



### **Mise en garde**

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

### **Warning**

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

UNIVERSITÉ DU QUÉBEC EN ABITIBI-TÉMISCAMINGUE

SUPERVISION ET PREDICTION DES DEFAUTS DES  
TRANSFORMATEURS ELECTRIQUES EN UTILISANT LES  
TECHNIQUES DE MACHINE LEARNING

MÉMOIRE  
PRÉSENTÉ  
COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN INGÉNIERIE

PAR  
HANANE HADIKI

Juin 2024

## Remerciements

Avant d'entamer ce rapport, c'est avec un grand plaisir que je réserve ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à l'aboutissement de ce travail.

Je souhaite exprimer ma profonde gratitude envers mon directeur de recherche, le **Pr. FOUAD SLAOUI HASNAOUI**, pour sa guidance précieuse, sa motivation infaillible et le temps qu'il a généreusement consacré tout au long de cette période. Sans son soutien inestimable, ses judicieuses recommandations et ses conseils, la réalisation de ce mémoire aurait été impossible.

Je tiens à témoigner toute ma reconnaissance à tous le personnel et la Direction de l'UQAT pour m'avoir offert l'opportunité de vivre cette expérience enrichissante. Merci pour l'excellent accueil que vous m'avez réservé et merci d'avoir veillé directement à mon intégration et au bon déroulement de mes études. Je tiens également à remercier chaleureusement tous les professeurs du programme de maîtrise en ingénierie, concentration en génie électrique, pour leur formation de qualité et leurs conseils précieux au quotidien. Merci d'avoir grandement contribué à la réussite de mon projet de recherche.

Mes remerciements sont adressés également aux membres du jury pour leur bienveillance à vouloir évaluer mon travail, tout en espérant qu'ils trouveront dans ce rapport les qualités de clarté et de motivation qu'ils attendent.

Un grand merci à ma famille, pour l'amour qu'elle m'apporte au quotidien, les conseils ainsi que le soutien inconditionnel, à la fois moral et économique, m'ont permis de réaliser les études que j'ai toujours souhaité poursuivre.

## Dédicace

*À mes chers parents Driss et Khadija,*

Je ne saurai exprimer mon respect, mon amour et ma considération à travers une simple dédicace. Vos prières et vos bénédictions m'ont été d'un grand secours aux différents stades de ma vie. Autant de phrases aussi expressives soient-elles ne sauraient montrer le degré d'amour et d'affection que j'éprouve pour vous.

Vous m'avez comblé avec votre tendresse et affection tout au long de mon parcours. Puisse le tout puissant vous donner santé, bonheur et longue vie. Que ce travail soit l'exaucement de vos vœux tant allégués, le fruit de vos innombrables sacrifices et de votre soutien infaillible.

*À ma sœur Imane et mon frère Ayoub,*

En signe de l'affection et du grand amour que je vous porte, les mots sont insuffisants pour exprimer ma profonde affection et mon attachement indéfectible. Je vous dédie ce travail avec mes vœux de réussite, de prospérité et de bonheur. Que Dieu vous garde et vous protège.

*À mes chers amis,*

En souvenir des moments merveilleux que nous avons passés et aux liens solides qui nous unissent. Un grand merci pour votre soutien, vos encouragements, votre aide. J'ai trouvé en vous le refuge de mes chagrins et mes secrets. Avec toute mon affection et estime, je vous souhaite beaucoup de réussite et de bonheur, autant dans votre vie professionnelle que privée.

*À ma famille au Canada,*

Aux personnes qui sont devenues une famille pour moi, je tiens à vous remercier pour votre amour et votre soutien. Je vous suis toujours reconnaissante de tout cœur.

*À tous mes professeurs,*

*À vous chers lecteurs,*

# Table des matières

<b>Remerciement.....</b>	<b>2</b>
<b>Dédicace.....</b>	<b>3</b>
<b>Liste des figures .....</b>	<b>9</b>
<b>Liste des tableaux .....</b>	<b>13</b>
<b>Liste des abréviations.....</b>	<b>14</b>
<b>Résumé.....</b>	<b>15</b>
<b>Abstract.....</b>	<b>16</b>
<b>Chapitre 1 : Introduction .....</b>	<b>17</b>
1.1. Contexte général :.....	17
1.2. Problématique.....	20
1.3. Motivation et Objectifs.....	20
1.3.1. Objectif général.....	21
1.3.2. Objectifs spécifiques .....	21
1.4. Hypothèses de recherche .....	22
1.5. Méthodologie de recherche .....	22
1.6. Conclusion.....	23
<b>Chapitre 2 : Aperçu sur les transformateurs électriques, la détection et la prédiction des défauts .....</b>	<b>24</b>
2.1. Transformateurs électriques .....	24
2.1.1. Définition générale.....	24
2.2. Principe du fonctionnement d'un transformateur électrique.....	26
2.3. Types de transformateurs .....	26
2.3.1. Transformateur de distribution.....	27
2.3.2. Transformateurs de puissance .....	27

2.3.3.	Transformateurs élévateurs .....	28
2.3.4.	Les transformateurs basse tension.....	28
2.3.5.	Les transformateurs de courant (TC) .....	29
2.3.6.	Les transformateurs d'isolement.....	30
2.3.7.	Autotransformateur .....	31
2.3.8.	Transformateur triphasé .....	32
2.4.	Contextes d'utilisation des transformateurs .....	32
2.4.1.	Réseaux électriques :.....	33
2.4.2.	Production d'électricité .....	34
2.4.3.	Transport d'énergie électrique .....	34
2.4.4.	Distribution électrique.....	34
2.4.5.	Télécommunications .....	35
2.4.6.	Appareils Électroménagers .....	35
2.4.7.	Les systèmes d'éclairage .....	35
2.4.8.	Systèmes Ferroviaires .....	35
2.4.9.	Les moteurs électriques.....	36
2.4.10.	Applications Industrielles .....	36
2.5.	Défauts rencontrés par les transformateurs .....	36
2.6.	Défaillance de l'enroulement.....	37
2.6.1.	Défaillance des traversées .....	38
2.6.2.	Défaillance du changeur de prises.....	39
2.6.3.	Défaillance du noyau.....	39
2.6.4.	Défaillances du Système de Refroidissement .....	40

2.6.5.	Défaillances des Systèmes de Protection .....	41
2.6.6.	Défaillances des réservoirs.....	42
2.6.7.	Les défaillances diélectriques .....	43
2.7.	Conclusion.....	43
<b>Chapitre 3 : Maintenance prédictive et les concepts du Machine Learning .....</b>		<b>44</b>
3.1.	Introduction .....	44
3.2.	Concepts généraux et la Maintenance prédictive .....	44
3.2.1.	Maintenance corrective .....	44
3.2.2.	Maintenance préventive .....	45
3.2.3.	Maintenance prédictive .....	46
3.3.	Méthodes basées sur l'intelligence artificielle et prédiction des défauts .....	48
3.3.1.	Machine Learning (Apprentissage Automatique).....	50
3.3.2.	Le réseau de neurones artificiel et l'apprentissage profond (Deep Learning).....	57
3.3.4.	Prédiction des défauts .....	59
3.4.	Outils utilisés.....	71
3.4.1.	Matlab / Simulink.....	71
3.4.2.	Python .....	71
3.4.3.	Spyder .....	72
3.5.	Conclusion.....	72
<b>Chapitre 4 : Prédiction et localisation des défauts des transformateurs en utilisant des données générées.....</b>		<b>73</b>
4.1.	Stratégie d'apprentissage et processus .....	73
4.2.	Implémentation du modèle par simulation.....	74

4.3. Génération et prétraitement des données .....	76
4.4. Simulations du système électrique proposés .....	78
4.5. Prédiction des défauts des transformateurs électriques .....	81
4.5.1. Division du Dataset .....	81
4.5.2. Entrainement des modèles du Machine learning .....	82
4.5.3. Évaluation des modèles .....	83
4.5.4. Résultats et discussions .....	86
4.5.5. Évaluation des performances des algorithmes .....	109
4.6. Localisation des défauts des transformateurs .....	114
4.6.1. Données et stratégies d'apprentissage .....	114
4.6.2. Division et entrainement des algorithmes .....	115
4.6.3. Résultats obtenus et discussions .....	115
4.6.4. Évaluation des performances des algorithmes .....	133
4.7. Conclusion .....	136
<b>Chapitre 5 : Prédiction des défauts des transformateurs électriques par analyse prédictive des données de surveillance .....</b>	<b>137</b>
5.1. Introduction .....	137
5.2. Base de données .....	140
5.3. Prétraitement des données .....	142
5.4. Division des données .....	142
5.5. Entrainement des algorithmes du Machine Learning .....	143
5.6. Résultats obtenus et discussion .....	144
5.7. Évaluation globale des algorithmes .....	168
5.8. Conclusion .....	171



<b>Chapitre 6 : Prédiction des défauts des transformateurs électriques en utilisant l'apprentissage profond</b> .....	172
6.1. Introduction .....	172
6.2. Prédiction des défauts des transformateurs en utilisant des données de courant et de tension.....	172
6.2.1. Base de données .....	172
6.2.2. Division de la base de données .....	173
6.2.3. Prétraitement des données .....	174
6.2.4. Architecture du RNA .....	174
6.2.5. Entraînement du RNA .....	176
6.2.6. Évaluation du RNA.....	177
6.3. Prédiction des défauts des transformateurs en utilisant des données de Surveillance .....	183
6.3.1. Base de données .....	183
6.3.2. Base de données et prétraitement des données .....	184
6.3.3. Structure du RNA .....	184
6.3.4. Entraînement du RNA .....	185
6.3.5. Évaluation du RNA.....	186
6.4. Conclusion .....	191
<b>CONCLUSION</b> .....	192
<b>BIBLIOGRAPHIE</b> .....	194
<b>ANNEXE A EXPLOITATION DE LA BASE DES DONNÉES</b> .....	199
<b>ANNEXE B ALGORITHME KNN</b> .....	200

## Liste des figures

<b>Figure 1.1</b>	Architecture générale d'un système de supervision.
<b>Figure 2.1</b>	Architecture générale d'un transformateur électrique industriel.
<b>Figure 2.2</b>	Réseau électrique [6].
<b>Figure 3.1</b>	Les discipline de l'Intelligence Artificielle.
<b>Figure 3.2</b>	Apprentissage automatique vs intelligence artificielle vs. Apprentissage profond.
<b>Figure 3.3</b>	Schéma d'un apprentissage supervisé.
<b>Figure 3.4</b>	Terminologie de l'apprentissage automatique.
<b>Figure 3.5</b>	Schéma d'un apprentissage non supervisé.
<b>Figure 3.6</b>	Schéma d'un apprentissage par renforcement.
<b>Figure 3.7</b>	Schéma d'un apprentissage semi-supervisé.
<b>Figure 3.8</b>	L'algorithme Support Vector Machine.
<b>Figure 3.9</b>	L'apprentissage d'un algorithme KNN.
<b>Figure 3.10</b>	Structure d'un algorithme des arbres de décisions.
<b>Figure 3.11</b>	Structure d'un algorithme Random Forest.
<b>Figure 4.1</b>	Stratégie d'apprentissage.
<b>Figure 4.2</b>	Système électrique.
<b>Figure 4.3</b>	Extraction de données et traitement de signal.
<b>Figure 4.4</b>	Formes d'onde de tension et de courant du système triphasé, sans défaut.
<b>Figure 4.5</b>	Formes d'onde de tension et de courant dans le cas d'un défaut.
<b>Figure 4.6</b>	Matrice de confusion de l'algorithme Logistic Regression.
<b>Figure 4.7</b>	Rapport de classification de l'algorithme Logistic Regression.
<b>Figure 4.8</b>	Matrice de confusion de l'algorithme Support Vector Machine.
<b>Figure 4.9</b>	Rapport de classification de l'algorithme SVM.
<b>Figure 4.10</b>	Matrice de confusion de l'algorithme K-Nearest Neighbors.
<b>Figure 4.11</b>	Rapport de classification de l'algorithme K-Nearest Neighbors.
<b>Figure 4.12</b>	Matrice de confusion de l'algorithme Naïve Bayes.
<b>Figure 4.13</b>	Rapport de classification de l'algorithme Naïve Bayes.
<b>Figure 4.14</b>	Matrice de confusion de l'algorithme Decision Tree.
<b>Figure 4.15</b>	Rapport de classification de l'algorithme Decision Tree.
<b>Figure 4.16</b>	Matrice de confusion de l'algorithme Random Forest.
<b>Figure 4.17</b>	Rapport de classification de l'algorithme Random Forest.
<b>Figure 4.18</b>	Matrice de confusion de l'algorithme AdaBoost.

- Figure 4.19** Rapport de classification de l'algorithme AdaBoost.
- Figure 4.20** Matrice de confusion de l'algorithme XGBoost.
- Figure 4.21** Rapport de classification de l'algorithme XGBoost.
- Figure 4.22** Mesures de performance de divers modèles d'apprentissage automatique sur des données de formation et de test.
- Figure 4.23** Comparaison de l'accuracy des tests de modèles dans la classification par apprentissage automatique.
- Figure 4.24** Analyse comparative de l'accuracy de l'entraînement et du test pour différents modèles d'apprentissage automatique.
- Figure 4.25** Scores de l'accuracy de formation, de test et de validation croisée pour divers modèles de classification.
- Figure 4.26** Ensemble de matrices de confusion affichant les résultats de l'algorithme régression logistique.
- Figure 4.27** Rapport de classification de l'algorithme régression logistique MultiOutput.
- Figure 4.28** Ensemble de matrices de confusion affichant les résultats de l'algorithme SVM MultiOutput.
- Figure 4.29** Rapport de classification de l'algorithme SVM MultiOutput.
- Figure 4.30** Ensemble de matrices de confusion affichant les résultats de l'algorithme KNN MultiOutput.
- Figure 4.31** Rapport de classification de l'algorithme KNN MultiOutput.
- Figure 4.32** Ensemble de matrices de confusion affichant les résultats de l'algorithme Naïve Bayes MultiOutput.
- Figure 4.33** Rapport de classification de l'algorithme Naïve Bayes MultiOutput.
- Figure 4.34** Ensemble de matrices de confusion affichant les résultats de l'algorithme arbres de décisions MultiOutput.
- Figure 4.35** Rapport de classification de l'algorithme arbres de décisions MultiOutput.
- Figure 4.36** Ensemble de matrices de confusion affichant les résultats de l'algorithme forêt aléatoire MultiOutput.
- Figure 4.37** Rapport de classification de l'algorithme forêt aléatoire MultiOutput.
- Figure 4.38** Ensemble de matrices de confusion affichant les résultats de l'algorithme AdaBoost MultiOutput.
- Figure 4.39** Rapport de classification de l'algorithme AdaBoost MultiOutput.
- Figure 4.40** Ensemble de matrices de confusion affichant les résultats de l'algorithme XGBoost MultiOutput.
- Figure 4.41** Rapport de classification de l'algorithme XGBoost MultiOutput.

<b>Figure 4.42</b>	Mesures de performance de divers modèles d'apprentissage automatique sur des données de formation et de test.
<b>Figure 5.1</b>	Diagramme d'un Transformateur à Huile avec Système de Refroidissement et Protection.
<b>Figure 5.2</b>	Transformateur Électrique avec Système de Refroidissement et Terminaux de Tension.
<b>Figure 5.3</b>	Matrice de confusion du classificateur de régression logistique.
<b>Figure 5.4</b>	Rapport de classification de l'algorithme Logistic Regression.
<b>Figure 5.5</b>	Matrice de confusion du classificateur de SVM.
<b>Figure 5.6</b>	Rapport de classification de l'algorithme SVM.
<b>Figure 5.7</b>	Matrice de confusion du classificateur de KNN.
<b>Figure 5.8</b>	Rapport de classification de l'algorithme KNN.
<b>Figure 5.9</b>	Matrice de confusion du classificateur de Naïve Bayes.
<b>Figure 5.10</b>	Rapport de classification de l'algorithme Naïve Bayes.
<b>Figure 5.11</b>	Matrice de confusion du classificateur des arbres de décisions.
<b>Figure 5.12</b>	Rapport de classification de l'algorithme arbres de décisions.
<b>Figure 5.13</b>	Matrice de confusion du classificateur de Forêt aléatoire.
<b>Figure 5.14</b>	Rapport de classification de l'algorithme Forêt aléatoire.
<b>Figure 5.15</b>	Matrice de confusion du classificateur de AdaBoost.
<b>Figure 5.16</b>	Rapport de classification de l'algorithme AdaBoost.
<b>Figure 5.17</b>	Matrice de confusion du classificateur de XGBoost.
<b>Figure 5.18</b>	Rapport de classification de l'algorithme XGBoost.
<b>Figure 5.19</b>	Comparaison de l'accuracy des tests de modèles dans la classification par apprentissage automatique.
<b>Figure 5.20</b>	Comparaison des modèles de Machine Learning.
<b>Figure 6.1</b>	Architecture d'un réseau de neurones artificiels.
<b>Figure 6.2</b>	Matrice de confusion du réseau de neurones artificiels.
<b>Figure 6.3</b>	Rapport de classification du réseau de neurones artificiels.
<b>Figure 6.4</b>	Une courbe ROC (Receiver Operating Characteristic).
<b>Figure 6.5</b>	Tracés de la précision du modèle (Model Accuracy).
<b>Figure 6.6</b>	Tracés de la perte du modèle.
<b>Figure 6.7</b>	Tracés de l'erreur quadratique moyenne du modèle (MSE).
<b>Tableau 6.1</b>	Métriques d'évaluation.
<b>Figure 6.8</b>	Matrice de confusion du RNA.
<b>Figure 6.9</b>	Rapport de classification du RNA.
<b>Figure 6.10</b>	Courbe ROC (Receiver Operating Characteristic).
<b>Figure 6.11</b>	Graphique des précisions de validation pour un modèle sur cinq plis

de validation croisée K-fold.

**Figure 6.12** Graphique des erreurs quadratiques moyennes (MSE) pour un modèle sur cinq plis de validation croisée K-fold.

## Liste des tableaux

**Tableau 4.1** : Paramètres.

**Tableau 4.2** : Matrice de confusion.

**Tableau 6.1** : Métriques d'évaluation.

**Tableau 6.2** : Métriques d'évaluation du RNA.

## Liste des abréviations

**IA** : Intelligence Artificielle.

**ML** : Machine Learning.

**DL** : Deep Learning.

**ANN** : Artificial Neural Network.

**LG** : Logistic regression.

**SVM** : Support Vector Machine.

**KNN** : K-Nearest Neighbors.

**XGB** : XGBoost.

**NB** : Naïve Bayes.

**DT** : Decision Tree.

**RMS** : Root Mean Square.

**UQAT** : Université du Québec en Abitibi-Témiscamingue.

## Résumé

Ce rapport présente en détail le mémoire de recherche intitulé "Supervision et prédiction des défauts des transformateurs électriques via les techniques de Machine Learning". L'ambition de ce travail est de déployer une stratégie innovante fondée sur l'apprentissage automatique pour anticiper les pannes des transformateurs électriques, garantissant ainsi leur optimal fonctionnement.

Dans le cadre de cette recherche, notre approche s'est articulée autour de deux axes principaux. D'un côté, nous avons exploité des séries de données issues de simulations obtenues grâce à un modèle électrique que nous avons conçu, afin de modéliser les fluctuations de courant et de tension des transformateurs. Ces données de simulation ont été essentielles non seulement pour anticiper les pannes potentielles, mais également pour déterminer l'emplacement des irrégularités au sein des transformateurs. De l'autre côté, nous avons bénéficié de données réelles issues d'indicateurs de suivi mis à notre disposition, ce qui a ajouté une dimension pratique et concrète à notre recherche. Ces données, nous ont permis d'aborder la prédiction des défaillances des transformateurs sous un angle novateur. L'utilisation de techniques avancées d'apprentissage automatique et d'apprentissage en profondeur pour analyser ces ensembles de données s'est révélée cruciale pour prédire avec précision les défauts et les anomalies dans les transformateurs électriques.

Pour cette recherche, nous avons recours à Matlab/Simulink pour élaborer un modèle électrique fidèle des transformateurs. Par ailleurs, l'implémentation des modèles d'apprentissage automatique s'est appuyée sur l'utilisation de Python, un langage largement privilégié dans le domaine de la science des données.

**Mots clés :** Transformateurs électriques, apprentissage automatique, détection, prédiction des défauts, science des données.



## Abstract

This document elaborates on the research thesis titled "Machine Learning Approaches for Monitoring and Predicting Faults in Electrical Transformers." The core objective of this study is to leverage Machine Learning for early detection and prediction of faults within electrical transformers, ensuring their efficient functioning.

Our research methodology is bifurcated into two distinct segments. Initially, we utilized simulated datasets derived from an electrical model we developed, aiming to replicate the current and voltage variations observed in transformers. These simulations were pivotal for forecasting failures and pinpointing anomalies' locations. Furthermore, we incorporated actual operational data, gathered from monitoring systems, into our analysis, thereby grounding our investigation in real-world scenarios. This integration of simulated and empirical data facilitated a novel perspective on fault prediction in transformers. The employment of sophisticated machine learning and deep learning methodologies to dissect these datasets was instrumental in accurately identifying electrical system faults.

To construct a detailed electrical model of the transformers, we employed Matlab/Simulink. Additionally, the development of our machine learning models was conducted using Python, the preferred programming language in data science, enhancing our data analysis capabilities and enabling the selection of appropriate learning algorithms tailored to our research requirements.

**Keywords:** Electrical transformers, machine learning, detection, fault prediction, data science.

# Chapitre 1

## Introduction

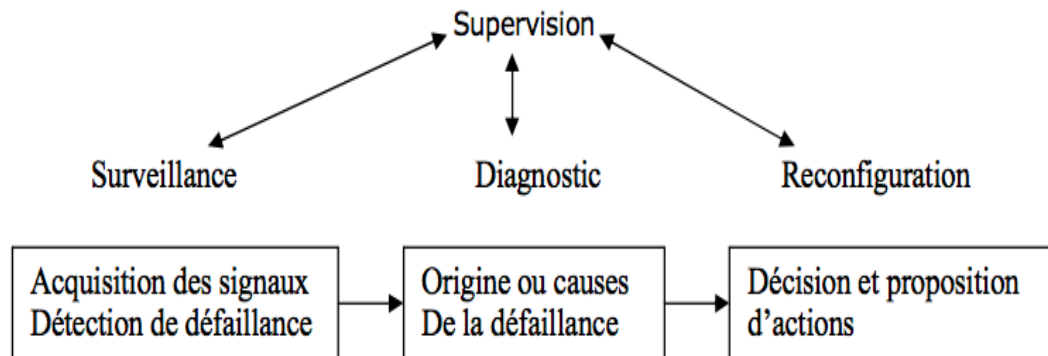
Aujourd'hui, dans le domaine industriel, toutes les entreprises s'efforcent de rendre leurs opérations autonomes et intelligentes, tout en réduisant les coûts et la dépendance humaine grâce à l'intelligence artificielle. Il est essentiel pour ces entreprises de rechercher des approches permettant d'optimiser les coûts et de maintenir leur compétitivité sur le marché. L'automatisation est devenue de plus en plus cruciale pour accroître la productivité en perfectionnant la gestion des procédés industriels. Toutefois, au-delà de ces évolutions, il s'avère primordial de détecter rapidement les anomalies dans les processus, de les isoler et d'identifier leurs origines afin de proposer des solutions adaptées. Ce chapitre introductif est consacré à la présentation du contexte général du projet et sa problématique, en mettant en évidence les objectifs établis et la méthodologie de recherche adoptée.

### 1.1. Contexte général :

La supervision et le pilotage des systèmes et des processus industriels sont une priorité pour chaque entreprise afin d'assurer le bon déroulement des activités, préserver le bon fonctionnement des équipements, garantir la sécurité des biens et du personnel et de protéger l'environnement.

La supervision est une technique de surveillance et de pilotage qui consiste à gérer et à surveiller l'exécution d'une opération ou d'un travail accompli par l'homme ou par une machine, puis à proposer des actions correctives en cas de besoin [1]. Elle permet de rapporter et d'alerter les fonctionnements normaux et anormaux et répond aux préoccupations suivantes :

- Technique : pilotage de l'infrastructure et des machines, surveillance du réseau et mesure des performances...
- Applicative : surveillance des applications et des processus métiers...



