Rapport sur les Axes Principaux du Projet MANTIS

Abderrahim Boussyf

21 décembre 2025

Table des matières

Introduction

Le projet MANTIS (MAiNtenance prédictive Temps-réel pour usines Intelligentes) est une plateforme modulaire conçue pour répondre aux défis de l'industrie 4.0. Ce rapport détaille les trois axes technologiques majeurs qui constituent le cœur du système :

1. **Data Mining** : L'exploration et la préparation des données industrielles.
2. **Machine Learning / Deep Learning** : La modélisation prédictive de la durée de vie utile (RUL).
3. **Architecture Microservices** : L'infrastructure logicielle distribuée et scalable.

Chapitre 1

Data Mining : Exploration et Préparation des Données

1.1 Description du Dataset NASA C-MAPSS

Les données utilisées proviennent du *Prognostics Data Repository* de la NASA, spécifiquement le dataset C-MAPSS. Ce jeu de données simule la dégradation de moteurs turbofans sous différentes conditions opérationnelles.

- **Unités** : Plusieurs moteurs suivis du début de leur vie jusqu'à la défaillance.
- **Capteurs** : 21 capteurs mesurant diverses grandeurs physiques (température, pression, vitesse, etc.).
- **Settings** : 3 conditions opératoires influençant les mesures.

1.2 Pipeline de Traitement des Données

Le processus de Data Mining suit un pipeline rigoureux illustré ci-dessous :

