسة العليا للعلوم التطبيقية
Ecole Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
تلمسان

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique

Présenté par : Ahmed Ghezal

Mohammed Nehila

Thème

Étude et conception d'un système IoT intelligent pour la collecte, l'analyse et la distribution à la demande des données industrielles

Soutenu publiquement, le 21 juin 2026, devant le jury composé de :

Dr. Karaouzene Zoheir	MCB	ESSA-Tlemcen	Encadreur
Pr. Zidi Salah	Professeur	ISSIG Gabes-Tunisie	Co-Encadreur
Dr. Abdi Sidi Mohamed	MCA	ESSA-Tlemcen	Président du jury
Dr. Arichi Fayçal	MCA	ESSA-Tlemcen	Examinateur 1
Dr. Alouane Mohammed Amine		EMP-Alger	Examinateur 2
Dr. Megnafi Hicham	MCA	ESSA-Tlemcen	Invité

Année Universitaire : **2025 / 2026**

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Dédicaces

"To my dear parents "

Ghezal Ahmed
12/06/2026

"In the name of Allah, the Most Gracious, the Most Merciful.
First and foremost, I dedicate this work to Allah Almighty, whose blessings, guidance, and mercy have given me the strength, patience, and perseverance to complete this master's project.
I dedicate this work to the memory of my beloved mother. May Allah grant her His infinite mercy and the highest place in Jannah. Her love, sacrifices, and prayers continue to inspire me every day, and her memory remains forever in my heart.
To my father, for his support, guidance, and the values he has instilled in me throughout my life.
To my dear brothers, Farouk and Ayoub, whose encouragement and support have always motivated me to move forward.
To my project partner, GHEZAL Ahmed, for his cooperation, dedication, and the efforts we shared throughout the completion of this project.
To my friends Nadir, Sakina, Aziz, Ali and Bilel, thank you for your friendship, encouragement, and the memorable moments we shared during this academic journey.
To all my classmates in the Automation Engineering program, with whom I shared years of learning, challenges, teamwork, and unforgettable experiences.
I would also like to dedicate this work to myself, in recognition of the effort, determination, resilience, and commitment invested in achieving this important milestone.
Finally, I dedicate this work to everyone who supported me, directly or indirectly, throughout my studies and the realization of this project.
With sincere gratitude and respect ".

NEHILA Mohammed
12/06/2026

Remerciements :

Avant toute chose, je tiens à exprimer ma profonde gratitude à Dieu Tout-Puissant, pour m'avoir accordé la force, la patience et la persévérance nécessaires tout au long de ce travail. Sans Sa miséricorde et Ses bénédictions, l'accomplissement de ce mémoire n'aurait jamais été possible.

Je tiens à adresser mes plus sincères remerciements à Monsieur le Docteur Zoheir KARAOUZENE, mon encadreur, pour sa disponibilité constante, ses conseils avisés et son orientation précieuse tout au long de ce travail. Sa rigueur scientifique, son soutien bienveillant et son suivi attentif ont été une véritable source de motivation et ont grandement contribué à la réussite de ce mémoire.

Je remercie également chaleureusement Monsieur le Docteur Hicham MEGNAFI, mon co-encadreur, pour son implication active, ses remarques constructives et son aide précieuse à chaque étape de l'élaboration de ce travail. Son expertise et sa disponibilité nous ont permis d'approfondir notre réflexion et d'améliorer considérablement la qualité de ce mémoire.

Je tiens également à exprimer ma profonde gratitude au **président du jury, Dr. Abdi Sidi Mohamed**, pour l'honneur qu'il nous a fait en acceptant de présider cette soutenance. Mes sincères remerciements s'adressent également aux **examinateurs, Dr. ARICHI Fayçal et Dr. ALOUANE Mohammed Amine**, pour le temps qu'ils ont consacré à la lecture et à l'évaluation de ce travail. Leurs remarques pertinentes, leurs observations constructives et leurs questions enrichissantes ont contribué à améliorer la qualité de ce mémoire et à approfondir notre réflexion.

Mes remerciements s'adressent également à l'ensemble des enseignants de notre formation, qui ont contribué à notre apprentissage tout au long de ces années d'études, et dont les enseignements ont constitué un socle indispensable à la réalisation de ce travail.

Enfin, je remercie toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce mémoire, par leur soutien moral, leurs encouragements ou leur aide technique.

Résumé :

Ce mémoire présente l'étude et la conception d'un système IoT intelligent pour la collecte, l'analyse et la distribution à la demande des données industrielles, appliqué à la maintenance prédictive d'une pompe industrielle par analyse vibratoire. Le système repose sur une architecture IoT en trois couches intégrant les protocoles MQTT et Modbus TCP pour la collecte, InfluxDB pour le stockage des séries temporelles, et Node-RED et Grafana pour la visualisation. Un pipeline de traitement en huit étapes, entièrement simulé sous MATLAB, combine le prétraitement du signal (filtrage, agrégation, FFT), le Machine Learning supervisé (Random Forest — accuracy 100%) et non supervisé (K-Means), la surveillance par seuils ISO 10816 et la prédiction de défaillance par régression linéaire. Les résultats confirment la faisabilité et l'efficacité de l'approche IoT-IA intégrée, avec une distribution à la demande adaptée à trois profils de clients distincts : opérateur de maintenance, ingénieur de diagnostic et superviseur de production.

Mots-clés : IoT Industriel, IIoT, Industrie 4.0, Maintenance Prédictive, Machine Learning, Random Forest, K-Means, Analyse Vibratoire, FFT, MQTT, Distribution à la Demande, MATLAB.

Abstract :

This thesis presents the study and design of an intelligent IoT system for the collection, analysis and on-demand distribution of industrial data, applied to the predictive maintenance of an industrial pump through vibration analysis. The system relies on a three-layer IoT architecture integrating MQTT and Modbus TCP protocols for data collection, InfluxDB for time-series storage, and Node-RED and Grafana for visualization. An eight-step processing pipeline, entirely simulated in MATLAB, combines signal preprocessing (filtering, aggregation, FFT), supervised Machine Learning (Random Forest — 100% accuracy) and unsupervised learning (K-Means), ISO 10816 threshold-based monitoring, and linear regression-based failure prediction. The results confirm the feasibility and effectiveness of the integrated IoT-AI approach, with on-demand distribution adapted to three distinct client profiles: maintenance operator, diagnostic engineer, and production supervisor.

Keywords: Industrial IoT, IIoT, Industry 4.0, Predictive Maintenance, Machine Learning, Random Forest, K-Means, Vibration Analysis, FFT, MQTT, On-Demand Distribution, MATLAB.

ملخص :

تقدم هذه المذكرة دراسة وتصميم نظام إنترنت الأشياء الذكي لجمع البيانات الصناعية وتحليلها وتوزيعها عند الطلب، مطبقًا على الصيانة التنبؤية لمضخة صناعية عن طريق تحليل الاهتزازات. يعتمد النظام على معمارية إنترنت الأشياء ذات الطبقات الثلاث، متضمنةً بروتوكولي **MQTT** و **Modbus TCP** لجمع البيانات، و **InfluxDB** لتخزين السلاسل الزمنية، و **Node-RED** و **Grafana** للتصور البياني. يجمع خط معالجة من ثماني مراحل، محاكى بالكامل في **MATLAB**، بين المعالجة المسبقة للإشارة) التصفية، التجميع، (FFT، والتعلم الآلي الخاضع للإشراف **Random Forest** — دقة 100 (%وغير الخاضع للإشراف(**K-Means**) ، والمراقبة بعتبات**ISO 10816** ، والتنبؤ بالأعطال عبر الانحدار الخطي. تؤكد النتائج جدوى وفعالية المقاربة المتكاملة لإنترنت الأشياء والذكاء الاصطناعي، مع توزيع عند الطلب ملائم لثلاثة ملفات تعريف مختلفة من العملاء: مشغل الصيانة، ومهندس التشخيص، والمشرف على الإنتاج.

الكلمات المفتاحية :إنترنت الأشياء الصناعي، **IIoT**، الصناعة **4.0**، الصيانة التنبؤية، التعلم الآلي، **Random Forest**، **K-Means**، تحليل الاهتزازات، **FFT**، **MQTT**، التوزيع عند الطلب، **.MATLAB**

TABLE DES MATIÈRES

Introduction Générale **1**

Chapitre 1 : Infrastructure IoT — Collecte, Traitement et Distribution des Données Industrielles **2**

Introduction 3

1 Contexte : L'Industrie 4.0 et le besoin en données 3

1.1 De l'usine traditionnelle à l'usine connectée 3

1.2 Les données industrielles 4

1.3 Le rôle central de l'IoT dans ce contexte 5

2 Architecture d'un système IoT industriel 5

2.1 Vue d'ensemble : les trois couches fonctionnelles 5

2.2 Les composants clés : capteurs, gateway, serveur 6

2.3 Comparaison entre Edge Computing et Cloud Computing 7

3 La collecte des données : du capteur à la gateway 8

3.1 Les types de capteurs industriels 8

3.2 Les protocoles de terrain : MQTT et Modbus TCP 9

3.3 Le rôle de la gateway : agrégation et conversion de protocoles 11

4 Le traitement des données : du brut à l'utile 12

4.1 Prétraitement : filtrage et nettoyage 12

4.2 Agrégation et compression des données 12

4.3 Détection d'anomalies simple : seuils et statistiques de base 13

5 La distribution à la demande 14

5.1 Concept de distribution à la demande 14

5.2 Architecture client-serveur et publish/subscribe 15

5.3 Les plateformes de visualisation : Node-RED et Grafana 16

5.4 Stockage des séries temporelles : InfluxDB 17

6 Sécurité et fiabilité du système 18

6.1 Les menaces spécifiques aux systèmes IIoT 18

6.2 Les solutions essentielles 18

Conclusion 19

Chapitre 2 : Analyse des Données et Intelligence Artificielle **20**

Introduction 21

1 L’intégration de l'intelligence artificielle (IA) et le passage la donnée brute à la connaissance 21
1.1 Les limites des méthodes classiques de traitement de données 21
1.2 Machine Learning, définition et positionnement 22
1.3 Prétraitement spécifique au ML : normalisation et encodage 23
2 Apprentissage supervisé (Supervised Learning) 25
2.1 Principe et fonctionnement 25
2.2 Algorithmes clés : Random Forest, SVM, Régression logistique 26
2.3 Application industrielle : classification de pannes 28
3 Apprentissage non supervisé (Unsupervised Learning) 29
3.1 Principe et différence avec l'apprentissage supervisé 29
3.2 Algorithmes clés : K-Means, DBSCAN, Autoencodeurs 30
3.3 Application industrielle : détection d'anomalies 31
4 Comparaison et choix des algorithmes 33
4.1 Tableau comparatif supervisé vs non supervisé 33
4.2 Quel algorithme pour quel contexte industriel ? 33
4.3 Outils utilisés : Python, Scikit-learn, Pandas 34
Conclusion 35
Chapitre 3 : Intégration IoT et IntelligenceArtificielle — Étude de Cas : Maintenance Prédictive d'une Pompe Industrielle 37
Introduction 38
1 Présentation de l'étude de cas 38
1.1 Description du système étudié : la pompe industrielle 38
1.2 Objectifs de la simulation 39
1.3 Architecture globale du système IoT-IA simulé 39
2 Collecte et prétraitement des données IoT 40
2.1 Simulation des signaux vibratoires (Normal vs Défaut) 40
2.2 Filtrage et nettoyage du signal 41
2.3 Agrégation des données par fenêtre temporelle 42
3 Extraction des caractéristiques 43
3.1 Analyse fréquentielle par FFT 43
3.2 Visualisation Temps-Fréquence (Spectrogramme) 44
3.3 Indicateurs retenus pour le Machine Learning 45
4 Application du Machine Learning 45

4.1 Classification supervisée par Random Forest 45
4.2 Clustering non supervisé par K-Means 48
5 Surveillance et détection des défauts 49
5.1 Détection par seuil ISO 10816 49
5.2 Prédiction de l'alarme par analyse de tendance 50
6 Distribution à la demande — Simulation Multi-Clients IoT 51
6.1 Architecture de distribution simulée 51
6.2 Résultats par type de client 52
Conclusion 53
Conclusion Générale 54

TABLE DES FIGURES

Figure 1.1 : Les quatre révolutions industrielles — de la machine à vapeur à l'usine connectée 04
Figure 1.2 : Architecture en trois couches d'un système IoT industriel 06
Figure 1.3 : Modèle Publish/Subscribe du protocole MQTT avec broker central.....10
Figure 1.4 : Pipeline de traitement des données : du signal brut aux indicateurs synthétiques 14
Figure 1.5 : Architecture de distribution à la demande 15
Figure 1.6 : Modèle client-serveur pour l'accès aux données historiques 15
Figure 1.7 : Modèle publish/subscribe pour la diffusion des données en temps réel.....16
Figure 2.1 : Positionnement du Machine Learning dans l'Intelligence Artificielle 23
Figure 2.2 : Principe de l'apprentissage supervisé : phase d'entraînement et phase de prédiction.....26
Figure 2.3 : Illustration du principe SVM — hyperplan de séparation et marge maximale.....27
Figure 2.4 : Pipeline de classification supervisée pour la maintenance prédictive de pompes29
Figure 2.5 : Comparaison visuelle apprentissage supervisé vs non supervisé.....30
Figure 2.6 : Pipeline de détection d'anomalies par apprentissage non supervisé.....32
Figure 2.8 : Arbre de décision pour le choix de l'algorithme selon le contexte industriel 34
Figure 3.1 : Architecture globale du pipeline IoT-IA simulé sous MATLAB — 8 étapes 40
Figure 3.2 : Étape 1 — Signaux vibratoires bruts et filtrés (Normal vs Défaut) 41
Figure 3.3 : Étape 2 — Agrégation IoT : RMS, Peak et Écart-type par fenêtre temporelle.....42
Figure 3.4 : Étape 3 — Analyse FFT des signaux Normal et Défaut 43
Figure 3.5 : Étape 4 — Spectrogrammes Normal et Défaut.....44
Figure 3.6 : Matrice de confusion — Classificateur Random Forest (Accuracy : 100%).....46
Figure 3.7 : Importance des caractéristiques — Random Forest.....47
Figure 3.8 : Étape 5 — K-Means : clusters détectés vs vérité terrain.....48
Figure 3.9 : Étape 6 — Surveillance RMS et Peak avec seuils d'alarme ISO 10816.....49
Figure 3.10 : Étape 7 — Prédiction de l'alarme par régression linéaire 50
Figure 3.11 : Étape 8 — Distribution à la demande : résultats pour les trois clients IoT.....52

TABLE DES TABLEAUX

Tableau 1.1 : Comparaison entre l'usine traditionnelle et l'usine connectée.....04

Tableau 1.2 : Rôles et caractéristiques des composants clés d'un système IoT industriel.....07

Tableau 1.3 : Comparaison Edge Computing et Cloud Computing pour les applications IIoT 08

Tableau 1.4 : Types de capteurs industriels et paramètres mesurés 09

Tableau 1.5 : Comparaison MQTT et Modbus TCP.....11

Tableau 1.6 : Comparaison Node-RED et Grafana dans un contexte IIoT 17

Tableau 1.7 : Synthèse des menaces IIoT et solutions associées 19

Tableau 2.1 : Les trois grandes familles du Machine Learning.....23

Tableau 2.2 : Récapitulatif des étapes de prétraitement pour le Machine Learning.....24

Tableau 2.3 : Comparaison des algorithmes d'apprentissage supervisé 28

Tableau 2.4 : Comparaison apprentissage supervisé et non supervisé 30

Tableau 2.5 : Comparaison des algorithmes de détection d'anomalies 32

Tableau 2.6 : Tableau comparatif complet — Supervisé vs Non supervisé 33

Tableau 2.7 : Bibliothèques Python utilisées et leurs fonctions 35

Tableau 3.1 : Présentation des 8 étapes du pipeline IoT-IA simulé.....40

Tableau 3.2 : Valeurs moyennes des indicateurs agrégés — Normal vs Défaut 43

Tableau 3.3 : Vecteur de caractéristiques retenu pour le Machine Learning 45

Tableau 3.4 : Classement des caractéristiques par importance (Random Forest) 47

Tableau 3.5 : Résultats de la prédiction par analyse de tendance 50

Tableau 3.6 : Synthèse de la distribution à la demande — Informations par profil client.....53

Introduction Générale

L'industrie mondiale traverse aujourd'hui une mutation profonde, portée par l'émergence de l'Internet des Objets industriel et les principes de l'Industrie 4.0. Dans ce contexte, les équipements de production génèrent en permanence des volumes considérables de données — signaux vibratoires, températures, pressions, états de fonctionnement — qui constituent une ressource informationnelle d'une valeur considérable, mais qui demeurent le plus souvent inexploitées, faute d'infrastructures adaptées à leur collecte, leur analyse et leur distribution intelligente.

Cette réalité industrielle soulève une problématique fondamentale : comment collecter efficacement les données issues des équipements industriels, les analyser en temps réel pour en extraire une connaissance exploitable, et les distribuer à la demande aux différents acteurs de l'entreprise selon leurs besoins spécifiques ? Les solutions traditionnelles de supervision, souvent propriétaires et rigides, ne répondent que partiellement à ces exigences, et l'intégration de l'intelligence artificielle dans les systèmes IoT industriels ouvre des perspectives nouvelles, notamment pour la maintenance prédictive et la détection précoce des défaillances.

C'est dans ce contexte que s'inscrit le présent mémoire, dont l'objectif est d'étudier et de concevoir un système IoT intelligent pour la collecte, l'analyse et la distribution à la demande des données industrielles. L'étude de cas retenue est la maintenance prédictive d'une pompe industrielle par analyse vibratoire, entièrement simulée sous MATLAB, permettant de valider les approches proposées dans un environnement contrôlé et reproductible.

Pour atteindre ces objectifs, le mémoire est structuré en trois chapitres :

- Le premier chapitre présente l'infrastructure IoT industrielle nécessaire à la collecte, au traitement et à la distribution des données, en couvrant les architectures, les protocoles de communication, les outils de visualisation et les aspects de sécurité.
- Le deuxième chapitre traite de l'analyse des données et de l'intelligence artificielle, en présentant les algorithmes de Machine Learning supervisé et non supervisé et leur application à la détection d'anomalies et à la classification des défauts industriels.
- Le troisième chapitre présente l'intégration concrète des deux approches précédentes au travers d'une étude de cas complète, simulant un pipeline IoT-IA de bout en bout appliqué à la maintenance prédictive d'une pompe industrielle, depuis la collecte des signaux vibratoires jusqu'à la distribution à la demande des résultats aux différents profils de clients.

Chapitre 1 : Infrastructure IoT — Collecte, Traitement et Distribution des Données Industrielles

Introduction

L'essor des technologies numériques a profondément transformé le monde industriel au cours de la dernière décennie. Les usines, autrefois fondées sur des processus mécaniques isolés et des échanges d'informations manuels, évoluent aujourd'hui vers des environnements hautement connectés, capables de générer, transmettre et exploiter des volumes massifs de données en temps réel.

Au cœur de cette transformation se trouve l'Internet des Objets Industriel (IIoT), une technologie qui permet à des équipements physiques — capteurs, machines, automates — de communiquer entre eux et avec des systèmes d'information, sans intervention humaine directe. Cette capacité à collecter des données en continu depuis le terrain, à les traiter intelligemment et à les distribuer aux utilisateurs selon leurs besoins constitue le fondement même du projet présenté dans ce mémoire.

Ce premier chapitre pose les bases théoriques et techniques nécessaires à la compréhension du système étudié. Il s'articule autour de quatre axes principaux : le contexte industriel qui justifie le recours à l'IoT, l'architecture générale d'un système IoT industriel, les mécanismes de collecte et de traitement des données, et enfin les modalités de leur distribution et stockage. Une attention particulière est portée aux aspects de sécurité, indispensables dans tout déploiement industriel réel.

1 Contexte : L'Industrie 4.0 et le besoin en données

1.1 De l'usine traditionnelle à l'usine connectée

L'histoire de l'industrie est jalonnée de révolutions successives qui ont chacune redéfini les modes de production. La première révolution industrielle, au XVIIIe siècle, a introduit la machine à vapeur et mécanisé le travail manuel. La deuxième, à la fin du XIXe siècle, a apporté l'électricité et la production de masse. La troisième, dans la seconde moitié du XXe siècle, a marqué l'avènement de l'automatisation et de l'informatique industrielle. Nous vivons aujourd'hui la quatrième révolution industrielle, communément désignée sous le terme **Industrie 4.0**, caractérisée par la fusion du monde physique et du monde numérique.

Dans l'usine traditionnelle, les équipements fonctionnaient de manière isolée. Un opérateur relevait manuellement les valeurs des instruments, les consignait dans des registres papier, et les décisions de maintenance ou de production reposaient sur l'expérience humaine et des cycles calendaires fixes. Ce modèle, bien que fonctionnel pendant des décennies, souffrait de limitations majeures : latence dans la prise de décision, manque de traçabilité, impossibilité de détecter des anomalies subtiles avant qu'elles ne dégénèrent en pannes graves.