**Figure 1.1** Architecture générale d'un système de supervision.

La surveillance consiste à l'acquisition des données en temps réel afin de connaître l'état de fonctionnement réel (normal ou anormal). Le but du diagnostic est de détecter les pannes et les problèmes de certains composants du système conduisant à un comportement anormal, ainsi que d'identifier les causes qui ont conduit à cette situation. La reconfiguration agit sur le système en adaptant la configuration matérielle, et prend la décision qui détermine l'état à atteindre et la séquence d'actions correctives à réaliser pour y arriver [1, 2].

La maintenance est l'ensemble des techniques et d'actions de gestion associées qui ont pour objectif de maintenir et d'assurer le bon fonctionnement des systèmes. Diminuer les coûts de production et/ou d'utilisation des équipements sont parmi les buts principaux de la maintenance. Il existe différents types de maintenance : la maintenance corrective, la maintenance préventive systématique, la maintenance préventive conditionnelle et la maintenance prédictive. La fonction de supervision permet de définir des règles de surveillance et d'y associer des scénarios de remédiation (la maintenance) [3].

Au cours des dernières années, l'intégration croissante de l'intelligence artificielle (IA) a considérablement façonné le paysage de la supervision et de la maintenance des processus industriels. Deux approches distinctes se démarquent dans ce domaine : celles reposant sur l'existence d'un modèle formel de l'équipement à surveiller, et celles qui se basent uniquement sur l'analyse des variables de surveillance [4].

Les méthodologies fondées sur les données, également connues sous le nom de méthodes Data-Driven, ont gagné en popularité et représentent actuellement l'approche privilégiée pour la supervision, le pilotage, la détection et la prédiction des défaillances [2, 5]. Ces méthodes tirent parti de l'utilisation d'ensembles massifs de données et d'algorithmes complexes associés à l'apprentissage automatique (Machine Learning), ce qui leur confère un avantage particulier. Cette approche leur permet d'extraire des informations pertinentes et exploitables des données collectées. En tirant parti d'une surveillance continue et d'une organisation méthodique des données de surveillance, ces méthodes sont privilégiées. En parallèle, l'apprentissage profond (Deep Learning) représente une subdivision hautement efficace de l'apprentissage automatique, utilisant des réseaux de neurones artificiels multicouches pour analyser et interpréter des données complexes. En outre, l'apprentissage ensembliste (Ensemble Learning) émerge comme une approche significative où plusieurs modèles d'apprentissage sont combinés pour améliorer la précision globale des prédictions. Cette méthode cherche à minimiser les erreurs individuelles et à garantir des performances plus solides et fiables.

Ces méthodes d'intelligence artificielle offrent des capacités avancées pour la surveillance, le monitoring, la détection et la prédiction des anomalies dans les processus industriels. Les méthodes de supervision traditionnelles sont de plus en plus soutenues par des techniques d'intelligence artificielle pour améliorer la précision, la rapidité et la fiabilité de la surveillance des processus.

Vu l'importance des transformateurs dans les réseaux électriques, l'étude en cours se focalisera sur la prédiction des défauts et dysfonctionnements qui pourraient survenir sur ces transformateurs.

## **1.2. Problématique**

La qualité, l'efficacité et la rentabilité ont toujours suscité l'intérêt des industries. Aujourd'hui, avec l'évolution des technologies, les industries ont ajouté un aspect très important à leurs priorités : la capacité de détecter, d'isoler voire prédire les défauts et les anomalies. Les transformateurs électriques constituent des composants très importants dans les réseaux électriques. Bien qu'ils jouent un rôle essentiel dans le transport et la distribution de l'électricité, comme tous autres composants, les transformateurs peuvent subir des pannes. Ces dernières sont peu fréquentes mais sont généralement problématiques par rapport au coût de réparation et encore plus critiques au niveau de l'indisponibilité du réseau électrique concerné [6].

Afin d'assurer la fiabilité du réseau électrique, il est important d'évaluer les transformateurs par le biais d'évaluation de leur fonctionnement, leur protection, leur maintenance et réhabilitation. Dans ce contexte, des nouvelles méthodes intelligentes sont de plus en plus utilisées pour garantir la supervision et la maintenance de ces équipements électriques (transformateurs), permettant de prédire les défauts et les dysfonctionnements qui s'imposent, tout en se basant sur des algorithmes de Machine Learning et de Deep Learning pour réaliser les actions de détection et de prédiction des défauts. A noter par ailleurs, que ces méthodes présentent des avantages en termes de fiabilité d'information, de la rapidité sur l'obtention de l'information et du gain sur les coûts d'exploitation (de maintenance et de réparation).

## **1.3. Motivation et Objectifs**

Dans le domaine de l'électricité, les transformateurs sont des éléments cruciaux et particulièrement coûteux. Il revêt une importance capitale de prévenir les éventuelles

défaillances susceptibles de se produire, d'où la nécessité de porter tout l'intérêt à ces équipements par l'adoption des meilleures pratiques d'exploitation et de prise de décision, basées sur l'utilisation des différentes méthodes de l'intelligence artificielle, plus précisément l'apprentissage automatique (Machine Learning), permettant la prédiction et la localisation des défauts.

Entre autres, le suivi instantané de la variation de courant, de la tension électrique ou d'autres paramètres, s'avère important pour l'exploitation optimale d'un transformateur, particulièrement pour la détection des défauts et dysfonctionnement potentiels, d'où la nécessité de bien exploiter les résultats de ces deux paramètres, afin de bien concevoir et choisir les modèles prédictifs optimaux.

### **1.3.1. Objectif général**

L'objectif principal de cette étude est d'appliquer des méthodes d'intelligence artificielle pour la supervision et la maintenance prédictive des transformateurs électriques. Cette approche vise à tirer parti des capacités des algorithmes d'intelligence artificielle, tels que l'apprentissage automatique (Machine Learning) et l'apprentissage profond (Deep Learning), pour analyser les données de surveillance. En utilisant ces méthodes avancées qui ont pour but de détecter à l'avance les signes de défaillance, d'anticiper les anomalies potentielles et de mettre en œuvre des stratégies de maintenance proactive. En se concentrant sur l'application de ces technologies émergentes dans le domaine spécifique des transformateurs électriques, cette recherche vise à améliorer la fiabilité opérationnelle, à réduire les temps d'arrêt imprévus et à optimiser les coûts de maintenance associés à ces composants critiques dans les réseaux électriques.

### **1.3.2. Objectifs spécifiques**

- Prédiction des défauts de transformateurs en se basant sur différentes bases de données.
- Localisation des défauts des transformateurs.
- Choisir les méthodes d'apprentissage automatique (Machine Learning) et d'apprentissage profond (Deep Learning) pour effectuer la prédiction des défauts de transformateurs.
- Implémentation des méthodes en python.
- Évaluer des méthodes de l'apprentissage automatique afin de sélectionner le modèle qui offre la meilleure performance.

#### **1.4. Hypothèses de recherche**

- Étant donné que le fonctionnement des transformateurs est étroitement lié aux valeurs de tension et de courant, nous partons du principe que des anomalies dans ces valeurs indiquent des défaillances au niveau des transformateurs.
- On considère aussi que les données concernant le courant et la tension sont mutuellement dépendantes.
- On suppose que les seuls équipements sujets à des pannes sont les transformateurs.

#### **1.5. Méthodologie de recherche**

Afin d'atteindre les objectifs déterminés, par le logiciel Matlab, on constitue un réseau électrique contenant des lignes de transmissions, des générateurs et des transformateurs. Dans cette étude on s'intéresse aux transformateurs auxquelles un défaut électrique est appliqué.

On commence tout d'abord par la rédaction de la revue de littérature à partir d'articles scientifiques, d'actes de conférences, des livres et d'autres sources Internet. Ensuite, on passe à la partie de génération de données à partir des simulations de notre système déjà choisie. Ultérieurement, on effectue l'exploration des données

générées, l'implémentation des modèles intelligents qui permettront la détection et la prédiction des défauts dans les transformateurs. Et enfin, l'évaluation des méthodes utilisés et le déploiement.

La démarche suivie dans cette étude :

- La génération, la collecte et l'acquisition des données,
- L'exploration et l'analyse des données collectées,
- Découpage des données en données d'entraînement et données tests,
- Préparation des données,
- Implémentation des modèles de Machine Learning et Deep Learning en python pour la prédiction des défauts ;
- Évaluation des modèles,
- Comparaison des résultats obtenus par les sept modèles de machine Learning
- Déploiement des modèles.

## **1.6. Conclusion**

Ce chapitre a abordé l'introduction du sujet en détaillant sa problématique, en définissant les objectifs fixés et en exposant la méthodologie de recherche adoptée.



## **Chapitre 2**

### **Aperçu sur les transformateurs électriques, la détection et la prédiction des défauts**

Ce chapitre sera consacré aux définitions générales, les méthodes de maintenance traditionnelles et aux défauts des transformateurs électriques.

#### **2.1. Transformateurs électriques**

##### **2.1.1. Définition générale**

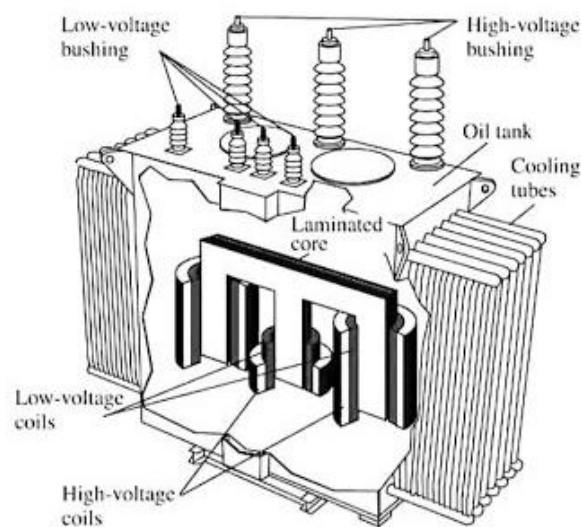
Le transformateur électrique est un convertisseur permettant de modifier les niveaux de tension et de l'intensité du courant délivrées par une source d'énergie électrique alternative en un système de tension et de courant de valeurs différentes mais de même fréquence et de même forme. Les transformateurs sont des équipements coûteux qui contribuent au bon fonctionnement des réseaux électriques. Ils jouent un rôle important dans le transport et la distribution de l'électricité. Les transformateurs électriques couplent par induction des dispositifs électromagnétiques pour transférer l'énergie électrique d'un circuit à un autre [3, 7-9].

Les fabricants de transformateurs électriques configurent des transformateurs pour répondre à tous les types d'applications électriques dans une variété de tailles, de la taille du pouce pour les lampes électriques ou les microphones aux énormes transformateurs stationnaires dans les centrales électriques [10].

Les transformateurs jouent un rôle crucial dans les centrales électriques. Pour envoyer l'électricité plus efficacement dans les lignes de transmissions, la puissance doit être augmentée à haute tension à l'aide des transformateurs élévateurs. Cela aide à réduire la quantité d'énergie perdue en cours de route. Ensuite, à l'aide des transformateurs de distribution, elle sera abaissée à de basses tensions pour être distribuée. Leur

fonctionnement est donc vital dans le processus de transmission efficace de l'électricité. Les transformateurs élévateurs augmentent la puissance à haute tension pour améliorer l'efficacité de transmission dans les lignes, réduisant ainsi les pertes d'énergie sur la route. Par la suite, les transformateurs de distribution abaissent cette tension à des niveaux adaptés à une distribution plus large et plus locale ou à des appareils tels que les variateurs de lumière et les petits appareils électroniques [9].

Un transformateur électrique permet donc de faire transiter la puissance électrique en courant alternatif (AC) depuis la centrale de production d'électricité jusqu'à son utilisateur final avec un minimum de pertes, à différents niveaux de tensions [10].



**Figure 2.1** Architecture générale d'un transformateur électrique industriel [11].

- **Laminated core** : Circuit magnétique.
- **Low-voltage coil** : Enroulement basse tension.
- **High-voltage coil** : Enroulement haute tension.
- **Low-voltage bushing** : Traversées basse tension.
- **High-voltage bushing** : Traversées haute tension.

- *Oil tank Cuve* : Cooling tubes Tubes de refroidissement.

## **2.2. Principe du fonctionnement d'un transformateur électrique**

L'idée fondamentale à l'origine des transformateurs découle des avancées de Michael Faraday, qui a établi un lien entre l'électricité et le magnétisme. Sa théorie a suggéré que la variation du champ magnétique peut générer de l'électricité, formant ainsi le principe essentiel sur lequel reposent les transformateurs.

Les transformateurs modifient les niveaux de tension en manipulant les propriétés électriques. Tout changement de magnétisme génère une tension dans les circuits électriques, provoquant le flux d'électricité dans les fils à proximité. Ils fonctionnent comme suit : L'électricité à haute tension traverse des bobines enroulées autour d'un noyau, induisant un magnétisme alternatif. Ce magnétisme fluctuant génère un courant dans un fil de sortie enroulé avec moins de bobines [8].

Pour la transmission d'électricité, des tensions élevées (de 155 000 à 770 000 volts) sont utilisées, puis abaissées aux points de réception pour un usage pratique. Les sous-stations réduisent ces tensions, et d'autres transformateurs locaux les ajustent aux niveaux standard pour les foyers. Les transformateurs varient la tension en utilisant des ratios de bobines primaires et secondaires. Des pertes d'énergie se produisent sous forme de chaleur, gérées par des méthodes de refroidissement à l'air ou à l'huile pour éviter les dommages.

En résumé, les transformateurs ajustent les tensions pour divers besoins, permettant une transmission efficace de l'électricité sur de longues distances et un usage sécurisé.

## **2.3. Types de transformateurs**

Chaque type de transformateur est conçu avec des caractéristiques et des fonctionnalités spécifiques, adaptées pour répondre à diverses exigences en termes de tension, de courant et de puissance dans différents systèmes électriques et applications.

### **2.3.1. Transformateur de distribution**

Ces transformateurs sont utilisés dans les réseaux de distribution électrique pour réduire l'électricité à haute tension provenant des lignes de transmission à des niveaux de tension plus bas adaptés à la consommation des utilisateurs finaux. Ils sont généralement installés sur des poteaux ou des socles et alimentent les réseaux de distribution locaux. Donc, les transformateurs de distribution sont des machines qui abaissent la tension du réseau électrique, afin qu'elle soit plus adaptée aux utilisateurs finaux [10, 12].

### **2.3.2. Transformateurs de puissance**

Les transformateurs de puissance sont employés dans les réseaux de transmission électrique pour modifier les niveaux de tension, soit en augmentant soit en diminuant la tension, permettant un transfert efficace de l'électricité sur de longues distances. Les transformateurs de puissance facilitent la transmission de l'énergie électrique entre les circuits sans en modifier la fréquence. Ces transformateurs utilisent l'induction électromagnétique pour transporter l'énergie électrique des générateurs et des circuits primaires de distribution. Fonctionnant comme des dispositifs statiques sans pièces mobiles, les transformateurs de puissance augmentent ou diminuent la tension dans les réseaux de distribution.

Généralement de grande taille, les transformateurs de puissance jouent un rôle crucial dans la transmission de l'énergie aux sous-stations, agissant comme un lien vital entre les générateurs et le réseau. Ils sont classés en trois catégories - petits, moyens et grands - avec des puissances nominales allant de 30KVA à 700KVA. En raison de leur taille, les transformateurs de puissance ont besoin d'un agent de refroidissement, généralement de l'huile dans laquelle ils sont immergés. Utilisant un système triphasé, ils offrent un meilleur rendement que les systèmes monophasés, bien que les versions plus petites puissent fonctionner en monophasé.

Les transformateurs de puissance sont constitués de bobines de cuivre ou d'aluminium, le cuivre étant préféré pour sa conductivité et sa ductilité élevées. Les noyaux laminés, construits à partir de feuilles de fer ou d'acier, sont utilisés pour atténuer les pertes par courants de Foucault et les pertes par hystérésis. Les changeurs de prise connectés à l'enroulement haute tension permettent une régulation précise de la tension et minimisent les pertes du noyau. Pour réduire le risque d'étincelles qui pourraient enflammer l'huile de refroidissement, le courant dans l'enroulement haute tension est maintenu à un niveau bas [12].

### **2.3.3. Transformateurs élévateurs**

Les transformateurs élévateurs sont fréquemment utilisés dans les tâches impliquant la transformation et l'ajustement de la puissance. Ils sont préférés pour leur capacité à convertir des tensions plus faibles en tensions plus élevées tout en diminuant l'ampérage et en contrant les effets de la résistance. Cette conversion implique que l'électricité passe par deux phases de bobine. La deuxième phase de la bobine comporte un plus grand nombre d'enroulements, ce qui constitue un ensemble complet de bobines. Si certaines applications exigent un seul enroulement, d'autres peuvent nécessiter des enroulements tertiaires supplémentaires. Cependant, pour le fonctionnement des transformateurs électriques, deux ensembles de bobines sont essentiels. Ces enroulements sont placés autour d'un noyau constitué de composés de ferrite ou de matériaux stratifiés. Le noyau est enroulé avec des bobines émaillées ou du cuivre non recouvert. Les bobines des transformateurs élévateurs sont généralement composées de matériaux tels que l'aluminium, le nickel, le chrome, les alliages d'acier et le cuivre. Parmi ces matériaux, le cuivre est l'option la plus coûteuse mais la plus efficace sur le plan énergétique [10, 12].

### **2.3.4. Les transformateurs basse tension**

Les transformateurs basse tension sont utilisés pour ajuster la capacité de tension des lignes de transmission électrique basse tension. Généralement, les transformateurs basse tension convertissent 120 volts en 12 ou 24 volts. Contrairement aux transformateurs haute tension, qui peuvent peser plusieurs centaines de livres dans les cas extrêmes, les transformateurs basse tension peuvent être suffisamment petits pour être tenus entre deux doigts.

L'objectif d'un transformateur basse tension est de réduire cette tension d'entrée à une valeur beaucoup plus faible. Cette tension plus faible permet de faire fonctionner des équipements, d'installer des ampoules et d'effectuer diverses tâches en toute sécurité grâce à l'électricité. Dans les transformateurs basse tension, le nombre d'enroulements dans l'enroulement secondaire est nettement inférieur à celui de l'enroulement primaire. Les deux enroulements sont enroulés sur un noyau magnétique en fer mais restent séparés. Le noyau, constitué d'un matériau magnétique doux avec des tôles liées, minimise les pertes causées par le flux magnétique alternatif, connu sous le nom de perte d'énergie dans le noyau. Lorsqu'il est connecté à l'alimentation électrique, le courant circule dans l'enroulement primaire, induisant un champ magnétique. Une partie de ce champ est reliée à l'enroulement secondaire, ce qui entraîne un flux de courant et une tension du côté de la charge secondaire. Cette tension est proportionnelle au nombre de tours de l'enroulement secondaire par rapport au côté primaire [12].

### **2.3.5. Les transformateurs de courant (TC)**

Les transformateurs de courant (TC) sont des dispositifs utilisés pour mesurer le courant, ou le flux d'électricité, dans les lignes de transmission électrique. Pour comprendre ce qu'est un transformateur de courant, il faut comprendre ce qu'est le courant électrique, quantifié en ampères, un ampère étant égal au mouvement de  $6,241 \times 10^{18}$  électrons passant par un point donné par seconde. Alors que les transformateurs de puissance gèrent la tension, les transformateurs de courant servent

à mesurer et à réguler le flux d'électricité à des fins de mesure et à contrôler l'électricité fournie aux appareils électroniques.

L'enroulement primaire d'un transformateur de courant se compose généralement de très peu d'enroulements - soit une seule spire plate de fil robuste enroulée autour d'un noyau, soit une barre omnibus passant par un trou central. Les transformateurs de courant réduisent le courant haute tension à une valeur inférieure, ce qui permet de contrôler le courant dans une ligne à courant alternatif à l'aide d'un ampèremètre standard. Ils sont appelés transformateurs en série parce que les enroulements primaires sont en série avec un conducteur porteur de courant alimentant la charge. Contrairement à l'enroulement primaire, les enroulements secondaires d'un transformateur de courant peuvent avoir des enroulements multiples autour d'un noyau laminé fait d'un matériau magnétique à faible perte. La section du noyau est conçue pour maintenir une faible densité de flux magnétique. L'enroulement secondaire alimente en courant un court-circuit sous la forme d'un ampèremètre ou une charge résistive [12].

### **2.3.6. Les transformateurs d'isolement**

Les transformateurs d'isolement jouent un rôle crucial dans l'acheminement de l'énergie électrique d'une source à un appareil. En termes techniques, tous les transformateurs peuvent être considérés comme des transformateurs d'isolement, car ils utilisent l'induction pour convertir l'électricité plutôt qu'un conducteur. Ce qui distingue les transformateurs d'isolement des autres types de transformateurs est leur capacité à isoler l'électricité ou à isoler un transformateur d'une source d'énergie. Ces transformateurs trouvent des applications dans divers domaines, notamment les alimentations sans interruption (ASI), les systèmes de test et de mesure, la robotique, les commandes de moteur, les luminaires, l'électronique médicale, les panneaux de commande industriels, les communications de données, l'avionique et les systèmes audio [12].

La principale caractéristique des transformateurs d'isolement est la séparation des enroulements primaires et secondaires afin de maintenir les circuits distincts. Ils transmettent l'énergie électrique entre les circuits par induction magnétique, en utilisant un aimant pour générer un champ électromagnétique (EMF) dans un circuit sans impact sur la fréquence. En isolant les connexions à la terre des lignes électriques, ces transformateurs éliminent les boucles de terre et suppriment les bruits à haute fréquence [12].

Les transformateurs d'isolation offrent une séparation entre les lignes principales d'alimentation en courant alternatif et l'équipement alimenté, empêchant un chemin de courant continu entre les enroulements. Ils isolent également le secondaire de la terre, offrent des capacités d'élévation ou de réduction de la tension et réduisent le bruit de la ligne.

### **2.3.7. Autotransformateur**

Les autotransformateurs ont un enroulement unique qui sert à la fois de primaire et de secondaire, interconnecté magnétiquement et électriquement, avec trois prises pour les connexions électriques. Ils sont plus petits, plus légers et plus économiques, et sont couramment utilisés dans des systèmes à tensions différentes. Ces transformateurs comprennent un seul enroulement ainsi que des circuits d'entrée et de sortie, la source de tension et la charge étant connectées à deux prises. Une autre prise à l'extrémité de l'enroulement sert de connexion commune aux circuits de la source et de la charge [10].

Les autotransformateurs se distinguent des autres types de transformateurs par la liaison conductrice entre leurs enroulements primaire et secondaire. L'enroulement qui précède la charge est le primaire, tandis que l'enroulement parallèle à la charge est appelé secondaire ou commun. Qu'il s'agisse d'augmenter ou de diminuer la tension, un autotransformateur suit les mêmes principes, bien qu'il en résulte une tension de charge plus importante que la tension de source [12].



En fonctionnement, les autotransformateurs fonctionnent de la même manière que les autres types de transformateurs, où le courant primaire induit un courant variable dans l'enroulement secondaire en fonction du rapport des tours de l'enroulement. L'enroulement secondaire est taraudé en différents points sur sa longueur pour offrir une sortie réglable en différents paliers et pourcentages [12].

### **2.3.8. Transformateur triphasé**

Un transformateur triphasé fait partie de la catégorie des transformateurs polyphasés. Bien qu'il soit possible de relier trois transformateurs monophasés pour créer une unité triphasée, il est plus rentable et plus efficace sur le plan opérationnel d'intégrer les trois phases dans une seule unité. Ces transformateurs comprennent trois ensembles d'enroulements primaires et secondaires, qui peuvent être configurés en étoile ou en triangle. Les configurations varient : delta-delta, étoile-étoile ou autres combinaisons. Pour gérer un système de transmission électrique triphasé, ces transformateurs sont essentiels pour transformer et canaliser le courant électrique des lignes électriques vers les maisons et les établissements commerciaux. Les transformateurs de courant régulent le flux de courant d'une ligne électrique vers les systèmes pour utilisation. Les transformateurs de puissance jouent un rôle crucial en assurant le fonctionnement sûr et efficace des systèmes électriques et des appareils individuels [10, 12].