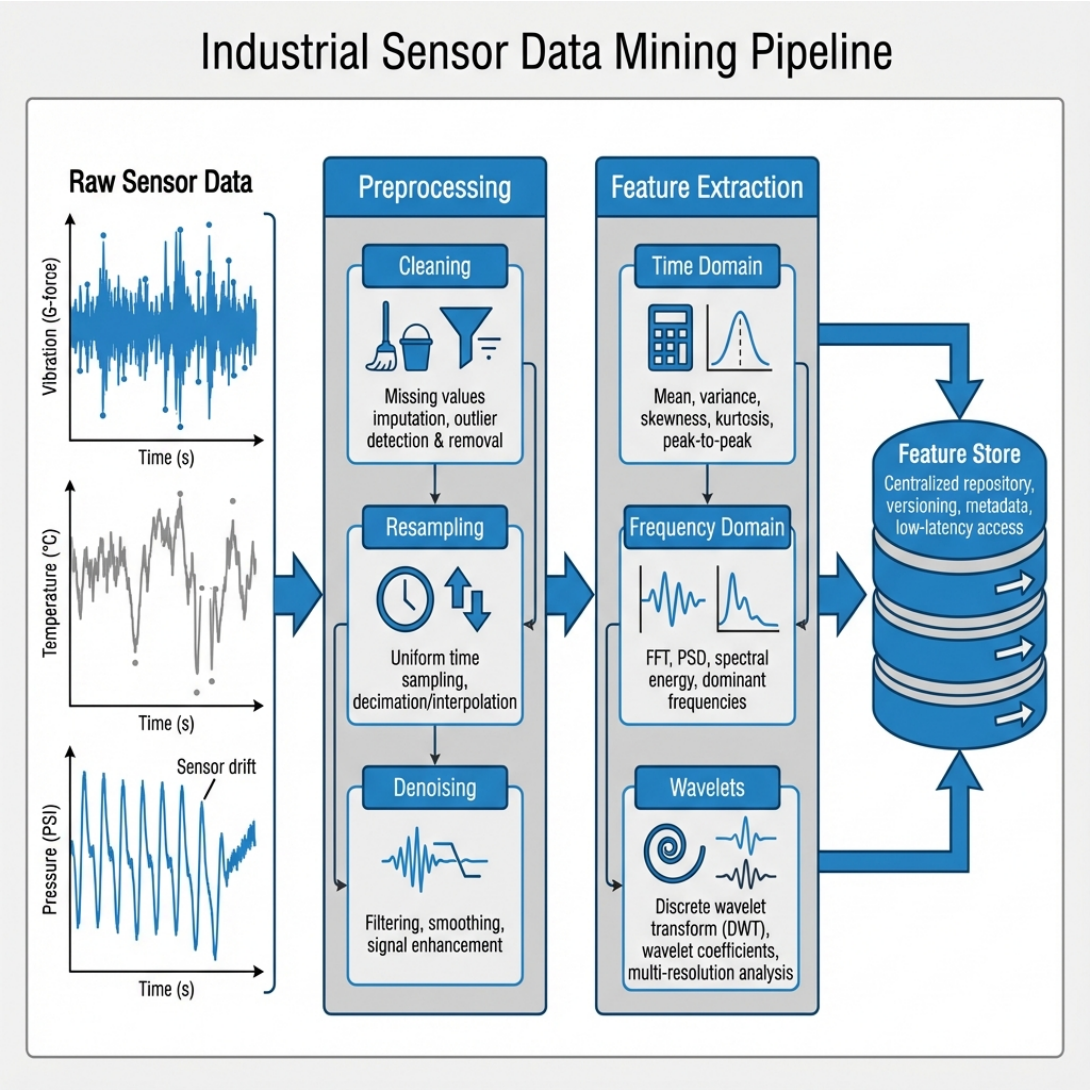

FIGURE 1.1 – Pipeline de Data Mining pour les données capteurs industrielles.

Ce pipeline comprend :

1. **Nettoyage** : Gestion des valeurs manquantes et suppression des outliers statistiques.
2. **Rééchantillonnage** : Alignement temporel des séries provenant de capteurs à fréquences variées.
3. **Denoising** : Application de filtres (ex : moyenne mobile, filtres de Kalman) pour réduire le bruit de mesure.

1.3 Ingénierie des Fonctionnalités (Feature Engineering)

Afin d'alimenter les modèles d'apprentissage, des caractéristiques (features) pertinentes sont extraites des signaux bruts :

— **Domaine Temporel** : Moyenne, écart-type, kurtosis, skewness, valeur efficace (RMS).

— **Domaine Fréquentiel** : Transformée de Fourier Rapide (FFT) pour identifier les fréquences dominantes de vibration.

— **Feature Store** : Stockage centralisé des features calculées pour garantir la cohérence entre l'entraînement et l'inférence (utilisation de Feast).

Chapitre 2

Machine Learning et Deep Learning

2.1 Objectif : Prédiction de la RUL

L'objectif principal est la régression de la *Remaining Useful Life* (RUL), c'est-à-dire le nombre de cycles restants avant la défaillance du moteur. C'est un problème critique pour la maintenance prédictive.

2.2 Architectures de Modèles

Nous avons exploré plusieurs architectures de Deep Learning, notamment les Réseaux de Neurones Récurrents (RNN) adaptés aux séries temporelles.