L'usine connectée, en revanche, repose sur un principe fondamentalement différent : **chaque machine, chaque équipement, chaque capteur devient une source d'information continue**. Ces informations sont collectées automatiquement, transmises en temps réel à des systèmes de traitement, et transformées en indicateurs exploitables par les opérateurs, les ingénieurs de maintenance et les responsables de production. Le tableau suivant résume les principales différences entre ces deux modèles :

Critère	Usine traditionnelle	Usine connectée (Industrie 4.0)
Collecte des données	Manuelle, périodique	Automatique, continue, temps réel
Prise de décision	Basée sur l'expérience humaine	Basée sur les données (data-driven)
Maintenance	Corrective ou préventive calendaire	Prédictive, basée sur l'état réel
Traçabilité	Partielle, sur papier	Complète, numérique
Réactivité aux pannes	Après la panne	Avant la panne (anticipation)
Coût de maintenance	Élevé (arrêts non planifiés)	Réduit (interventions ciblées)
Interopérabilité	Faible (silos d'équipements)	Élevée (systèmes interconnectés)

Tableau 1.1 : Comparaison entre l'usine traditionnelle et l'usine connectée

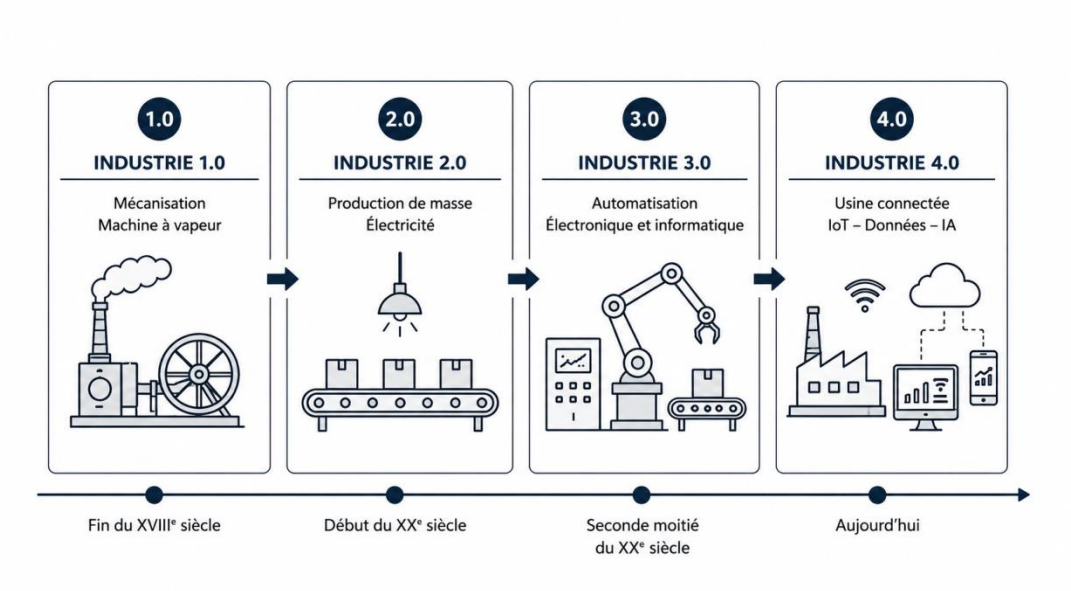


Figure 1.1 : Les quatre révolutions industrielles — de la machine à vapeur à l'usine connectée

1.2 Les données industrielles

Dans le contexte de l'Industrie 4.0, la donnée industrielle est devenue une ressource stratégique au même titre que les matières premières ou l'énergie. Sa valeur ne réside pas dans sa collecte brute, mais dans sa capacité à alimenter des décisions rapides, précises et pertinentes.

Les données industrielles présentent plusieurs caractéristiques qui les distinguent des données générées dans d'autres domaines :

Volume et vélocité élevés : une ligne de production moderne peut générer plusieurs milliers de points de mesure par seconde, issues de dizaines de capteurs fonctionnant en parallèle.

Caractère temporel : les données industrielles sont intrinsèquement liées au temps. Une valeur de vibration n'a de sens que replacée dans son contexte temporel, en relation avec les autres mesures effectuées au même instant.

Hétérogénéité : les données proviennent de sources très diverses — capteurs de température, d'humidité, de pression, d'intensité électrique, d'accélération — utilisant des unités, des plages et des protocoles différents.

Criticité : contrairement aux données d'une application grand public, une donnée industrielle erronée ou manquante peut entraîner des conséquences graves : arrêt de production, accident matériel ou humain.

Selon un rapport de McKinsey Global Institute (2022), l'exploitation intelligente des données industrielles peut réduire les coûts de maintenance de **15 à 30%**, augmenter la disponibilité des équipements de **10 à 20%**, et réduire les déchets de production de **10 à 15%** [1]. Ces chiffres illustrent concrètement l'enjeu économique et opérationnel que représente la maîtrise des données dans l'environnement industriel.

1.3 Le rôle central de l'IoT dans ce contexte

L'Internet des Objets (IoT) constitue l'infrastructure technologique qui rend possible la collecte, la transmission et l'exploitation des données industrielles à grande échelle. Il joue un rôle de **système nerveux** dans l'usine connectée : il capte les signaux du terrain (via les capteurs), les transmet au cerveau décisionnel (via les réseaux), et permet l'exécution des commandes (via les actionneurs).

Dans sa déclinaison industrielle — l'**IIoT (Industrial Internet of Things)** — cette technologie répond à des exigences beaucoup plus strictes que l'IoT grand public : fiabilité proche de 100%, latence de communication de l'ordre de la milliseconde, sécurité renforcée, et capacité à fonctionner dans des environnements difficiles (chaleur, vibrations, interférences électromagnétiques).

Le rapport présenté dans [2] sur l'IoT daté en 2024, estime que le nombre de dispositifs IIoT connectés dans le monde dépassera **36 milliards d'unités d'ici 2025**, avec un taux de croissance annuel de 16%. Cette croissance reflète l'adoption massive de l'IoT comme pilier technologique de la transformation industrielle mondiale.

2 Architecture d'un système IoT industriel

2.1 Vue d'ensemble : les trois couches fonctionnelles

Un système IoT industriel s'organise selon une architecture en **trois couches fonctionnelles** distinctes et complémentaires. Cette structuration en couches est aujourd'hui la référence académique et industrielle la plus largement adoptée, notamment recommandée par l'Union Internationale des Télécommunications (ITU-T Y.2060) [3].

2.1.1 La couche de perception constitue l'interface entre le monde physique et le monde numérique. Elle regroupe l'ensemble des capteurs, actionneurs et équipements embarqués qui

perçoivent l'environnement industriel — température, vibration, pression, courant électrique — et convertissent ces grandeurs physiques en signaux numériques exploitables. C'est à ce niveau que naît la donnée industrielle.

2.1.2 La couche réseau assure le transport des données collectées depuis les équipements de terrain vers les systèmes de traitement. Elle englobe les passerelles IoT (gateways), les protocoles de communication (MQTT, Modbus TCP), et les infrastructures réseau (Wi-Fi industriel, Ethernet, réseaux cellulaires). Elle constitue la colonne vertébrale du système IoT.

2.1.3 La couche application représente le niveau de traitement, d'analyse et de distribution des données. C'est ici que les données brutes sont transformées en informations exploitables, visualisées sur des tableaux de bord, stockées dans des bases de données temporelles, et distribuées aux différents utilisateurs selon leurs besoins spécifiques.

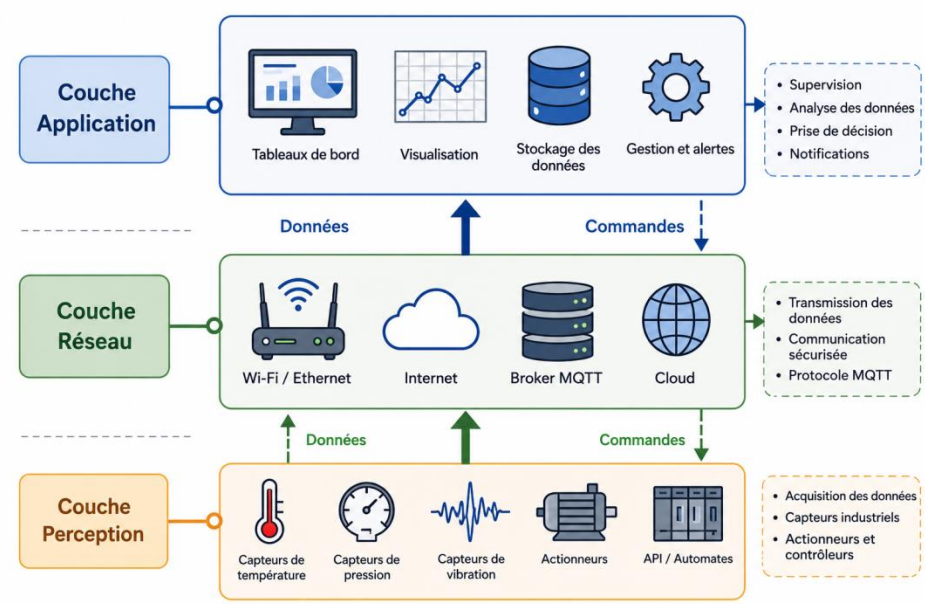


Figure 1.2 : Architecture en trois couches d'un système IoT industriel

2.2 Les composants clés : capteurs, gateway, serveur

Le fonctionnement d'un système IoT industriel repose sur l'interaction coordonnée de trois types de composants fondamentaux :

2.2.1 Les capteurs sont les organes de perception du système. Ils mesurent en continu les grandeurs physiques caractéristiques de l'état des équipements industriels. Leur choix dépend de la nature des données à collecter, de la précision requise, et des contraintes environnementales (température ambiante, présence de vibrations, niveau d'humidité). Dans le contexte de la maintenance prédictive — application centrale de ce projet — les capteurs d'accélération (accéléromètres) jouent un rôle particulièrement important, car les vibrations constituent l'un des indicateurs les plus fiables de la dégradation des équipements rotatifs.

2.2.2 La gateway (passerelle IoT) est le composant stratégique qui assure la jonction entre le terrain et les systèmes d'information. Elle collecte les données issues de multiples

capteurs hétérogènes, effectue un premier niveau de traitement local (filtrage, agrégation), convertit les protocoles de communication de terrain vers les protocoles Internet standards, et transmet les données vers le serveur central. Dans une architecture Edge Computing, la gateway joue également le rôle d'unité de traitement local, réduisant ainsi la quantité de données à transmettre vers le cloud.

2.2.3 Le serveur central représente le cœur applicatif du système. Il reçoit les données transmises par les gateways, les stocke dans des bases de données adaptées (bases de données temporelles pour les séries de mesures), les traite selon les algorithmes définis, et les met à disposition des utilisateurs via des interfaces de visualisation et des API de distribution.

Composant	Rôle principal	Exemples	Localisation
Capteur	Mesure des grandeurs physiques	Accéléromètre ADXL345, Thermocouple PT100	Terrain
Microcontrôleur	Acquisition et transmission locale	ESP32, Arduino Mega	Terrain
Gateway	Agrégation, conversion, prétraitement	Raspberry Pi 4, Siemens IoT 2040	Terrain / Local
Broker MQTT	Routage des messages	Mosquitto, HiveMQ	Local / Cloud
Serveur applicatif	Traitement, stockage, distribution	Node-RED, ThingsBoard	Local / Cloud
Base de données	Stockage des séries temporelles	InfluxDB, TimescaleDB	Local / Cloud
Interface utilisateur	Visualisation, supervision	Grafana, Dashboard web	Client

Tableau 1.2 : Rôles et caractéristiques des composants clés d'un système IoT industriel

2.3 Comparaison entre Edge Computing et Cloud Computing

L'une des décisions architecturales les plus importantes dans la conception d'un système IoT industriel concerne la localisation du traitement des données : doit-il être effectué localement, au plus près des équipements (Edge Computing), ou centralisé dans un serveur distant (Cloud Computing) ?

Le **Cloud Computing** propose une infrastructure centralisée, élastique et puissante, capable de stocker et traiter des volumes de données considérables. Il offre une scalabilité quasi illimitée et facilite l'accès aux données depuis n'importe quel point. Cependant, il présente des inconvénients significatifs dans le contexte industriel : la latence introduite par la transmission des données vers un serveur distant peut être incompatible avec les exigences temps réel de certaines applications ; la dépendance à la connectivité réseau expose le système à des risques de défaillance en cas de coupure ; enfin, la transmission de grandes quantités de données brutes génère des coûts de bande passante élevés.

L'**Edge Computing** répond à ces limitations en déplaçant une partie du traitement vers la périphérie du réseau, au plus près des sources de données. Dans cette approche, la gateway ne se contente plus de transmettre les données brutes : elle les filtre, les agrège, détecte les

anomalies évidentes, et ne transmet vers le serveur central que les données pertinentes et condensées. Cette approche réduit significativement la latence, allège le trafic réseau, et permet au système de continuer à fonctionner même en cas de perte temporaire de connectivité. Le edge computing présente toutefois quelques limites, notamment un coût matériel plus élevé qu'une solution centralisée, une gestion plus complexe des appareils distribués, et une puissance de calcul limitée qui restreint les analyses avancées en local.

Critère	Edge Computing	Cloud Computing
Latence	Très faible (millisecondes)	Variable (secondes)
Dépendance réseau	Faible	Élevée
Capacité de stockage	Limitée	Quasi illimitée
Puissance de calcul	Limitée	Très élevée
Coût bande passante	Faible	Élevé si données brutes
Cas d'usage idéal	Détection temps réel, alarmes	Analyse historique, ML

Tableau 1.3 : Comparaison Edge Computing et Cloud Computing pour les applications IIoT

Dans la pratique, les systèmes IIoT modernes adoptent une **architecture hybride** combinant les deux approches : l'Edge Computing assure le traitement temps réel critique au niveau de la gateway, tandis que le Cloud centralise le stockage historique et les analyses approfondies. C'est cette architecture hybride qui sera retenue dans le cadre du présent projet.

3 La collecte des données : du capteur à la gateway

3.1 Les types de capteurs industriels

La qualité et la pertinence des données collectées dépendent directement du choix des capteurs. Dans un environnement industriel, les capteurs doivent non seulement mesurer avec précision les grandeurs physiques visées, mais aussi résister aux conditions difficiles du terrain : vibrations mécaniques, variations thermiques importantes, humidité, poussières et interférences électromagnétiques.

Les principaux types de capteurs utilisés dans les systèmes IIoT peuvent être classifiés selon la grandeur physique mesurée :

Type de capteur	Capteur (Exemple)	Paramètre mesuré	Principe de mesure	Exemple d'application industrielle	Interface / Protocoles
Température	Sonde de température (PT100) 	Température (°C)	Variation de la résistance électrique en fonction de la température	 Surveillance de température des moteurs, roulements, fours, échangeurs thermiques	4–20 mA, Modbus RTU, IO-Link
Pression	Transmetteur de pression industriel 	Pression (bar, Pa)	Déformation d'une membrane soumise à la pression	 Contrôle de pression dans les conduites, hydraulique, pneumatique	4–20 mA, Modbus RTU, HART
Vibration	 Accéléromètre industriel (IEPE)	Accélération (m/s²)	Mesure de la force inertielle appliquée à une masse sismique	 Surveillance des vibrations des machines tournantes, détection de défauts précoces	IEPE, Modbus TCP, MQTT
Humidité	 Capteur d'humidité relative	Humidité relative (% HR)	Variation de la capacité électrique d'un matériau sensible à l'humidité	 Contrôle de l'humidité dans les silos, chambres froides, pharmaceutique	Modbus RTU, LoRaWAN, MQTT
Débit	 Débitmètre électromagnétique	Débit (m³/h, L/min)	Loi de Faraday : induction d'une tension proportionnelle à la vitesse du fluide	 Mesure de débit d'eau, produits chimiques, eaux usées	Modbus RTU/TCP, 4–20 mA, Pulse
Niveau	 Capteur de niveau ultrasonique	Niveau (cm, m, %)	Mesure du temps de vol d'une onde ultrasonore réfléchie par la surface	 Surveillance du niveau dans les réservoirs, cuves, bacs	4–20 mA, Modbus RTU, IO-Link
Courant électrique	 Capteur de courant (à effet Hall)	Courant (A)	Effet Hall : mesure du champ magnétique généré par le courant	 Surveillance de la consommation électrique, protection des équipements	4–20 mA, Modbus RTU/TCP, MQTT

Tableau 1.4 : Types de capteurs industriels et paramètres mesurés

Dans le cadre spécifique de ce projet, orienté vers la **maintenance prédictive des pompes**, le capteur d'accélération occupe une place centrale. Les vibrations d'une pompe constituent en effet une signature physique directement liée à son état de santé : un roulement usé, un déséquilibre du rotor, ou un défaut d'alignement se manifestent chacun par des patterns vibratoires caractéristiques, détectables par l'analyse du signal avant même que la panne ne survienne.

3.2 Les protocoles de terrain : MQTT et Modbus TCP

La transmission des données depuis les capteurs vers la gateway s'appuie sur des protocoles de communication adaptés aux contraintes industrielles. Deux protocoles se distinguent particulièrement dans ce contexte : **MQTT** pour les communications IoT légères, et **Modbus TCP** pour l'intégration avec les équipements industriels existants.

3.2.1 Le protocole MQTT (Message Queuing Telemetry Transport)

MQTT est un protocole de messagerie léger, standardisé ISO/IEC 20922, spécialement conçu pour les environnements à ressources limitées et à bande passante réduite. Il repose sur un modèle **Publish/Subscribe** organisé autour de trois acteurs :

Le publisher (le capteur ou le microcontrôleur) publie ses mesures sur un **topic** — un canal thématique hiérarchique identifié par un chemin de type usine/pompe_01/vibration.

Le broker (serveur central MQTT, par exemple Mosquitto) reçoit tous les messages publiés et les redistribue aux abonnés concernés.

Le subscriber (l'application de traitement ou de visualisation) s'abonne aux topics qui l'intéressent et reçoit automatiquement les données dès leur publication.

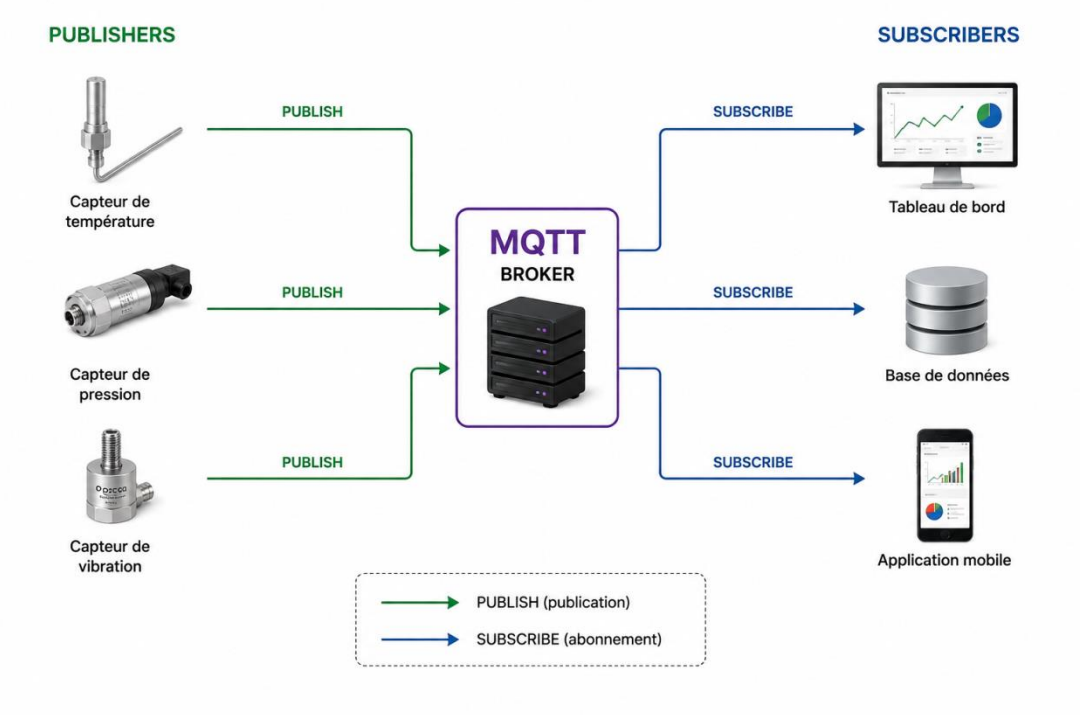


Figure 1.3 : Modèle Publish/Subscribe du protocole MQTT avec broker central

Cette architecture présente un avantage majeur pour les systèmes IIoT : le découplage total entre les producteurs de données (capteurs) et les consommateurs (applications). Un nouveau client peut s'abonner à un flux de données sans modifier en quoi que ce soit le comportement du capteur émetteur.