Dans les réseaux industriels l'énergie électrique est principalement transportée en triphasé, où sont par suite utilisés. Ils sont installés dans : Les centrales ; les postes d'interconnexion ; de distribution ; sur les poteaux ; dans les zones de distribution.

## **2.4. Contextes d'utilisation des transformateurs**

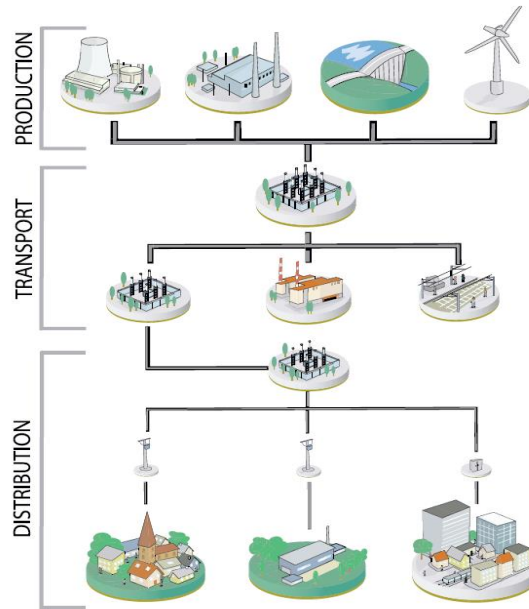
Les transformateurs sont devenus des composants indispensables de la société moderne. Ils jouent un rôle crucial dans la production et la distribution de l'électricité. Leur fonction principale est d'élever ou de réduire le courant alternatif. La

polyvalence des transformateurs électriques provient de leur capacité à ajuster la tension et le courant dans diverses applications. Ces équipements peuvent être utilisés dans différents contextes ;

#### **2.4.1. Réseaux électriques :**

Un réseau électrique désigne un ensemble d'infrastructures énergétiques permettant le transport de l'électricité des centres de production vers les consommateurs. Il est composé de lignes électriques opérant à différentes tensions et interconnectées dans des postes électriques. Ces postes facilitent la distribution de l'électricité et permettent le changement de tension à l'aide de transformateurs. Le réseau électrique doit également assurer la gestion dynamique de l'ensemble de la production, du transport et de la consommation, mettant en œuvre des ajustements visant à garantir la stabilité globale. Dans le langage courant, le terme « secteur » est souvent utilisé comme synonyme de « réseau électrique », bien que techniquement il désigne une subdivision d'un tel réseau.

Le réseau électrique a pour rôle de transmettre l'énergie électrique des sources aux utilisateurs. Il est découpé en trois parties ; la production, le transport et la distribution. Différents transformateurs sont utilisés dans ce réseau. Différents types de transformateurs électriques sont largement utilisés dans les réseaux électriques pour modifier les niveaux de tension et faciliter la transmission et la distribution efficaces de l'énergie électrique.



**Figure 2.2 : Réseau électrique [10].**

### **2.4.2. Production d'électricité**

Les générateurs produisent de l'électricité à différents niveaux de tension. Les transformateurs élévateurs sont utilisés en sortie des centrales de production d'électricité pour abaisser l'intensité (en augmentant la tension pour éviter les pertes de charges) [12].

### **2.4.3. Transport d'énergie électrique**

Le réseau de transport d'énergie électrique assure la répartition de l'électricité depuis les grands centres de production jusqu'aux réseaux de distribution qui sont connectés aux consommateurs. Il comporte majoritairement des transformateurs d'interconnexion, ceux-ci adaptent les différents niveaux de tensions du transport [10, 12].

### **2.4.4. Distribution électrique**

Le réseau de distribution électrique assure la liaison entre le réseau de transport et le consommateur final. Les transformateurs abaisseurs sont utilisés pour adapter la tension du réseau électrique de transport. Et les transformateurs de distribution sont utilisés pour abaisser les tensions du réseau de distribution et l'adapter aux consommateurs [10, 12].

#### **2.4.5. Télécommunications**

Les transformateurs sont employés dans les systèmes d'alimentation électrique des équipements de télécommunications afin de s'assurer que la tension est appropriée pour assurer le bon fonctionnement des dispositifs électroniques [12].

#### **2.4.6. Appareils Électroménagers**

De nombreux appareils tels que les réfrigérateurs, les climatiseurs et les ordinateurs nécessitent des niveaux de tension spécifiques pour un fonctionnement sûr. Les transformateurs sont utilisés pour fournir la tension requise à ces appareils [12].

#### **2.4.7. Les systèmes d'éclairage**

Les systèmes d'éclairage utilisent souvent des transformateurs pour réduire la tension d'une source d'énergie, ce qui les rend appropriés pour les appareils d'éclairage. Dans les systèmes d'éclairage à basse tension, ces transformateurs réduisent la tension de l'alimentation résidentielle standard de 120 volts à 12 ou 24 volts. L'objectif des systèmes d'éclairage incorporant des transformateurs est d'assurer l'isolation électrique, régulant ainsi la tension de sortie de manière cohérente, indépendamment des fluctuations de la tension d'entrée. Ce point est crucial car de nombreux appareils d'éclairage ont besoin d'une tension spécifique pour fonctionner de manière optimale [12].

#### **2.4.8. Systèmes Ferroviaires**

Les transformateurs sont utilisés dans les systèmes ferroviaires pour fournir de l'énergie aux trains. Ils peuvent augmenter la tension pour une transmission efficace sur de longues distances et la réduire pour une utilisation dans les sous-stations ferroviaires [10].

#### **2.4.9. Les moteurs électriques**

Les moteurs électriques peuvent être considérés comme une extension des transformateurs, partageant un principe similaire. Alors que les transformateurs modifient la tension ou le courant d'une alimentation électrique pour répondre aux exigences d'une application, les moteurs électriques transforment l'énergie électrique en énergie mécanique et en mouvement de rotation. Les transformateurs et les moteurs fonctionnent selon la loi de Faraday sur l'induction électromagnétique [12].

#### **2.4.10. Applications Industrielles**

Les industries ont souvent besoin de niveaux de tension spécifiques pour leurs équipements. Les transformateurs sont utilisés pour élever ou abaisser la tension afin de répondre aux exigences des machines et des processus industriels [10, 12].

### **2.5. Défauts rencontrés par les transformateurs**

Les défauts se produisent également dans les transformateurs, comme dans toutes autres appareils électriques. Certaines des défaillances courantes qui se produisent dans les transformateurs de distribution sont les défaillances de l'enroulement, comme les défauts diélectriques, les pertes thermiques dues à la résistance du cuivre, les défauts mécaniques dans la distorsion de l'enroulement, les défaillances des douilles, les défaillances des changeurs de prises, les défaillances du noyau, les défaillances des réservoirs, les défaillances des systèmes de protection, les défaillances du système de refroidissement, etc [7].

Les défauts et les défaillances peuvent se produire dans différentes parties et composants du transformateur en raison de contraintes mécaniques, électriques ou thermiques dues à différentes conditions. Certaines des défaillances les plus courantes du transformateur et leurs causes sont énumérées ci-dessous [13].

## **2.6. Défaillance de l'enroulement**

Les enroulements sont une partie essentielle des transformateurs. Les transformateurs de distribution ont généralement deux enroulements. Un sur le côté primaire et le second sur le côté secondaire.

La haute tension et le courant faible circulent dans l'enroulement primaire et, par induction électromagnétique, la tension est abaissée et le courant est augmenté dans l'enroulement secondaire. Ces enroulements supportent des contraintes diélectriques, thermiques et mécaniques au cours de ce processus. Les défauts qui se produisent dans l'enroulement sont dus à ces contraintes. Cela provoque la rupture des enroulements ou le burn-out [13].

- Les défauts diélectriques se produisent dans l'enroulement en raison d'une rupture d'isolation entre les spires. Il s'agit de l'isolation entre les spires de l'enroulement. Le claquage de l'isolation se produit généralement en raison d'un courant et d'une tension élevés, supérieurs aux valeurs nominales. La rupture de l'isolation entraîne l'embrassement des spires de l'enroulement et provoque un court-circuit. Les deux raisons pour lesquelles la valeur nominale est élevée sont : L'attaque par la foudre sans parafoudre et les tensions de défaut.
- Les enroulements sont généralement en cuivre. En raison de la résistance de la ligne en cuivre, des pertes thermiques se produisent. Ces pertes thermiques créent des points chauds dans l'enroulement en raison d'un mauvais entretien ou d'un manque d'entretien. Avec le temps, cela provoque une usure et une diminution de la résistance physique jusqu'à la rupture de l'enroulement.

- Les défauts mécaniques sont la déformation, le desserrage ou le déplacement des enroulements. Il en résulte une diminution des performances du transformateur et une détérioration du rapport entre les spires. Les principales raisons qui provoquent ce défaut sont la réparation inadéquate, le mauvais entretien, la corrosion, les défauts de fabrication, les vibrations et les mouvements mécaniques dans le transformateur.

### **2.6.1. Défaillance des traversées**

Les traversées sont des dispositifs isolants qui permettent à un conducteur électrique haute tension de passer dans un conducteur de terre. Dans les transformateurs, elles fournissent un chemin de courant à travers la paroi de la cuve. À l'intérieur du transformateur, on utilise des isolateurs en papier qui sont entourés d'huile qui fournissent une isolation supplémentaire. La défaillance des traversées se produit généralement avec le temps [14]. Certaines des principales raisons de la défaillance des traversées sont présentées ci-dessous.

- Le relâchement des conducteurs est causé par les vibrations du transformateur qui entraînent une surchauffe. Cette chaleur endommage le papier isolant et l'huile utilisée.
- Des tensions de défaut élevées et soudaines provoquent des décharges partielles (rupture des isolants électriques solides/liquides) qui endommagent les traversées et provoquent leur dégénérescence et leur rupture complète en quelques heures.
- La rupture de l'étanchéité des douilles est due à la pénétration d'eau, au vieillissement ou à des pertes diélectriques excessives. Ce défaut entraîne la défaillance du noyau du transformateur.
- Le non-remplacement de l'huile usagée pendant une longue période ou son insuffisance en raison d'une fuite provoque des surchauffes internes [13].

### **2.6.2. Défaillance du changeur de prises**

La fonction du changeur de prises dans le transformateur est de réguler le niveau de tension. Cela se fait en ajoutant ou en enlevant des tours à l'enroulement secondaire du transformateur. C'est la partie la plus complexe du transformateur et aussi la plus importante. Même le plus petit défaut entraîne une mauvaise production d'énergie [15]. Voici quelques défauts et leurs causes :

- Dans le cas d'un défaut de type Run-Through, le changeur de prises prend du temps et, après un délai, change le rapport de transformation. La principale raison en est que le relais responsable du changement de prise a un flux résiduel à cause de l'huile polluée, et prend donc du temps à changer. L'autre raison de ce défaut est que le ressort se fragilise avec le temps.
- Le manque d'entretien fait que l'arbre de liaison entre le robinet et le moteur du changeur de robinet n'est pas synchrone. De ce fait, le changeur de prises n'est pas dans la position où il doit être.
- De vieux condensateurs ou des condensateurs brûlés dans le moteur font que le changeur de prises ne contrôle pas son mouvement de direction.
- L'utilisation régulière du changeur de prises fait que le ressort de celui-ci se fragilise lentement avec le temps et finit par se casser. Pour cette raison, le changeur de prises n'est pas en mesure de modifier le rapport de rotation de l'enroulement.
- Une panne du moteur du changeur de prises en raison d'une surtension ou d'une mauvaise utilisation empêche également le changeur de prises de changer le rapport de transformation de l'enroulement [13].

### **2.6.3. Défaillance du noyau**

Les transformateurs ont des noyaux en acier laminé au milieu, entourés par les enroulements du transformateur. La fonction du noyau est de concentrer le flux magnétique. Une défaillance du noyau affecte directement les enroulements du



transformateur, provoquant des défauts dans ces derniers. Les noyaux des transformateurs sont laminés pour réduire les courants de Foucault. La stratification du noyau peut être défectueuse en raison d'un mauvais entretien, d'une vieille huile ou de la corrosion. La rupture de la plus petite partie de la stratification entraîne une augmentation de la chaleur thermique due aux courants de Foucault [16]. Les effets de cette surchauffe sont

- La surchauffe atteint la surface du noyau qui est en contact direct avec les enroulements. Il en résulte que les enroulements sont endommagés par la chaleur.
- Cette chaleur endommage également l'huile dans les transformateurs, ce qui entraîne la libération d'un gaz de l'huile qui endommage d'autres parties du transformateur [13].

#### **2.6.4. Défaillances du Système de Refroidissement**

Le système de refroidissement joue un rôle crucial dans la gestion de la chaleur générée dans les transformateurs en raison des pertes de cuivre et de fer. Ce système comprend des ventilateurs de refroidissement, des pompes à huile et des échangeurs de chaleur refroidis à l'eau. Une défaillance du système de refroidissement conduit à une accumulation de chaleur dans le transformateur, affectant divers composants et pouvant entraîner une augmentation de la pression des gaz, susceptible de causer une défaillance du transformateur. Plusieurs raisons clés de la défaillance du système de refroidissement sont détaillées ci-dessous [13] :

- Fuite dans les conduites d'huile/d'eau : L'une des principales raisons de la défaillance du système de refroidissement est la présence de fuites dans les conduites d'huile ou d'eau. Cela entraîne une diminution des fluides, provoquant un échange de chaleur insuffisant, préjudiciable au transformateur. Les fuites peuvent résulter du stress environnemental, de la corrosion, d'une humidité élevée et de l'exposition au soleil.

- Défaillances des ventilateurs de refroidissement : Les ventilateurs de refroidissement jouent un rôle vital en introduisant de l'air frais dans les cuves des transformateurs à des fins de refroidissement. Les défaillances des ventilateurs de refroidissement peuvent survenir en raison d'un entretien insuffisant, d'une utilisation excessive ou de l'usure du moteur.
- Dysfonctionnement des thermostats : Les dysfonctionnements du système de refroidissement peuvent également être imputés à des thermostats défectueux, chargés de mesurer la température dans le transformateur. Lorsque les thermostats ne sont pas précis, ils fournissent des lectures de température incorrectes, entraînant un fonctionnement inapproprié du système de refroidissement.

### **2.6.5. Défaillances des Systèmes de Protection**

Le rôle principal du système de protection est de repérer et de réagir rapidement aux défauts dans le transformateur afin de prévenir d'éventuels dommages. Les mesures de protection incluent la protection Buchholz, les circuits de soupape de sûreté, la protection contre les surtensions et les relais à pression soudaine, avec une fréquence de défaillance élevée (PN entre 22 et 64) [13, 15].

- Protection Buchholz : Ce dispositif détecte les défauts diélectriques dans le transformateur. Une surchauffe peut se produire en raison de l'accumulation de gaz, ce qui diminue la sensibilité aux défauts. Des niveaux d'huile bas, causés par des fuites, peuvent déclencher une activation inutile de la protection Buchholz, entraînant un gaspillage d'énergie.
- Circuit de soupape de décharge de pression : Conçu pour prévenir les explosions de transformateurs dues à la pression du gaz, ce circuit réduit lentement la pression du gaz. Une défaillance peut résulter de la fragilité du ressort au fil du temps, empêchant une réduction rapide de la pression. Les

augmentations rapides de la pression du gaz peuvent également dépasser la capacité du circuit à relâcher progressivement la pression.

- Protecteur de surtension : Il protège contre les surtensions en permettant à une tension spécifique d'atteindre le transformateur et en détournant l'excédent par un autre chemin. L'humidité, la chaleur et la corrosion contribuent à la défaillance de la protection contre les surtensions, provoquant des surchauffes et des courts-circuits.
- Relais à pression soudaine : Ils protègent les transformateurs contre les explosions dues à des augmentations soudaines et exponentielles de la pression du gaz. S'il ne parvient pas à relâcher la pression soudaine, le transformateur peut exploser. La défaillance du relais peut être due à l'humidité qui affecte les circuits internes.

#### **2.6.6. Défaillances des réservoirs**

Les défaillances des réservoirs dans les transformateurs surviennent lorsque le réservoir, conçu pour contenir l'huile du transformateur, présente des problèmes tels que des fuites ou des fissures dans ses parois. L'huile dans le réservoir remplit des fonctions essentielles d'isolation et de refroidissement, et le réservoir peut également servir de support pour d'autres équipements dans le transformateur [17]. Les contraintes environnementales, la corrosion, l'humidité élevée et le rayonnement solaire contribuent aux défauts du réservoir, entraînant des fuites ou des fissures dans ses parois [15]. Ces problèmes provoquent un déversement d'huile du réservoir, entraînant une diminution des niveaux d'huile [13].

- La réduction du niveau d'huile compromet l'isolation du transformateur et affecte les enroulements.
- L'huile étant essentielle au refroidissement, une diminution du niveau d'huile entraîne une surchauffe, endommageant différentes parties du transformateur.

### **2.6.7. Les défaillances diélectriques**

Les défaillances diélectriques dans les transformateurs électriques se produisent lorsque les matériaux isolants du transformateur ne parviennent pas à résister aux contraintes électriques, provoquant des dégradations dans l'isolation électrique. Ces défaillances peuvent être causées par divers facteurs, notamment [13]:

- Surcharges électriques : Les défaillances diélectriques peuvent être provoquées par des surtensions temporaires ou permanentes, générant des arcs électriques, des décharges partielles ou d'autres phénomènes susceptibles d'altérer l'isolation.
- Contamination de l'huile isolante : Si l'huile isolante du transformateur est contaminée par des particules étrangères, de l'humidité ou d'autres impuretés, cela peut causer des défaillances diélectriques en réduisant la résistance diélectrique de l'huile.
- Vieillessement du matériau isolant : Avec le temps, les matériaux isolants du transformateur peuvent subir un vieillissement naturel, entraînant une diminution de leurs propriétés diélectriques et éventuellement des défaillances.
- Problèmes de conception ou de fabrication : Des défauts dans la conception ou la fabrication du transformateur, tels que des bulles d'air emprisonnées dans l'isolant, peuvent créer des points faibles susceptibles de déclencher des défaillances diélectriques.

### **2.7. Conclusion**

Cette section offre un résumé de la synthèse bibliographique effectuée, mettant l'accent sur les transformateurs électriques, les divers types de transformateurs, leurs applications respectives, ainsi que les divers défauts susceptibles de survenir dans ces équipements. Dans ce rapport, les méthodes du Machine Learning sont proposées pour effectuer la prédiction des défauts des transformateurs électriques.

# **Chapitre 3**

## **Maintenance prédictive et les concepts du Machine Learning**

### **3.1. Introduction**

Aujourd'hui, les entreprises optent à détecter les défauts avant même qu'ils se produisent. On appelle cela, la maintenance prédictive. Ce concept devient de plus en plus commun. La maintenance prédictive est la surveillance intelligente de l'état de santé des équipements afin d'éviter toute défaillance future. Ce chapitre donc, présentera le concept de la maintenance et ses différents types ainsi il mettra le point sur l'importance de la maintenance prédictive et les méthodes basées sur l'intelligence artificielle pour faire la détection et la prédiction des défauts de transformateurs.

### **3.2. Concepts généraux et la Maintenance prédictive**

La maintenance est l'ensemble des techniques et actions de gestion associées ayant pour objectif de maintenir ou de rétablir un équipement dans un état lui permettant de réaliser sa fonction. Elle a pour but principale de diminuer les coûts de production et/ou d'utilisation des équipements [18, 19]. Les stratégies de maintenance les plus utilisées dépendent de divers facteurs, notamment de l'industrie, du type d'équipement et des objectifs organisationnels. Cependant, certaines stratégies de maintenance sont largement adoptées dans différents secteurs en raison de leur efficacité. Voici quelques-unes des stratégies de maintenance les plus utilisées et populaires :

#### **3.2.1. Maintenance corrective**

La maintenance corrective fait référence aux activités effectuées pour remettre l'équipement ou les systèmes dans un état de fonctionnement normal après qu'un défaut ou un dysfonctionnement a été identifié. Celles-ci impliquent généralement

une intervention immédiate pour réparer ou remplacer les composants défectueux afin de restaurer les performances normales de l'appareil.

La maintenance corrective est souvent effectuée en réponse à des pannes inattendues et vise à minimiser les temps d'arrêt et à restaurer la fonctionnalité de l'équipement le plus rapidement possible [18].

### **3.2.2. Maintenance préventive**

La maintenance préventive est une stratégie planifiée et proactive visant à minimiser les risques de défaillance et à prolonger la durée de vie utile des équipements. Elle consiste en des actions préventives programmées et régulières, telles que l'inspection, la lubrification, le remplacement de pièces usées et d'autres activités planifiées. L'objectif principal de la maintenance préventive est d'anticiper les problèmes potentiels avant qu'ils ne se traduisent par des pannes coûteuses [18].

- Planification régulière : Les activités de maintenance préventive sont planifiées à intervalles réguliers, souvent en fonction du temps d'utilisation, du nombre d'opérations, du kilométrage, ou d'autres critères reflétant l'état de l'équipement.
- Lubrification : L'application régulière de lubrifiants est courante dans la maintenance préventive pour réduire l'usure des pièces mobiles et prolonger la durée de vie des équipements.
- Remplacement des pièces usées : Les pièces usées sont remplacées avant qu'elles n'atteignent un état critique, réduisant ainsi le risque de panne.
- Tests et ajustements : Des tests fonctionnels et des ajustements peuvent être effectués pour garantir que l'équipement fonctionne conformément aux spécifications.
- Historique de l'équipement : Pour une gestion de maintenance plus efficace, des enregistrements détaillés des activités de maintenance préventive, des inspections et des réparations passées sont généralement conservés.

- Réduction des coûts de réparation : En anticipant les problèmes potentiels, la maintenance préventive vise à minimiser les coûts de réparation imprévus et à éviter les temps d'arrêt non planifiés.
- Fiabilité améliorée : En identifiant et en corrigeant les problèmes avant qu'ils ne deviennent critiques, la maintenance préventive contribue à améliorer la fiabilité globale de votre équipement.

En veillant à ce que les équipements soient entretenus de manière régulière, les organisations peuvent minimiser les risques d'arrêts imprévus et prolonger la durée de vie utile de leurs actifs. Les actions de maintenance préventive se basent sur un critère simple qui reflète l'état de l'équipement, par exemple le temps d'utilisation, le nombre d'opérations, le kilométrage, etc.

### **3.2.3. Maintenance prédictive**

La maintenance prédictive est une approche de gestion des actifs qui utilise des données et des technologies avancées pour anticiper les défaillances et planifier les activités de maintenance de manière plus précise [18]. Contrairement à la maintenance préventive, qui est basée sur un calendrier fixe, la maintenance prédictive utilise des outils tels que la surveillance continue, les capteurs, l'analyse de données et l'intelligence artificielle pour évaluer en temps réel l'état des équipements. La maintenance prédictive fait beaucoup de promesses, qu'il s'agisse de réduire les temps d'arrêt des machines, d'éliminer les opérations de maintenance inutiles ou de générer des revenus supplémentaires pour les fournisseurs d'équipements [20, 21]. Elle consiste à surveiller d'une manière intelligente l'état de santé des équipements afin d'éviter toute défaillance future. Aujourd'hui, les organisations peuvent identifier les problèmes avant que l'équipement ne tombe en panne, déterminer la cause profonde de la panne grâce à la maintenance prédictive [22-24]. Les caractéristiques clés de la maintenance prédictive incluent :

- Surveillance continue : Les équipements sont équipés de capteurs et d'instruments de surveillance pour collecter en temps réel des données sur divers paramètres tels que la température, la pression, les vibrations, la tension, etc.
- Analyse des données : Les données recueillies sont analysées à l'aide d'outils d'analyse avancés, de techniques de modélisation statistique et d'algorithmes d'intelligence artificielle pour détecter les tendances, les anomalies ou les signes précurseurs de défaillance.
- Prévision des défaillances : Sur la base des analyses, des modèles prédictifs sont développés pour anticiper les défaillances potentielles, permettant ainsi une planification précise des activités de maintenance.
- Optimisation des ressources : En anticipant les défaillances et en planifiant la maintenance au moment opportun, la maintenance prédictive permet une utilisation optimale des ressources, évitant les interventions non nécessaires.
- Réduction des temps d'arrêt : En intervenant de manière ciblée avant qu'une défaillance ne survienne, la maintenance prédictive contribue à réduire les temps d'arrêt imprévus.
- Allongement de la durée de vie des équipements : En identifiant et en corrigeant les problèmes à un stade précoce, la maintenance prédictive peut contribuer à prolonger la durée de vie utile des équipements.
- Coûts réduits : En évitant les réparations d'urgence et en optimisant l'utilisation des ressources, la maintenance prédictive peut contribuer à réduire les coûts globaux de maintenance.
- Amélioration de la fiabilité : En assurant une maintenance basée sur les besoins réels, la fiabilité opérationnelle des équipements peut être considérablement améliorée.

