

# Application de l'IA en Mécanique : Maintenance Prédictive

Projet AI en Mcanique

ARKHIS M'HAMMED RADOUANE ELARFAOUI FARID ELABISISI  
AMYNE ED-DARIF SABANE MOHAMMED KHALID LOULIJET

École Nationale des Sciences Appliquées  
Département Génie Mécanique  
Université Sidi Ben Abdellah  
-Fes-

11 décembre 2025

# Plan de la présentation

- 1 Introduction
- 2 Contexte et Objectifs
- 3 Types d'Apprentissage
- 4 Types d'Architectures Neuronales
- 5 Méthodologie
- 6 Résultats et Analyse
- 7 Conclusion

# Introduction

# Introduction

- **Contexte** : Convergence de l'ingénierie mécanique et de l'intelligence artificielle (Industrie 4.0).
- **Sujet** : Maintenance prédictive des machines industrielles.
- **Objectif** : Anticiper les pannes pour optimiser la production et réduire les coûts.
- **Approche** : Utilisation de réseaux de neurones artificiels (RNA) sous MATLAB.

# Contexte et Objectifs

# Problématique

Les approches traditionnelles de maintenance présentent des limites :

- **Maintenance Corrective :**

- Réactive (après la panne).
- Coûts d'arrêt imprévisibles et élevés.

- **Maintenance Préventive :**

- Planifiée (intervalles fixes).
- Gaspillage potentiel (remplacement de pièces saines).

**Question :** Comment intervenir au *bon moment* ?

# Objectif du Projet

Développer un système de **Maintenance Prédictive Intelligent**.

## Mission

Concevoir et entraîner un modèle de réseau de neurones capable de classifier l'état de la machine en temps réel :

- **0** : Fonctionnement Normal
- **1** : Panne Détectée

Basé sur les données de capteurs (température, vitesse, couple, etc.).

# Types d'Apprentissage



# Types d'Apprentissage

L'intelligence artificielle utilise différents paradigmes d'apprentissage :

## 1. Apprentissage Supervisé

- Données étiquetées (entrées + sorties connues).
- Objectif : Apprendre une fonction de mapping.
- Exemples : Classification, Régression.

## 2. Apprentissage Non Supervisé

- Données non étiquetées.
- Objectif : Découvrir des patterns cachés.
- Exemples : Clustering, Réduction de dimensionnalité.

# Types d'Apprentissage (suite)

## 3. Apprentissage par Renforcement

- Agent interagit avec un environnement.
- Apprend par essais-erreurs avec récompenses/pénalités.
- Exemples : Jeux, Robotique, Contrôle optimal.

## 4. Apprentissage Semi-Supervisé

- Combinaison de données étiquetées et non étiquetées.
- Utile quand l'étiquetage est coûteux.

## Notre Projet

Utilise l'**Apprentissage Supervisé** (classification binaire : Normal/Panne).

# Types d'Architectures Neuronales

# Régression Linéaire

**Régression Linéaire** : Modèle le plus simple pour prédire une valeur continue.

## Formulation

$$y = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

où  $w_i$  sont les poids et  $x_i$  les features.

- **Avantages** : Simple, interprétable, rapide.
- **Limites** : Hypothèse de linéarité, ne capture pas les relations complexes.

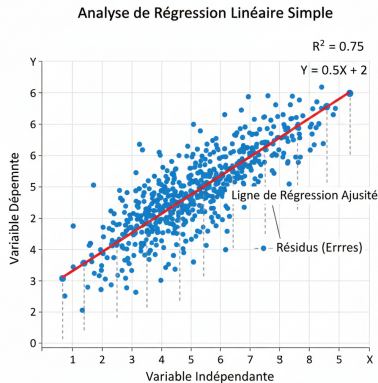


Figure – Visualisation de la régression linéaire

# Réseaux de Neurones Artificiels (RNA)

## Perceptron Multicouche (MLP) : Architecture de base des RNA.

### Composants :

- Couche d'entrée
- Couches cachées
- Couche de sortie
- Fonctions d'activation

### Avantages :

- Capture des relations non-linéaires
- Apprentissage automatique de features
- Puissance d'approximation universelle

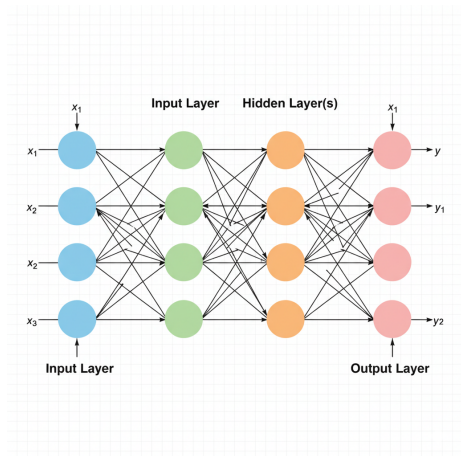


Figure – Architecture d'un RNA

# Autres Architectures Neuronales

## 1. Réseaux de Neurones Convolutifs (CNN)

- Spécialisés pour les images et données spatiales.
- Utilisent des filtres convolutifs.
- Exemples : Reconnaissance d'images, Vision par ordinateur.

## 2. Réseaux de Neurones Récurrents (RNN/LSTM)

- Conçus pour les séries temporelles.
- Mémoire des états précédents.
- Exemples : Prédiction de séquences, Analyse de signaux temporels.

## 3. Deep Learning

- Réseaux avec plusieurs couches profondes.
- Capacité d'apprentissage hiérarchique de représentations.