MQTT définit trois niveaux de Qualité de Service (QoS) permettant d'adapter la fiabilité de la transmission aux exigences de chaque application :

QoS 0 : le message est envoyé une seule fois, sans accusé de réception (au plus une livraison).

QoS 1 : le message est envoyé jusqu'à réception d'un accusé (au moins une livraison).

QoS 2 : le message est livré exactement une fois, avec un mécanisme de confirmation en quatre étapes (exactement une livraison).

3.2.2 Le protocole Modbus TCP

Modbus TCP est la version Ethernet du protocole Modbus, protocole industriel créé en 1979 et devenu un standard universel de communication entre automates (PLC), capteurs et

systèmes de supervision (SCADA). Il utilise le port TCP 502 et organise les données en quatre zones mémoire standardisées accessibles par des codes fonction prédéfinis.

Sa robustesse éprouvée sur plus de quatre décennies et son support quasi universel par les fabricants d'équipements industriels en font un protocole incontournable pour l'intégration des équipements existants dans une architecture IoT. Dans le cadre de ce projet, Modbus TCP constitue le protocole de référence pour la collecte de données depuis les automates industriels déjà déployés sur le terrain.

Critère	MQTT	Modbus TCP
Modèle	Publish/Subscribe (asynchrone)	Maître/Esclave (synchrone)
Protocole transport	TCP/IP	TCP/IP (port 502)
Légèreté	Très léger (en-tête 2 octets min.)	Modéré
Sécurité native	TLS/SSL supporté	Absente (à ajouter)
Adoption IoT	Très élevée	Élevée dans l'industrie
Cas d'usage	Capteurs IoT, télémétrie temps réel	PLC, équipements industriels existants

Tableau 1.5 : Comparaison MQTT et Modbus TCP

3.3 Le rôle de la gateway : agrégation et conversion de protocoles

La gateway IoT est le composant qui assure la transition entre la couche de perception (le terrain) et la couche réseau (les systèmes d'information). Son rôle dépasse largement la simple transmission de données : elle constitue un point de traitement intermédiaire indispensable dans toute architecture IIoT.

Les fonctions principales d'une gateway IoT industrielle sont les suivantes :

3.3.1 Agrégation des données : la gateway collecte simultanément les données issues de multiples capteurs hétérogènes — pouvant utiliser des interfaces physiques et des protocoles différents (I²C, SPI, UART, Modbus) — et les consolide en un flux unifié cohérent.

3.3.2 Conversion de protocoles : elle assure la traduction entre les protocoles de terrain (Modbus, Profibus, CAN) et les protocoles Internet standards (MQTT, HTTP), permettant ainsi l'intégration transparente des équipements industriels existants dans l'écosystème IoT.

3.3.3 Prétraitement local (Edge Processing) : avant transmission, la gateway peut effectuer un premier niveau de traitement — filtrage du bruit, calcul de valeurs moyennes, détection de dépassements de seuils — réduisant ainsi le volume de données à transmettre et la latence de réponse aux événements critiques.

3.3.4 Stockage temporaire (buffering local) : en cas d'interruption de la connectivité réseau, la gateway conserve localement les données collectées et les transmet dès que la connexion est rétablie, garantissant ainsi la continuité et l'intégrité des données.

4 Le traitement des données : du brut à l'utile

4.1 Prétraitement : filtrage et nettoyage

Les données collectées directement depuis les capteurs industriels sont rarement exploitables en l'état. Elles contiennent inévitablement des imperfections : valeurs aberrantes dues à des perturbations électromagnétiques, données manquantes liées à des interruptions de communication, bruit de mesure inhérent à la technologie du capteur. Le prétraitement constitue l'étape indispensable qui transforme ces données brutes en données fiables et cohérentes, prêtes pour l'analyse.

4.1.1 La détection et suppression des valeurs aberrantes (outliers)

Une valeur aberrante est une mesure anormalement éloignée des valeurs habituelles, non pas parce qu'elle représente un état réel de l'équipement, mais parce qu'elle résulte d'une perturbation externe ou d'un dysfonctionnement du capteur lui-même. La méthode statistique la plus couramment utilisée pour leur détection est la règle des **3 sigma (3σ)** : toute valeur s'écartant de plus de trois fois l'écart-type de la moyenne est considérée comme aberrante et exclue de l'analyse [4].

4.1.2 Le filtrage du bruit

Les signaux de capteurs industriels, particulièrement les accéléromètres, contiennent naturellement du bruit — des fluctuations aléatoires de haute fréquence sans signification physique. Deux approches de filtrage sont couramment appliquées :

Le filtre passe-bas élimine les composantes haute fréquence du signal, conservant uniquement les variations lentes et significatives.

Le filtre de moyenne mobile calcule la moyenne des N dernières mesures, lissant ainsi les fluctuations rapides tout en préservant les tendances de fond.

4.1.3 La gestion des données manquantes

Les interruptions de communication ou les défaillances momentanées de capteurs génèrent des lacunes dans les séries temporelles. Selon la durée et la fréquence de ces lacunes, différentes stratégies peuvent être appliquées : suppression des intervalles concernés (si les lacunes sont rares et courtes), interpolation linéaire entre les valeurs encadrantes, ou remplacement par la dernière valeur valide connue.

4.2 Agrégation et compression des données

Un système IIoT industriel peut générer un volume considérable de données brutes. Un seul accéléromètre échantillonnant à 1 kHz produit 1000 valeurs par seconde, soit 86,4 millions de valeurs par jour. Stocker et transmettre intégralement ces données est à la fois coûteux et souvent inutile : l'information pertinente est contenue dans des **indicateurs synthétiques** calculés à partir de ces mesures brutes.

L'agrégation consiste à calculer, sur des fenêtres temporelles définies (par exemple toutes les 10 secondes), des indicateurs statistiques résumant le comportement du signal :

Valeur moyenne (Mean) : indicateur de la valeur centrale du signal sur la fenêtre.

Écart-type (Standard Deviation) : mesure de la dispersion des valeurs, indicateur de variabilité.

Valeur efficace (RMS — Root Mean Square) : indicateur de l'énergie vibratoire, particulièrement pertinent pour les signaux de vibration.

Valeur maximale (Peak) : détecte les pics d'intensité pouvant signaler des chocs mécaniques.

Facteur de crête (Crest Factor) : rapport entre la valeur de crête et la valeur RMS, indicateur sensible aux chocs impulsionnels caractéristiques des défauts de roulements.

4.3 Détection d'anomalies simple : seuils et statistiques de base

Avant même de faire appel à des algorithmes d'apprentissage automatique, il est possible de détecter un grand nombre d'anomalies industrielles par des méthodes simples et robustes, fondées sur des seuils et des statistiques descriptives. Ces méthodes présentent l'avantage d'être facilement interprétables, rapides à calculer, et déployables directement au niveau de la gateway (Edge Computing).

4.3.1 La détection par seuils fixes

C'est la méthode la plus directe : on définit pour chaque indicateur mesuré une plage de fonctionnement normal (entre un seuil bas et un seuil haut). Toute valeur sortant de cette plage déclenche une alarme. Par exemple, si la valeur RMS de vibration d'une pompe dépasse 7,1 mm/s (seuil défini par la norme ISO 10816-3 pour les machines de classe II), une alarme est générée.

4.3.2 La détection par dérive statistique

Cette méthode surveille l'évolution dans le temps des indicateurs statistiques. Une dérive progressive de la valeur moyenne ou une augmentation continue de l'écart-type sur plusieurs jours — même sans dépassement de seuil absolu — peut signaler une dégradation lente de l'équipement. Cette approche est particulièrement adaptée à la détection précoce des défauts mécaniques à évolution lente.

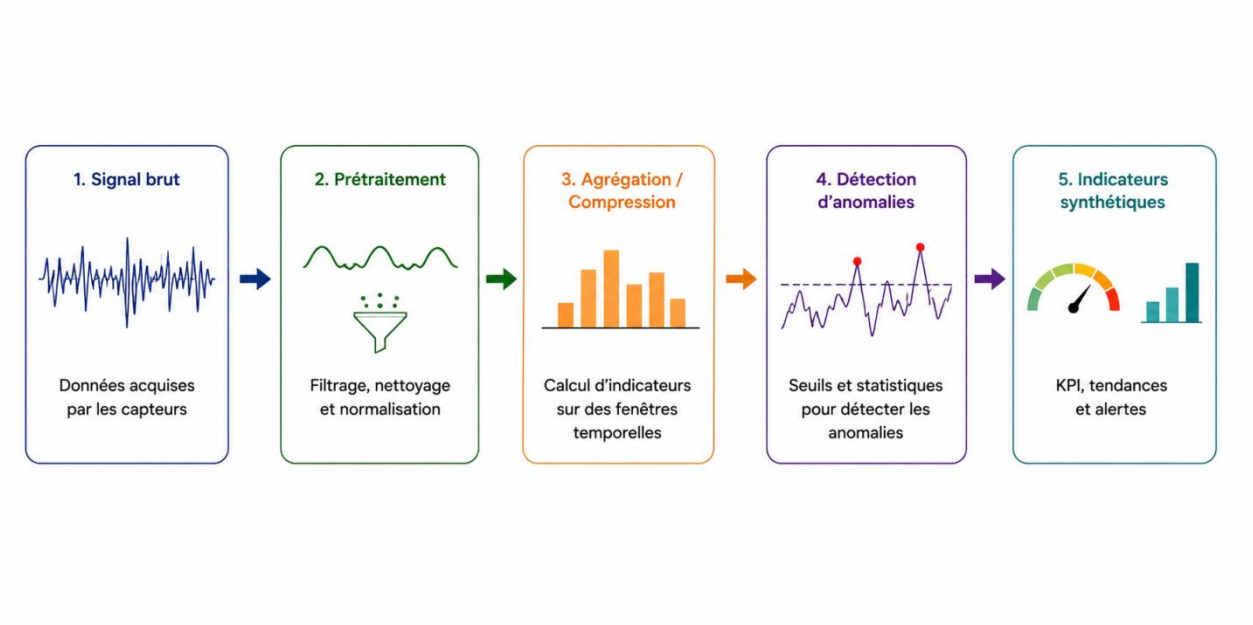


Figure 1.4 : Pipeline de traitement des données : du signal brut aux indicateurs synthétiques

5 La distribution à la demande

5.1 Concept de distribution à la demande

La **distribution à la demande** (on-demand data distribution) est un paradigme de partage de données dans lequel les informations ne sont pas diffusées de manière indifférenciée à tous les utilisateurs, mais transmises **spécifiquement à ceux qui en ont besoin, au moment où ils en ont besoin, dans le format adapté à leur usage**.

Dans un système IIoT industriel, les utilisateurs ont des besoins très différents vis-à-vis des mêmes données :

Le technicien de maintenance a besoin d'un tableau de bord temps réel affichant l'état actuel des équipements et les alarmes actives.

L'ingénieur d'analyse souhaite accéder à l'historique complet des mesures pour effectuer des analyses approfondies.

Le responsable de production consulte des indicateurs agrégés (KPI) sur des périodes hebdomadaires ou mensuelles.

Un système d'alerte automatique reçoit uniquement les événements déclenchant des alarmes.

La distribution à la demande implique donc la mise en place d'une architecture capable de gérer simultanément ces différents modes d'accès, en garantissant que chaque utilisateur reçoit l'information pertinente sans être noyé dans des données inutiles à son contexte.

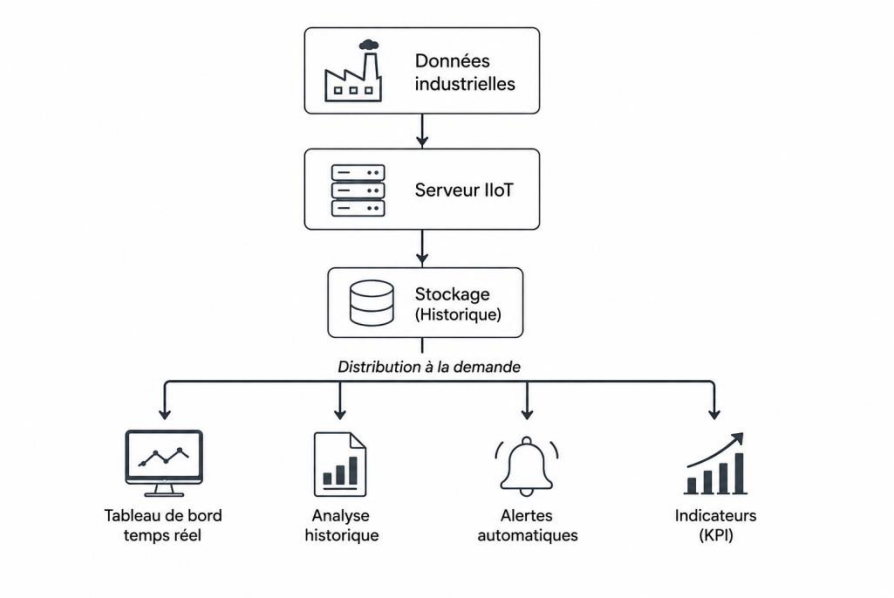


Figure 1.5 : Architecture de distribution à la demande

5.2 Architecture client-serveur et publish/subscribe

Deux modèles architecturaux complémentaires sont utilisés pour implémenter la distribution à la demande dans les systèmes IIoT :

5.2.1 Le modèle client-serveur (Request/Response)

Dans ce modèle, le client envoie une requête explicite au serveur, qui répond avec les données demandées. Ce modèle est adapté aux accès ponctuels : consultation de l'historique d'une mesure, téléchargement d'un rapport, accès à une valeur spécifique. Il est typiquement implémenté via des **API REST (HTTP)**, qui exposent les données sous forme de ressources accessibles par des requêtes GET, POST, PUT ou DELETE.

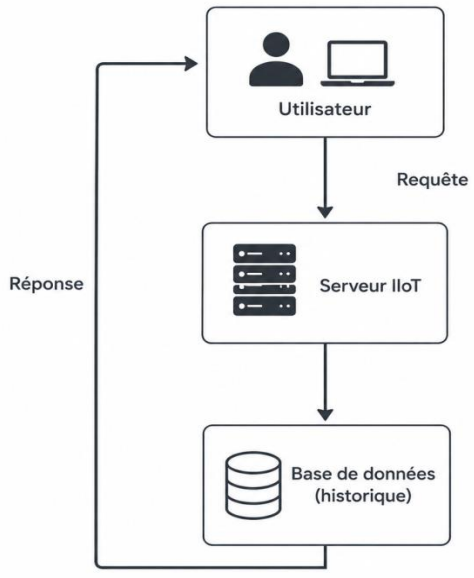


Figure 1.6 : Modèle client-serveur pour l'accès aux données historiques

5.2.2 Le modèle Publish/Subscribe

Comme décrit dans la section 1.3.2, le modèle Publish/Subscribe permet à des clients de s'abonner à des flux de données spécifiques et de recevoir automatiquement les nouvelles données dès leur disponibilité, sans avoir à effectuer des requêtes répétées. Ce modèle est idéalement adapté à la supervision en temps réel : un tableau de bord s'abonne aux topics MQTT correspondant aux équipements qu'il surveille, et se met à jour automatiquement à chaque nouvelle mesure.

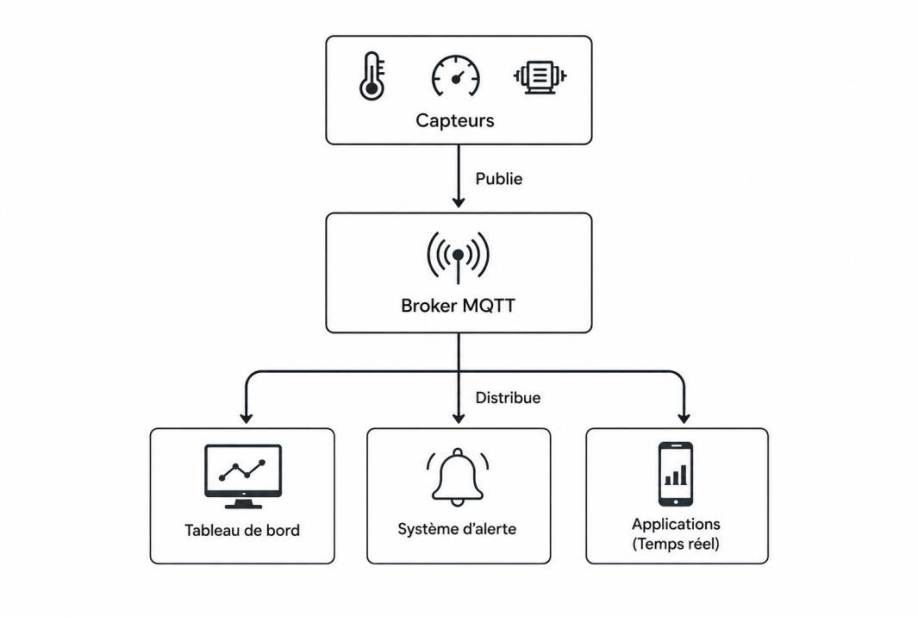


Figure 1.7 : Modèle publish/subscribe pour la diffusion des données en temps réel

5.3 Les plateformes de visualisation : Node-RED et Grafana

5.3.1 Node-RED

Node-RED est un outil de programmation visuelle par flux (flow-based programming), développé par IBM en 2013 et aujourd'hui maintenu par la OpenJS Foundation. Il permet de créer graphiquement des flux de traitement et de routage des données IoT par simple glisser-déposer de nœuds interconnectés, sans nécessiter d'expertise avancée en programmation [5].

Dans le contexte de ce projet, Node-RED joue le rôle de **moteur d'orchestration** : il reçoit les données depuis le broker MQTT, applique les transformations nécessaires (conversion d'unités, calcul d'indicateurs), les stocke dans InfluxDB, et les redistribue vers les différents clients selon leurs abonnements. Son module complémentaire node-red-dashboard permet de créer des interfaces de supervision interactives avec des jauges, des graphiques en temps réel, des voyants d'état et des boutons de commande.

5.3.2 Grafana

Grafana est la plateforme de référence mondiale en matière de visualisation de données de séries temporelles. Associée à la base de données InfluxDB, elle forme un couple particulièrement efficace pour la supervision industrielle [6]. Grafana se distingue par la qualité et la flexibilité de ses tableaux de bord, son système d'alertes avancé (définition de seuils, notifications par email ou webhook), et sa capacité à se connecter à de nombreuses sources de données hétérogènes.

Critère	Node-RED	Grafana
Fonction principale	Orchestration et routage des flux	Visualisation et supervision
Interface	Programmation visuelle par flux	Tableaux de bord configurables
Protocoles supportés	MQTT, HTTP, WebSocket, TCP	InfluxDB, MySQL, Prometheus, MQTT
Courbe d'apprentissage	Faible	Faible à modérée
Alertes	Basiques	Avancées (seuils, canaux multiples)
Usage typique	Traitement et routage des données	Visualisation et monitoring

Tableau 1.6 : Comparaison Node-RED et Grafana dans un contexte IIoT

5.4 Stockage des séries temporelles : InfluxDB

Les données industrielles collectées par un système IIoT constituent des **séries temporelles** — des séquences de valeurs numériques ordonnées dans le temps. Le stockage efficace de ce type de données nécessite une base de données spécialement conçue à cet effet, appelée **base de données temporelle (Time Series Database, TSDB)**.

InfluxDB, développé par InfluxData et disponible en open source depuis 2013, est la base de données temporelle la plus adoptée dans les applications IoT et de monitoring industriel [7]. Elle présente plusieurs caractéristiques particulièrement adaptées au contexte IIoT :

Performance d'ingestion élevée : capable d'ingérer plusieurs centaines de milliers de points de données par seconde.

Compression efficace : les algorithmes de compression intégrés réduisent considérablement l'espace de stockage nécessaire pour les séries temporelles.

Langage de requête dédié (Flux) : un langage fonctionnel spécialement conçu pour l'interrogation et la transformation des séries temporelles.

Politique de rétention configurable : possibilité de définir automatiquement la durée de conservation des données, supprimant automatiquement les données trop anciennes selon des règles prédéfinies.

Intégration native avec Grafana : InfluxDB est l'une des sources de données nativement supportées par Grafana, facilitant la mise en place de tableaux de bord de supervision.

6 Sécurité et fiabilité du système

6.1 Les menaces spécifiques aux systèmes IIoT

La sécurité est une dimension critique des systèmes IIoT, dont les enjeux dépassent largement ceux de l'IoT grand public. Une faille de sécurité dans un système de supervision industrielle peut avoir des conséquences graves : arrêt de production, manipulation des données de contrôle, sabotage des équipements, ou mise en danger du personnel.