La maintenance prédictive donc repose sur une approche proactive qui tire parti des avancées technologiques pour prendre des décisions éclairées en matière de



maintenance. En surveillant en temps réel l'état des équipements, les organisations peuvent améliorer considérablement la gestion de leurs actifs et optimiser les performances opérationnelles [21].

### **3.3. Méthodes basées sur l'intelligence artificielle et prédiction des défauts**

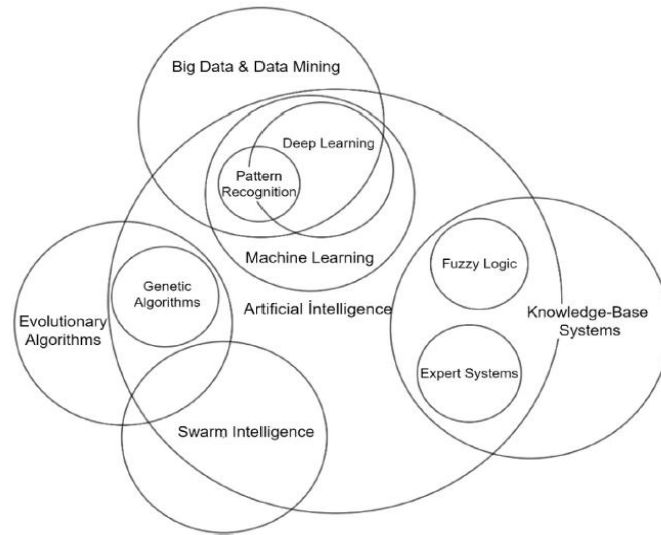
Aujourd'hui, les applications de l'intelligence artificielle émergent progressivement des environnements sécurisés des laboratoires de recherche pour s'intégrer à notre vie quotidienne [25].

L'intelligence artificielle (IA) est une famille de technologies en évolution rapide qui peut apporter une vaste gamme d'avantages économiques et sociaux à travers l'ensemble des industries et des activités sociales. En améliorant les prédictions, en optimisant les opérations et l'allocation des ressources, et en personnalisant la prestation de services, l'utilisation de l'intelligence artificielle peut soutenir des résultats socialement et environnementalement bénéfiques tout en offrant des avantages concurrentiels clés aux entreprises et à l'économie [26]. L'intelligence artificielle (IA) peut être définie comme des techniques informatiques reproduisant des activités mentales humaines telles que la réflexion, l'inférence, la classification, l'interprétation, l'estimation et la prise de décision. Ce domaine interdisciplinaire fusionne des connaissances issues de différentes disciplines comme les mathématiques, l'informatique, la neurologie, la biologie, la génétique et l'ingénierie. La recherche en IA vise à effectuer rapidement, efficacement et précisément des tâches et des activités mentales spécifiques à l'homme à l'aide d'ordinateurs. À la différence des logiciels traditionnels, l'IA peut traiter des données incomplètes et incertaines en créant des relations sémantiques entre les données, permettant ainsi des déductions sur le passé et des prédictions pour l'avenir. Son utilisation s'étend désormais largement, touchant non seulement les domaines de la conception et de

l'ingénierie, mais également l'éducation, la santé, le trafic ferroviaire, l'économie, la juridiction et la production [25, 27].

L'intelligence artificielle désigne le développement de systèmes informatiques capables d'accomplir des tâches qui exigent généralement l'intelligence humaine. Ces tâches comprennent l'apprentissage, le raisonnement, la résolution de problèmes, la compréhension du langage naturel, la reconnaissance vocale et la perception visuelle [28]. L'objectif de l'IA est de créer des machines ou des logiciels capables de reproduire des fonctions cognitives, leur permettant d'exécuter des activités complexes et de s'adapter à des situations changeantes sans programmation explicite. Il existe deux principaux types d'IA : l'IA étroite ou faible, conçue pour effectuer des tâches spécifiques, et l'IA générale ou forte, ayant la capacité de comprendre, d'apprendre et d'appliquer des connaissances dans divers domaines, à l'instar de l'intelligence humaine.

Les applications de l'IA sont variées et englobent différents secteurs tels que les soins de santé, la finance, l'éducation, la fabrication, etc. Le machine learning, une sous-catégorie de l'IA, utilise des algorithmes permettant aux systèmes d'apprendre et de s'améliorer de manière autonome à partir de l'expérience, sans nécessiter de programmation explicite. De plus, l'IA excelle par rapport à l'humain dans l'exécution de ces tâches en raison de sa capacité de calcul élevée, du traitement efficace de volumes massifs de données, et de sa prise de décision objective. Parmi les nombreuses techniques d'IA, les systèmes experts, la logique floue, les réseaux neuronaux artificiels et les algorithmes génétiques sont les méthodes classiques les plus couramment utilisées dans les processus d'évaluation et d'optimisation depuis très longtemps. Néanmoins, l'utilisation de l'approche basée sur les données, telle que l'apprentissage automatique et l'apprentissage profond, s'est renforcée dans plusieurs domaines au cours des dernières années [27, 29].



**Figure 3.1 :** Les discipline de l’Intelligence Artificielle [27].

### 3.3.1. Machine Learning (Apprentissage Automatique)

Avec l'émergence du big data et les progrès de la puissance de calcul, le Machine Learning est devenu de plus en plus populaire dans les applications d'ingénierie, car elle peut aider les ingénieurs à prendre des décisions plus éclairées et à optimiser des systèmes complexes plus efficacement.

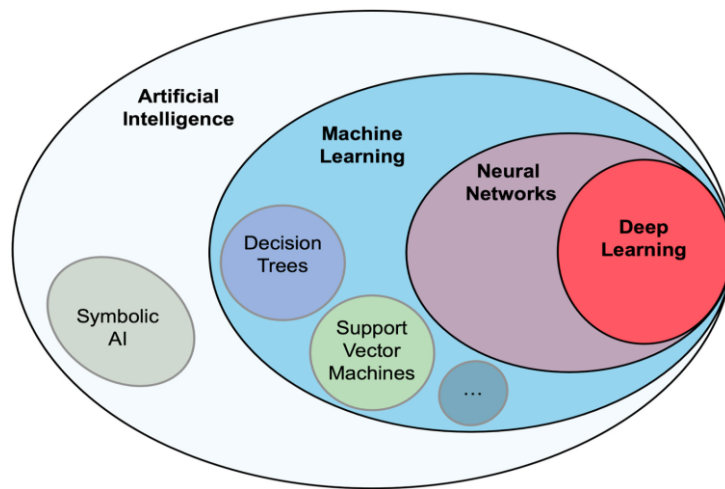
Le Machine Learning ou l'apprentissage automatique se définit comme une série de méthodes statistiques relevant de l'intelligence artificielle, visant à effectuer des prédictions et des déductions à partir de données. Il peut parfois être confondu avec la programmation classique, bien que l'apprentissage automatique ne nécessite pas d'algorithme explicite pour apprendre, contrairement à la programmation. Il repère plutôt des similitudes et des motifs au sein d'un ensemble de données donné, utilisant ces informations pour interpréter les données futures [27, 30, 31]. L'apprentissage automatique est un sous-ensemble de l'intelligence artificielle qui se concentre sur le développement d'algorithmes et de modèles permettant aux ordinateurs d'apprendre à partir de données et de faire des prédictions ou de prendre des décisions sans être explicitement programmés pour une tâche spécifique. L'idée centrale de

l'apprentissage automatique est de permettre aux ordinateurs de reconnaître des modèles, d'extraire des informations significatives et d'améliorer leurs performances au fil du temps sur la base des données auxquelles ils sont exposés et prendre des décisions avec un minimum d'intervention humaine [19, 30, 32].

L'apprentissage automatique est appliqué dans divers domaines, tels que le traitement du langage naturel, la vision par ordinateur, la reconnaissance vocale, les systèmes de recommandation et bien d'autres encore. Il joue un rôle crucial dans l'automatisation de tâches complexes et dans l'élaboration de prédictions basées sur de vastes ensembles de données. L'apprentissage statistique, qui forme la base des algorithmes modernes d'apprentissage automatique, est utilisé pour comprendre et interpréter les données. Bien que les concepts d'apprentissage statistique et d'apprentissage automatique se chevauchent, ils présentent des différences, notamment en termes d'interprétabilité. L'apprentissage statistique est appliqué dans des problèmes où il existe une relation linéaire entre les données d'entrée (variables indépendantes) et les données de sortie (variables dépendantes), comme dans la régression linéaire et la régression logistique. En raison de cette relation linéaire, il est possible de comprendre comment les changements dans les variables indépendantes affecteront la variable dépendante, ce qui donne une grande capacité explicative. L'apprentissage automatique, en revanche, est aujourd'hui la branche dominante de l'IA et se concentre sur l'amélioration des performances par l'acquisition d'expérience en termes de données. Parmi les nombreuses approches algorithmiques possibles de l'apprentissage automatique, les réseaux neuronaux sont devenus les plus populaires depuis 2010 environ, car leurs performances ne semblent pas saturer aussi facilement que celles d'autres méthodes. Les réseaux neuronaux comportant de nombreuses couches sont appelés réseaux neuronaux profonds et constituent la base de l'apprentissage profond [31, 33].

En revanche, dans l'apprentissage automatique, les résultats sont souvent plus difficiles à interpréter en raison de l'absence de relation linéaire claire entre les

variables d'entrée et de sortie. Dans certains cas, l'interprétabilité peut être négligée en raison de la facilité d'application et du succès de l'apprentissage automatique, en particulier sur de grands ensembles de données. Ces algorithmes améliorent leur capacité d'interprétation à mesure que la quantité de données augmente, de manière similaire au mécanisme d'apprentissage humain [34, 35].



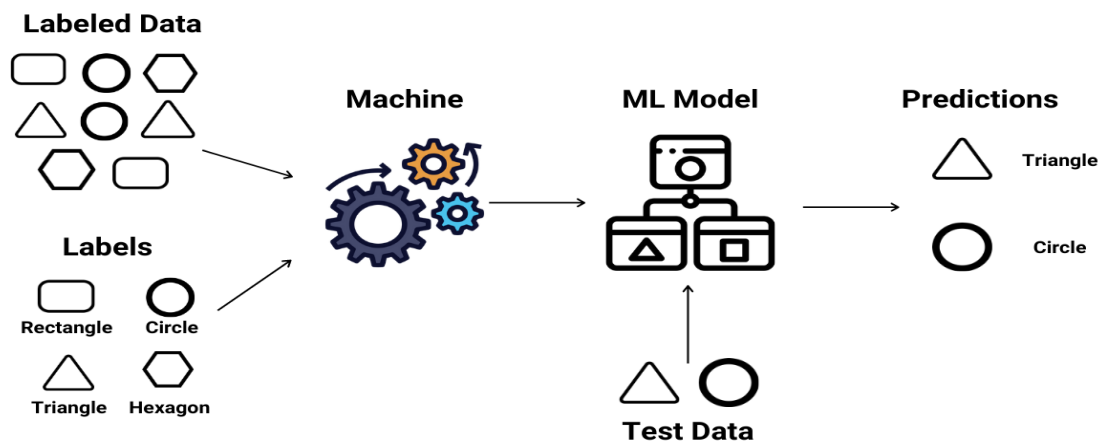
**Figure 3.2 :** Apprentissage automatique vs intelligence artificielle vs. Apprentissage profond.

Il existe quatre types de Machine Learning : l'apprentissage supervisé, non supervisé, semi-supervisé et par renforcement [34, 36].

#### **a. L'apprentissage supervisé**

L'apprentissage supervisé vise à obtenir la meilleure fonction (modèle) qui décrit le modèle entre les variables d'entrée et les variables de sortie étiquetées. En d'autres termes, l'apprentissage s'effectue en déterminant la relation entre les valeurs d'entrée et les valeurs d'étiquetage. Grâce à l'apprentissage supervisé, il est possible d'effectuer la classification de résultats composés d'un certain nombre de catégories (mauvais-moyen-bon, féminin-masculin, acceptation-rejet, etc.) ou des tâches de régression composées de nombres continus (température de l'air, salaire, poids, etc.). Les ensembles de données utilisés sont divisés en deux parties : entraînement et test.

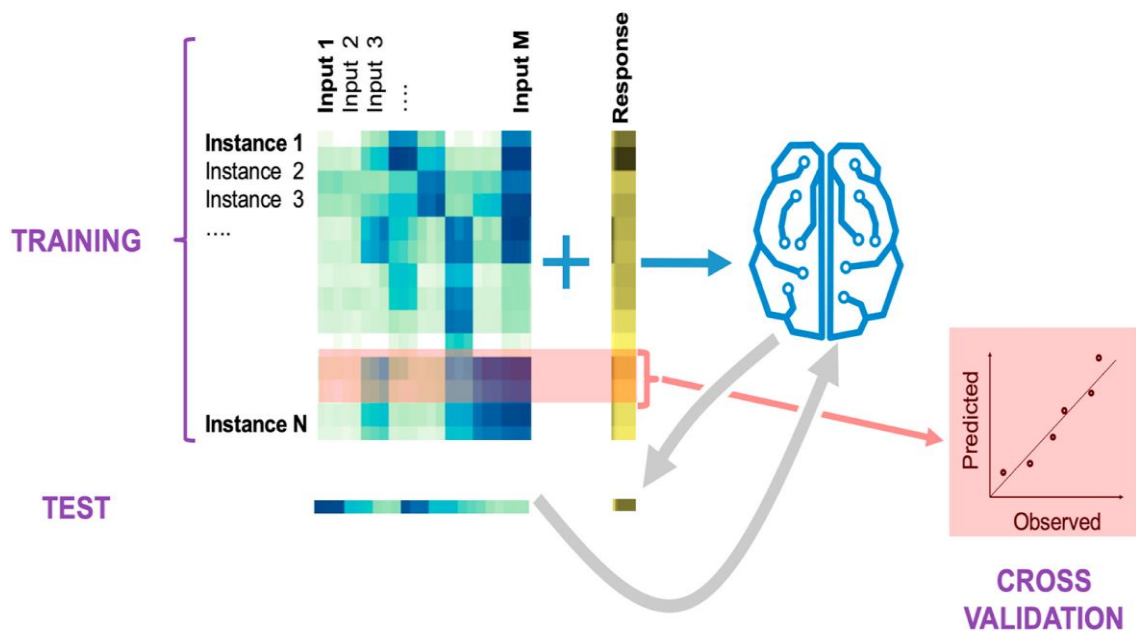
Alors que l'ensemble de données d'entraînement est utilisé pour la sélection des caractéristiques du modèle et l'ajustement des poids du réseau, le succès du modèle d'apprentissage automatique est mesuré à l'aide de l'ensemble de données de test. Il convient de noter ici que la création d'ensembles de données étiquetées est fastidieuse et que les relations d'entrée-sortie entre les données doivent être claires. Cette technique d'apprentissage présente un taux de réussite très élevé avec une marge d'erreur minimale. Les réseaux neuronaux artificiels, la régression logistique, la régression linéaire, les K-voisins les plus proches, les arbres de décision et les machines à vecteurs de support sont les algorithmes d'apprentissage automatique supervisé les plus courants [30-33, 37].



**Figure 3.3 :** Schéma d'un apprentissage supervisé [37].

Le flux de travail standard pour l'apprentissage automatique supervisé implique d'abord l'utilisation d'un ensemble de données de formation (comprenant les entrées, ou caractéristiques, et les réponses correspondantes, ou étiquettes) pour former l'algorithme choisi. L'ensemble de données d'apprentissage est composé d'instances ou d'exemples des entrées et des réponses à apprendre. Les instances dépendent du problème à apprendre : il peut s'agir de différentes souches et conditions, de points dans le temps. L'objectif est que l'algorithme soit capable de prédire la réponse pour des entrées qu'il n'a jamais vues auparavant (c'est-à-dire qui ne faisaient pas partie de l'ensemble d'apprentissage), ce qui constitue le test ultime de ses performances. Un

moyen de prévoir les performances de l'algorithme dans le cadre d'un tel test consiste à n'utiliser qu'une partie de l'ensemble de données d'apprentissage pour l'apprentissage, puis à vérifier les prédictions pour les entrées restantes, afin de les comparer avec les réponses connues. Cette procédure est appelée validation et, si elle est effectuée plusieurs fois en conservant au hasard une fraction de l'ensemble de données d'apprentissage, elle prend le nom de validation croisée. La validation croisée est un bon moyen de déterminer la complexité nécessaire de l'algorithme [37].

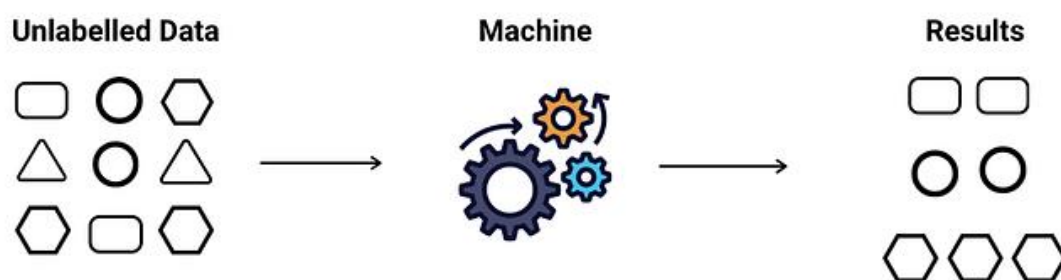


**Figure 3.4 :** Terminologie de l'apprentissage automatique [37].

### **b. L'apprentissage non supervisé**

L'algorithme d'apprentissage non supervisé permet au modèle d'IA de découvrir et de détecter lui-même les données présentant des caractéristiques communes à partir d'un ensemble de données non étiquetées et de révéler des modèles cachés parmi les données. Contrairement à l'apprentissage supervisé, il n'est pas nécessaire d'avoir un superviseur pour l'apprentissage du modèle. Cependant, il est difficile d'évaluer objectivement les performances des modèles. Ces algorithmes sont généralement

utilisés pour détecter la distribution générale des données, le regroupement et la réduction de la taille. Le regroupement K-moyennes, les règles d'association, l'auto-encodeur et l'analyse en composantes principales sont les algorithmes d'apprentissage non supervisé les plus couramment utilisés. Le regroupement K-means permet de diviser les données en groupes significatifs en tenant compte des distances entre les données. D'autre part, les règles d'association tentent de définir les relations entre les données. L'analyse en composantes principales et l'auto-encodeur permettent de réduire la taille des données afin d'augmenter la vitesse de traitement et de réduire la charge de calcul inutile [30-32, 37].



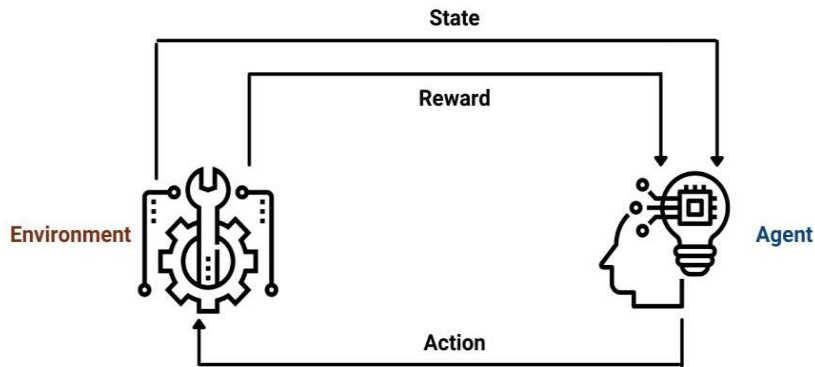
**Figure 3.5 :** Schéma d'un apprentissage non supervisé [37].

### c. L'apprentissage par renforcement

L'apprentissage par renforcement est très prometteur, tout comme la méthode traditionnelle d'essai et d'erreur. Cet apprentissage consiste à faire apprendre aux ordinateurs à partir de leurs expériences grâce à un système de récompense ou de pénalité. Dans ce cas, une série d'actions sont entreprises pour effectuer certaines tâches. Les actions qui aboutissent au bon résultat sont récompensées (approuvées), et celles qui aboutissent au mauvais résultat sont punies (rejetées). Par exemple, la consommation d'énergie et de temps d'un robot qui tente d'atteindre sa cible par le chemin le plus court peut être considérée comme une récompense ou une punition. L'objectif principal de l'apprentissage par renforcement est de développer des tactiques de haut niveau en commençant par des essais aléatoires. Cette méthode



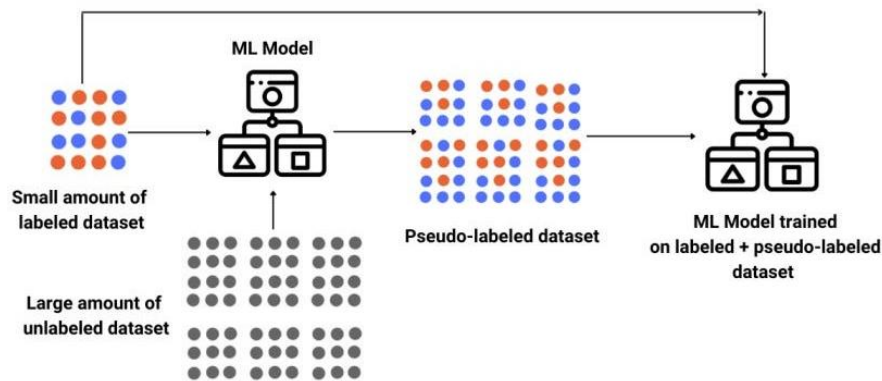
explore des solutions de manière autonome, mais le développeur doit définir des objectifs clairs et fixer des récompenses et des pénalités. Lorsque tout cela est pris en compte, le modèle d'apprentissage par renforcement présente le potentiel de créativité le plus élevé, car il peut trouver différentes façons de résoudre un problème [30, 37].



**Figure 3.6 :** Schéma d'un apprentissage par renforcement [37].

#### **d. L'apprentissage semi-supervisé**

L'apprentissage semi-supervisé est une approche intermédiaire entre l'apprentissage supervisé et l'apprentissage non supervisé. Dans ce type d'apprentissage, le modèle est formé sur un ensemble de données qui contient à la fois des exemples étiquetés et des exemples non étiquetés. L'objectif est d'utiliser à la fois les informations fournies par les exemples étiquetés et la structure sous-jacente des exemples non étiquetés pour améliorer les performances du modèle. Contrairement à l'apprentissage supervisé, qui utilise uniquement des exemples étiquetés pour former le modèle, et à l'apprentissage non supervisé, qui utilise des données non étiquetées sans référence à des étiquettes, l'apprentissage semi-supervisé combine les deux approches. Cela peut être particulièrement utile lorsque l'acquisition d'étiquettes pour l'ensemble des données est coûteuse ou impraticable [30, 34, 37].