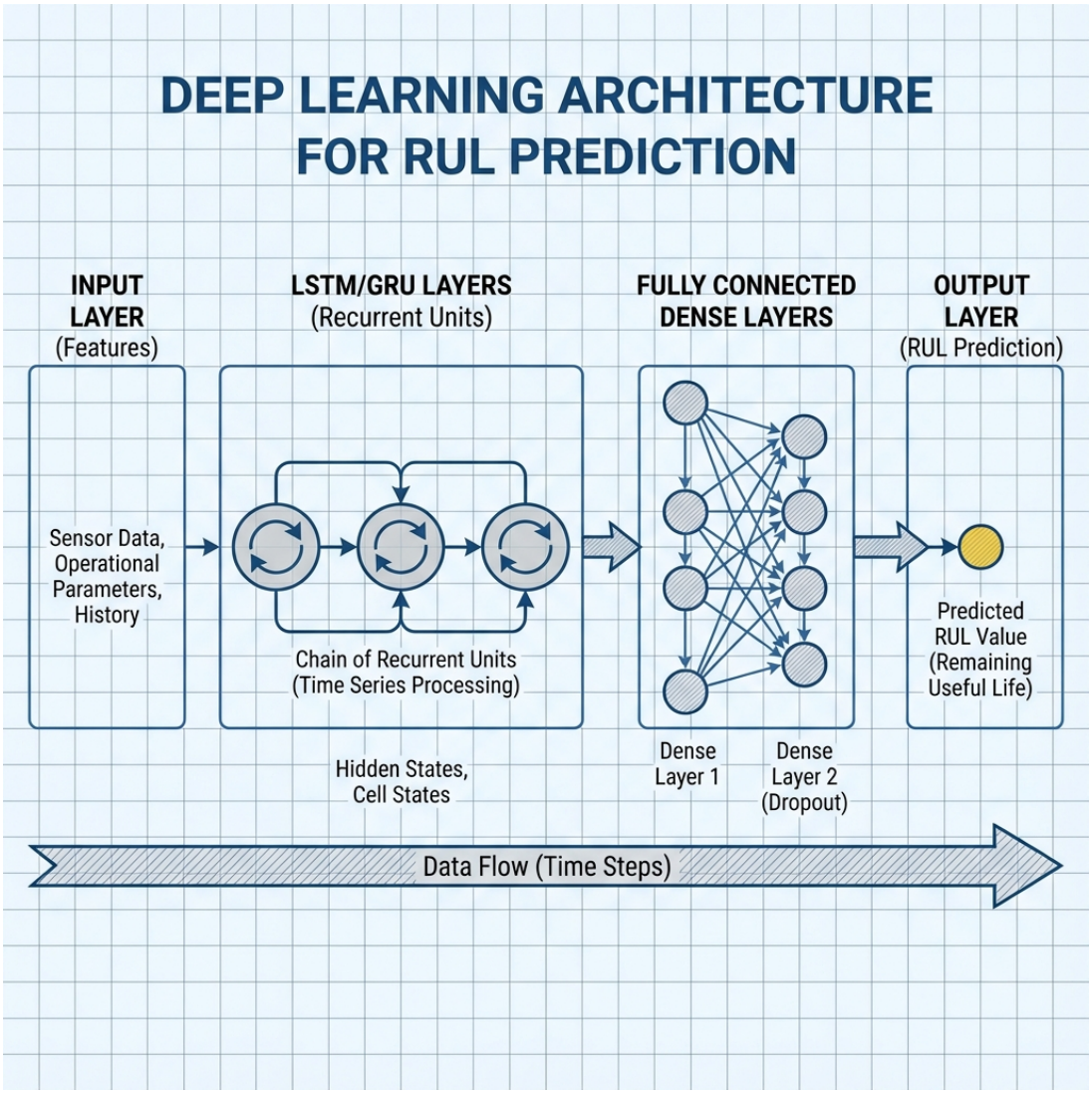

FIGURE 2.1 – Architecture Deep Learning pour la prédiction de RUL (LSTM/GRU).

Les modèles utilisés incluent :

— **LSTM (Long Short-Term Memory)** : Capables de capturer les dépendances à long terme dans les séquences de dégradation.

— **GRU (Gated Recurrent Unit)** : Variante plus légère des LSTM, souvent plus rapide à entraîner.

— **TCN (Temporal Convolutional Networks)** : Utilisation de convolutions dilatées pour une grande fenêtre de réception.

2.3 Évaluation et Métriques

La performance des modèles est évaluée sur un jeu de test indépendant en utilisant :

— **RMSE (Root Mean Squared Error)** : Pénalise fortement les grandes erreurs de prédiction.

— **Score NASA** : Fonction de coût asymétrique qui pénalise davantage les prédictions optimistes (prédire une RUL plus longue que la réalité) car elles posent un risque de sécurité.

Chapitre 3

Architecture Microservices

3.1 Approche Modulaire et Polyglotte

L'architecture de MANTIS est basée sur des microservices autonomes, permettant une scalabilité indépendante et l'utilisation de la technologie la plus adaptée pour chaque tâche. Le système est **polyglotte** :

- **Java / Spring Boot** : Pour les composants nécessitant haute performance et robustesse (Ingestion, Orchestrator).
- **Python / FastAPI** : Pour les composants Data Science et ML (Preprocessing, Prediction, Anomaly Detection).

3.2 Vue d'Ensemble du Système

L'architecture globale est centrée autour d'un bus d'événements Kafka pour le couplage lâche.

FIGURE 3.1 – Architecture Microservices de la plateforme MANTIS.

3.3 Composants Principaux

1. **Ingestion Service** : Collecte les données via OPC UA et MQTT.
2. **Kafka Broker** : Assure la communication asynchrone et résiliente entre les services.
3. **Processing Services** : Chaîne de traitement (Preprocessing → Feature Extraction → Prediction).
4. **Orchestrator** : Moteur de règles (Drools) qui déclenche des actions de maintenance basées sur les prédictions.
5. **Dashboard** : Interface utilisateur React pour la visualisation temps-réel.

Conclusion

Le projet MANTIS démontre l'intégration réussie de techniques avancées de Data Mining et de Deep Learning au sein d'une architecture Microservices robuste. Cette combinaison permet de transformer des données capteurs brutes en décisions de maintenance actionnables, répondant ainsi aux exigences de fiabilité et d'efficacité de l'industrie moderne.