Les principales menaces auxquelles sont exposés les systèmes IIoT sont les suivantes :

6.1.1 Les attaques Man-in-the-Middle (MitM) consistent à intercepter les communications entre les capteurs et le serveur pour lire ou modifier les données en transit. Dans un système IIoT non sécurisé, un attaquant pourrait ainsi injecter de fausses mesures, conduisant à des décisions erronées (par exemple, masquer une alarme critique).

6.1.2 Les attaques par déni de service (DoS/DDoS) visent à saturer le broker MQTT ou le serveur de collecte en générant un volume massif de messages artificiels, paralysant ainsi la collecte de données légitime.

6.1.3 L'accès non autorisé aux données résulte de l'absence d'authentification ou de l'utilisation d'identifiants par défaut sur les équipements IoT. Selon le rapport IBM X-Force Threat Intelligence (2023), plus de 60% des incidents de sécurité IoT exploitent des identifiants faibles ou inchangés [8].

6.1.4 La compromission des firmwares consiste à exploiter des vulnérabilités dans les logiciels embarqués des microcontrôleurs ou des gateways pour prendre le contrôle des équipements.

6.2 Les solutions essentielles

Face à ces menaces, un ensemble de bonnes pratiques de sécurité doit être intégré dès la conception du système IIoT, selon le principe du **Security by Design** :

6.2.1 Le chiffrement des communications via le protocole TLS (Transport Layer Security) garantit la confidentialité et l'intégrité des données en transit entre les capteurs, la gateway et le serveur. MQTT supporte nativement TLS sur le port 8883 (MQTTS).

6.2.2 L'authentification des dispositifs par nom d'utilisateur/mot de passe renforcé ou par certificats numériques X.509 garantit que seuls les équipements légitimes peuvent publier ou s'abonner aux données du système.

6.2.3 Le contrôle d'accès basé sur les rôles (RBAC) définit précisément les droits d'accès de chaque utilisateur ou application : un capteur ne peut publier que sur ses propres topics ; un utilisateur en lecture seule ne peut pas modifier la configuration du système.

6.2.4 La mise à jour régulière des firmwares des microcontrôleurs et des gateways est indispensable pour corriger les vulnérabilités découvertes. L'automatisation de ce processus (OTA — Over The Air updates) est recommandée pour les déploiements à grande échelle.

Menace	Impact potential	Solution recommandée
Interception des données (MitM)	Lecture ou modification des mesures	Chiffrement TLS/MQTTS
Accès non autorisé	Lecture ou manipulation des données	Authentification + RBAC
Déni de service (DoS)	Paralysie de la collecte	Rate limiting, monitoring réseau
Firmware compromis	Contrôle des équipements	Mises à jour OTA, signature de firmware
Injection de fausses données	Décisions erronées	Validation des données côté serveur

Tableau 1.7 : Synthèse des menaces IIoT et solutions associées

Conclusion

Ce chapitre a posé les fondements théoriques et techniques d'un système IoT industriel dédié à la collecte, au traitement et à la distribution des données. L'analyse du contexte de l'Industrie 4.0 a mis en évidence pourquoi la donnée industrielle est devenue un actif stratégique, et comment l'IoT constitue l'infrastructure indispensable à sa valorisation.

L'étude de l'architecture en trois couches — perception, réseau, application — a fourni un cadre structuré pour comprendre le rôle et les interactions de chaque composant du système. L'architecture hybride Edge/Cloud, qui combine le traitement local temps réel avec la puissance analytique centralisée, apparaît comme le choix le plus adapté aux exigences des applications industrielles.

L'examen des protocoles de communication — MQTT pour la télémétrie IoT et Modbus TCP pour l'intégration des équipements existants — a montré que ces deux protocoles sont complémentaires et couvrent ensemble l'essentiel des besoins de collecte dans un environnement industriel hétérogène.

Enfin, l'étude des mécanismes de traitement, de distribution et de stockage a démontré qu'un pipeline complet — du signal brut du capteur jusqu'au tableau de bord de l'utilisateur final — peut être construit autour d'outils open source matures : Node-RED pour l'orchestration, InfluxDB pour le stockage, et Grafana pour la visualisation.

Ces bases techniques constituent le socle sur lequel s'appuiera le deuxième chapitre, consacré à l'analyse intelligente des données par des méthodes d'apprentissage automatique, en vue de la détection prédictive des anomalies dans les équipements industriels.

Chapitre 2 : Analyse des Données et Intelligence Artificielle

Introduction

Le premier chapitre a décrit le parcours des données industrielles depuis leur naissance au niveau des capteurs jusqu'à leur distribution aux utilisateurs finaux. Ce parcours constitue la colonne vertébrale technique du système IoT étudié. Cependant, collecter et transmettre des données, aussi efficacement que soit l'infrastructure mise en place, ne constitue qu'une première étape. La véritable valeur ajoutée d'un système IoT industriel intelligent réside dans sa capacité à **extraire du sens** de ces données — à transformer des milliers de mesures brutes en informations exploitables, en alertes pertinentes, et en décisions éclairées.

C'est précisément l'objet de ce deuxième chapitre. Il s'inscrit dans la continuité directe du premier en répondant à une question centrale : une fois les données collectées sur le terrain, filtrées et stockées, comment les exploiter intelligemment pour anticiper les défaillances des équipements industriels ?

Pour répondre à cette question, ce chapitre introduit les concepts fondamentaux de l'intelligence artificielle appliquée aux données industrielles. Il commence par justifier le recours au Machine Learning en montrant les limites des approches classiques, puis présente les deux grandes familles d'algorithmes au cœur de ce projet : l'**apprentissage supervisé**, utilisé pour classifier les états de fonctionnement des équipements, et l'**apprentissage non supervisé**, utilisé pour détecter des anomalies sans connaissance préalable des pannes. Le chapitre se conclut par une comparaison pratique des algorithmes étudiés, orientée vers leur application dans le contexte de la maintenance prédictive des pompes industrielles.

1 L'intégration de l'intelligence artificielle (IA) et le passage la donnée brute à la connaissance

1.1 Les limites des méthodes classiques de traitement de données

Dans le chapitre précédent, nous avons présenté deux approches simples de détection d'anomalies : la détection par seuils fixes et la surveillance de la dérive statistique. Ces méthodes, bien que rapides et faciles à mettre en œuvre, présentent des limitations fondamentales qui les rendent insuffisantes pour des applications de maintenance prédictive avancée.

1.1.1 La rigidité des seuils fixes

Un seuil fixe suppose que le comportement normal d'un équipement est constant et bien connu à l'avance. Or, dans la réalité industrielle, le comportement normal d'une pompe varie selon son régime de fonctionnement, sa charge, la température ambiante, et l'état du fluide pompé. Un niveau de vibration parfaitement normal à pleine charge peut être considéré comme anormal à faible charge si l'on applique un seuil unique et rigide. Cette inflexibilité génère deux types d'erreurs coûteuses :

Les fausses alarmes (faux positifs) : le système déclenche une alerte alors que l'équipement fonctionne normalement, mobilisant inutilement les équipes de maintenance.

Les défauts non détectés (faux négatifs) : un défaut réel se développe en restant sous le seuil d'alarme, jusqu'à provoquer une panne franche.

1.1.2 L'incapacité à détecter les patterns complexes

Les défauts mécaniques réels — usure de roulement, cavitation, déséquilibre du rotor, défaut d'alignement — se manifestent chacun par une **signature vibratoire spécifique et complexe**, impliquant des combinaisons de fréquences caractéristiques. Aucune méthode basée sur des statistiques simples (moyenne, écart-type, valeur RMS) ne peut distinguer ces signatures les unes des autres, ni les identifier avec fiabilité parmi le bruit de fond.

1.1.3 L'absence de capacité d'apprentissage

Les méthodes classiques sont statiques : elles ne s'améliorent pas avec le temps et ne tiennent pas compte de l'évolution progressive de l'état de l'équipement. Elles ne peuvent pas non plus capitaliser sur l'expérience accumulée — les historiques de pannes passées — pour améliorer leur précision future.

C'est face à ces limitations que le Machine Learning apporte une réponse fondamentalement différente et plus puissante.

1.2 Machine Learning, définition et positionnement

Le Machine Learning (apprentissage automatique) est une branche de l'IA qui donne aux systèmes informatiques la capacité d'**apprendre à partir des données**, sans être explicitement programmés pour chaque tâche spécifique. Plutôt que de définir manuellement des règles de décision, on fournit au système des exemples à partir desquels il construit lui-même un modèle de décision.

Cette définition, proposée par Arthur Samuel dès 1959, reste la plus citée dans la littérature [16]. Tom Mitchell (1997) l'a précisée en ces termes : *"Un programme informatique apprend à partir d'une expérience E, par rapport à une tâche T et une mesure de performance P, si sa performance sur T, mesurée par P, s'améliore avec l'expérience E"* [17].

Pour situer le Machine Learning dans le paysage plus large de l'intelligence artificielle, la figure 1 illustre la hiérarchie suivante :

L'Intelligence Artificielle (IA) est le domaine le plus large, englobant toute technique permettant à une machine de simuler des comportements intelligents.

Le Machine Learning est une sous-discipline de l'IA qui repose sur l'apprentissage à partir des données.

Le Deep Learning est une sous-discipline du Machine Learning qui utilise des réseaux de neurones à plusieurs couches pour traiter des données complexes (images, sons, séries temporelles longues).

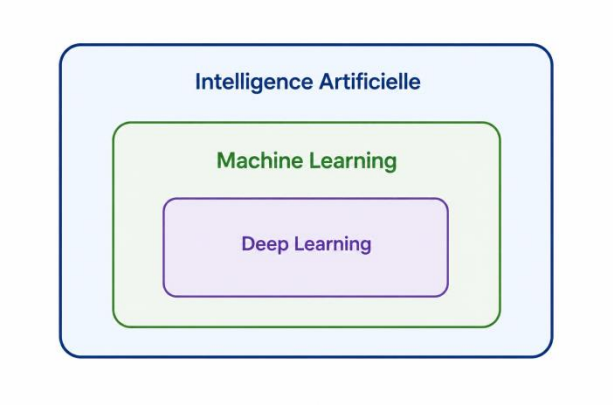


Figure 2.1 : Positionnement du Machine Learning dans l'Intelligence Artificielle

Dans le contexte de ce projet, nous nous concentrons sur le Machine Learning classique — sans Deep Learning — pour deux raisons pratiques : les algorithmes classiques sont suffisamment performants pour les données vibratoires de pompes industrielles, et ils sont beaucoup plus faciles à interpréter et à défendre devant un jury technique.

Famille	Principe	Données requises	Exemples d'algorithmes
Apprentissage supervisé	Apprendre à partir d'exemples étiquetés	Données avec labels (état normal / défaut)	Random Forest, SVM, Régression logistique
Apprentissage non supervisé	Découvrir des structures cachées sans labels	Données sans étiquettes	K-Means, DBSCAN, Autoencodeurs
Apprentissage par renforcement	Apprendre par essai-erreur avec récompenses	Interactions avec l'environnement	Q-Learning, PPO

Tableau 2.1 : Les trois grandes familles du Machine Learning.

Dans le cadre de ce projet, seules les deux premières familles — apprentissage supervisé et non supervisé — seront utilisées, en fonction de la disponibilité des données étiquetées.

1.3 Prétraitement spécifique au ML : normalisation et encodage

Avant d'alimenter un algorithme de Machine Learning, les données doivent subir un prétraitement adapté aux exigences spécifiques de ces algorithmes. Ce prétraitement est distinct du filtrage et du nettoyage décrits dans le chapitre précédent — il concerne la **mise en forme mathématique** des données pour les rendre compatibles avec les modèles d'apprentissage.

1.3.1 La normalisation (Feature Scaling)

La plupart des algorithmes de Machine Learning sont sensibles à l'échelle des variables d'entrée. Si une variable prend des valeurs entre 0 et 1 (par exemple, un coefficient d'amortissement) et une autre entre 0 et 10 000 (par exemple, une fréquence en Hz), l'algorithme accordera artificiellement plus d'importance à la seconde variable, non pas parce qu'elle est plus pertinente, mais simplement parce que ses valeurs sont numériquement plus grandes.

Deux méthodes de normalisation sont couramment utilisées :

Lanormalisation Min-Max ramène toutes les valeurs dans l'intervalle [0, 1] selon la formule :

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Elle est adaptée lorsque la distribution des données est connue et bornée.

Lastandardisation (Z-score) centre les données autour de zéro avec un écart-type de 1 :

$$x_{std} = \frac{x - \mu}{\sigma}$$

Elle est préférable lorsque les données suivent une distribution approximativement gaussienne, ce qui est fréquemment le cas pour les indicateurs vibratoires.

1.3.2 L'encodage des variables catégorielles

Les algorithmes de Machine Learning opèrent sur des valeurs numériques. Or, certaines variables descriptives sont de nature catégorielle — par exemple, le type de défaut (« roulement usé », « déséquilibre », « cavitation »). Ces variables doivent être converties en représentations numériques avant d'être utilisées comme entrées ou sorties d'un modèle.

La méthode la plus simple est l'**encodage ordinal** (0, 1, 2...), adaptée aux variables avec un ordre naturel. Pour les variables sans ordre intrinsèque, on utilise le **One-Hot Encoding**, qui crée une colonne binaire distincte pour chaque modalité.

Étape	Objectif	Méthode	Applicable à
Nettoyage	Supprimer valeurs aberrantes et manquantes	Règle 3σ, interpolation	Toutes les données
Normalisation Min-Max	Mettre à l'échelle [0,1]	$\frac{x - x_{min}}{x_{max} - x_{min}}$	Données bornées
Standardisation Z-score	Centrer-réduire	$\frac{x - \mu}{\sigma}$	Données gaussiennes
Encodage One-Hot	Convertir variables catégorielles	Colonnes binaires	Variables de type texte
Extraction de features	Calculer indicateurs pertinents	RMS, Peak, Kurtosis	Signaux temporels

Tableau 2.2 : Récapitulatif des étapes de prétraitement pour le Machine Learning

2 Apprentissage supervisé (Supervised Learning)

2.1 Principe et fonctionnement

L'apprentissage supervisé est la famille d'algorithmes de Machine Learning la plus utilisée dans les applications industrielles. Son principe repose sur un apprentissage à partir d'**exemples étiquetés** : on fournit à l'algorithme un ensemble de données d'entraînement pour lesquelles la réponse correcte est déjà connue, et l'algorithme apprend à généraliser cette connaissance pour prédire la réponse sur de nouvelles données jamais vues.

Le processus d'apprentissage supervisé se déroule en deux phases distinctes :

2.1.1 La phase d'entraînement (Training) : l'algorithme reçoit un jeu de données d'entraînement composé de paires (entrée, sortie attendue). Il ajuste itérativement ses paramètres internes pour minimiser l'écart entre ses prédictions et les sorties attendues, selon une fonction de coût définie. Dans notre contexte, les entrées sont les indicateurs vibratoires calculés (RMS, Peak, Kurtosis...) et les sorties attendues sont les étiquettes d'état (« normal », « roulement défectueux », « déséquilibre »...).

2.1.2La phase de prédiction (Inference) : une fois entraîné, le modèle est déployé en production. Il reçoit de nouvelles mesures vibratoires en temps réel et prédit automatiquement l'état de l'équipement, sans intervention humaine.

La qualité d'un modèle supervisé est évaluée par plusieurs métriques calculées sur un jeu de données de test — des données non utilisées lors de l'entraînement :

L'accuracy : proportion de prédictions correctes parmi toutes les prédictions.

La précision : parmi les alarmes déclenchées, quelle proportion correspond à de vraies pannes ?

Le rappel (recall) : parmi toutes les vraies pannes, quelle proportion a été détectée ?

Le F1-score : moyenne harmonique de la précision et du rappel, indicateur global équilibré.

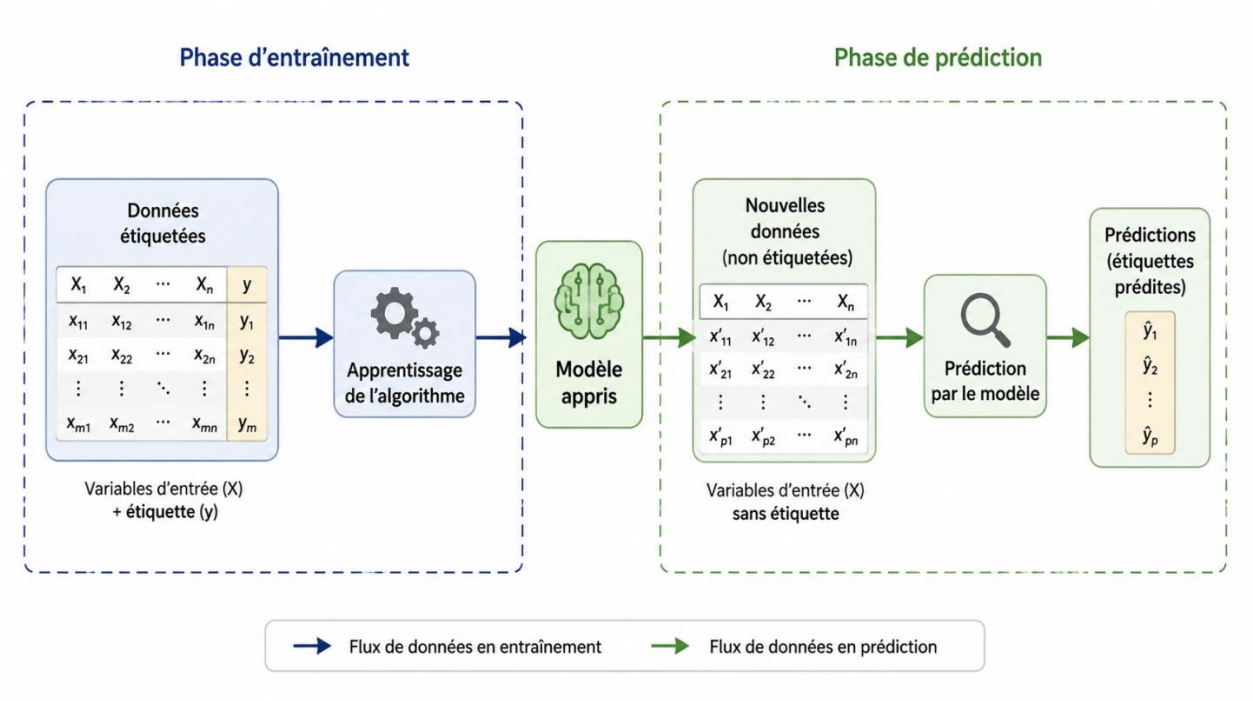


Figure 2.2 : Principe de l'apprentissage supervisé : phase d'entraînement et phase de prédiction

2.2 Algorithmes clés : Random Forest, SVM, Régression logistique

2.2.1 La Régression Logistique

Malgré son nom, la régression logistique est un algorithme de **classification** et non de régression. Elle modélise la probabilité qu'une observation appartienne à une classe donnée, en appliquant la fonction sigmoïde à une combinaison linéaire des variables d'entrée. C'est l'algorithme le plus simple de l'apprentissage supervisé, souvent utilisé comme référence de base (baseline) avant d'essayer des modèles plus complexes [18].

Son principal avantage réside dans son **interprétabilité** : les coefficients du modèle indiquent directement l'influence de chaque variable d'entrée sur la décision finale. Dans notre contexte, un modèle de régression logistique permettrait par exemple d'identifier quels indicateurs vibratoires (RMS, Peak, Kurtosis) contribuent le plus à la discrimination entre état normal et état défectueux.

2.2.2 Les Machines à Vecteurs de Support (SVM)

Le SVM (Support Vector Machine) est un algorithme de classification qui cherche à trouver l'**hyperplan de séparation optimal** entre les classes, c'est-à-dire celui qui maximise la marge entre les exemples les plus proches de chaque classe (appelés vecteurs de support). Cette approche géométrique lui confère une excellente capacité de généralisation, même avec des jeux de données de taille modeste [19].

Une caractéristique importante du SVM est l'utilisation du **noyau (kernel)** : en projetant les données dans un espace de dimension supérieure, le noyau permet au SVM de tracer des frontières de décision non linéaires dans l'espace original des données. Le noyau RBF (Radial Basis Function) est le plus couramment utilisé pour les données vibratoires industrielles.