**Figure 3.7 :** Schéma d'un apprentissage semi-supervisé [37].

### 3.3.2. Le réseau de neurones artificiel et l'apprentissage profond (Deep Learning)

**Les réseaux de neurones artificiels** (Artificial Neural Network ; ANN) est basé sur la structure biologique des réseaux neuronaux des organismes vivants [19-20]. Les réseaux neuronaux artificiels (RNA) sont des modèles informatiques inspirés de la structure et du fonctionnement du cerveau humain. Il s'agit d'un sous-ensemble d'algorithmes d'apprentissage automatique conçus pour reconnaître des modèles, faire des prédictions et effectuer diverses tâches liées à l'analyse des données. La structure d'un réseau neuronal artificiel consiste en des couches de neurones interconnectées. Ces couches comprennent généralement une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque neurone est associé à des poids et à une fonction d'activation qui déterminent sa contribution à la sortie du réseau. Le processus d'apprentissage dans les réseaux neuronaux artificiels implique souvent une rétropropagation, où le modèle ajuste les poids de connexion sur la base des erreurs entre ses prédictions et les étiquettes réelles dans les données d'apprentissage. La principale caractéristique de l'ANN est sa capacité à résoudre des problèmes complexes et non formulés grâce à l'optimisation des poids du réseau entre les neurones interconnectés où l'information est activée [19].

Les réseaux neuronaux artificiels sont appliqués dans divers domaines, tels que la vision artificielle, le traitement du langage naturel, la reconnaissance vocale et la prédiction de séries temporelles. Ils sont particulièrement efficaces pour résoudre des problèmes complexes où des modèles et des relations non linéaires existent dans les données. Les réseaux de neurones artificiels sont parmi les algorithmes de classification mathématique les plus largement utilisés dans les problèmes de détection et de diagnostic des défauts. Les ANN sont connues pour leurs capacités d'auto-apprentissage efficaces des associations complexes qui existent dans les problèmes de détection et de diagnostic des défauts. Les ANN effectuent une extraction automatique des caractéristiques en désignant des poids négligeables aux caractéristiques non pertinentes [38].

Le **Deep Learning**, ou "apprentissage profond" en français, est un sous-ensemble de l'apprentissage automatique qui se concentre sur l'utilisation de réseaux neuronaux artificiels pour modéliser et résoudre des problèmes complexes. L'objectif est de former des réseaux neuronaux profonds avec plusieurs couches (d'où le terme "profond") pour apprendre automatiquement des représentations hiérarchiques des données. Ces réseaux ont la capacité de découvrir automatiquement des modèles, des caractéristiques et des représentations à partir de données brutes sans programmation explicite.

### **3.3.3. Ensemble Learning**

L'apprentissage d'ensemble ou Ensemble Learning est une méthode d'apprentissage automatique qui vise à capturer la sagesse d'une décision de groupe. L'apprentissage d'ensemble consiste à combiner les prédictions de plusieurs modèles pour améliorer la performance globale. Plutôt que de s'appuyer sur un seul modèle, l'idée est d'exploiter la diversité des modèles individuels pour obtenir des prédictions plus robustes et précises [6, 31, 39-42].

Il existe plusieurs techniques d'ensemble learning, dont les deux principales sont [31, 39, 41, 42] :

- **Bagging (Bootstrap Aggregating)** : Le "bagging", abréviation de "Bootstrap aggregating", est principalement utilisé dans les domaines de la classification et de la régression. Dans le bagging, plusieurs instances du même modèle sont entraînées sur des sous-ensembles aléatoires des données d'entraînement, souvent obtenus par un échantillonnage avec remplacement (Bootstrap). Les prédictions de chaque modèle sont ensuite combinées, souvent par moyenne (pour la régression) ou par vote majoritaire (pour la classification).
- **Boosting** : Le Boosting est une technique d'ensemble qui apprend des erreurs des prédicteurs précédents pour faire de meilleures prédictions à l'avenir. Cette technique combine plusieurs modèles faibles de base pour former un modèle fort, augmentant significativement la capacité prédictive. Le Boosting opère en disposant les modèles faibles dans une séquence, permettant à chaque modèle faible d'apprendre des erreurs de celui qui le suit dans la séquence, contribuant ainsi à la création de modèles plus performants.

Les avantages de l'apprentissage d'ensemble incluent une meilleure robustesse contre le surajustement, une généralisation améliorée et une réduction des risques associés à la variation des données. Les algorithmes d'apprentissage d'ensemble courants incluent les Random Forest (bagging) et Gradient Boosting (Boosting), tels que XGBoost et AdaBoost. Ces techniques sont largement utilisées dans des domaines tels que la classification, la régression, et même le classement.

### 3.3.4. Prédiction des défauts

La détection et l'isolation des défauts sont des domaines clés de l'ingénierie des procédés modernes. Les défaillances ou défauts peuvent être définies comme des écarts par rapport aux plages acceptables pour les variables observées ou les

paramètres calculés associés au processus. Aujourd'hui, la détection et la prédiction des défauts est très importante pour les entreprises afin d'optimiser les coûts de maintenance et éviter les pannes soudaines. La détection et la prédiction des défauts fournit un avertissement précieux sur les problèmes qui peuvent survenir dans les systèmes. Ainsi, le diagnostic des pannes consiste à déterminer le type, l'ampleur et la localisation de la panne.

La prédiction de défauts, ou "Fault prediction", fait référence au processus d'utilisation des données et des analyses pour prévoir ou anticiper les défaillances ou les pannes potentielles d'un système, d'une machine ou d'un équipement. Elle consiste à prédire, analyser et mesurer les défaillances possibles sur la base de l'état actuel de l'équipement, afin d'aider les décideurs à comprendre les informations relatives aux défaillances de l'équipement et à élaborer à l'avance des stratégies de maintenance efficaces. Cette approche proactive vise à identifier les problèmes avant qu'ils ne surviennent, ce qui permet d'effectuer des opérations de maintenance ou de prendre des mesures correctives en temps voulu pour éviter les temps d'arrêt, réduire les coûts et améliorer la fiabilité globale du système.

Avec le développement des technologies de l'information, les modèles de prévision des défaillances basés sur les données remplacent les méthodes traditionnelles basées sur les modèles. La prédiction des défaillances implique souvent l'utilisation d'analyses prédictives, d'algorithmes d'apprentissage automatique et de données historiques pour analyser les schémas et les tendances susceptibles d'indiquer une défaillance imminente.

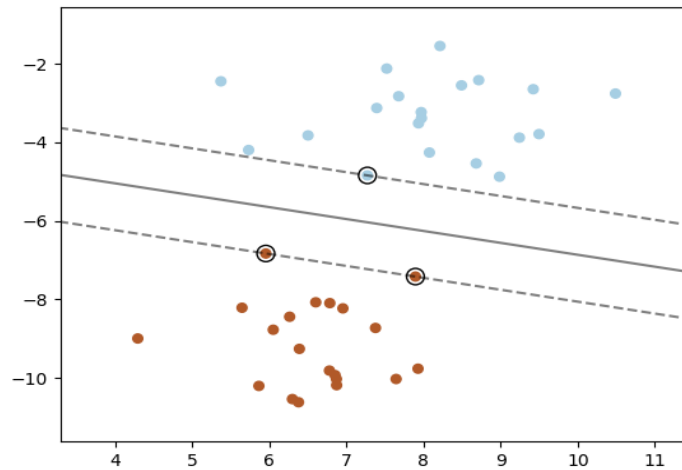
Dans les recherches existantes sur les méthodes de prédiction des défaillances basées sur les données, l'accent est mis sur la prédiction de la probabilité de défaillance, du type de défaillance ou de l'emplacement de la défaillance. Cependant, tout défaut passe par une série de changements d'état depuis l'apparition du signal jusqu'à

l'apparition complète, et les mesures de maintenance et les exigences d'urgence sont différentes aux différents stades de détérioration d'un même défaut.

Dans ce qui suit, nous présenterons les méthodes d'apprentissage automatique qui permettent de prédire les défauts et qui seront utilisées dans notre étude ;

#### **3.3.4.1. Support Vector Machine (SVM) :**

Support Vector Machine est un algorithme de l'apprentissage supervisé. Cet algorithme de classification a été introduit par V. Vapnik en 1992 [10, 31]. Cet algorithme était destiné pour les tâches de classification mais aujourd'hui grâce aux développements qui ont été apportés, il peut être utilisées pour effectuer la classification ainsi que la régression. Il s'agit d'une méthode de classification binaire dont l'objectif principale est de trouver un hyperplan séparant au mieux les données et dont la séparation (ou "marge", i.e. la distance séparant l'hyperplan du plus proche exemple) est la plus grande possible. Cette recherche correspond à un problème d'optimisation au cours duquel les exemples plus proches de l'hyperplan, appelés vecteurs supports, sont sélectionnés. L'hyperplan calculé permet ainsi de séparer l'espace en deux zones. Intuitivement, une bonne séparation est obtenue par l'hyperplan qui a la plus grande distance avec les points de données d'apprentissage les plus proches de n'importe quelle classe (ce qu'on appelle la marge fonctionnelle), car en général, plus la marge est grande, plus l'erreur de généralisation du classificateur est faible. La figure ci-dessous montre la fonction de décision pour un problème linéairement séparable, avec trois échantillons sur les limites de la marge, appelés "vecteurs de support" [19, 31, 38, 39, 42-45];



**Figure 3.8 :** L'algorithme Support Vector Machine [43].

Les avantages des machines à vecteurs de support sont les suivants [43] :

- Efficaces dans les espaces à haute dimension.
- Toujours efficaces dans les cas où le nombre de dimensions est supérieur au nombre d'échantillons.
- Elles utilisent un sous-ensemble de points d'apprentissage dans la fonction de décision (appelés vecteurs de support), ce qui les rend également efficaces en termes de mémoire.
- Polyvalent : différentes fonctions de noyau peuvent être spécifiées pour la fonction de décision. Les noyaux courants sont fournis, mais il est également possible de spécifier des noyaux personnalisés.

Les inconvénients des machines à vecteurs de support sont les suivants :

- Si le nombre de caractéristiques est beaucoup plus grand que le nombre d'échantillons, il est crucial d'éviter le surajustement dans le choix des fonctions Kernel et du terme de régularisation.
- Les SVM ne fournissent pas directement d'estimations de probabilité, celles-ci sont calculées à l'aide d'une validation croisée cinq fois coûteuse.

### 3.3.4.2. Logistic Regression (LR)

La régression logistique est l'un des algorithmes fondamentaux et populaires pour résoudre un problème de classification. LR est un algorithme de classification d'apprentissage automatique supervisé utilisé pour observer un ensemble discret de classes. La régression logistique est une méthode statistique utilisée pour modéliser la probabilité d'un événement binaire en fonction d'une ou plusieurs variables indépendantes. On l'appelle "régression logistique" parce que la technique sous-jacente est assez semblable à la régression linéaire. Le terme "logistique" provient de la fonction Logit qui est utilisée dans cette méthode de classification. Contrairement à la régression linéaire, qui prédit des valeurs continues, la régression logistique prédit la probabilité qu'un événement appartienne à la classe positive (1) ou à la classe négative (0) [39, 45, 46].

Le modèle de régression logistique utilise une fonction logistique pour transformer une combinaison linéaire des variables indépendantes en une probabilité comprise entre 0 et 1. Cette fonction logistique (ou sigmoïde) s'ajuste bien aux problèmes de classification binaire. L'équation de la régression logistique peut être exprimée comme suit, où  $P(Y = 1)$  est la probabilité que l'événement soit de la classe positive :

$$P(Y = 1) = \frac{1}{1 + e^{-(b_0 + b_1x_1 + b_2x_2 + \dots + b_kx_k)}}$$

Dans cette équation,  $b_0, b_1, b_2, \dots, b_k$  sont les coefficients du modèle, et  $x_1, x_2, \dots, x_k$  sont les variables indépendantes. L'estimation des coefficients se fait généralement par des méthodes d'optimisation [45].

### 3.3.4.3. K-Nearest Neighbors (K-voisins les plus proches)

L'algorithme des K-voisins les plus proches (K-NNA) est un algorithme d'apprentissage automatique de base qui suit les techniques d'apprentissage supervisé. L'un des avantages de l'algorithme KNN est qu'il peut traiter à la fois les tâches de classification et de régression. Pour les tâches de régression, la moyenne des



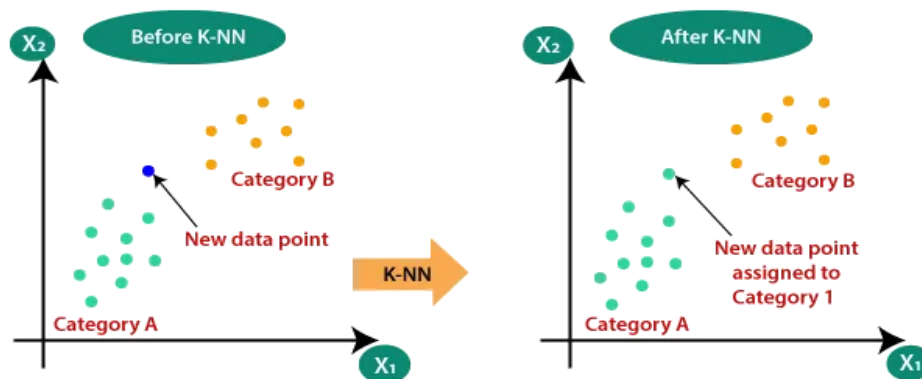
étiquettes des K plus proches voisins est utilisée pour prédire l'étiquette du nouveau point de données. Un autre avantage de l'algorithme KNN est qu'il est simple à comprendre et à mettre en œuvre [31, 47].

Cet algorithme suppose qu'il existe une similitude entre le nouvel ensemble de données sur les défauts et l'ensemble de données sur les défauts déjà disponibles, et il classera le nouvel ensemble de données dans les catégories disponibles les plus similaires. Il stocke toutes les données disponibles sur les défauts et les classe en fonction de leur similarité. Lorsque de nouvelles données apparaissent, l'algorithme peut facilement les classer en les comparant aux ensembles de données existants. Pour déterminer la similarité entre deux objets, l'algorithme utilise la méthode de la distance euclidienne [39, 42, 44, 45, 47, 48].

Voici les principaux concepts de l'algorithme k-NN :

- Voisinage : L'algorithme se base sur la proximité entre les points dans l'espace des caractéristiques. Le voisinage d'un point est déterminé par la distance entre les points, généralement mesurée par la distance euclidienne.
- Paramètre k : Le nombre de voisins (k) est un paramètre crucial de l'algorithme. Pour chaque point à classer ou pour chaque prédiction à faire, l'algorithme regarde les k voisins les plus proches.

Pour la classification, l'algorithme attribue la classe majoritaire parmi les k voisins les plus proches. Par exemple, si la majorité des k voisins appartiennent à la classe A, le point d'intérêt sera classé comme appartenant à la classe A. L'algorithme k-NN est simple et efficace, mais le choix de la valeur de k et de la mesure de distance peuvent affecter ses performances. Il est souvent utilisé dans des applications telles que la recommandation de produits, la reconnaissance de motifs et la classification d'images [47, 48].



**Figure 3.9 :** L'apprentissage d'un algorithme KNN [48].

#### 3.3.4.4. Naïve Bayes (NB)

L'algorithme de Naïve Bayes est largement utilisé dans l'apprentissage automatique pour des tâches de classification. Fondé sur le théorème de Bayes, qui permet de mettre à jour la probabilité d'une hypothèse en fonction de nouvelles preuves, Naïve Bayes est un algorithme simple qui se révèle efficace avec de grands ensembles de données. Il est fréquemment employé dans des domaines tels que le traitement du langage naturel, le filtrage des spams et l'analyse des sentiments. La caractéristique "naïve" de l'algorithme provient de son hypothèse selon laquelle chaque caractéristique de l'ensemble de données est indépendante des autres, ce qui simplifie considérablement le processus. Bien que cette simplification puisse sembler trop optimiste, Naïve Bayes donne souvent des résultats satisfaisants en pratique, surtout avec des ensembles de données volumineux contenant de nombreuses caractéristiques. L'algorithme calcule la probabilité conditionnelle de chaque classe en se basant sur les valeurs des caractéristiques. Il commence par estimer la probabilité a priori de chaque classe à partir des données d'apprentissage, puis détermine la probabilité conditionnelle de chaque caractéristique pour chaque classe. Enfin, il utilise le théorème de Bayes pour combiner ces probabilités et déterminer la probabilité finale de chaque classe en fonction des valeurs des caractéristiques [19, 39, 42, 49-51].

Mathématiquement, l'algorithme cherche à calculer la probabilité conditionnelle d'appartenir à une classe donnée ( $P(C|X)$ ), où  $C$  représente la classe et  $X$  les caractéristiques de l'exemple. En utilisant le théorème de Bayes, cette probabilité peut être formulée comme suit [19, 51] :

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)}$$

Dans cette équation :

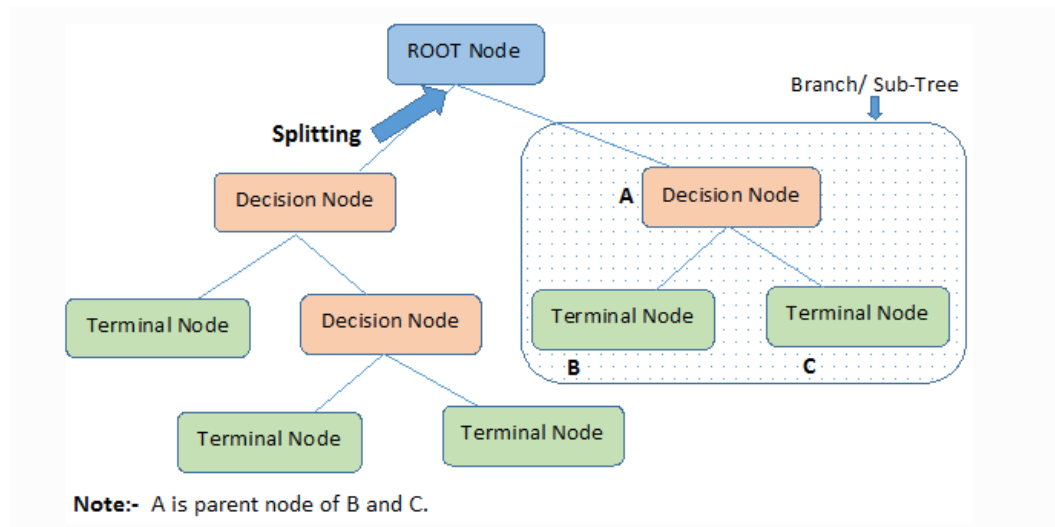
- $P(C|X)$  est la probabilité conditionnelle de la classe  $C$  sachant les caractéristiques  $X$ .
- $P(X|C)$  est la probabilité conditionnelle des caractéristiques  $X$  sachant la classe  $C$ .
- $P(C)$  est la probabilité a priori de la classe  $C$ .
- $P(X)$  est la probabilité marginale des caractéristiques  $X$ .

L'algorithme attribue l'exemple à la classe avec la probabilité conditionnelle la plus élevée. La simplicité de Naïve Bayes et ses performances acceptables en font un choix populaire dans de nombreuses applications.

### **3.3.4.5. Decision Tree (Arbres de décision)**

Les arbres de décision (DT) constituent une catégorie d'algorithmes d'apprentissage automatique fréquemment employés pour la classification et la prédiction. Ils opèrent dans le contexte de l'apprentissage supervisé et utilisent une structure arborescente pour prendre des décisions fondées sur un ensemble de conditions. Les arbres de décision sont particulièrement prisés dans des secteurs tels que les soins de santé et la finance, où l'interprétabilité et la transparence sont des critères importants. Un arbre de décision est constitué d'un nœud racine, de nœuds internes et de nœuds feuilles. Le nœud racine représente l'ensemble des données, chaque nœud interne symbolise une décision basée sur un attribut spécifique, les branches issues de chaque nœud interne

représentent les résultats possibles de cette décision, et les nœuds feuilles représentent la classification ou la prédiction finale. Dans les arbres de décision, deux types de nœuds existent : les nœuds de décision et les nœuds feuilles. Les nœuds de décision sont utilisés pour prendre des décisions et contiennent des branches menant vers d'autres nœuds de décision ou nœuds feuilles. Les nœuds feuilles, quant à eux, ne possèdent pas de branches et représentent la sortie finale ou la prédiction. L'algorithme des arbres de décision prend des décisions en fonction des caractéristiques de l'ensemble de données fourni. Il pose des questions et, en fonction des réponses (oui ou non), il divise l'ensemble de données en sous-arbres. L'algorithme vise à maximiser la valeur du gain d'information, divisant en premier le nœud présentant le gain d'information le plus élevé. Les arbres de décision peuvent être utilisés pour la classification et résoudre des problèmes de régression lorsqu'ils sont associés à un algorithme de régression au niveau des feuilles [6, 9, 19, 39, 42, 52, 53].

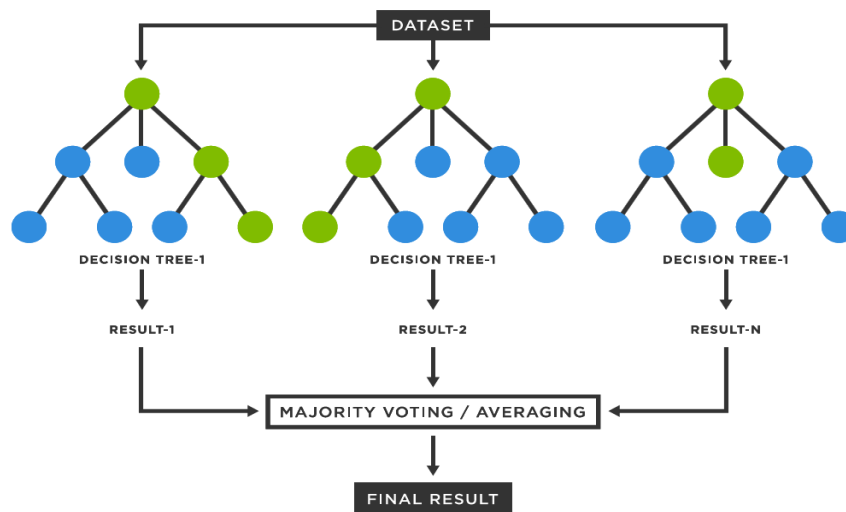


**Figure 3.10 :** Structure d'un algorithme des arbres de décisions [53].

#### **3.3.4.6. Random Forest**

Forêt aléatoire ou Random Forest est une méthode d'apprentissage ensembliste utilisée pour la classification et la régression. Il est appelé "forêt" parce qu'il développe une forêt d'arbres de décision. Cet algorithme construit plusieurs arbres de décision en sélectionnant aléatoirement des caractéristiques et en utilisant un échantillonnage bootstrap, introduisant ainsi de la diversité. La prédiction finale résulte d'un vote majoritaire (classification) ou d'une moyenne (régression) (voir la figure). Alors qu'un arbre de décision unique présente un seul résultat et une gamme étroite de groupes, la forêt garantit un résultat plus précis avec un plus grand nombre de groupes et de décisions. La popularité de l'algorithme Random Forest provient de sa convivialité, sa robustesse, ses performances élevées et de son adaptabilité, qui lui permettent de résoudre de manière efficace les problèmes de classification et de régression. La puissance de cet algorithme réside dans sa capacité à manipuler des ensembles de données complexes tout en atténuant le surajustement, en faisant ainsi un outil précieux pour une variété de tâches prédictives dans le domaine de l'apprentissage automatique [45, 52, 54, 55].

Une caractéristique clé de l'algorithme Random Forest est sa capacité à traiter des ensembles de données comprenant à la fois des variables continues, comme dans le cas de la régression, et des variables catégorielles, comme dans le cas de la classification. Sa performance élevée dans les tâches de classification et de régression en fait un choix privilégié pour ces types de problèmes [54].



**Figure 3.11 :** Structure d'un algorithme Random Forest [54].

### 3.3.4.7. Gradient Boost

Le Gradient Tree Boosting, également appelé Gradient Boosted Trees (GBT), est un algorithme d'apprentissage automatique qui fusionne plusieurs modèles d'arbres de décision pour former un modèle global plus performant et résistant. Il fait partie des méthodes d'ensemble, où plusieurs modèles sont entraînés de manière séquentielle et leurs résultats sont combinés pour améliorer les performances globales. À chaque itération, des ajustements sont apportés aux coefficients, poids ou biais associés à chaque variable d'entrée utilisée pour prédire la valeur cible. L'objectif est de minimiser la fonction de perte, qui mesure la différence entre la valeur cible prédite et la valeur cible réelle. Le gradient représente les ajustements progressifs effectués à chaque étape, et l'amplification est une méthode visant à accélérer l'amélioration de la précision prédictive jusqu'à atteindre une valeur optimale [39, 55, 56].