# Comparaison des Approches

| Modèle              | Complexité | Interprétabilité | Performance                     |
|---------------------|------------|------------------|---------------------------------|
| Régression Linéaire | Faible     | Élevée           | Limitée                         |
| RNA (MLP)           | Moyenne    | Moyenne          | Bonne                           |
| CNN                 | Élevée     | Faible           | Excellente (images)             |
| LSTM                | Élevée     | Faible           | Excellente (séries temporelles) |

Table – Comparaison des architectures pour la maintenance prédictive

## Choix pour notre projet

RNA (MLP) : Bon compromis entre complexité et performance pour données tabulaires.

# Méthodologie



# Description des Données

Données : `machine_failure.csv` (Conditions de fonctionnement d'une machine-outil).

## Variables d'entrée (Features) :

- **Type de produit** (L, M, H) : Qualité/Contraintes.
- **Température de l'air [K]** : Ambiance.
- **Température du processus [K]** : Pièce/Outil.
- **Vitesse de rotation [rpm]** : Broche.
- **Couple [Nm]** : Force de torsion.
- **Usure de l'outil [min]** : Temps d'utilisation cumulé.

## Variable Cible (Target) :

- **Machine Failure** (0 ou 1).

# Implémentation sous MATLAB

**Outils :** MATLAB + Neural Network Toolbox.

**Étapes clés :**

- ❶ **Préparation** : Chargement CSV, conversion 'Type' (catégoriel → numérique), transposition (Features  $\times$  Samples).
- ❷ **Architecture** : patternnet (Reconnaissance de formes).
  - 2 couches cachées : 100 et 50 neurones.
- ❸ **Entraînement** :
  - Répartition : 70% Train, 15% Val, 15% Test.
  - Algorithme : Scaled Conjugate Gradient (trainscg).

# Architecture du Réseau

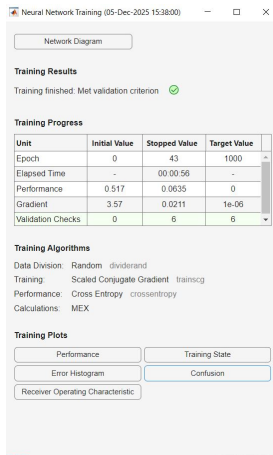


Figure – l'architecture et de l'entraînement (MATLAB)

## Résultats et Analyse

# Performance Globale

## Résultats sur l'ensemble de Test

Le modèle atteint une précision (*Accuracy*) de **97.5%**.

Analyse par classe :

- **Classe 0 (Normal)** : Excellente détection.
  - Précision : 98.0%
  - Rappel : 99.4%
- **Classe 1 (Panne)** : Plus complexe (déséquilibre des classes).
  - Précision : 69.2%
  - Rappel : 38.3%

# Matrices de Confusion

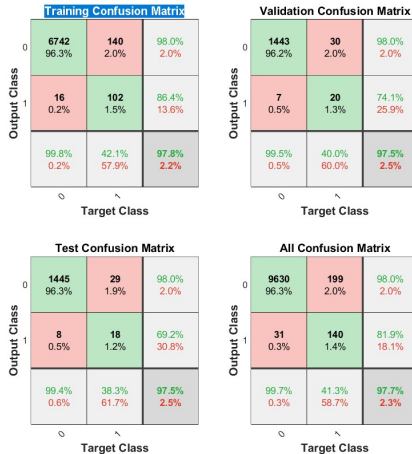


Figure – Matrices de confusion (Train, Val, Test, All)

# Courbes ROC

La courbe ROC montre la capacité de discrimination du modèle.

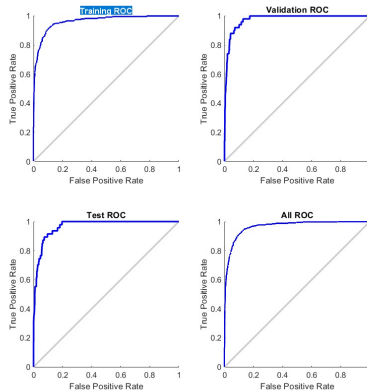


Figure – Courbes ROC : Proches du coin supérieur gauche (Excellente performance).

# Validation et Erreur

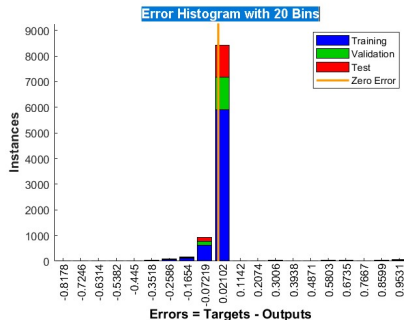
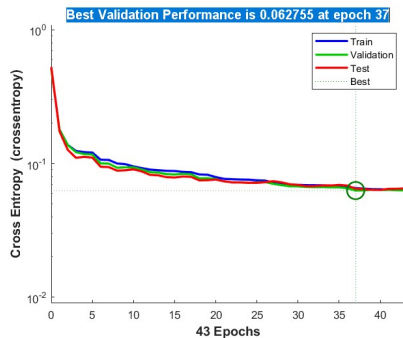


Figure – Performance (Cross-Entropy)

Figure – Histogramme des erreurs

Pas de sur-apprentissage majeur (arrêt précoce à l'époque 37).



# Conclusion

# Conclusion

- **Réussite technique** : Mise en place d'un réseau de neurones performant (97.5% de précision globale) avec MATLAB.
- **Maintenance Prédictive** : Preuve de concept validée pour l'anticipation des pannes à partir de données capteurs.
- **Perspectives** :
  - Améliorer le rappel sur la classe "Panne" (gestion du déséquilibre des données).
  - Tester d'autres architectures (Deep Learning, LSTM pour les séries temporelles).
  - Déploiement sur un système temps réel.

# Merci de votre attention

Avez-vous des questions ?