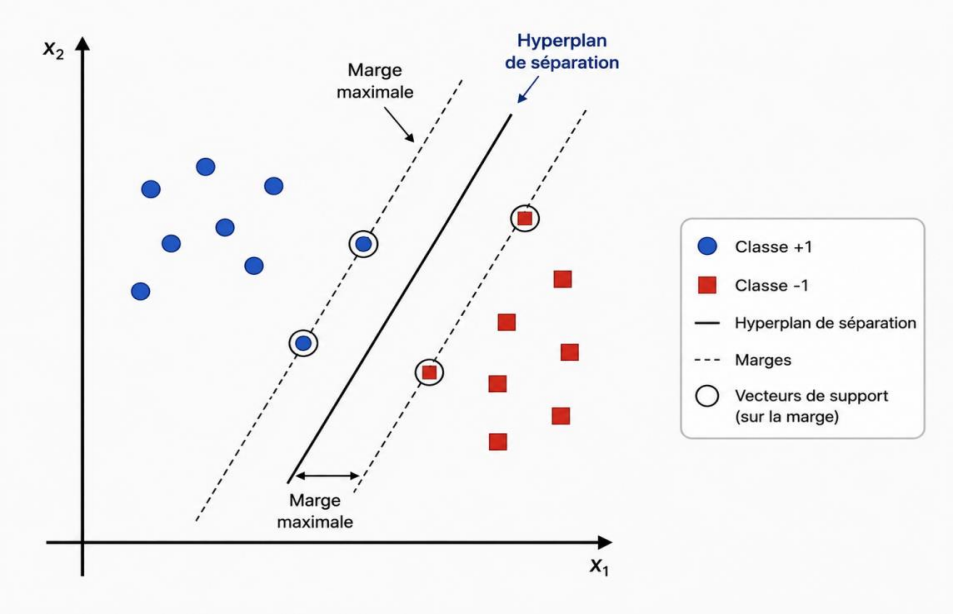


Figure 2.3 : Illustration du principe SVM — hyperplan de séparation et marge maximale

2.2.3 La Forêt Aléatoire (Random Forest)

Le Random Forest est une méthode d'**ensemble learning** qui construit un grand nombre d'arbres de décision indépendants lors de l'entraînement, puis agrège leurs prédictions par vote majoritaire pour la classification finale. Chaque arbre est entraîné sur un sous-échantillon aléatoire des données et utilise un sous-ensemble aléatoire des variables, ce qui garantit la diversité des arbres et réduit le risque de surapprentissage (overfitting) [20].

Le Random Forest présente plusieurs avantages particulièrement appréciables dans le contexte industriel :

Robustesse au bruit : la moyenne des prédictions de nombreux arbres élimine l'influence des données aberrantes résiduelles.

Importance des variables (Feature Importance) : le modèle calcule automatiquement l'importance relative de chaque variable d'entrée, fournissant une information précieuse sur les indicateurs vibratoires les plus pertinents.

Peu de paramètres à régler : contrairement au SVM, le Random Forest fonctionne généralement bien avec ses paramètres par défaut.

Algorithme	Complexité	Interprétabilité	Performance	Sensibilité au bruit	Cas d'usage idéal
Régression logistique	Faible	Très élevée	Modérée	Modérée	Baseline, données linéaires
SVM	Modérée	Faible	Élevée	Faible	Données de taille modeste
Random Forest	Modérée	Modérée	Très élevée	Très faible	Données industrielles hétérogènes

Tableau 2.3 : Comparaison des algorithmes d'apprentissage supervisé

2.3 Application industrielle : classification de pannes

Dans le cadre de la maintenance prédictive des équipements rotatifs — et des pompes en particulier — l'apprentissage supervisé est utilisé pour résoudre un problème de **classification multi-classes** : à partir des indicateurs vibratoires calculés en temps réel, déterminer automatiquement à quelle classe d'état appartient l'équipement.

Les classes typiquement définies pour une pompe industrielle sont :

État normal : les vibrations sont dans les plages de référence, aucune dégradation détectée.

Défaut de roulement : usure ou écaillage des billes ou des pistes, se manifestant par des fréquences caractéristiques liées aux fréquences de passage des billes (BPFO, BPFI).

Déséquilibre du rotor : masse asymétrique sur l'axe rotatif, générant une composante vibratoire dominante à la fréquence de rotation.

Défaut d'alignement : désalignement angulaire ou parallèle entre la pompe et son moteur d'entraînement.

Cavitation : formation et implosion de bulles de vapeur dans le fluide pompé, générant un bruit large bande caractéristique.

La figure suivante illustre le pipeline complet d'un système de classification supervisée appliqué à la surveillance vibratoire d'une pompe :

Un point critique dans l'application de l'apprentissage supervisé à la maintenance industrielle est la **disponibilité des données étiquetées**. En effet, pour entraîner un modèle, il faut disposer d'un historique de mesures pour lequel l'état réel de l'équipement est connu avec certitude. Cet historique est souvent difficile à constituer, car les pannes industrielles sont — heureusement — relativement rares. Cette limitation est précisément ce qui motive l'utilisation complémentaire de l'apprentissage non supervisé, présenté dans la section suivante.

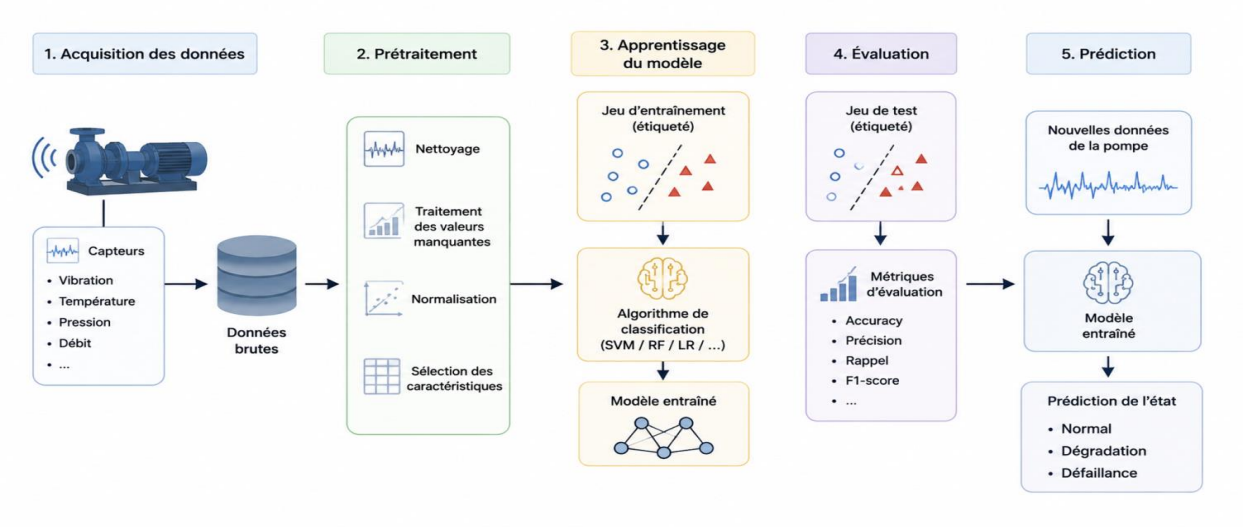


Figure 2.4 : Pipeline de classification supervisée pour la maintenance prédictive de pompes

3 Apprentissage non supervisé (Unsupervised Learning)

3.1 Principe et différence avec l'apprentissage supervisé

L'apprentissage non supervisé aborde le problème de l'analyse des données sous un angle fondamentalement différent : il n'utilise **aucune étiquette**, aucune réponse correcte préalablement définie. L'algorithme explore librement la structure des données pour y découvrir des patterns, des groupements naturels, ou des points inhabituels — sans avoir été guidé vers une réponse attendue.

Cette différence fondamentale est illustrée par une analogie simple : l'apprentissage supervisé ressemble à un élève qui apprend avec un professeur qui corrige ses exercices (il connaît les bonnes réponses) ; l'apprentissage non supervisé ressemble à un explorateur qui découvre une nouvelle région sans carte ni guide — il identifie lui-même les reliefs, les regroupements géographiques, et les zones inhabituelles.

Dans le contexte de la maintenance industrielle, l'apprentissage non supervisé est particulièrement précieux dans deux situations :

Absence d'historique de pannes étiquetées : lorsqu'un équipement est nouveau ou que les pannes passées n'ont pas été documentées, il est impossible d'entraîner un modèle supervisé. L'apprentissage non supervisé permet néanmoins de détecter des comportements anormaux en s'appuyant uniquement sur la connaissance du fonctionnement normal.

Détection de pannes inconnues : un modèle supervisé ne peut détecter que les types de pannes présents dans ses données d'entraînement. L'apprentissage non supervisé peut en revanche signaler tout comportement déviant de la normale, même s'il correspond à un type de défaut jamais rencontré auparavant.

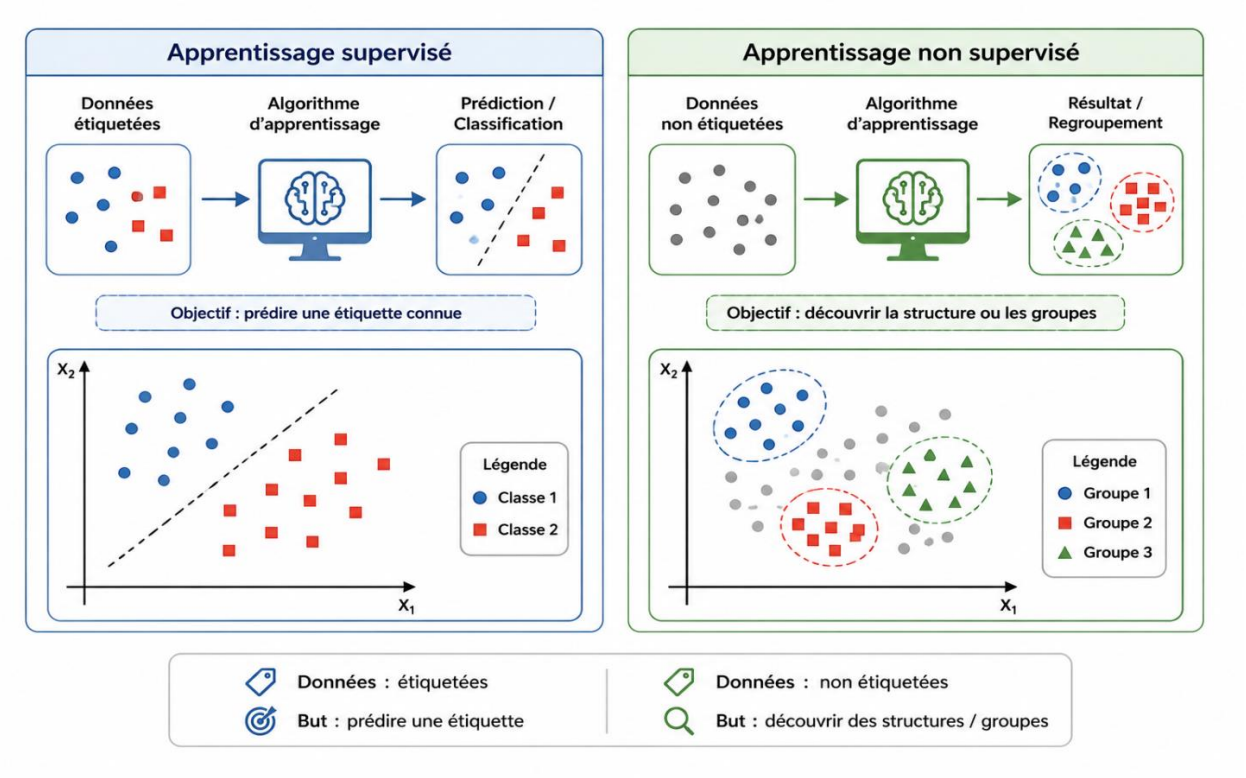


Figure 2.5 : Comparaison visuelle apprentissage supervisé vs non supervisé

Critère	Apprentissage supervisé	Apprentissage non supervisé
Données requises	Étiquetées (labels obligatoires)	Non étiquetées
Objectif	Prédire une classe ou une valeur	Découvrir des structures cachées
Interprétation	Directe (classe prédite connue)	Nécessite une analyse humaine
Détection de pannes inconnues	Non	Oui
Exemples d'application	Classification d'état, diagnostic	Détection d'anomalies, clustering

Tableau 2.4 : Comparaison apprentissage supervisé et non supervisé

3.2 Algorithmes clés : K-Means, DBSCAN, Autoencodeurs

3.2.1 K-Means

K-Means est l'algorithme de clustering (regroupement) le plus connu et le plus utilisé. Son principe est simple : il partitionne les données en **K groupes (clusters)** en minimisant la distance entre chaque point et le centre (centroïde) de son cluster. L'algorithme procède de manière itérative en quatre étapes : initialisation aléatoire des K centroïdes, affectation de chaque point au centroïde le plus proche, recalcul des centroïdes comme la moyenne des points affectés, et répétition jusqu'à convergence [21].

Dans le contexte de la maintenance prédictive, K-Means peut être utilisé pour regrouper automatiquement les données vibratoires en clusters correspondant à différents modes de fonctionnement. Un point qui n'appartient clairement à aucun cluster établi — ou qui se retrouve dans un cluster de très petite taille — est un candidat sérieux à l'anomalie.

La principale limitation de K-Means est la nécessité de **définir à l'avance le nombre K de clusters**, ce qui requiert une connaissance préalable du nombre d'états possibles de l'équipement. De plus, K-Means suppose que les clusters sont sphériques et de taille similaire, ce qui n'est pas toujours le cas dans la réalité.

3.2.2 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un algorithme de clustering fondé sur la **densité locale** des données plutôt que sur la distance aux centroïdes. Il définit un cluster comme une région de l'espace des données où les points sont suffisamment denses (plus de MinPts voisins dans un rayon ε). Les points isolés, qui n'appartiennent à aucune région dense, sont automatiquement étiquetés comme **bruit (noise)** — et constituent précisément les anomalies recherchées [22].

DBSCAN présente deux avantages décisifs sur K-Means pour la détection d'anomalies industrielles :

Il **ne requiert pas de spécifier le nombre de clusters** à l'avance.

Il **identifie nativement les points aberrants** comme une classe à part entière, ce qui en fait un outil naturellement adapté à la détection d'anomalies.

3.2.3 Les Autoencodeurs

Un autoencodeur est un type de réseau de neurones entraîné à **reconstruire ses propres données d'entrée** après les avoir compressées dans une représentation de dimension réduite (appelée espace latent). Il se compose de deux parties symétriques : l'encodeur, qui compresse l'entrée, et le décodeur, qui reconstruit l'entrée originale à partir de la représentation compressée.

L'idée centrale pour la détection d'anomalies est la suivante : un autoencodeur entraîné **exclusivement sur des données normales** apprend à reconstruire fidèlement les patterns du fonctionnement normal. Lorsqu'il est confronté à une donnée anormale — correspondant à un état de défaut — il ne sait pas la reconstruire correctement, car ce pattern ne figurait pas dans ses données d'entraînement. **L'erreur de reconstruction** (écart entre l'entrée et la sortie) devient ainsi un indicateur direct d'anomalie : une erreur élevée signale un comportement inhabituel [23].

3.3 Application industrielle : détection d'anomalies

La détection d'anomalies (Anomaly Detection) est l'application phare de l'apprentissage non supervisé dans le contexte de la maintenance prédictive industrielle. Elle vise à identifier automatiquement les comportements inhabituels des équipements avant qu'ils ne se traduisent par des pannes franches.

Le pipeline typique d'un système de détection d'anomalies par apprentissage non supervisé se déroule en trois phases :

Phase 1 — Apprentissage du comportement normal : pendant une période de fonctionnement normal bien documentée, l'algorithme (K-Means, DBSCAN ou Autoencodeur) apprend les caractéristiques du fonctionnement sain de l'équipement. Cette phase ne nécessite aucune étiquette — seule la certitude que l'équipement fonctionne normalement durant cette période est requise.

Phase 2 — Définition du seuil d'anomalie: sur la base des données d'entraînement, on définit un seuil au-delà duquel un comportement est considéré comme anormal. Pour K-Means, ce seuil est une distance maximale au centroïde ; pour l'Autoencodeur, c'est une erreur de reconstruction maximale.

Phase 3 — Surveillance en temps réel : en production, chaque nouvelle mesure est évaluée par le modèle. Si le score d'anomalie dépasse le seuil défini, une alerte est générée et transmise aux équipes de maintenance via le système de distribution à la demande décrit dans le chapitre précédent.

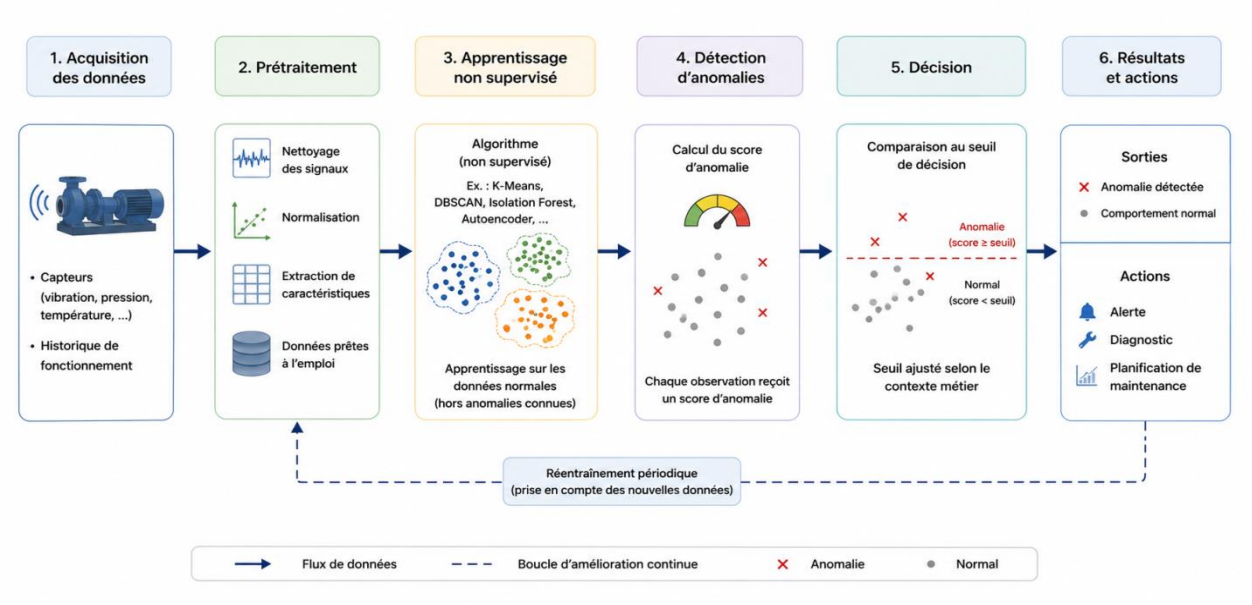


Figure 2.6 : Pipeline de détection d'anomalies par apprentissage non supervisé

Algorithme	Type	Avantages	Limitations	Adapté pour
K-Means	Clustering	Simple, rapide	K à définir, clusters sphériques	Données bien séparées
DBSCAN	Clustering densité	Détecte les outliers nativement	Sensible aux paramètres ε et MinPts	Données avec bruit
Autoencodeur	Réseau de neurons	Capture patterns complexes	Plus complexe à entraîner	Séries temporelles longues

Tableau 2.5 : Comparaison des algorithmes de détection d'anomalies

4 Comparaison et choix des algorithmes

4.1 Tableau comparatif supervisé vs non supervisé

Le choix entre apprentissage supervisé et non supervisé ne doit pas être perçu comme une opposition, mais comme une **complémentarité** dictée par la nature des données disponibles et les objectifs opérationnels. Le tableau suivant synthétise les critères de choix entre ces deux approches dans le contexte de la maintenance industrielle :

Critère	Supervisé	Non supervisé
Données nécessaires	Historique étiqueté (pannes connues)	Données de fonctionnement normal uniquement
Types de pannes détectables	Uniquement les pannes connues	Tout comportement déviant, y compris pannes inconnues
Précision de diagnostic	Élevée (identifie le type de panne)	Modérée (signale l'anomalie sans l'identifier)
Facilité de mise en œuvre	Dépend de la qualité des labels	Plus facile à déployer sur équipement neuf
Interprétabilité	Élevée	Modérée
Évolution dans le temps	Nécessite ré-entraînement	S'adapte naturellement
Exemple d'algorithme	Random Forest, SVM	DBSCAN, Autoencodeur

Tableau 2.6 : Tableau comparatif complet — Supervisé vs Non supervisé

4.2 Quel algorithme pour quel contexte industriel ?

Le choix d'un algorithme spécifique au sein de chaque famille dépend de plusieurs facteurs pratiques : la taille du jeu de données disponible, la nature des données (linéaires ou non), les ressources de calcul disponibles, et surtout l'exigence d'interprétabilité du modèle — car dans un contexte industriel, un technicien de maintenance doit pouvoir comprendre et faire confiance aux recommandations du système.

Pour un système de maintenance prédictive de pompes industrielles, la stratégie recommandée est une **approche hybride en deux niveaux** :

Niveau 1 — Détection (Non supervisé) : un modèle DBSCAN ou un Autoencodeur surveille en permanence le flux de données vibratoires et signale tout comportement déviant. Ce niveau fonctionne en continu, sans intervention humaine, et constitue le premier filet de sécurité.

Niveau 2 — Diagnostic (Supervisé): lorsqu'une anomalie est détectée au niveau 1, un modèle Random Forest classifie l'anomalie détectée parmi les types de défauts connus, permettant aux équipes de maintenance d'intervenir avec une connaissance précise de la nature du problème.