### 3.3.4.8. AdaBoost

AdaBoost (Adaptive Boosting) est un algorithme d'apprentissage ensembliste utilisé pour les tâches de classification. Il a été conçu pour améliorer la précision des classificateurs faibles en les combinant en un classificateur fort. Adaboost utilise des

souches de décision, qui sont des arbres de décision avec une seule division, comme apprenants faibles. Son efficacité réside dans sa capacité de se concentrer sur les instances difficiles à classer, en adaptant le modèle pour qu'il soit performant sur l'ensemble de la base de données. Il assigne des poids aux observations, donnant plus de poids aux cas difficiles à classer et moins de poids aux cas faciles. L'objectif est de se concentrer sur les cas difficiles à classer pour chaque nouvel apprenant faible, et le résultat final est la moyenne des résultats pondérés de tous les apprenants individuels. Cependant, il peut être sensible aux données bruitées et aux valeurs aberrantes. L'algorithme a été utilisé avec succès dans diverses applications, et son concept de renforcement a inspiré d'autres méthodes d'apprentissage d'ensemble [31, 39, 55].

En revanche, l'algorithme de gradient boosting, bien qu'il utilise également des arbres de décision, diffère légèrement d'Adaboost. Au lieu d'utiliser une moyenne pondérée pour les résultats finaux, il minimise une fonction de perte pour converger vers une valeur de sortie finale. L'optimisation de la fonction de perte se fait par descente de gradient, d'où le nom "gradient boosting". De plus, le gradient boosting utilise des arbres de décision courts et moins complexes au lieu de souches de décision.

#### **3.3.4.9. XGBoost**

XGBoost, qui signifie eXtreme Gradient Boosting, est un algorithme très puissant d'apprentissage automatique connu pour son efficacité dans une variété de tâches, en particulier dans les données structurées/tabulaires. Il appartient à la famille des algorithmes de boosting de gradient et est une version améliorée qui répond à certaines limites du boosting de gradient traditionnel. Cet algorithme appartient à la catégorie de l'apprentissage ensembliste. Il utilise des arbres de décision comme apprenants de base et emploie des techniques de régularisation pour améliorer la généralisation du modèle. À chaque étape, il corrige les erreurs du modèle précédent en se concentrant sur les exemples mal prédits. XGBoost intègre une régularisation pour éviter le surajustement, améliorant ainsi la généralisation du modèle. Contrairement au boosting traditionnel, XGBoost utilise des approximations dérivées

pour minimiser la perte de manière plus efficace. Grâce à une mise en œuvre optimisée et à la distribution parallèle, XGBoost permet un apprentissage plus rapide et est largement utilisé dans diverses applications pour ses performances exceptionnelles. Connu pour son efficacité de calcul, son analyse de l'importance des caractéristiques et sa gestion des valeurs manquantes, XGBoost est largement utilisé pour des tâches telles que la régression, la classification [39, 46, 55, 56].

### **3.4. Outils utilisés**

#### **3.4.1. Matlab / Simulink**

MATLAB est un environnement de programmation et un langage de script de haut niveau développé par MathWorks. Il est largement utilisé par les ingénieurs et les scientifiques pour la modélisation numérique, le traitement de signal, l'analyse de données, l'algèbre linéaire, l'optimisation, et bien d'autres domaines liés au calcul mathématique et à l'ingénierie. MATLAB facilite la manipulation des matrices, la visualisation de données sous forme de graphiques, l'implémentation d'algorithmes, la création d'interfaces utilisateur, et l'interfaçage avec des programmes écrits dans d'autres langages, y compris C, C++, Java, et Python.

MATLAB est accompagné de diverses toolboxes spécialisées qui étendent ses fonctionnalités dans des domaines spécifiques, comme le traitement de signal, l'apprentissage automatique, le contrôle de systèmes, et bien d'autres. Pour notre étude, nous avons choisi de nous appuyer sur la "Signal Processing Toolbox".

Simulink, un outil complémentaire à MATLAB, a été essentiel dans la conception et la simulation de notre système électrique. Grâce à son interface graphique basée sur des schémas blocs, Simulink a facilité la modélisation précise de notre système, permettant des simulations dynamiques et la génération de données de simulation.

#### **3.4.2. Python**

Nous avons choisi Python comme langage de programmation pour développer nos



algorithmes d'apprentissage automatique. Ce choix s'est avéré judicieux en raison de la richesse de l'écosystème Python, qui offre une variété de bibliothèques spécialisées telles que scikit-learn, TensorFlow, et PyTorch. Ces outils facilitent grandement l'implémentation, le test et l'optimisation des modèles d'apprentissage automatique, permettant ainsi une exploration approfondie des données et une modélisation précise. Python, reconnu pour sa syntaxe claire et sa facilité d'utilisation, a également contribué à accélérer le développement de nos algorithmes et à améliorer la collaboration au sein de notre équipe de recherche.

### **3.4.3. Spyder**

Pour notre étude, nous avons sélectionné Spyder comme environnement de développement intégré (IDE) de choix. Spyder, spécifiquement conçu pour les scientifiques, les ingénieurs et les analystes de données travaillant avec le langage de programmation Python, s'est révélé être un outil exceptionnellement adapté à nos besoins. Avec ses fonctionnalités avancées telles que l'analyse de code en temps réel, la visualisation de données intégrée, et un explorateur de variables interactif, Spyder a facilité la programmation, le débogage, et l'analyse des données. Sa capacité à intégrer directement des bibliothèques scientifiques et d'apprentissage automatique comme NumPy, SciPy, Matplotlib, et scikit-learn dans un seul et même environnement a considérablement optimisé notre flux de travail, permettant une exploration et une expérimentation efficaces avec nos modèles d'apprentissage automatique.

## **3.5. Conclusion**

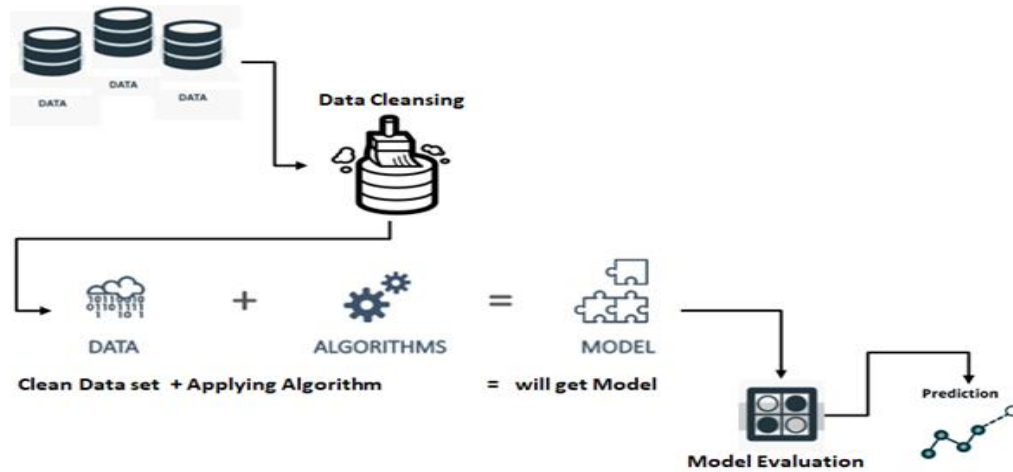
Ce chapitre a offert une exploration détaillée de la méthodologie employée pour la prédiction des défaillances au sein des transformateurs électriques, soulignant l'approche méthodique adoptée ainsi que les outils sélectionnés pour mener à bien cette recherche.

## **Chapitre 4**

# **Prédiction et localisation des défauts des transformateurs en utilisant des données générées**

### **4.1. Stratégie d'apprentissage et processus**

Dans le processus d'apprentissage automatique, nous amorçons notre étude en générant et en acquérant les données nécessaires. Dans cette démarche, un modèle électrique triphasé a été mis en place pour faciliter la génération des données. Cependant, les données issues de ce modèle peuvent initialement être non structurées, demandant ainsi un prétraitement pour résoudre des problèmes potentiels tels que les valeurs manquantes et les informations non pertinentes. Des modules de prétraitement sont disponibles grâce aux outils d'apprentissage automatique. Cette étape cruciale transforme les données générées par le modèle électrique simulé en un format structuré, adapté à l'algorithme d'apprentissage automatique ultérieur. Par la suite, les données sont séparées en ensembles d'entraînement et de test. Ensuite, l'entraînement du modèle est réalisé en utilisant les données d'entraînement, tandis que les données de test sont employées pour effectuer des prédictions. Enfin, une évaluation comparative des résultats est menée.



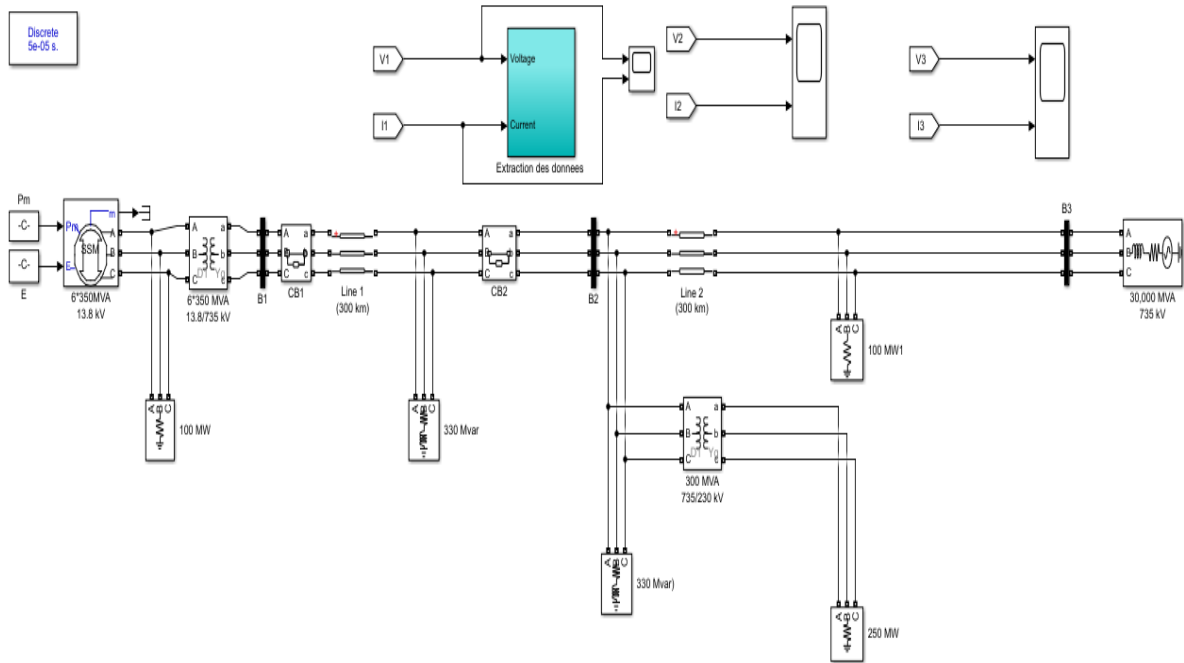
**Figure 4.1 :** Stratégie d'apprentissage [57].

## 4.2. Implémentation du modèle par simulation

Le modèle de réseau électrique triphasé considéré est simulé sous l'environnement MATLAB/Simulink. Il s'agit d'un système de 735kV. Il se compose d'une ligne de transmission qui transmet une puissance de 3000 MW d'une centrale de production (6 générateurs de 350 MVA) à un réseau équivalent ayant un niveau de court-circuit de 30 GVA. La centrale de production est simulée avec une machine synchrone simplifiée (réactance subtransitoire de 0,22 pu). La machine est connectée au réseau de transmission par un transformateur Wye-Delta de 13,8 kV/735 kV.

Le système se compose des disjoncteurs, d'appareillage de mesure, des charges, un bloc pour le traitement des signaux et extraction des caractéristiques fondamentales.

Le système choisi pour notre étude est le suivant :



**Figure 4.2 : Système électrique.**

L'alimentation électrique du réseau est générée par la source de tensions triphasées et la génératrice, et transmise aux charges à travers les lignes de transport électrique, les transformateurs servent à adapter le niveau de tension aux nœuds de connexion, c'est-à-dire desservir les charges avec la tension appropriée. Les disjoncteurs CB1 et CB2 servent à établir ou interrompre l'alimentation électrique entre les parties du réseau, dans notre cas, ils servent au cas de l'occurrence du défaut à isoler les transformateurs en défaut pour éviter des dommages à l'ensemble du reste du système électrique. Cet élément joue un rôle essentiel dans la protection des équipements électriques. Les blocs B1, B2 et B3 servent à mesurer les valeurs triphasées du courant et tension, ces dernières vont être introduites au bloc 'Extraction des données' afin d'être traitées et filtrées pour générer la base des données.

### 4.3. Génération et prétraitement des données

Une fois que le modèle électrique est construit, celui-ci a été utilisé pour créer la base de données de défauts servant à concevoir les modèles de Machine Learning destinés à la prédiction et la localisation des défauts de transformateurs électriques. Cette base de données est constituée des valeurs triphasées de tension ( $V_A, V_B, V_C$ ) et de courant ( $I_A, I_B, I_C$ ) mesurées au niveau du bus de mesure triphasé, tant pour les conditions normales que pendant le défaut.

Divers paramètres et conditions, tels que la résistance, la localisation et le type de défaut, ont été modifiés afin de créer la base de données. Le tableau présente les différentes valeurs qui ont été variées.

<b>Résistance de défaut <math>R_f</math> (<math>\Omega</math>)</b>	0.25, 0.75, 1
<b>Localisation de défaut</b>	Transformateur 1 (T1), Transformateur 2 (T2)
<b>Types de défaut</b>	Non-fault, AG, BG, CG, AB,AC,BC, ABG,ACG,BCG,ABC

**Tableau 4.1 :** Paramètres.

L'ensemble de données collecté comprend 3000 instances, englobant une diversité de types de défauts. Cette variété permet d'enrichir l'apprentissage des modèles, en prenant en compte différentes conditions et scénarios représentatifs du fonctionnement réel du système électrique. Les données sont ensuite enregistrées sous format de fichier Excel nommé « Dataset » afin de simplifier leur utilisation.

Dans le contexte de la classification supervisée, il est essentiel que nos algorithmes disposent d'étiquettes, qui seront utilisées pour l'entraînement. Les étiquettes sont définies comme suit : ( $Y_d$ ) indique la présence ou l'absence du défaut, ( $Y_{t1}$ ) indique la présence ou l'absence du défaut dans le transformateur 1, et ( $Y_{t2}$ ) indique la présence ou l'absence du défaut dans le transformateur 2.

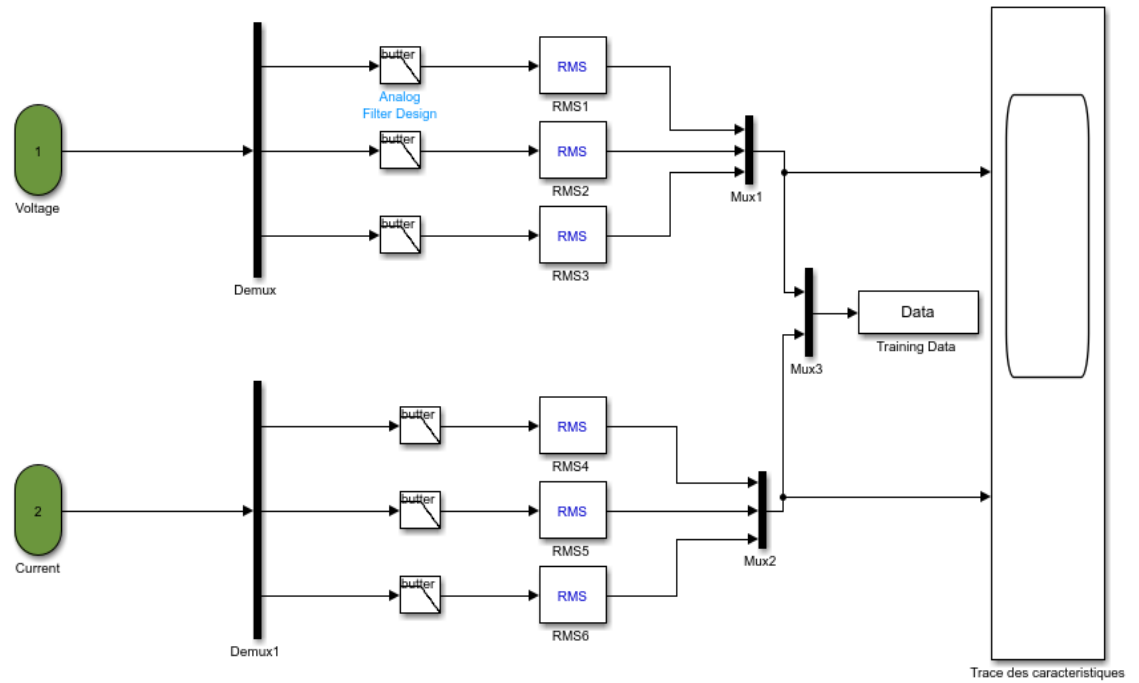
Cependant, au cours de la phase de collecte des données, les mesures de tension et de courant peuvent être affectées par diverses perturbations telles que le bruit et les harmoniques parasites. Ainsi, pour améliorer la qualité des données, le prétraitement des données et le filtrage du signal analogique s'avèrent essentiels afin d'éliminer les harmoniques indésirables et de réduire le bruit.

Le prétraitement des données constitue une étape cruciale visant à nettoyer et préparer les données avant leur utilisation. Cette technique permet de convertir les données en un jeu plus propre et plus facile à traiter. Son application est fondamentale avant d'entreprendre l'analyse et la création du modèle. Dans notre contexte, le prétraitement des données a été réalisé au moyen du bloc illustré ci-dessous.

Les différentes étapes du processus de prétraitement des données comprennent :

- Le filtrage qui vise à éliminer les bruits à haute fréquence. Cela a été réalisé en utilisant un filtre passe-bas Butterworth du 10e ordre avec une fréquence de coupure de 480 Hz.
- L'échantillonnage, effectué à une fréquence de 2000 hertz, correspondant à la fréquence à laquelle les formes d'onde de tension et de courant ont été capturées. Étant donné que la fréquence du système est de 60 Hz, le choix de la fréquence d'échantillonnage offre une meilleure résolution.
- L'utilisation de la méthode de la racine carrée moyenne (RMS) pour extraire les caractéristiques les plus significatives des données d'apprentissage.

Ces différentes étapes ont été mises en œuvre sous Simulink, comme illustré dans la figure ci-dessous.

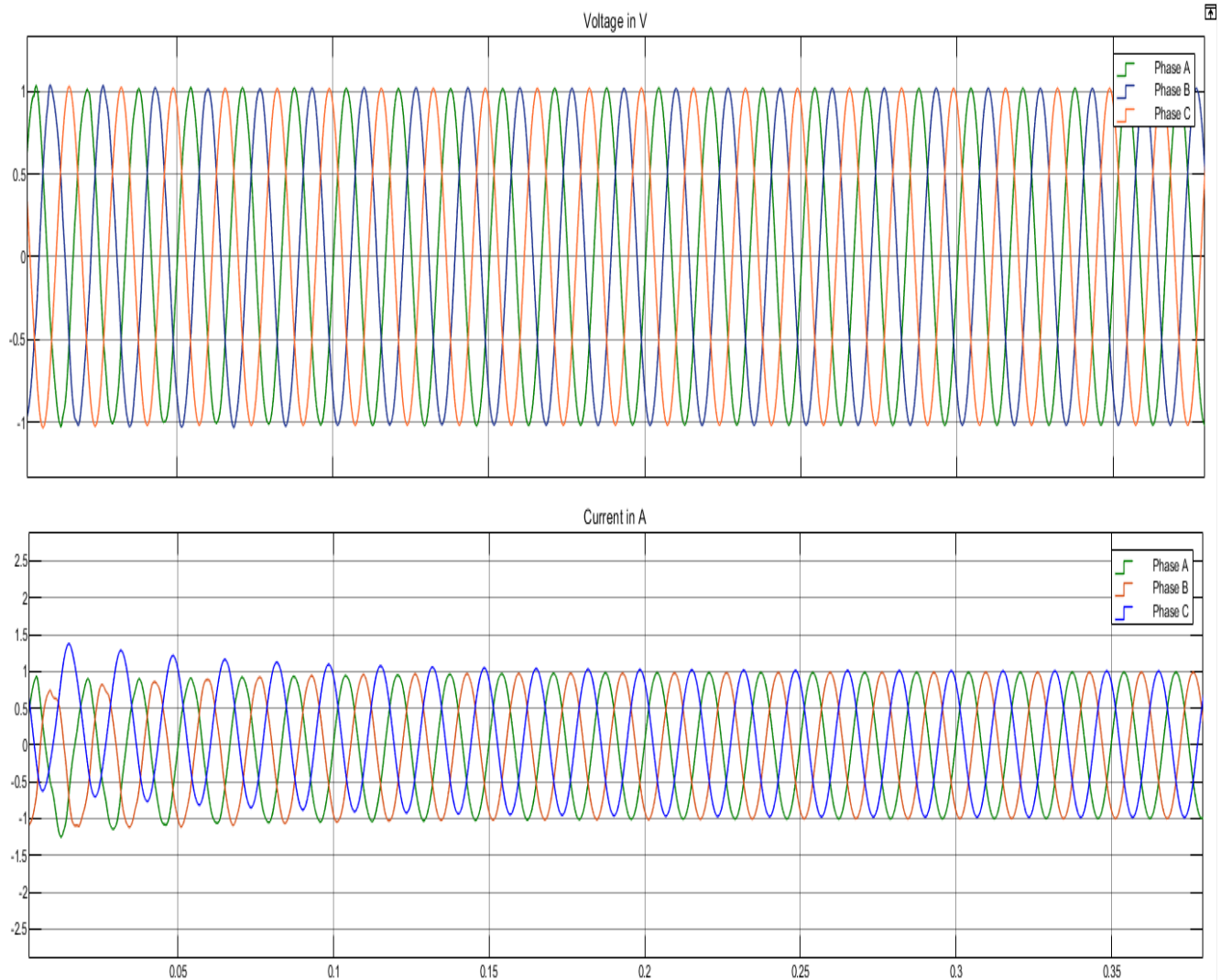


**Figure 4.3 :** Extraction de données et traitement de signal.

#### 4.4. Simulations du système électrique proposés

L'objectif principal de cette partie est de visualiser les résultats obtenus de la simulation du réseau électrique. Il est très important de valider la performance du système électrique afin de s'assurer de son bon fonctionnement.

