

Application de l'IA en Mécanique : Maintenance Prédictive

Projet de fin d'étude

Étudiant

Département de Génie Mécanique

9 décembre 2025

Plan de la présentation

- 1 Introduction
- 2 Contexte et Objectifs
- 3 Méthodologie
- 4 Résultats et Analyse
- 5 Conclusion

- **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.

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* ?

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.).

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).

Outils : MATLAB + Neural Network Toolbox.

Étapes clés :

- ① **Préparation** : Chargement CSV, conversion 'Type' (catégoriel → numérique), transposition (Features \times Samples).
- ② **Architecture** : patternnet (Reconnaissance de formes).
 - 2 couches cachées : 100 et 50 neurones.
- ③ **Entraînement** :
 - Répartition : 70% Train, 15% Val, 15% Test.
 - Algorithme : Scaled Conjugate Gradient (trainscg).


Neural Network Training (05-Dec-2025 15:38:00)

Network Diagram

Training Results

Training finished: Met validation criterion 

Training Progress

Unit	Initial Value	Stopped Value	Target Value	
Epoch	0	43	1000	
Elapsed Time	-	00:00:56	-	
Performance	0.517	0.0635	0	
Gradient	3.57	0.0211	1e-06	

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

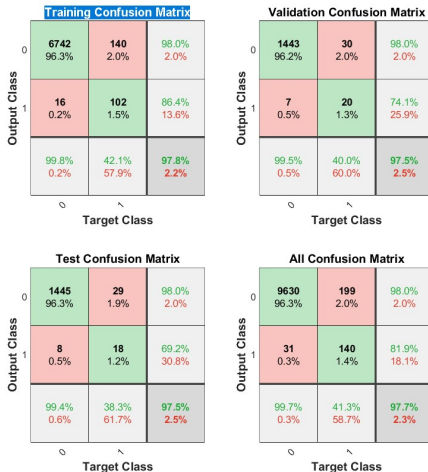


Figure – Matrices de confusion (Train, Val, Test, All)

Courbes ROC

La courbe ROC montre la capacité de discrimination du modèle.

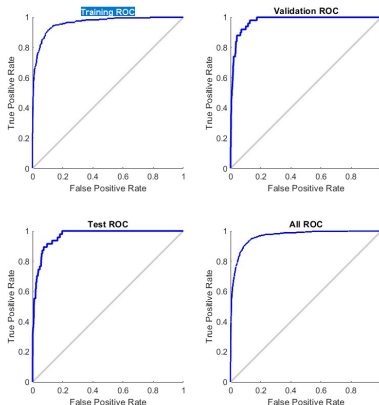
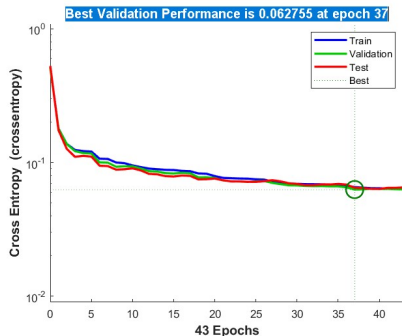
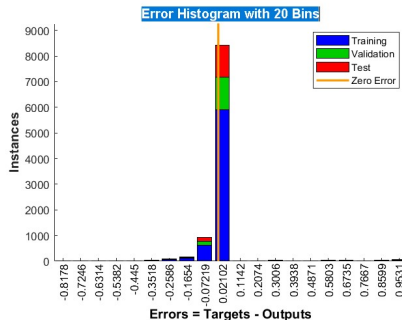


Figure – Courbes ROC : Proches du coin supérieur gauche (Excellente performance).

Validation et Erreur



Performance (Cross-Entropy)



Histogramme des erreurs

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

- **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 ?