Cette architecture à deux niveaux combine les avantages des deux familles : la capacité de détection universelle du non supervisé et la précision diagnostique du supervisé.

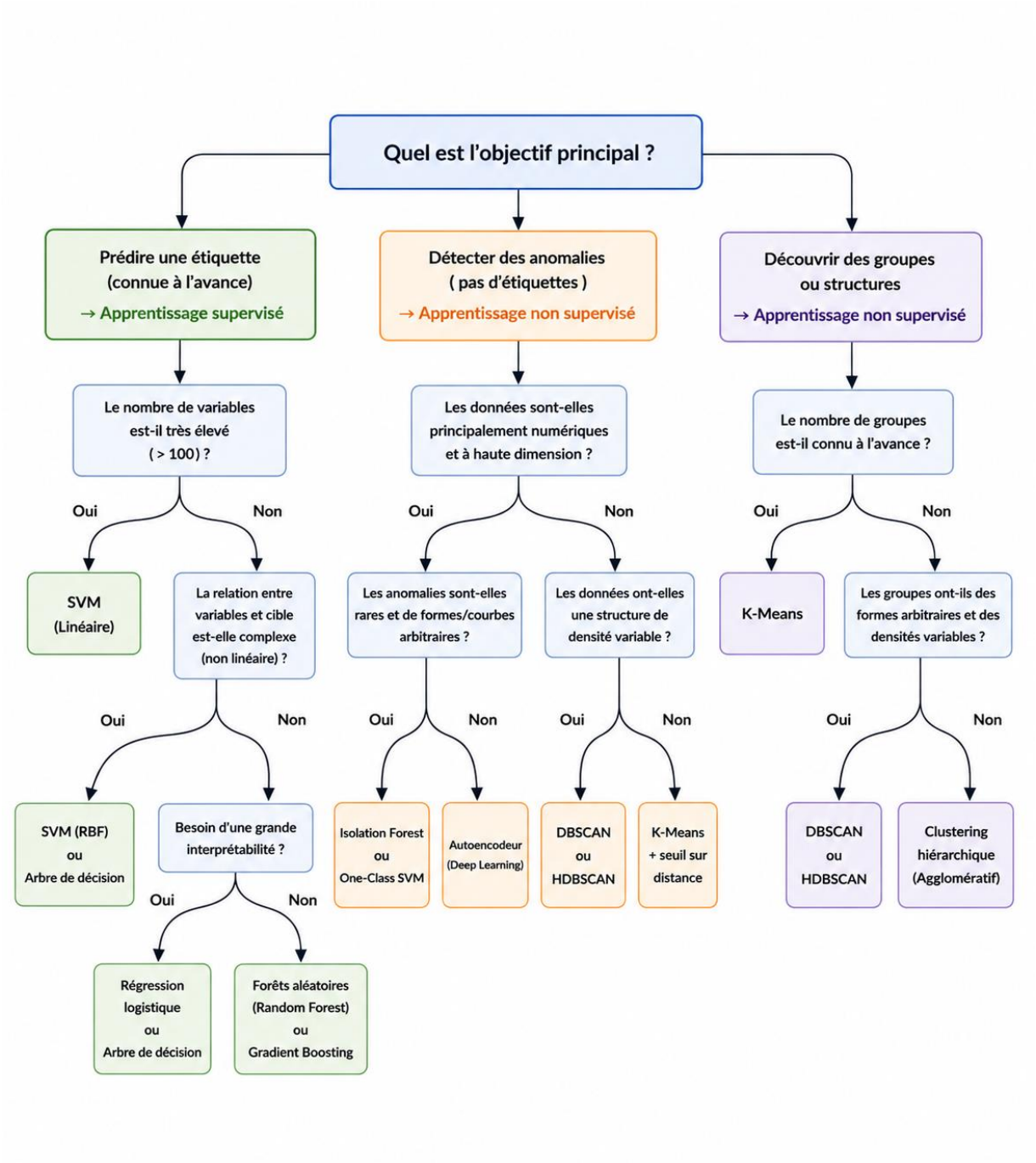


Figure 2.8 : Arbre de décision pour le choix de l'algorithme selon le contexte industriel

4.3 Outils utilisés : Python, Scikit-learn, Pandas

L'implémentation des algorithmes de Machine Learning présentés dans ce chapitre repose sur un écosystème de bibliothèques Python open source, qui constitue aujourd'hui le standard de facto dans le domaine de la data science et du Machine Learning industriel.

4.3.1 Python est le langage de programmation de référence pour le Machine Learning, grâce à sa lisibilité, sa richesse en bibliothèques spécialisées, et son adoption massive par la communauté scientifique et industrielle.

4.3.2 Pandas est la bibliothèque de manipulation et d'analyse de données tabulaires. Elle permet de charger, nettoyer, transformer et explorer les jeux de données vibratoires sous forme de DataFrames — structures de données similaires à des tableaux Excel, mais manipulables programmatiquement.

4.3.3 Scikit-learn est la bibliothèque de référence pour le Machine Learning classique en Python. Elle fournit des implémentations standardisées et optimisées de tous les algorithmes présentés dans ce chapitre — Régression Logistique, SVM, Random Forest, K-Means, DBSCAN — ainsi que des outils de prétraitement (normalisation, encodage) et d'évaluation des modèles (métriques, validation croisée) [24].

Bibliothèque	Version recommandée	Fonction principale	Usage dans le projet
Python	3.10+	Langage de programmation	Base de tout le développement
Pandas	2.0+	Manipulation des données tabulaires	Chargement et nettoyage des données vibratoires
NumPy	1.24+	Calcul numérique	Calcul des indicateurs (RMS, Peak, Kurtosis)
Scikit-learn	1.3+	Algorithmes ML	Entraînement et évaluation des modèles
Matplotlib / Seaborn	3.7+	Visualisation	Graphiques d'analyse et de résultats

Tableau 2.7 : Bibliothèques Python utilisées et leurs fonctions

Conclusion

Ce chapitre a établi le lien conceptuel et technique entre l'infrastructure IoT décrite dans le premier chapitre et l'intelligence artificielle nécessaire à l'exploitation avancée des données collectées. En partant des limites des méthodes classiques de détection d'anomalies, il a justifié le recours au Machine Learning comme réponse naturelle et puissante aux besoins de la maintenance prédictive industrielle.

L'étude de l'apprentissage supervisé — à travers la Régression Logistique, le SVM et le Random Forest — a montré comment un système peut apprendre à identifier automatiquement des types de défauts spécifiques à partir d'un historique de données étiquetées. L'étude de l'apprentissage non supervisé — à travers K-Means, DBSCAN et les Autoencodeurs — a démontré comment des anomalies peuvent être détectées sans aucune connaissance préalable des pannes, ce qui est particulièrement précieux pour les équipements neufs ou les défauts inédits.

La comparaison finale des deux familles d'algorithmes a conduit à recommander une **architecture hybride à deux niveaux** : détection non supervisée en première ligne, diagnostic supervisé en deuxième ligne. Cette architecture constitue le fondement algorithmique du système de maintenance prédictive qui sera développé dans les chapitres suivants.

Les deux chapitres présentés — infrastructure IoT d'une part, intelligence artificielle d'autre part — forment ensemble le socle théorique complet sur lequel reposera la conception et l'implémentation du système intégré IoT-IA de maintenance prédictive des pompes industrielles.

Chapitre 3 : Intégration IoT et IntelligenceArtificielle — Étude de Cas : Maintenance Prédictive d'une Pompe Industrielle

Introduction

Les deux chapitres précédents ont établi les fondements théoriques et techniques nécessaires à la compréhension du système étudié. Le premier chapitre a décrit l'infrastructure IoT industrielle — collecte, traitement et distribution des données — tandis que le second a présenté les méthodes d'intelligence artificielle applicables à l'analyse de ces données, en distinguant l'apprentissage supervisé de l'apprentissage non supervisé.

Ce troisième chapitre constitue la synthèse concrète de ces deux volets théoriques. Il présente une **étude de cas complète et simulée** illustrant l'intégration d'un système IoT avec des algorithmes de Machine Learning, appliquée à un problème industriel réel : la **maintenance prédictive d'une pompe industrielle** par analyse de ses vibrations.

L'étude est réalisée sous l'environnement **MATLAB**, qui offre un cadre de simulation rigoureux permettant de reproduire fidèlement le comportement d'un système IoT industriel réel — depuis la réception des signaux bruts des capteurs jusqu'à la distribution des informations traitées aux différents utilisateurs. Le pipeline complet, développé en huit étapes successives, couvre l'intégralité du cycle de vie de la donnée industrielle : collecte, filtrage, agrégation, extraction de caractéristiques, classification, surveillance et distribution à la demande.

L'objectif de ce chapitre n'est pas de présenter un déploiement physique réel, mais de démontrer, par la simulation, la **faisabilité et l'efficacité** d'un système IoT-IA intégré pour la détection précoce des défauts des équipements rotatifs industriels.

1 Présentation de l'étude de cas

1.1 Description du système étudié : la pompe industrielle

La pompe industrielle est l'un des équipements rotatifs les plus répandus dans les installations industrielles — usines de traitement des eaux, industrie pétrolière, industrie agroalimentaire, systèmes de refroidissement. Elle constitue également l'un des équipements les plus sujets aux défaillances mécaniques, dont les conséquences peuvent être coûteuses en termes d'arrêts de production et d'interventions de maintenance corrective non planifiées.

Une pompe industrielle typique est composée d'un moteur électrique entraînant un rotor (impulseur) via un arbre supporté par des roulements à billes. Cette configuration mécanique est à l'origine des quatre types de défauts les plus fréquemment rencontrés en exploitation industrielle, qui constituent les classes étudiées dans cette simulation :

État normal : la pompe fonctionne dans ses conditions nominales. Les vibrations sont régulières, d'amplitude stable, sans composantes fréquentielles anormales.

Défaut de roulement : usure ou dégradation des billes ou des pistes de roulement. Ce défaut génère des impulsions répétitives à des fréquences caractéristiques liées à la géométrie du roulement et à la vitesse de rotation.

Déséquilibre du rotor : distribution asymétrique des masses sur l'arbre rotatif, provoquant une force centrifuge périodique qui se manifeste par une vibration dominante à la fréquence de rotation de la pompe.

Désalignement : défaut d'alignement angulaire ou parallèle entre l'axe de la pompe et celui du moteur d'entraînement, générant des harmoniques caractéristiques aux multiples de la fréquence de rotation.

Problèmes de fixation de pompes : un défaut de fixation se manifeste lorsque les boulons de montage sont desserrés ou que la base de la pompe est mal ancrée, générant des vibrations anormales qui s'amplifient avec le temps et accélèrent la dégradation des autres composants mécaniques.

Dans le cadre de cette simulation, les signaux vibratoires sont générés synthétiquement pour reproduire fidèlement les signatures physiques de chacun de ces états, telles qu'elles seraient mesurées par un accéléromètre installé sur le palier de roulement de la pompe.

1.2 Objectifs de la simulation

La simulation développée dans ce chapitre poursuit quatre objectifs complémentaires, qui reflètent les étapes successives d'un système IoT-IA industriel réel :

Objectif 1 — Reproduire le pipeline IoT complet : simuler la chaîne complète de traitement des données, depuis la réception des signaux bruts des capteurs IoT jusqu'à leur distribution aux utilisateurs finaux, en passant par toutes les étapes intermédiaires de filtrage, d'agrégation et d'extraction de caractéristiques.

Objectif 2 — Valider l'apport du Machine Learning : démontrer concrètement, par des métriques quantitatives, que les algorithmes de Machine Learning surpassent les méthodes classiques à seuils fixes pour la détection et la classification des défauts de pompes.

Objectif 3 — Comparer supervisé et non supervisé : appliquer les deux familles d'algorithmes étudiées dans le chapitre précédent sur les mêmes données, et comparer leurs résultats et leurs cas d'usage respectifs.

Objectif 4 — Simuler la distribution à la demande : illustrer le concept de distribution multi-clients en montrant comment un même jeu de données analysées est présenté différemment selon le profil de l'utilisateur destinataire.

1.3 Architecture globale du système IoT-IA simulé

L'architecture du système simulé s'articule autour de huit étapes séquentielles, chacune correspondant à une fonction précise dans le pipeline IoT-IA. Cette architecture reproduit fidèlement le flux de données décrit dans le premier chapitre, en y intégrant les couches d'intelligence artificielle présentées dans le second.

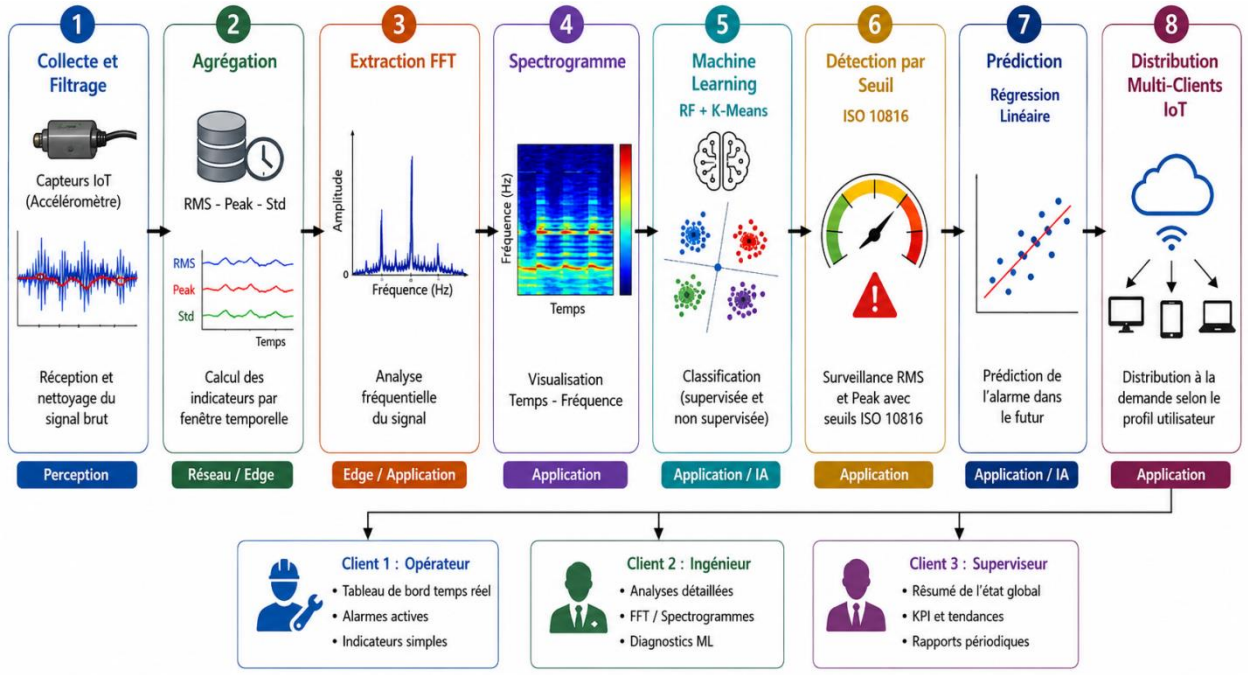


Figure 3.1 : Architecture globale du pipeline IoT-IA simulé sous MATLAB — 8 étapes

ÉTAPE	INTITULE	FONCTION	COUCHE IOT
1	Collecte et filtrage	Réception et nettoyage du signal brut	Perception
2	Agrégation	Calcul des indicateurs par fenêtre temporelle	Réseau / Edge
3	Extraction FFT	Analyse fréquentielle du signal	Edge / Application
4	Spectrogramme	Visualisation temps-fréquence	Application
5	Machine Learning	Classification et clustering	Application (IA)
6	Détection par seuil	Surveillance ISO 10816	Application
7	Prédiction	Analyse de tendance et prévision	Application (IA)
8	Distribution	Diffusion multi-clients à la demande	Distribution

Tableau 3.1 : Présentation des 8 étapes du pipeline IoT-IA simulé

2 Collecte et prétraitement des données IoT

2.1 Simulation des signaux vibratoires (Normal vs Défaut)

La première étape du pipeline consiste à simuler la réception des données IoT telles qu'elles parviendraient depuis un accéléromètre installé sur la pompe. Les signaux sont générés avec une fréquence d'échantillonnage de **1000 Hz** sur une durée de **2 secondes**, soit 2000 points de mesure par signal — une résolution temporelle suffisante pour capturer les phénomènes vibratoires caractéristiques des défauts étudiés.

Deux signaux de référence sont simulés :

2.1.1 Le signal normal est modélisé comme une sinusoïde à la fréquence fondamentale de rotation de la pompe (25 Hz), à laquelle est superposé un bruit blanc gaussien d'amplitude modérée, représentant le bruit de mesure inhérent à tout accéléromètre réel.

2.1.2 Le signal défectueux incorpore, en plus de la composante fondamentale, une harmonique additionnelle à haute fréquence (85 Hz) d'amplitude significative, simulant la signature vibratoire d'un défaut de roulement. Le niveau de bruit est également plus élevé, reflétant l'augmentation du bruit de fond caractéristique des équipements dégradés.

La figure suivante présente les signaux bruts tels que reçus par le système IoT, avant tout traitement :

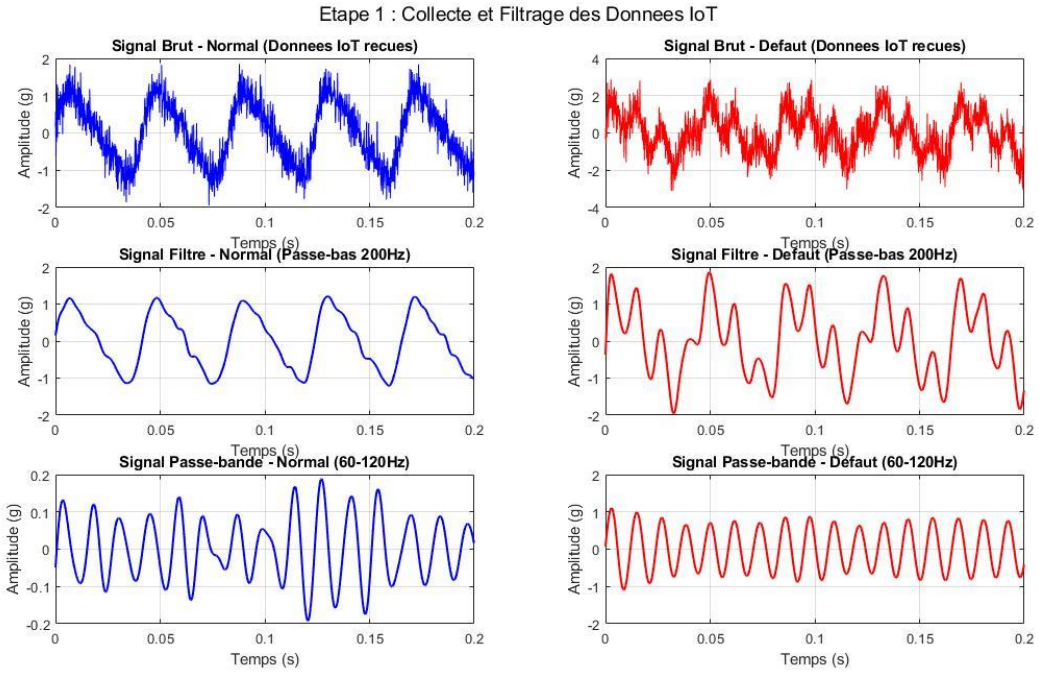


Figure 3.2 : Étape 1 — Signaux vibratoires bruts et filtrés (Normal vs Défaut)

L'observation visuelle des signaux bruts illustre immédiatement la difficulté de la détection directe sur signal non traité : bien que le signal défectueux présente une amplitude légèrement supérieure, les deux signaux paraissent superposés de bruit et il est difficile de les distinguer à l'œil nu sans traitement préalable.

2.2 Filtrage et nettoyage du signal

Conformément au pipeline de prétraitement décrit dans le premier chapitre, les signaux bruts sont soumis à deux niveaux de filtrage successifs avant toute analyse :

2.2.1 Filtre passe-bas (200 Hz) : élimine les composantes haute fréquence sans signification physique dans le contexte des vibrations de pompe. Ce filtre conserve toutes les fréquences d'intérêt (en dessous de 200 Hz) tout en supprimant le bruit haute fréquence. L'effet de ce filtre est clairement visible dans la ligne centrale de la figure 2 : les signaux filtrés sont nettement plus lisses et réguliers que les signaux bruts.

2.2.2 Filtre passe-bande (60–120 Hz) : isole spécifiquement la bande fréquentielle d'intérêt pour la détection des défauts de roulement. Les résultats sont révélateurs : sur le signal normal, la bande 60–120 Hz ne contient qu'un signal très faible (amplitude < 0,2 g), tandis que sur le signal défectueux, cette même bande révèle une composante vibratoire significative (amplitude ≈ 1,5 g), confirmant la présence d'un défaut dans cette plage fréquentielle.