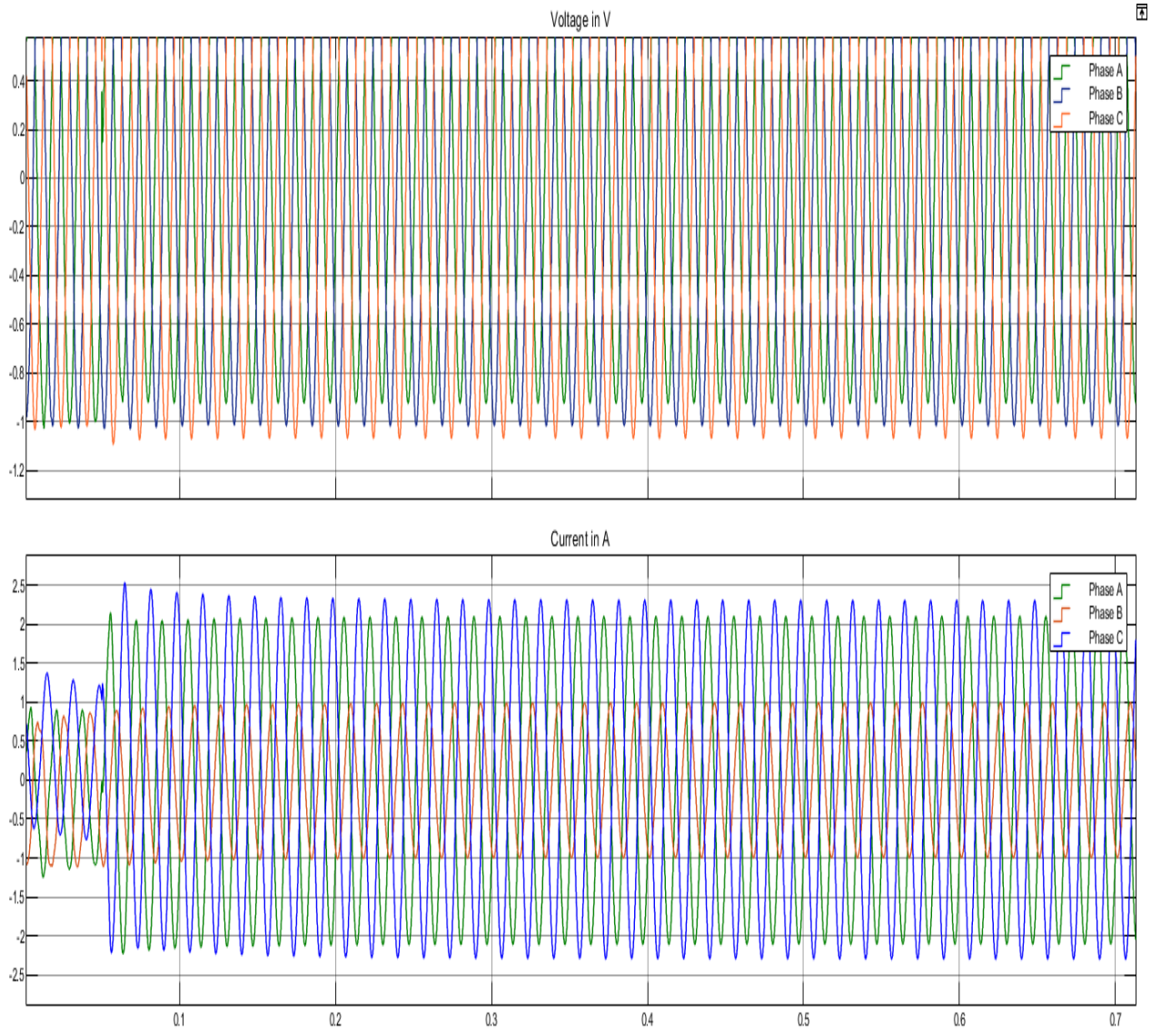
La figure suivante présente les formes d'ondes de tensions et courants obtenues dans le cas du non-défaut ;



**Figure 4.4 :** Formes d'onde de tension et de courant du système triphasé, sans défaut.



La figure suivante présente les formes d'ondes de tensions et courants obtenues dans le cas du défaut ;



**Figure 4.5 :** Formes d'onde de tension et de courant dans le cas d'un défaut.

## **4.5. Prédiction des défauts des transformateurs électriques**

### **4.5.1. Division du Dataset**

Cette étape consiste en diviser les données en un ensemble d'entraînement (Training data) et un ensemble de test (Test data). Le processus de division des données est une pratique fondamentale dans le domaine de l'apprentissage automatique. Cette étape cruciale vise à évaluer les performances d'un modèle sur des données qui n'ont pas été utilisées pendant la phase d'entraînement, offrant ainsi une mesure objective de sa capacité à généraliser à de nouvelles données. Cette procédure contribue à évaluer la robustesse et l'efficacité du modèle en dehors de l'environnement contrôlé de l'ensemble d'entraînement, où il a été initialement développé.

Dans cette étude, les données sont divisées en 75 % de données d'entraînement et 25 % de données de test.

**Ensemble d'entraînement (Training Set) :** Ici, 75 % de l'ensemble des données sera utilisée pour entraîner le modèle. Le modèle ajuste ses paramètres en se basant sur cet ensemble pour apprendre les relations entre les caractéristiques et les étiquettes (ou les résultats attendus).

**Ensemble de test (Test Set) :** Ici, 25 % de l'ensemble des données réservée à l'évaluation du modèle. Une fois que le modèle est entraîné, il est évalué sur cet ensemble pour estimer ses performances en termes de précision, de rappel, de F1-score, etc.

**Randomisation :** Les données sont souvent mélangées de manière aléatoire avant la division pour éviter tout biais dans l'ordre des données. Cela garantit que l'ensemble d'entraînement et l'ensemble de test sont représentatifs de la distribution globale des données.

#### 4.5.2. Entraînement des modèles du Machine learning

Comme précédemment mentionné, nous sommes confrontés à un problème de classification dans cette étude. Par conséquent, nous opterons pour une approche d'apprentissage supervisé, utilisant des algorithmes spécifiquement conçus pour la classification. Ce choix stratégique découle de la nature du problème, où notre objectif principal est de classer les données en différentes classes ; absence du défaut ou présence du défaut.

Dans le cadre de l'apprentissage supervisé, l'algorithme assimile des connaissances à partir d'un ensemble de données étiqueté, composé de mesures de courant ( $I_A, I_B, I_C$ ) et de tension ( $V_A, V_B, V_C$ ). Chaque observation de cet ensemble est associée à une classe prédéfinie, où la classe 0 indique l'absence de défaut et la classe 1 signale la présence d'un défaut. En fournissant ces mesures à l'algorithme, il apprend à établir des relations entre les caractéristiques électriques spécifiques, représentées par les mesures de courant et de tension, et les classes correspondantes, déterminant ainsi si un défaut est présent ou non. Cette phase d'entraînement permet à l'algorithme de généraliser ces relations, lui permettant de classer de manière autonome de nouvelles données non étiquetées en fonction de leurs caractéristiques électriques.

Dans le cadre de cette étude, nous avons choisi d'exploiter plusieurs algorithmes de Machine Learning pour la prédiction et la localisation des défauts des transformateurs électriques. Ces algorithmes comprennent :

- Régression Logistique (Logistic Regression)
- Machine à Vecteurs de Support (Support Vector Machine)
- K-Nearest Neighbors (K-NN)
- Naïve Bayes
- Arbre de Décision (Decision Tree)
- Forêt Aléatoire (Random Forest)
- AdaBoost

- Extreme Gradient Boosting (XGBoost)

Le processus d'entraînement de ces modèles implique l'introduction des données dans chaque algorithme, permettant ainsi au modèle d'ajuster ses paramètres internes afin de mieux prédire les résultats. Cependant, il est crucial de surveiller le surajustement (overfitting) et le sous-ajustement (underfitting) au cours de l'entraînement.

Le surajustement doit être évité, car il se produit lorsque le modèle s'adapte trop précisément aux données d'entraînement spécifiques, risquant ainsi de mal généraliser à de nouvelles données. De même, le sous-ajustement doit être évité, car il indique que le modèle n'a pas réussi à capturer les structures complexes des données d'entraînement.

L'évaluation de ces modèles sera basée sur diverses métriques, telles que la précision, la sensibilité, la spécificité, et d'autres, pour assurer une évaluation complète de leur performance. Ce processus permettra de sélectionner le modèle qui offre le meilleur compromis entre la capacité de s'adapter aux données d'entraînement et la généralisation aux nouvelles données.

### **4.5.3. Évaluation des modèles**

L'évaluation du modèle constitue une étape cruciale dans le développement du projet, permettant de déterminer la capacité du modèle sélectionné à prédire correctement la variable cible lorsqu'il est exposé à de nouvelles données. La valeur ajoutée de la division du lot de données est d'obtenir une partie qui représente 25 % des données destinées au test. Cela facilitera l'utilisation de ces données pour effectuer des prévisions précises et conduire des analyses statistiques approfondies.

La science des données et de l'apprentissage automatique s'appuie fortement sur les principes mathématiques. Lors de la phase d'évaluation, des statistiques prédéfinies sont utilisées à l'aide du module "metrics" de la bibliothèque Scikit-learn. Ce module met en œuvre plusieurs métriques, notamment la précision, le rappel, la f-mesure et le

support. Ces métriques seront calculées pour chaque catégorie afin d'évaluer le modèle. Avant d'explorer ces métriques, il est essentiel de définir quelques notions fondamentales.

La Matrice de Confusion (Confusion Matrix) est couramment employée dans les problèmes de classification. Elle met en évidence les prédictions correctes et incorrectes du modèle, offrant ainsi un aperçu détaillé des types d'erreurs commises. Pour renforcer la fiabilité de l'évaluation, la technique de validation croisée sera également intégrée. Afin de prévenir le surajustement, nous adopterons cette approche pour confirmer et évaluer nos modèles. La validation croisée (cross-validation) est une technique robuste qui contribue à évaluer la performance d'un modèle en le testant sur plusieurs sous-ensembles de données, permettant ainsi d'obtenir une évaluation plus fiable de la capacité de généralisation du modèle. Cette méthode est particulièrement utile pour s'assurer que le modèle n'est pas simplement adapté aux données d'entraînement spécifiques, mais qu'il est également capable de faire des prédictions précises sur de nouvelles données. En utilisant la validation croisée, nous nous assurons d'une évaluation robuste et d'une meilleure compréhension de la performance du modèle sur des données inconnues.

Avant de passer à la définition des métriques nous allons commencer par définir quelques notions :

**Matrice de confusion (Confusion Matrix) :** Dans les problèmes de classification, les résultats prédits doivent être comparés à la réalité pour mesurer leur degré de performance. On utilise la matrice de confusion, celle-là met en valeur les prédictions correctes et incorrectes et donne aussi un indice sur le type des erreurs commises.

Actual Class	Predicted Class		
		Class = Yes	Class = No
	Class = Yes	True positive	False Negative
	Class = No	False Positive	True Negative

**Tableau 4.2 :** Matrice de confusion.

- **True Positive (Vrai positif) :** Ce sont les prédictions qui forment un match avec la valeur réelle. C'est le cas équivaut à la situation suivante : la valeur réelle est : « il y'a un défaut » et la prédiction est : « il y'a un défaut ».
- **False Negative (Faux négatif) :** la prédiction est négative alors que la valeur réelle est négative. C'est le cas équivaut à la situation suivante : la valeur réelle est : « il y a un défaut » et la prédiction est : « il n'y a pas de défaut ».
- **False Positive (Faux positif) :** la prédiction est positive alors que la valeur réelle est négative. C'est le cas équivaut à la situation suivante : la valeur réelle est : « il n'y a aucun défaut » et la prédiction est : « il y'a un défaut ».
- **True Negative (Vrai négatif) :** La prédiction et la valeur réelle sont négatives. C'est le cas équivaut à la situation suivante : la valeur réelle est : « il n'y a aucun défaut » et la prédiction est : « il n'y a aucun défaut ».

Cela dit, nous allons procéder à la définition de nos métriques, en se basant sur les notions citées précédemment.

**La précision :** parmi toutes les classes que nous avons prédites comme positives, combien sont réellement positives. Ceci se traduit par l'opération suivante :

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

**Le rappel (sensitivity) :** Parmi tous les cas positifs, quel pourcentage est prédit positif ? La sensibilité (parfois appelée rappel) mesure la capacité du modèle à prédire les positifs.

Cela signifie qu'elle examine les vrais positifs et les faux négatifs (qui sont des positifs qui ont été prédits à tort comme négatifs). Ceci se traduit par l'opération

$$\text{suivante : } \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}.$$

**F-mesure** : C'est une mesure qui combine la précision et le rappel, et est leur moyenne harmonique. Ceci se traduit par l'opération suivante :

$$\frac{2 * \text{precision} * \text{rappel}}{\text{precision} + \text{rappel}}.$$

**Accuracy** : Quoiqu'il n'existe pas de mot français équivalent avec précision à la mesure « Accuracy », elle reste très différente de la précision avec laquelle cette mesure est souvent confondue. Parmi toutes les classes (positives et négatives), combien d'entre elles avons-nous prédit correctement. Elle est calculée pour la totalité du modèle, et est très intuitive.

Elle se traduit par l'opération suivante :

$$\frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Negative} + \text{True Negative} + \text{False Positive}}.$$

#### 4.5.4. Résultats et discussions

En anticipant les défaillances avant qu'elles ne surviennent, il devient possible de planifier la maintenance de manière proactive, évitant ainsi les périodes d'arrêt imprévues et réduisant les coûts liés à la maintenance. La prédiction des défaillances contribue également à renforcer la sécurité en identifiant les risques avant qu'ils ne se traduisent par des accidents ou des dommages à l'équipement [23-24].

L'apprentissage automatique englobe le développement, l'analyse et la mise en œuvre de méthodes permettant à une machine de se perfectionner par le biais d'un processus d'apprentissage. Dans le cadre de ce rapport, notre focus s'est porté sur l'apprentissage automatique supervisé appliqué à un problème de classification, spécifiquement dans la prédiction des défauts de transformateurs.

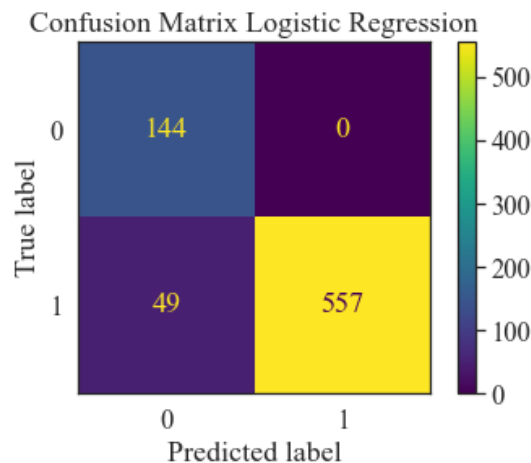
Dans cette étude axée sur la prédiction des défauts de transformateurs, nous avons minutieusement choisi un ensemble d'algorithmes d'apprentissage automatique supervisé pour affiner la performance du modèle. Ces algorithmes ont été sélectionnés en tenant compte de leurs capacités à traiter des données complexes, à fournir des prédictions précises et à gérer des dimensions importantes. Chaque algorithme a été entraîné sur 75% des données collectées et testé sur 25% des données et reçoit six entrées ( $V_A, V_B, V_C, I_A, I_B, I_C$ ) et donne une sortie. La sortie du modèle prend une valeur soit 0 indiquant l'absence du défaut ou 1 sa présence.

Pour aller jusqu'au bout du process, il est crucial de détailler l'évaluation de chaque algorithme. Dans la suite, nous exposons les résultats obtenus ainsi que les matrices de confusion pour chaque algorithme :

- **Régression Logistique (Logistic Regression) :**

**Raison du choix :** Adapté aux problèmes de classification binaire, la régression logistique fournit des prédictions binaires (défaut/non-défaut) de manière efficace.

**Matrice de confusion :**



**Figure 4.6 :** Matrice de confusion de l'algorithme Logistic Regression.



La matrice de confusion indique les résultats d'une classification effectuée par le modèle de régression logistique. Voici les détails de l'interprétation :

- Vrais Négatifs (VN) : Le modèle a correctement prédit 144 instances comme n'ayant pas de défaut (classe 0).
- Faux Positifs (FP) : Il n'y a aucun faux positif; le modèle n'a pas incorrectement prédit de défauts (classe 1) pour les instances sans défaut (classe 0).
- Faux Négatifs (FN) : Le modèle a incorrectement prédit 49 instances qui avaient un défaut (classe 1) comme étant sans défaut (classe 0).
- Vrais Positifs (VP) : Le modèle a correctement identifié 557 instances comme ayant un défaut (classe 1).

Cette matrice de confusion montre que le modèle de régression logistique est très performant pour identifier les instances sans défaut, comme en témoigne l'absence de faux positifs. Cependant, le modèle a manqué un nombre significatif de défauts réels, ce qui est indiqué par le nombre de faux négatifs plus élevé par rapport à d'autres modèles examinés précédemment. Cela peut suggérer que la régression logistique, dans ce cas, a une sensibilité plus faible pour détecter les cas positifs.

#### Évaluation et interprétation des résultats :

Training Accuracy : 94.22 %				
Model Accuracy Score : 93.47 %				
-----				
Classification_Report Logistic Regression:				
	precision	recall	f1-score	support
0	0.75	1.00	0.85	144
1	1.00	0.92	0.96	606
accuracy			0.93	750
macro avg	0.87	0.96	0.91	750
weighted avg	0.95	0.93	0.94	750

**Figure 4.7 :** Rapport de classification de l'algorithme Logistic Regression.

Le rapport de classification affiche les performances d'un modèle de régression logistique. Voici l'interprétation des résultats :

**Précision d'entraînement (Training Accuracy):** Le modèle a une précision de 94,22 % avec les données d'entraînement, ce qui indique une bonne adaptation du modèle aux données sur lesquelles il a été formé.

**Précision globale (Model Accuracy score) :** La précision du modèle sur les données de test est de 93,47 %, ce qui montre que le modèle est assez précis lorsqu'il est testé avec de nouvelles données.

**Rapport de classification pour la régression logistique :**

- Pour la classe 0 (absence de défaut) :

Précision : 0,75, ce qui signifie que 75 % des prédictions de la classe 0 par le modèle étaient correctes.

Rappel : 1,00, indiquant que le modèle a identifié correctement toutes les instances réelles de la classe 0.

Score F1 : 0,85, reflétant un équilibre entre la précision et le rappel pour la classe 0, mais affecté par la précision relativement basse.

- Pour la classe 1 (présence de défaut) :

Précision : 1,00, signifiant que toutes les prédictions de la classe 1 étaient correctes.

Rappel : 0,92, ce qui signifie que le modèle a identifié 92 % des instances réelles de la classe 1.

Score F1 : 0,96, indiquant un excellent équilibre entre la précision et le rappel.

Moyenne macro: Les moyennes des scores pour la précision, le rappel et le score F1 sont de 0,87 à 0,96, indiquant des performances générales moins uniformes entre les classes.

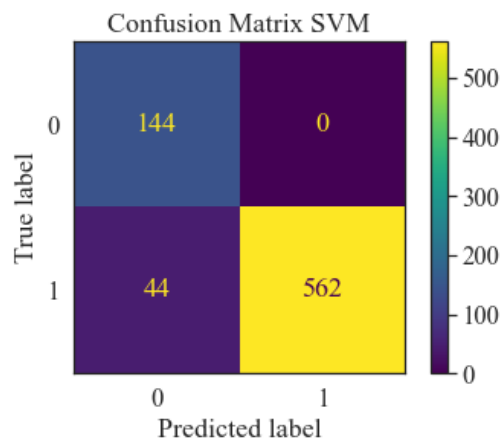
Moyenne pondérée : Les moyennes pondérées par le support de chaque classe sont de 0,95 pour la précision et de 0,93 pour le rappel et le score F1, ce qui prend en compte la fréquence de chaque classe dans l'ensemble de test.

Ces résultats indiquent que le modèle de régression logistique est très précis pour prédire la classe 1, mais moins précis pour la classe 0, comme en témoigne la précision plus faible pour cette classe. Cela peut suggérer que des améliorations sont nécessaires pour augmenter la capacité du modèle à distinguer correctement les instances non défectueuses.

- **Machine à Vecteurs de Support (Support Vector Machine) :**

**Raison du choix :** Capacité à gérer des dimensions importantes et à traiter des données non linéaires pour une prédiction précise des défauts.

**Matrice de confusion :**



**Figure 4.8 :** Matrice de confusion de l'algorithme Support Vector Machine.

La matrice de confusion fournie représente les performances d'un modèle de classification basé sur les Machines à Vecteurs de Support (SVM). Voici l'interprétation des résultats :

- Vrais Négatifs (VN) : Le modèle a correctement identifié 144 instances sans défaut (classe 0).
- Faux Positifs (FP) : Il n'y a pas eu de faux positifs; le modèle n'a incorrectement prédit aucune instance sans défaut comme un défaut (classe 1).
- Faux Négatifs (FN) : Le modèle a incorrectement prédit 44 instances qui avaient un défaut (classe 1) comme étant sans défaut (classe 0).
- Vrais Positifs (VP) : Il a correctement identifié 562 instances comme ayant un défaut (classe 1).

Cette matrice indique que le modèle SVM est très fiable pour prédire l'absence de défaut, comme en témoigne l'absence de faux positifs. Cependant, il présente un nombre relativement élevé de faux négatifs, ce qui signifie qu'il a manqué un certain nombre d'instances défectueuses, les classant à tort comme non défectueuses. Cela pourrait indiquer que le modèle SVM pourrait être amélioré en termes de sensibilité pour détecter les cas positifs.

#### Évaluation et interprétation des résultats :

```

Training Accuracy      : 94.88 %
Model Accuracy Score  : 94.13 %
-----
Classification_Report SVM :
      precision    recall  f1-score   support

     0       0.77      1.00      0.87      144
     1       1.00      0.93      0.96      606

   accuracy          0.94      750
  macro avg          0.88      750
 weighted avg          0.96      750

```

**Figure 4.9** : Rapport de classification de l'algorithme SVM.

Le rapport de classification affiche les performances du modèle de classification basé sur les Machines à Vecteurs de Support (SVM). Voici l'interprétation des résultats :

**Précision d'entraînement (Training Accuracy):** Le modèle a une précision de 94,88 % sur les données d'entraînement, ce qui suggère une bonne performance durant l'apprentissage.

**Précision globale (Model Accuracy score) :** La précision du modèle est de 94,13 % sur les données de test, indiquant que le modèle généralise correctement à de nouvelles données, mais avec une marge d'erreur plus importante que celle observée dans les autres modèles évalués précédemment.

**Rapport de classification pour SVM :**

- Pour la classe 0 (absence de défaut) :

Précision : 0,77, indiquant que seulement 77 % des prédictions de la classe 0 par le modèle étaient correctes.

Rappel : 1,00, signifiant que le modèle a identifié correctement toutes les instances réelles de la classe 0.

Score F1 : 0,87, reflétant un équilibre entre la précision et le rappel pour la classe 0, mais diminué par la faible précision.

- Pour la classe 1 (présence de défaut) :

Précision : 1,00, signifiant que toutes les prédictions étaient correctes (aucun faux positif).

Rappel : 0,93, indiquant que le modèle a identifié 93 % des instances réelles de la classe 1.

Score F1 : 0,96, indiquant un bon équilibre entre la précision et le rappel pour la classe 1.

Moyenne macro : Les moyennes des scores pour la précision, le rappel et le score F1 sont respectivement de 0,88 et 0,96, ce qui indique des performances générales moins uniformes entre les classes.

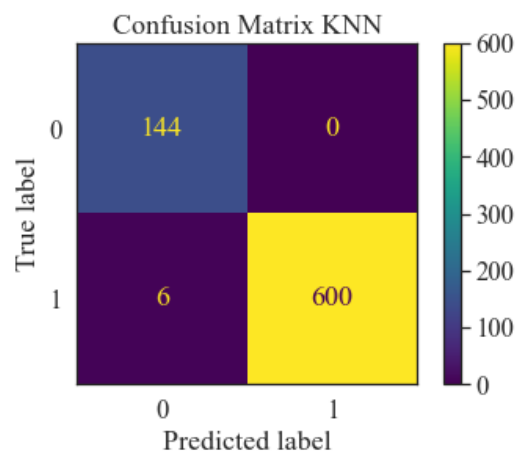
Moyenne pondérée : Les moyennes pondérées par le support pour chaque classe sont de 0,96 pour la précision et le rappel, et de 0,94 pour le score F1, ce qui prend en compte la fréquence de chaque classe dans l'ensemble de test.

Ces résultats suggèrent que le modèle SVM est très fiable pour prédire la présence de défauts (classe 1) mais moins précis pour prédire l'absence de défauts (classe 0), comme en témoigne la précision relativement basse pour cette classe. Le modèle pourrait nécessiter une amélioration pour augmenter sa capacité à distinguer les instances non défectueuses.

- **K-Nearest Neighbors (K-NN) :**

**Raison du choix :** Méthode simple basée sur la proximité des voisins, est fréquemment utilisé pour des tâches de classification avec des ensembles de données variés.