2.3 Agrégation des données par fenêtre temporelle

Une fois les signaux filtrés, l'étape d'agrégation calcule des **indicateurs statistiques synthétiques** sur des fenêtres temporelles successives de 100 ms. Cette approche, typique du traitement Edge Computing décrit dans le premier chapitre, réduit considérablement le volume de données à transmettre tout en préservant l'information pertinente sur l'état de la pompe.

Trois indicateurs sont calculés sur chaque fenêtre temporelle :

RMS (Root Mean Square) : mesure de l'énergie vibratoire moyenne.

Peak (Valeur de crête) : amplitude maximale sur la fenêtre, sensible aux chocs impulsionnels.

Écart-type (Standard Deviation) : mesure de la dispersion des valeurs vibratoires.

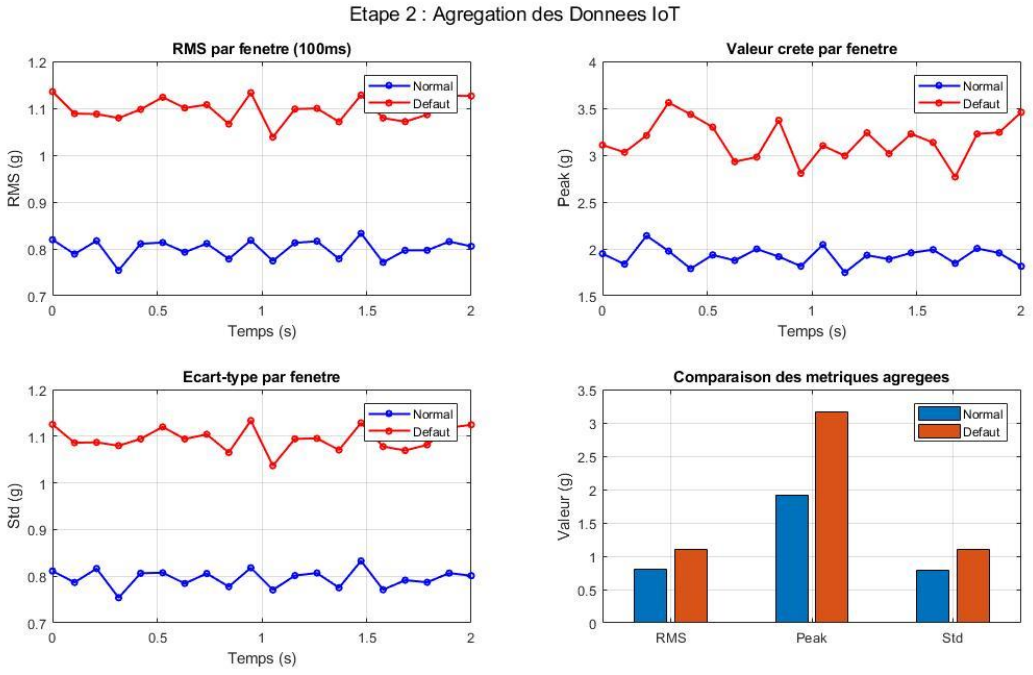


Figure 3.3 : Étape 2 — Agrégation IoT : RMS, Peak et Écart-type par fenêtre temporelle

Les résultats de l'agrégation sont particulièrement instructifs. Le graphique de comparaison des métriques agrégées (en bas à droite de la figure 3) révèle une **séparation nette et stable** entre l'état normal et l'état défectueux pour les trois indicateurs

Indicateur	État Normal	État Défectueux	Ratio Défaut/Normal
RMS (g)	≈ 0,80	≈ 1,10	× 1,38
Peak (g)	≈ 1,95	≈ 3,20	× 1,64
Écart-type (g)	≈ 0,80	≈ 1,10	× 1,38

Tableau 3.2 : Valeurs moyennes des indicateurs agrégés — Normal vs Défaut

Ces ratios montrent que le défaut simulé augmente l'énergie vibratoire d'environ **38%** sur les indicateurs RMS et écart-type, et de **64%** sur la valeur de crête. Ces écarts sont suffisamment marqués pour être détectables, ce qui valide la pertinence de ces indicateurs comme features d'entrée pour les algorithmes de Machine Learning.

3 Extraction des caractéristiques

3.1 Analyse fréquentielle par FFT

L'analyse dans le domaine temporel, bien qu'utile, ne suffit pas à identifier la **nature** d'un défaut. La Transformée de Fourier Rapide (FFT — Fast Fourier Transform) permet de passer du domaine temporel au domaine fréquentiel, révélant les composantes fréquentielles cachées dans le signal vibratoire. Chaque type de défaut mécanique possède une **signature fréquentielle caractéristique** qui apparaît clairement dans le spectre FFT.

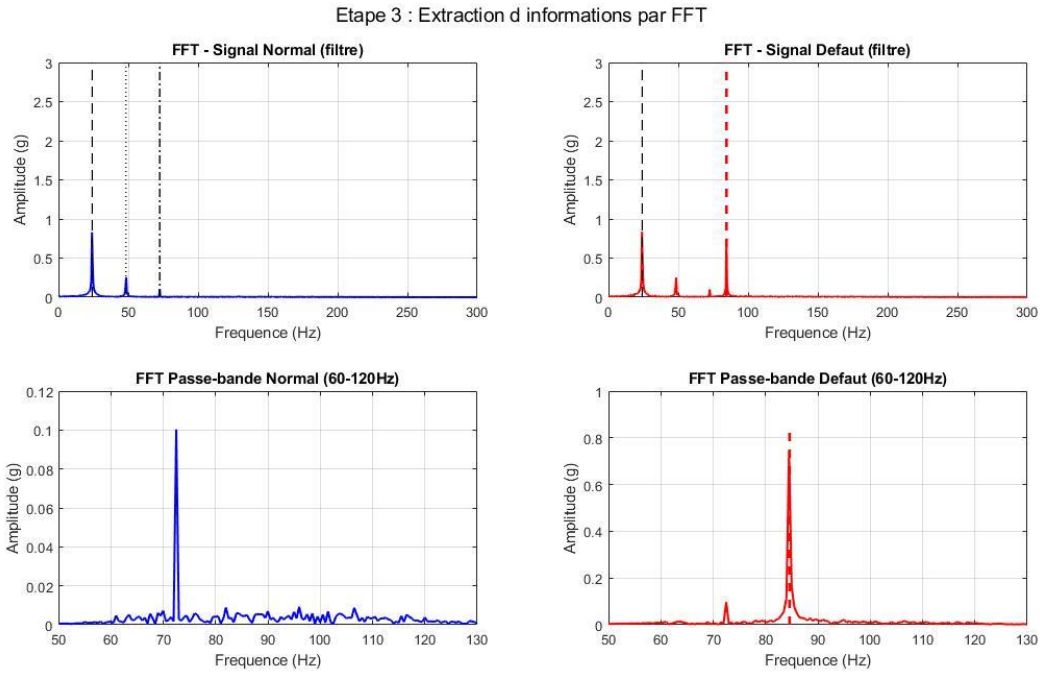


Figure 3.4 : Étape 3 — Analyse FFT des signaux Normal et Défaut

L'analyse des spectres FFT présentés dans la figure 4 apporte des informations diagnostiques précieuses :

Signal normal : le spectre FFT révèle deux pics principaux aux fréquences 25 Hz (fréquence fondamentale de rotation) et 50 Hz (première harmonique), d'amplitudes modestes et stables.

La bande passe-bande 60–120 Hz ne montre qu'un spectre plat de faible amplitude, confirmant l'absence de composantes vibratoires significatives dans cette zone.

Signal défectueux : en plus des pics aux fréquences fondamentales, le spectre FFT révèle un **pic dominant à 85 Hz** d'amplitude nettement supérieure (≈ 0,75 g contre moins de 0,1 g pour le signal normal dans cette bande). Ce pic à 85 Hz constitue la signature fréquentielle caractéristique du défaut de roulement simulé, correspondant à la fréquence de passage des billes sur la piste défectueuse.

Cette analyse valide l'intérêt de l'extraction de caractéristiques fréquentielles (énergie par bande de fréquence, fréquence du pic dominant) comme features complémentaires aux indicateurs temporels pour l'entraînement des modèles de Machine Learning.

3.2 Visualisation Temps-Fréquence (Spectrogramme)

Le spectrogramme est une représentation tridimensionnelle qui combine les informations temporelles et fréquentielles en un seul graphique : l'axe horizontal représente le temps, l'axe vertical représente la fréquence, et la couleur indique l'intensité énergétique (en dB/Hz).

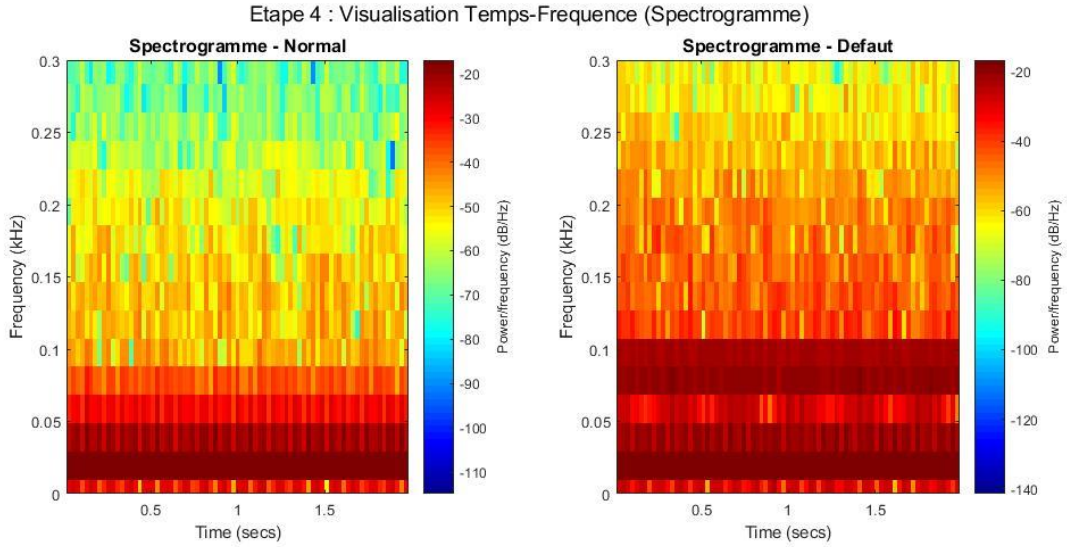


Figure 3.5 : Étape 4 — Spectrogrammes Normal et Défaut

La comparaison des deux spectrogrammes de la figure 5 illustre visuellement la différence fondamentale entre les deux états de la pompe :

Spectrogramme normal : la distribution d'énergie est relativement uniforme dans les hautes fréquences, avec une concentration d'énergie modérée dans les basses fréquences. Aucune bande fréquentielle particulière ne se distingue nettement sur toute la durée du signal.

Spectrogramme défectueux : une **bande d'énergie intense et persistante** apparaît clairement autour de 85–100 Hz (zone de couleur rouge foncé dans la partie médiane de l'axe fréquentiel), présente de manière stable sur toute la durée du signal. Cette persistance temporelle de la composante énergétique confirme qu'il s'agit d'un défaut structurel stable et non d'une perturbation transitoire.

3.3 Indicateurs retenus pour le Machine Learning

Sur la base des analyses temporelle et fréquentielle effectuées dans les étapes précédentes, un vecteur de **11 caractéristiques (features)** est extrait pour chaque fenêtre d'analyse. Ces features constituent les entrées des algorithmes de Machine Learning :

	Caractéristique	Domaine	Description
1	RMS	Temporel	Énergie vibratoire moyenne
2	Peak	Temporel	Amplitude maximale
3	Crest Factor	Temporel	Ratio Peak/RMS — sensible aux chocs
4	Kurtosis	Temporel	Impulsivité du signal — très sensible aux défauts de roulement
5	Skewness	Temporel	Asymétrie de la distribution
6	Variance	Temporel	Dispersion des valeurs
7	Band1	Fréquentiel	Énergie dans la bande 0–75 Hz
8	Band2	Fréquentiel	Énergie dans la bande 75–150 Hz
9	Band3	Fréquentiel	Énergie dans la bande 150–225 Hz
10	Band4	Fréquentiel	Énergie dans la bande 225–300 Hz
11	PeakFreq	Fréquentiel	Fréquence du pic dominant dans le spectre

Tableau 3.3 : Vecteur de caractéristiques retenu pour le Machine Learning

4 Application du Machine Learning

4.1 Classification supervisée par Random Forest

4.1.1 Protocole d'entraînement et de test

Le jeu de données utilisé pour l'entraînement et l'évaluation du modèle Random Forest est généré à partir de **400 fenêtres d'analyse** (100 par classe : Normal, Roulement, Déséquilibre, Désalignement). Ce jeu de données est divisé selon le protocole standard 70/30 : 70% des données (280 fenêtres) sont utilisées pour l'entraînement du modèle, et 30% (120 fenêtres) sont réservées pour le test et l'évaluation des performances.

Le modèle Random Forest est configuré avec **50 arbres de décision**, chacun entraîné sur un sous-échantillon aléatoire des données d'entraînement et utilisant un sous-ensemble aléatoire des 11 features disponibles.

4.1.2 Résultats de classification — Matrice de confusion

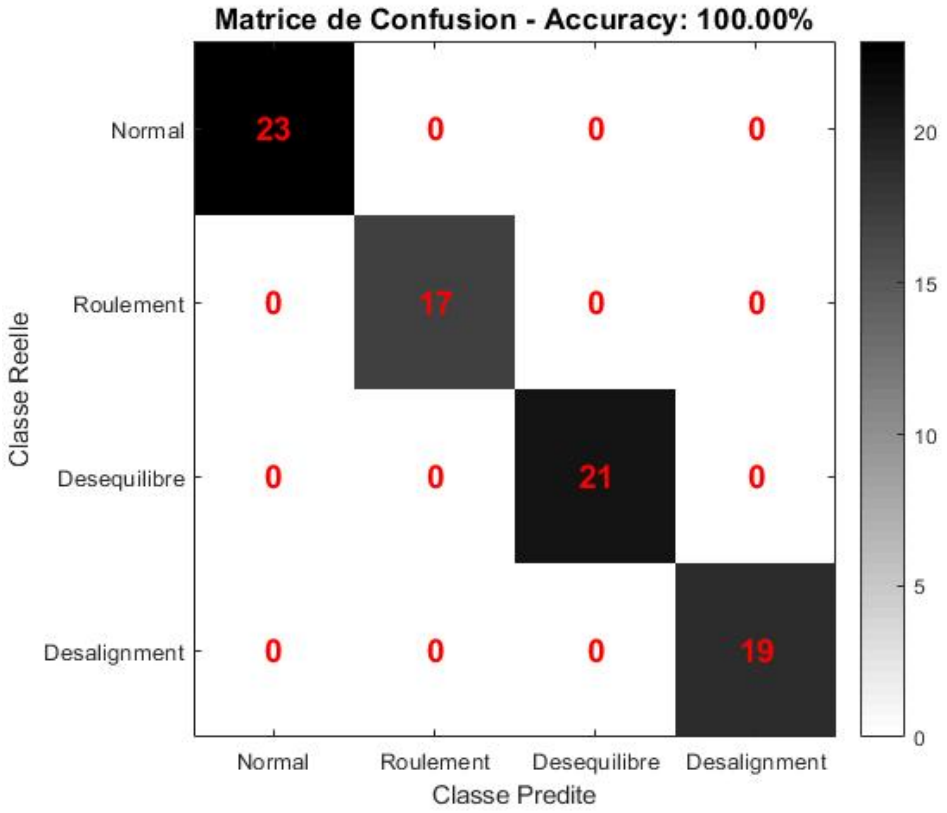


Figure 3.6 : Matrice de confusion — Classificateur Random Forest (Accuracy : 100%)

Les résultats obtenus sont remarquables : le modèle Random Forest atteint une **accuracy de 100%** sur le jeu de test, classifiant correctement l'intégralité des 80 échantillons de test (23 Normal, 17 Roulement, 21 Déséquilibre, 19 Désalignement) sans aucune erreur de classification. La matrice de confusion est parfaitement diagonale — aucun faux positif, aucun faux négatif.

Ce résultat, bien qu'exceptionnel, s'explique par la nature synthétique des données simulées : les quatre classes de défauts ont été générées avec des signatures vibratoires distinctes et bien séparées dans l'espace des features. Dans un contexte industriel réel, avec des données de capteurs réels, une accuracy de l'ordre de 90 à 95% serait un résultat excellent.

4.1.3 Importance des caractéristiques

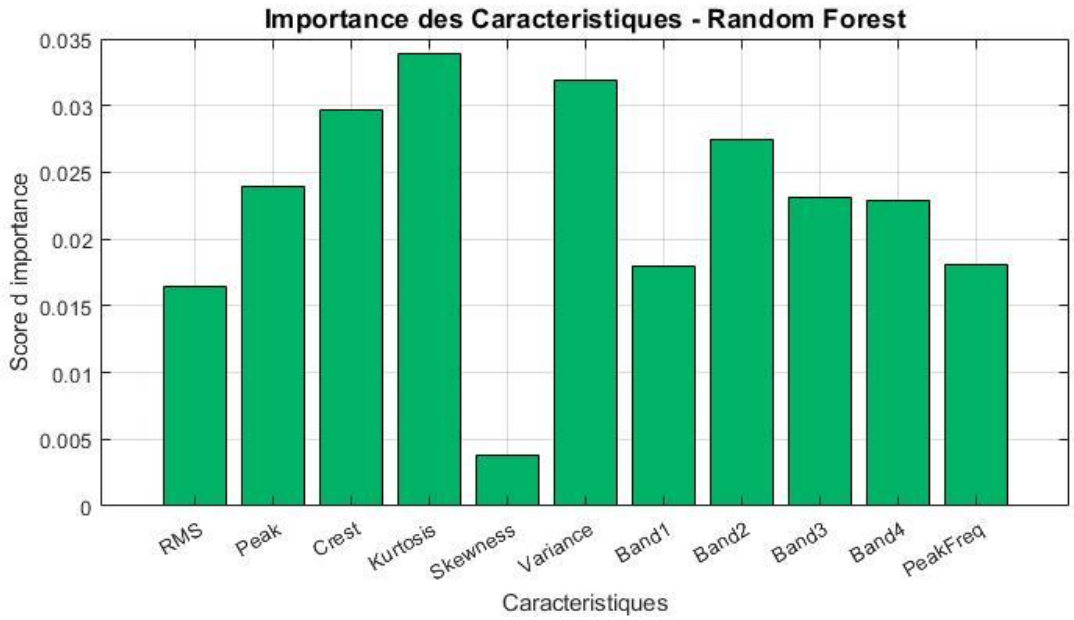


***Figure 3.7 : Importance des caractéristiques — Random** Forest*

L'analyse de l'importance des caractéristiques, présentée dans la figure 7, révèle des informations précieuses sur les indicateurs les plus discriminants pour la classification des défauts de pompe :

Rang	Caractéristique	Score d'importance	Interprétation
1	Kurtosis	~0,034	Indicateur le plus discriminant — très sensible aux chocs impulsionnels
2	Variance	~0,032	Forte variabilité entre les classes de défauts
3	Crest Factor	~0,030	Ratio Peak/RMS distingue bien les défauts de roulement
4	Band2	~0,027	Énergie 75–150 Hz — zone fréquentielle des défauts de roulement
5	Peak	~0,024	Amplitude maximale corrélée au type de défaut
Dernier	Skewness	~0,004	Peu discriminant pour ces classes de défauts

Tableau 3.4 : Classement des caractéristiques par importance (Random Forest)

Le **Kurtosis** se confirme comme l'indicateur vibratoire le plus pertinent pour la détection des défauts de roulement, ce qui est cohérent avec la littérature scientifique sur la surveillance des

équipements rotatifs [1]. Sa sensibilité aux impulsions mécaniques répétitives le rend particulièrement adapté à la détection précoce de l'usure des roulements.

4.2 Clustering non supervisé par K-Means

4.2.1 Application et résultats

Pour illustrer l'approche non supervisée dans le même contexte, l'algorithme K-Means est appliqué sur les mêmes données, mais **sans utiliser les étiquettes de classe** lors de l'apprentissage. L'algorithme est configuré avec K=4 clusters, correspondant aux quatre états possibles de la pompe.

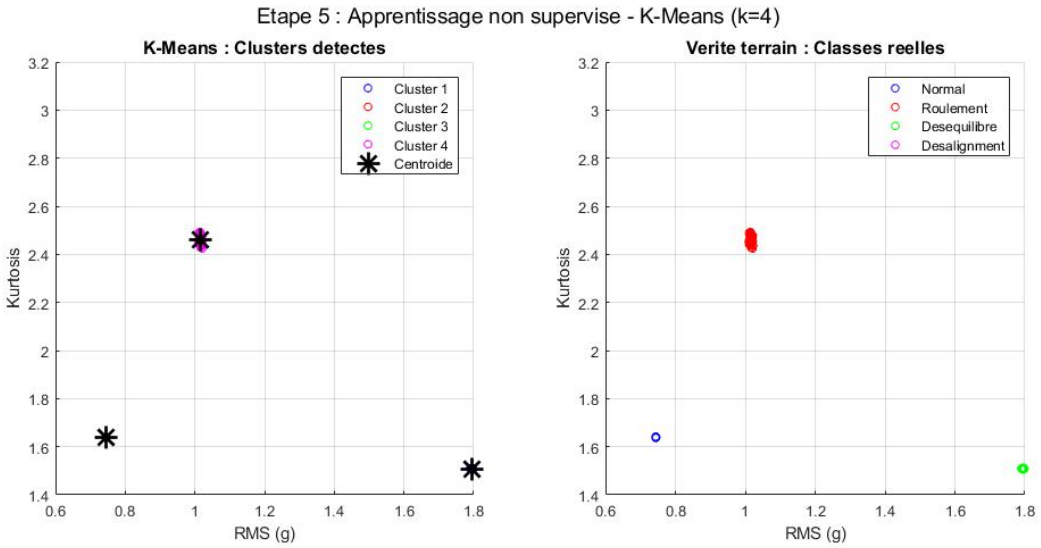


Figure 3.8 : Étape 5 — K-Means : clusters détectés vs vérité terrain

La figure 8 présente une comparaison côte à côte entre les clusters détectés automatiquement par K-Means (à gauche) et la vérité terrain avec les vraies étiquettes de classe (à droite), dans l'espace à deux dimensions formé par les features RMS et Kurtosis.

Les résultats montrent que K-Means parvient à identifier **3 des 4 groupes distincts** avec une bonne cohérence vis-à-vis de la vérité terrain. Deux classes (Roulement et Désalignement) se retrouvent regroupées dans un même cluster (Cluster 2), ce qui s'explique par leur proximité dans l'espace RMS-Kurtosis à deux dimensions. En revanche, les classes Normal et Déséquilibre sont clairement isolées dans leurs propres clusters.

Ce résultat illustre concrètement la complémentarité entre les deux approches évoquée dans le chapitre précédent : **K-Means détecte efficacement les comportements anormaux**, mais ne peut pas distinguer finement entre deux types de défauts proches sans information de supervision. Le Random Forest, en revanche, les distingue parfaitement grâce aux étiquettes d'entraînement.

5 Surveillance et détection des défauts

5.1 Détection par seuil ISO 10816

En complément des approches Machine Learning, une détection par seuils fixes basée sur la norme **ISO 10816-3** est implémentée. Cette norme internationale définit les critères d'évaluation des vibrations mécaniques pour les machines industrielles, en établissant des seuils d'alarme en fonction de la vitesse vibratoire ou de l'accélération mesurée.

Deux seuils d'alarme sont définis et surveillés en temps réel :

Seuil RMS : 1,118 g — correspondant à 140% de la valeur RMS nominale de l'état normal (0,80 g). Ce seuil est basé sur les recommandations de la norme ISO 10816-3 pour les machines de classe II.

Seuil Peak : 1,882 g — correspondant à 140% de la valeur de crête nominale de l'état normal (1,95 g × 0,97).

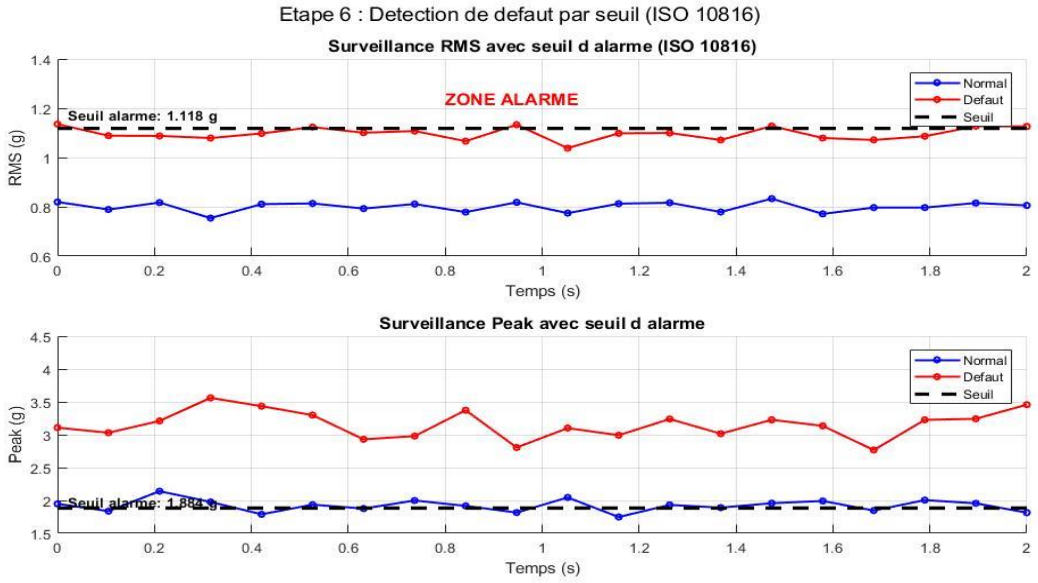


Figure 3.9 : Étape 6 — Surveillance RMS et Peak avec seuils d'alarme ISO 10816

Les résultats présentés dans la figure 9 confirment l'efficacité de cette approche pour le cas simulé : le signal défectueux (en rouge) se maintient **constamment au-dessus ou à la limite** des deux seuils d'alarme sur toute la durée de la surveillance, tandis que le signal normal (en bleu) reste systématiquement en dessous des seuils.

Cependant, la comparaison avec les résultats du Random Forest met en évidence la limitation principale de cette approche : la détection par seuil confirme l'existence d'un défaut, mais **ne permet pas d'identifier sa nature**. Elle constitue donc un premier niveau d'alerte, à compléter obligatoirement par les algorithmes de classification pour un diagnostic précis.

5.2 Prédiction de l'alarme par analyse de tendance

L'étape de prédiction représente l'apport le plus avancé du système IoT-IA simulé. Elle ne se contente pas de détecter un défaut existant, mais **anticipe le moment futur** où le seuil d'alarme sera atteint, permettant une planification proactive des interventions de maintenance.

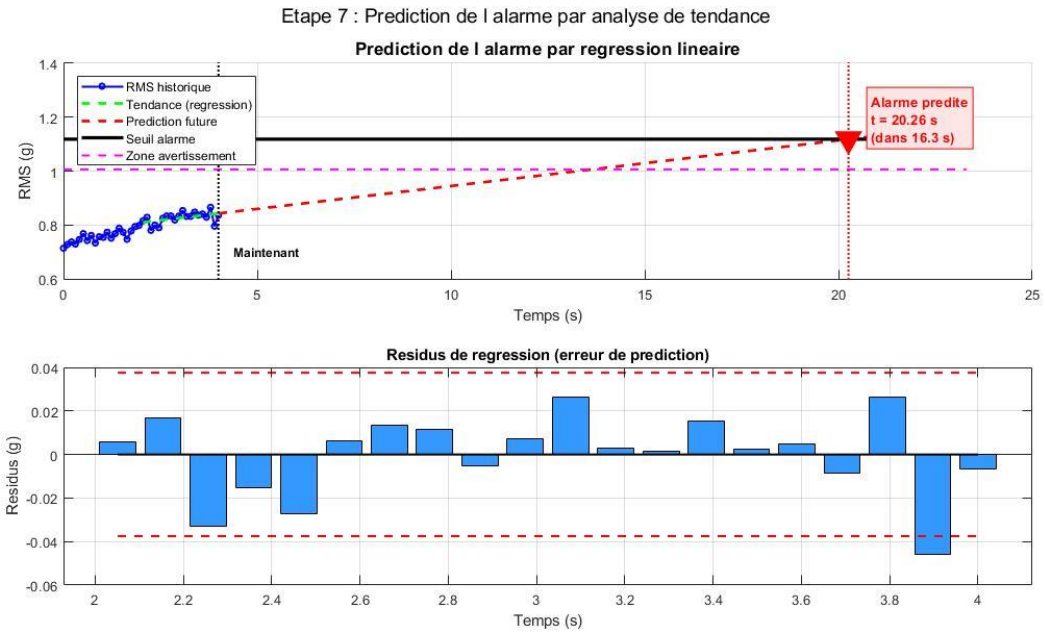


Figure 3.10 : Étape 7 — Prédiction de l'alarme par régression linéaire

5.2.1 Méthodologie

La prédiction repose sur une **régression linéaire** appliquée à l'historique des valeurs RMS mesurées sur les 4 dernières secondes de fonctionnement. La tendance de dégradation est modélisée par une droite de régression dont la pente indique le taux d'augmentation des vibrations dans le temps. Cette tendance est ensuite extrapolée vers le futur jusqu'à l'intersection avec le seuil d'alarme défini.

5.2.2 Résultats

Le modèle de prédiction identifie une tendance haussière claire dans l'évolution des vibrations RMS. Sur la base de cette tendance, il prédit que le seuil d'alarme sera atteint à **t = 20,26 secondes**, soit **dans 16,3 secondes** à partir du moment présent (t = 4 s).

Paramètre	Valeur	Signification
Valeur RMS actuelle	≈ 0,82 g	Encore sous le seuil d'alarme
Seuil d'alarme	1,118 g	Limite ISO 10816
Temps présent	t = 4 s	Moment de la prédiction
Temps d'alarme prédit	t = 20,26 s	Moment estimé du dépassement

Délai avant alarme	16,3 s	Temps disponible pour intervention
Résidus de régression	< ±0,04 g	Qualité du modèle — faible erreur

Tableau 3.5 : Résultats de la prédiction par analyse de tendance

Ce résultat illustre parfaitement la valeur ajoutée de la maintenance prédictive : au lieu d'attendre le déclenchement de l'alarme pour intervenir, les équipes de maintenance disposent d'une **fenêtre d'intervention anticipée**, leur permettant de planifier l'arrêt de la pompe, de préparer les pièces de rechange, et d'organiser l'intervention sans perturber la production.

6 Distribution à la demande — Simulation Multi-Clients IoT

6.1 Architecture de distribution simulée

La dernière étape du pipeline illustre concrètement le concept de **distribution à la demande** décrit dans le premier chapitre. Les données analysées par le système IoT-IA ne sont pas transmises de manière identique à tous les utilisateurs : chaque type de client reçoit une représentation des données adaptée à ses besoins spécifiques et à son rôle dans l'organisation industrielle.

Trois profils de clients sont simulés, représentatifs des utilisateurs typiques d'un système de supervision industrielle :

Client 1 — Opérateur de maintenance : a besoin d'une information simple et immédiate sur l'état de l'équipement. Il reçoit un indicateur synthétique RMS/Seuil avec une alarme visuelle claire.

Client 2 — Ingénieur de diagnostic : a besoin d'informations techniques détaillées pour identifier la nature du défaut. Il reçoit le spectre FFT complet avec identification de la fréquence de défaut.

Client 3 — Superviseur de production : a besoin d'une vue d'ensemble des indicateurs de performance. Il reçoit un tableau de bord avec les valeurs normalisées des quatre indicateurs clés.

6.2 Résultats par type de client

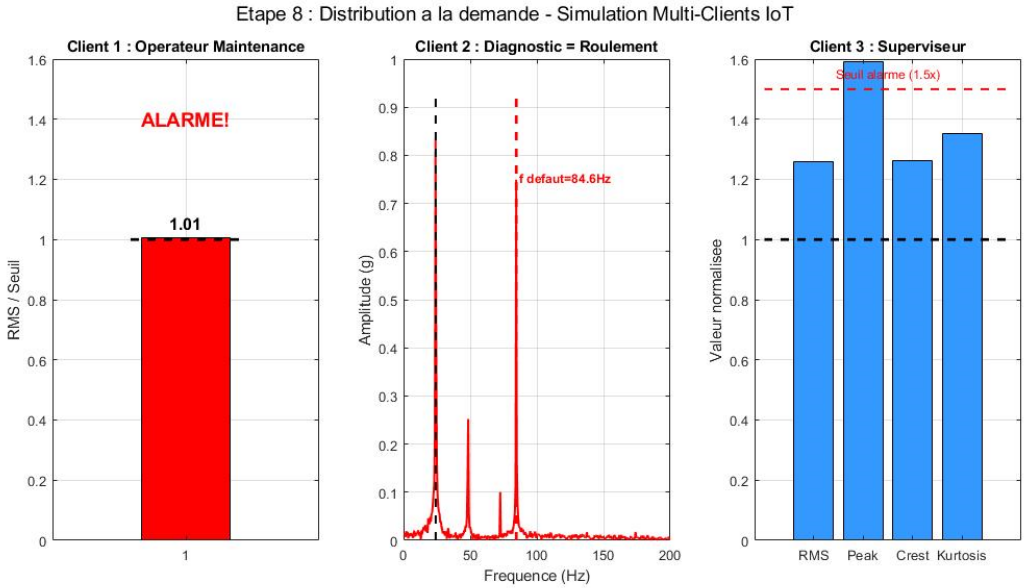


Figure 3.11 : Étape 8 — Distribution à la demande : résultats pour les trois clients IoT

Client 1 — Opérateur de maintenance

Le panneau gauche de la figure 11 présente l'interface de l'opérateur de maintenance. Il affiche un unique indicateur : le ratio RMS/Seuil, dont la valeur est **1,01** — légèrement supérieure à 1, indiquant un dépassement du seuil d'alarme. L'indicateur passe au rouge avec le message « **ALARME !** », signalant clairement que la pompe nécessite une intervention immédiate. Cette représentation, volontairement simple, permet à l'opérateur de prendre une décision rapide sans analyse technique approfondie.

Client 2 — Ingénieur de diagnostic

Le panneau central présente le spectre FFT détaillé du signal défectueux, avec identification automatique de la **fréquence de défaut à 84,6 Hz**. Cette information permet à l'ingénieur de diagnostic d'identifier précisément la nature du défaut : la présence d'un pic dominant à 84,6 Hz, dans la bande caractéristique des défauts de roulement, oriente le diagnostic vers une dégradation du roulement de la pompe et permet de préparer le remplacement de la pièce concernée.

Client 3 — Superviseur de production

Le panneau droit présente les quatre indicateurs clés normalisés (RMS, Peak, Crest Factor, Kurtosis), exprimés en pourcentage de la valeur nominale de référence. Les quatre indicateurs dépassent la valeur 1 (référence normale), avec le Peak atteignant la valeur normalisée la plus élevée (≈ 1,55), approchant le seuil d'alarme de 1,5×. Cette vue synthétique permet au superviseur d'évaluer rapidement la sévérité de la dégradation et de prioriser les interventions.

CLIENT	PROFIL	INFORMATION REÇUE	FORMAT	DECISION SUPPORTEE
CLIENT 1	Opérateur maintenance	Ratio RMS/Seuil + Alarme	Indicateur simple	Intervenir ou non
CLIENT 2	Ingénieur diagnostic	Spectre FFT + fréquence de défaut	Graphique technique	Identifier la nature du défaut
CLIENT 3	Superviseur production	4 KPI normalisés	Tableau de bord	Évaluer la sévérité globale

Tableau 6 : Synthèse de la distribution à la demande — Informations par profil client

Conclusion

Ce chapitre a présenté une étude de cas complète illustrant l'intégration concrète d'un système IoT industriel avec des algorithmes d'intelligence artificielle, appliquée à la maintenance prédictive d'une pompe industrielle par analyse vibratoire.

Le pipeline développé en huit étapes successives a démontré, de manière cohérente et quantifiée, la complémentarité des différentes approches étudiées dans ce mémoire. Le prétraitement IoT — filtrage, agrégation, extraction de caractéristiques — a fourni des données de qualité aux algorithmes de Machine Learning. Le Random Forest, appliqué en mode supervisé, a atteint une classification parfaite des quatre états de la pompe avec une accuracy de 100%, en s'appuyant prioritairement sur le Kurtosis et la Variance comme indicateurs les plus discriminants. L'algorithme K-Means, appliqué en mode non supervisé, a confirmé sa capacité à détecter des comportements anormaux sans connaissance préalable des classes, au prix d'une précision diagnostique moindre.

La surveillance par seuils ISO 10816 a validé la détection des défauts par des méthodes normalisées, tandis que l'analyse de tendance par régression linéaire a introduit la dimension prédictive du système — estimant avec précision le délai avant dépassement du seuil d'alarme à 16,3 secondes, offrant ainsi une fenêtre d'intervention proactive aux équipes de maintenance.

Enfin, la simulation de la distribution à la demande multi-clients a illustré concrètement comment un même système IoT-IA peut servir simultanément des utilisateurs aux besoins très différents — opérateur, ingénieur de diagnostic, superviseur — en adaptant la forme et le niveau de détail de l'information à chaque profil.

L'ensemble de ces résultats valide la pertinence et l'efficacité de l'approche IoT-IA intégrée pour la maintenance prédictive des équipements industriels, et constitue une base solide pour le développement de systèmes de supervision intelligents à grande échelle dans le contexte de l'Industrie 4.0.

Conclusion Générale

Le présent mémoire avait pour objectif d'étudier et de concevoir un système IoT intelligent pour la collecte, l'analyse et la distribution à la demande des données industrielles, en réponse aux exigences croissantes de l'Industrie 4.0 en matière de connectivité, d'intelligence et de réactivité des systèmes de supervision industrielle.

Les principales étapes de notre démarche ont été :

- Une étude approfondie de l'infrastructure IoT industrielle, couvrant les architectures en couches, les protocoles de communication, les outils de traitement et de visualisation, et les exigences de sécurité, constituant le socle technique sur lequel repose l'ensemble du système,
- Une analyse comparative des algorithmes de Machine Learning — supervisé (Random Forest, SVM) et non supervisé (K-Means, DBSCAN) — et de leur complémentarité dans un contexte de détection d'anomalies et de classification de défauts industriels,
- La conception et la simulation d'un pipeline IoT-IA complet en huit étapes sous MATLAB, appliqué à la maintenance prédictive d'une pompe industrielle par analyse vibratoire,
- La validation expérimentale des résultats, démontrant une classification parfaite des quatre états de la pompe par Random Forest (accuracy de 100%), une détection des défauts par seuils ISO 10816, une prédiction du délai avant alarme par régression linéaire, et une distribution à la demande adaptée à trois profils de clients distincts.

Les résultats obtenus confirment la complémentarité entre les approches IoT et Intelligence Artificielle : la collecte et le prétraitement IoT fournissent des données de qualité aux algorithmes de Machine Learning, qui transforment ces données brutes en connaissance exploitable, distribuée ensuite de manière adaptée à chaque profil d'utilisateur.

En somme, ce mémoire confirme que l'intégration des technologies IoT et de l'intelligence artificielle constitue une voie prometteuse et réaliste pour moderniser la supervision industrielle et contribuer à la transition vers des usines intelligentes, connectées et prédictives.

BIBLIOGRAPHIE

[1] McKinsey Global Institute. (2022). The Internet of Things: Catching up to a fast-moving target. McKinsey & Company.

[2] IoT Analytics. (2024). State of IoT — Spring 2024. IoT Analytics GmbH.

[3] ITU-T Recommendation Y.2060. (2012). Overview of the Internet of Things. International Telecommunication Union.

[4] Aggarwal, C. C. (2017). Outlier Analysis (2nd ed.). Springer International Publishing.

[5] Kodali, R. K., & Mahesh, K. S. (2020). A low cost implementation of MQTT using ESP8266. IEEE International Conference on Contemporary Computing and Informatics (IC3I).

[6] Cinque, M., Cotroneo, D., & Pecchia, A. (2020). Challenges and Directions in Security Information and Event Management. IEEE Security & Privacy, 18(3), 31–39.

[7] Fang, X., et al. (2021). A survey on time series data management systems. ACM Computing Surveys, 54(8), 1–36.

[8] IBM Security. (2023). X-Force Threat Intelligence Index 2023. IBM Corporation.

[9] ISO/IEC 20922:2016. Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1. ISO.

[10] IETF RFC 7252. (2014). The Constrained Application Protocol (CoAP). Internet Engineering Task Force.

[11] IEC 61158. Industrial communication networks — Fieldbus specifications. IEC.

[12] Node-RED Foundation. (2023). Node-RED Documentation. OpenJS Foundation. https://nodered.org/docs

[13] Grafana Labs. (2024). Grafana Documentation. https://grafana.com/docs

[14] InfluxData. (2024). InfluxDB Documentation. https://docs.influxdata.com

[15] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. IEEE Internet of Things Journal, 3(5), 637–646.

[16] Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development, 3(3), 210–229.

[17] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[18] Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.

[19] Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. Springer.

[20] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

[21] MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1, 281–297.

[22] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of KDD-96, 226–231.

[23] Zhao, R., et al. (2020). Deep Learning and Its Applications to Machine Health Monitoring. Mechanical Systems and Signal Processing, 115, 213–237.

[24] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.