**Matrice de confusion :**



**Figure 4.10 :** Matrice de confusion de l'algorithme K-Nearest Neighbors.

La matrice de confusion représente les performances d'un modèle de classification K-Nearest Neighbors (KNN). Voici l'interprétation des résultats affichés :

- Vrais Négatifs (VN): Le modèle a correctement prédit 144 instances comme n'ayant pas de défaut (classe 0).
- Faux Positifs (FP) : Il n'y a pas eu de faux positifs; le modèle n'a incorrectement prédit aucun cas de non-défaut comme un défaut (classe 1).
- Faux Négatifs (FN) : Le modèle a incorrectement prédit 6 instances comme n'ayant pas de défaut (classe 0), alors qu'elles avaient en réalité un défaut (classe 1).
- Vrais Positifs (VP) : Il a correctement identifié 600 instances comme ayant un défaut (classe 1).

Cette matrice de confusion indique que le modèle KNN a une très bonne performance, avec une excellente capacité à identifier correctement les instances sans défaut et une très bonne capacité à détecter les défauts, malgré quelques faux négatifs.

#### Évaluation et interprétation des résultats :

```

Training Accuracy   : 99.33 %
Model Accuracy Score : 99.2 %
-----
Classification_Report KNN:
      precision    recall  f1-score   support

     0       0.96      1.00      0.98        144
     1       1.00      0.99      1.00        606

 accuracy          0.99          0.99          0.99          750
 macro avg         0.98          1.00          0.99          750
 weighted avg      0.99          0.99          0.99          750

```

**Figure 4.11** : Rapport de classification de l'algorithme K-Nearest Neighbors.

Le rapport de classification affiché concerne un modèle de classification K-Nearest Neighbors (KNN). Voici l'analyse des résultats :

**Précision d'entraînement (Training Accuracy):** Le modèle a une précision de 99,33 % sur les données d'entraînement, indiquant qu'il a très bien appris à partir de l'ensemble d'entraînement.

**Précision globale (Model Accuracy score) :** La précision du modèle sur les données de test est de 99,2 %, ce qui montre une excellente généralisation du modèle à de nouvelles données.

**Rapport de classification pour KNN :**

- Pour la classe 0 (absence de défaut) :

Précision : 0,96, signifie que 96 % des instances que le modèle a prédit comme étant de la classe 0 étaient correctes.

Rappel : 1,00, indiquant que le modèle a identifié correctement toutes les instances réelles de la classe 0.

Score F1 : 0,98, reflétant un équilibre entre la précision et le rappel pour la classe 0.

- Pour la classe 1 (présence de défaut) :

Précision : 1,00, ce qui indique que toutes les prédictions étaient correctes.

Rappel : 0,99, montrant que le modèle a identifié 99 % des instances réelles de la classe 1.

Score F1: 1,00, indiquant un excellent équilibre entre la précision et le rappel.

Moyenne macro : La moyenne des scores pour la précision, le rappel et le score F1 est de 0,98 à 1,00, indiquant une performance globalement élevée et équilibrée du modèle pour les deux classes.

Moyenne pondérée : Les moyennes pondérées par le support pour chaque classe sont de 0,99 pour la précision, le rappel et le score F1, ce qui prend en compte la fréquence de chaque classe dans l'ensemble de test.

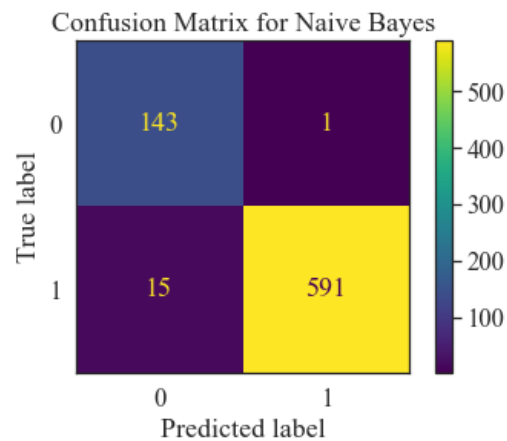


Ces résultats illustrent que le modèle KNN est extrêmement performant, avec une précision et un rappel élevés, signifiant qu'il est très fiable dans ses prédictions et qu'il est très efficace pour identifier correctement les instances positives et négatives.

- **Naïve Bayes :**

**Raison du choix :** Fondé sur le théorème de Bayes, Naïve Bayes offre une approche simple mais efficace, particulièrement adaptée à des ensembles de données volumineux et riches en caractéristiques.

**Matrice de confusion :**



**Figure 4.12 :** Matrice de confusion de l'algorithme Naïve Bayes.

La matrice de confusion fournie représente les résultats d'un modèle de classification utilisant l'algorithme Naïve Bayes. Voici l'interprétation des résultats :

- Vrais Négatifs (VN) : Le modèle a correctement prédit que 143 instances étaient sans défaut (classe 0).
- Faux Positifs (FP): Il y a eu une instance où le modèle a prédit un défaut (classe 1) alors que l'instance était en réalité sans défaut (classe 0).
- Faux Négatifs (FN) : Le modèle n'a pas identifié 15 instances comme étant défectueuses (classe 1) alors qu'elles l'étaient.

- Vrais Positifs (VP) : Le modèle a correctement identifié 591 instances comme étant défectueuses (classe 1).

Cette matrice indique que l'algorithme Naïve Bayes est assez précis pour détecter les instances sans défaut, mais il a manqué un nombre relativement plus élevé de défauts réels, comme en témoignent les faux négatifs. Cela peut indiquer une sensibilité inférieure de l'algorithme Naïve Bayes par rapport aux autres modèles pour la détection des cas de classe 1 dans cet ensemble de données spécifique.

#### Évaluation et interprétation des résultats :

Training Accuracy : 98.4 %				
Model Accuracy Score : 97.87 %				
-----				
Classification Report Naïve Bayes :				
	precision	recall	f1-score	support
0	0.91	0.99	0.95	144
1	1.00	0.98	0.99	606
accuracy			0.98	750
macro avg	0.95	0.98	0.97	750
weighted avg	0.98	0.98	0.98	750

**Figure 4.13 :** Rapport de classification de l’algorithme Naïve Bayes.

Le rapport de classification affiché concerne un modèle utilisant l'algorithme Naïve Bayes. Voici l'analyse des résultats présentés :

**Précision d’entraînement (Training Accuracy):** Le modèle a une précision de 98,4 % avec les données d'entraînement, indiquant une bonne performance durant l'entraînement.

**Précision globale (Model Accuracy score) :** La précision du modèle sur les données de test est de 97,87 %, ce qui montre une bonne généralisation du modèle aux nouvelles données.

Rapport de classification pour Naïve Bayes :

- Pour la classe 0 (absence de défaut) :

Précision : 0,91, signifiant que 91 % des prédictions du modèle sont correctes.

Rappel : 0,99, indiquant que le modèle a correctement identifié 99 % des cas réels.

Score F1 : 0,95, une moyenne harmonique de la précision et du rappel, montrant un bon équilibre entre ces deux mesures pour la classe 0.

- Pour la classe 1 (présence de défaut) :

Précision : 1,00, indiquant que le modèle a prédit avec précision 100 % des instances comme étant de la classe 1.

Rappel : 0,98, ce qui signifie que le modèle a identifié 98 % de tous les cas réels de défauts.

Score F1 : 0,99, indiquant un excellent équilibre entre la précision et le rappel.

Moyenne macro : Les moyennes des scores pour la précision, le rappel et le score F1 sont de 0,95 et 0,98 respectivement, ce qui indique une performance globalement élevée du modèle sur les deux classes sans pondération.

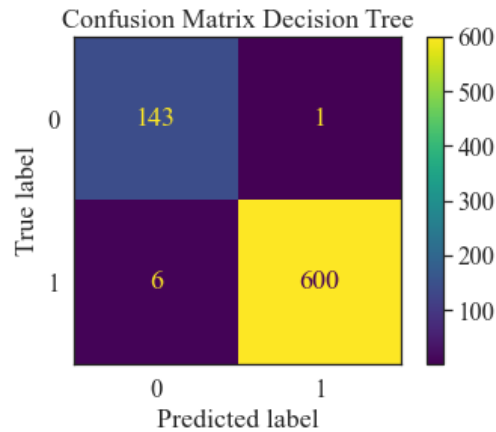
Moyenne pondérée : Les moyennes pondérées sont de 0,98 pour la précision, le rappel et le score F1, ce qui tient compte du nombre d'instances dans chaque classe.

Ces résultats indiquent que le modèle Naïve Bayes est performant, en particulier dans la précision de la prédiction de la classe 1. Le score F1 élevé pour les deux classes indique que le modèle maintient un bon équilibre entre la précision et le rappel.

- **Arbre de Décision (Decision Tree) :**

**Raison du choix :** Structure arborescente offrant une interprétation facile des résultats de la prédiction des défauts.

**Matrice de confusion :**



**Figure 4.14 :** Matrice de confusion de l’algorithme Decision Tree.

La matrice de confusion fournie montre les résultats d'un modèle de classification basé sur un arbre de décision. Voici l'interprétation des résultats :

- Vrais Négatifs (VN) : Le modèle a correctement prédit 143 instances sans défaut (classe 0).
- Faux Positifs (FP) : Le modèle a incorrectement prédit un défaut (classe 1) pour une instance qui était en réalité sans défaut (classe 0).
- Faux Négatifs (FN) : Il y a 6 instances où le modèle a prédit qu'il n'y avait pas de défaut (classe 0) alors que ces instances avaient un défaut (classe 1).
- Vrais Positifs (VP) : Le modèle a correctement identifié 600 instances avec défaut (classe 1).

Cette matrice indique que le modèle d'arbre de décision performe bien pour identifier les instances sans défaut, mais il a manqué quelques instances défectueuses (indiqué par les faux négatifs). Toutefois, la quantité de faux positifs est très basse, ce qui signifie que lorsque le modèle prédit un défaut, il est très probablement correct.

**Évaluation et interprétation des résultats :**

Training Accuracy	: 99.73 %			
Model Accuracy Score	: 99.07 %			
-----				
Classification_Report Decision Tree:				
	precision	recall	f1-score	support
0	0.96	0.99	0.98	144
1	1.00	0.99	0.99	606
accuracy			0.99	750
macro avg	0.98	0.99	0.99	750
weighted avg	0.99	0.99	0.99	750

**Figure 4.15 :** Rapport de classification de l'algorithme Decision Tree.

Le rapport de classification présenté correspond à un modèle de classification basé sur un arbre de décision. Voici les détails de l'interprétation :

**Précision d'entraînement (Training Accuracy):** Le modèle a une précision de 99,73 % sur les données d'entraînement, ce qui démontre une excellente performance pendant la phase d'apprentissage.

**Précision globale (Model Accuracy score) :** Sur les données de test, le modèle a une précision de 99,07 %, indiquant une très bonne capacité de généralisation à de nouvelles données.

#### **Rapport de classification pour l'arbre de décision :**

- Pour la classe 0 (absence de défaut) :

Précision : 0,96, signifiant que 96 % des prédictions du modèle sont correctes.

Rappel : 0,99, ce qui indique que le modèle a correctement identifié 99 % de tous les cas réels de la classe 0.

Score F1 : 0,98, qui est une mesure qui combine la précision et le rappel, indiquant un équilibre entre ces deux mesures pour la classe 0.

- Pour la classe 1 (présence de défaut) :

Précision : 1,00, indiquant que toutes les prédictions par le modèle sont correctes.

Rappel : 0,99, montrant que le modèle a identifié 99 % de tous les cas réels de la classe 1.

Score F1 : 0,99, suggérant un équilibre parfait entre la précision et le rappel.

Moyenne macro : Les moyennes des scores pour la précision, le rappel et le score F1 sont de 0,99 sans pondération par classe.

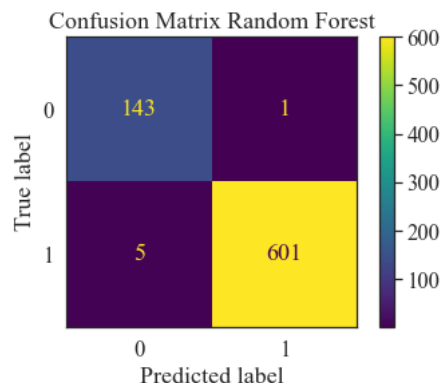
Moyenne pondérée : Les moyennes pondérées par le support de chaque classe sont également de 0,99 pour la précision, le rappel et le score F1.

Ces résultats indiquent que le modèle d'arbre de décision est très performant, avec une précision et un rappel élevés, signifiant qu'il est fiable dans ses prédictions et efficace pour identifier correctement la majorité des instances positives et négatives.

#### - Forêt Aléatoire (Random Forest) :

**Raison du choix :** Agrégation d'arbres de décision pour une prédiction robuste des défauts.

#### **Matrice de confusion :**



**Figure 4.16 :** Matrice de confusion de l'algorithme Random Forest.

La matrice de confusion fournie montre les performances du modèle de Forêt Aléatoire (Random Forest) pour une tâche de classification. Voici l'interprétation des résultats :

- Vrais Négatifs: Le modèle a correctement prédit 143 instances comme étant sans défaut (classe 0).
- Faux Positifs: Il y a eu 1 prédiction fausse où le modèle a indiqué la présence d'un défaut (classe 1) alors que l'instance était réellement sans défaut (classe 0).
- Faux Négatifs: Le modèle a échoué à identifier 5 instances comme étant défectueuses (classe 1), les classant incorrectement comme non défectueuses (classe 0).
- Vrais Positifs: Il a correctement identifié 601 instances comme étant défectueuses (classe 1).

Cette matrice suggère que le modèle de Forêt Aléatoire a une performance élevée, avec une très bonne capacité à identifier correctement les instances sans défaut, ainsi qu'une forte aptitude à reconnaître les instances défectueuses. Toutefois, il y a eu quelques cas où le modèle n'a pas détecté la présence d'un défaut, ce qui est indiqué par les faux négatifs.

#### Évaluation et interprétation des résultats :

```

Training Accuracy      : 99.73 %
Model Accuracy Score  : 99.33 %
-----
Classification_Report Random Forest :
      precision    recall  f1-score   support

     0       0.97      0.99      0.98        144
     1       1.00      0.99      1.00        606

   accuracy          0.99
  macro avg          0.99
 weighted avg          0.99
  
```

**Figure 4.17 :** Rapport de classification de l'algorithme Random Forest.

Le rapport de classification fourni affiche les mesures de performance pour le modèle de Forêt Aléatoire (Random Forest). Voici l'interprétation des résultats :

**Précision d'entraînement (Training Accuracy):** Le modèle a une précision de 99,73 % sur les données d'entraînement, indiquant une très bonne performance pendant la phase d'apprentissage.

**Précision globale (Model Accuracy score) :** La précision du modèle sur les données de test est de 99,33 %, montrant que le modèle est très performant lorsqu'il est appliqué à de nouvelles données.

**Rapport de classification pour Random Forest :**

- Pour la classe 0 (absence de défaut) :

Précision : 0,97, ce qui signifie que 97 % des instances prédites sont correctes.

Rappel : 0,99, indiquant que 99 % de toutes les instances réelles de la classe 0 ont été correctement identifiées par le modèle.

Score F1 : 0,98, une moyenne harmonique entre la précision et le rappel, suggérant un équilibre entre ces deux mesures pour la classe 0.

- Pour la classe 1 (présence de défaut) :

Précision : 1,00, indiquant que toutes les instances prédites comme étant de la classe 1 sont correctes.

Rappel : 0,99, montrant que le modèle a identifié 99 % de tous les cas réels de la classe 1.

Score F1: 1,00, indiquant un équilibre parfait entre la précision et le rappel pour la classe 1.

Moyenne macro : La moyenne des scores sans tenir compte du support par classe est de 0,99 pour la précision, le rappel et le score F1.



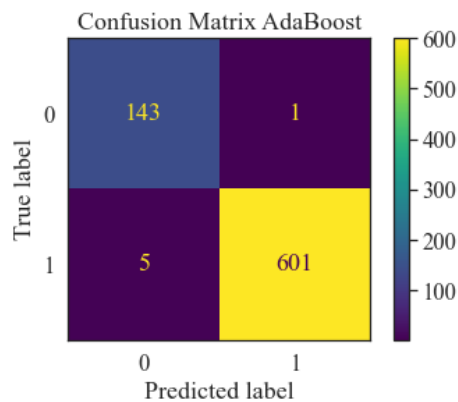
Moyenne pondérée : Les moyennes pondérées par le support de chaque classe sont également de 0,99 pour la précision, le rappel et le score F1.

Ces résultats indiquent que le modèle de Forêt Aléatoire a une excellente capacité de prédiction, avec une performance quasiment parfaite pour identifier la présence ou l'absence de défauts. La haute précision et le rappel élevé démontrent que le modèle est fiable et qu'il a une très bonne capacité à généraliser ses prédictions à de nouvelles données.

#### - AdaBoost :

**Raison du choix :** AdaBoost renforce la performance en accordant davantage de poids aux erreurs précédemment commises par le modèle, permettant ainsi d'apprendre des erreurs passées. Cet algorithme est souvent utilisé dans les problèmes de prédiction des défauts.

#### Matrice de confusion :



**Figure 4.18 :** Matrice de confusion de l'algorithme AdaBoost.

La matrice de confusion présentée montre les performances du modèle AdaBoost pour la classification. Voici l'interprétation des résultats :

- Vrais Négatifs (VN) : Le modèle a correctement prédit l'absence de défaut (classe 0) pour 143 instances.

- Faux Positifs (FP) : Le modèle a incorrectement prédit la présence de défaut (classe 1) pour 1 instance qui était en réalité sans défaut (classe 0).
- Faux Négatifs (FN) : Le modèle a incorrectement prédit l'absence de défaut (classe 0) pour 5 instances qui étaient en réalité défectueuses (classe 1).
- Vrais Positifs (VP) : Le modèle a correctement prédit la présence de défaut (classe 1) pour 601 instances.

L'interprétation de cette matrice de confusion est que le modèle AdaBoost a une très bonne performance globale, mais avec un peu plus de faux négatifs par rapport au modèle XGBoost (selon la matrice précédente). Cela signifie que le modèle AdaBoost est légèrement moins efficace pour identifier tous les cas réels de défauts par rapport au modèle XGBoost. Toutefois, le nombre de faux positifs reste très faible, ce qui indique que lorsque le modèle AdaBoost prédit la présence d'un défaut, il est très performant.

### Évaluation et interprétation des résultats :

```

Training Accuracy   : 99.73 %
Model Accuracy Score : 99.2 %
-----
Classification_Report AdaBoost :
      precision    recall  f1-score   support

     0       0.97       0.99       0.98        144
     1       1.00       0.99       1.00        606

 accuracy          0.99          0.99          0.99          750
  macro avg          0.98          0.99          0.99          750
 weighted avg          0.99          0.99          0.99          750

```

**Figure 4.19 :** Rapport de classification de l'algorithme AdaBoost.

Le rapport de classification fourni est pour le modèle AdaBoost. Voici les détails de l'interprétation :

**Précision d'entraînement (Training Accuracy):** Le modèle présente une précision de 99,73 % sur les données d'entraînement, ce qui montre qu'il a très bien appris à partir de ces données.

**Précision globale (Model Accuracy score) :** Le modèle a une précision de 99,2 % sur les données de test, indiquant une excellente capacité à généraliser les prédictions à de nouvelles données.

**- Rapport de classification pour AdaBoost :**

- Pour la classe 0 (absence de défaut) :

Précision : 0,97, indiquant que 97 % des prédictions du modèle sont correctes.

Rappel : 0,99, signifiant que le modèle a identifié 99 % de tous les cas réels de la classe 0.

Score F1 : 0,98, qui est la moyenne harmonique de la précision et du rappel, indiquant un équilibre entre ces deux mesures pour la classe 0.

- Pour la classe 1 (présence de défaut) :

Précision : 1,00, ce qui signifie que 100 % des prédictions par le modèle sont correctes.

Rappel : 0,99, indiquant que le modèle a identifié 99 % de tous les cas réels de la classe 1.

Score F1 : 1,00, montrant un équilibre parfait entre la précision et le rappel pour la classe 1.

Moyenne macro: La moyenne des scores sans pondération par classe est de 0,99 pour la précision, le rappel et le score F1.

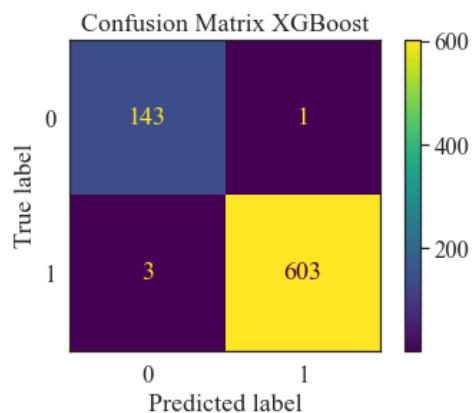
Moyenne pondérée : Les moyennes pondérées par le support de chaque classe sont également de 0,99 pour la précision, le rappel et le score F1.

Ces résultats montrent que le modèle AdaBoost a une performance très élevée, avec une excellente capacité de prédiction tant pour les cas de présence que d'absence de défaut. La précision et le rappel élevés indiquent que le modèle est à la fois précis dans ses prédictions positives et efficace pour identifier la majorité des cas positifs réels.

- **Extreme Gradient Boosting (XGBoost) :**

**Raison du choix :** XGBoost, une méthode de boosting, se distingue par sa haute performance et est couramment utilisé pour des tâches de classification.

**Matrice de confusion :**



**Figure 4.20 :** Matrice de confusion de l'algorithme XGBoost.

La figure montre une matrice de confusion pour un modèle de classification utilisant XGBoost.

- Vrai Positif : Le modèle a correctement prédit 603 instances où le défaut était présent (la classe 1).
- Vrai Négatif : Le modèle a correctement prédit 143 instances où le défaut était absent (la classe 0).
- Faux Positif : Le modèle a incorrectement prédit 1 instance comme étant un défaut (la classe 1) alors qu'en réalité, il n'y en avait pas (la classe 0).

- Faux Négatif : Le modèle a manqué 3 instances de défauts, les classant à tort comme absence de défaut (la classe 0).

La matrice indique que le modèle XGBoost est très précis dans ses prédictions, avec un nombre très faible de faux positifs et de faux négatifs. En termes de performance, cela signifie que le modèle est à la fois très sensible (il détecte la majorité des instances positives) et très spécifique (il identifie correctement la majorité des instances négatives).

### Évaluation et interprétation des résultats :

Training Accuracy : 99.73 %					
Model Accuracy Score : 99.47 %					
-----					
Classification_Report XGBoost :					
	precision	recall	f1-score	support	
0	0.98	0.99	0.99	144	
1	1.00	1.00	1.00	606	
accuracy			0.99	750	
macro avg	0.99	0.99	0.99	750	
weighted avg	0.99	0.99	0.99	750	

**Figure 4.21 :** Rapport de classification de l'algorithme XGBoost.

La figure contient un rapport de classification pour un modèle utilisant XGBoost. Le rapport fournit des mesures détaillées sur la précision, le rappel (recall), le score F1 et le support pour chaque classe ainsi que des moyennes pour l'ensemble du modèle. Voici l'interprétation des résultats obtenus :

**Précision d'entraînement (Training Accuracy) :** Le modèle a une précision de 99,73 % lors de l'entraînement, ce qui indique qu'il a très bien appris à partir des données d'entraînement fournies.

**Précision globale (Model Accuracy score) :** Sur les données de test, le modèle a une précision de 99,47 %, ce qui montre qu'il généralise bien les prédictions sur des données qu'il n'a pas vues pendant l'entraînement.

### **Rapport de classification :**

- Pour la classe 0 (absence de défaut) :

Précision : 0,98, indiquant que 98 % des prédictions de la classe 0 sont correctes.

Rappel : 0,99, signifiant que le modèle a identifié 99 % de tous les cas réels de la classe 0.

Score F1 : 0,99, une moyenne harmonique de la précision et du rappel, suggérant un équilibre entre les deux pour la classe 0.

- Pour la classe 1 (présence de défaut) :

Précision : 1,00, ce qui signifie que 100 % des prédictions sont correctes.

Rappel : 1,00, indiquant que le modèle a identifié 100 % de tous les cas réels.

Score F1 : 1,00, montrant un équilibre parfait entre la précision et le rappel.

Moyenne macro : 0,99 pour la précision, le rappel et le score F1, ce qui suggère une performance élevée à travers les classes sans tenir compte du support.

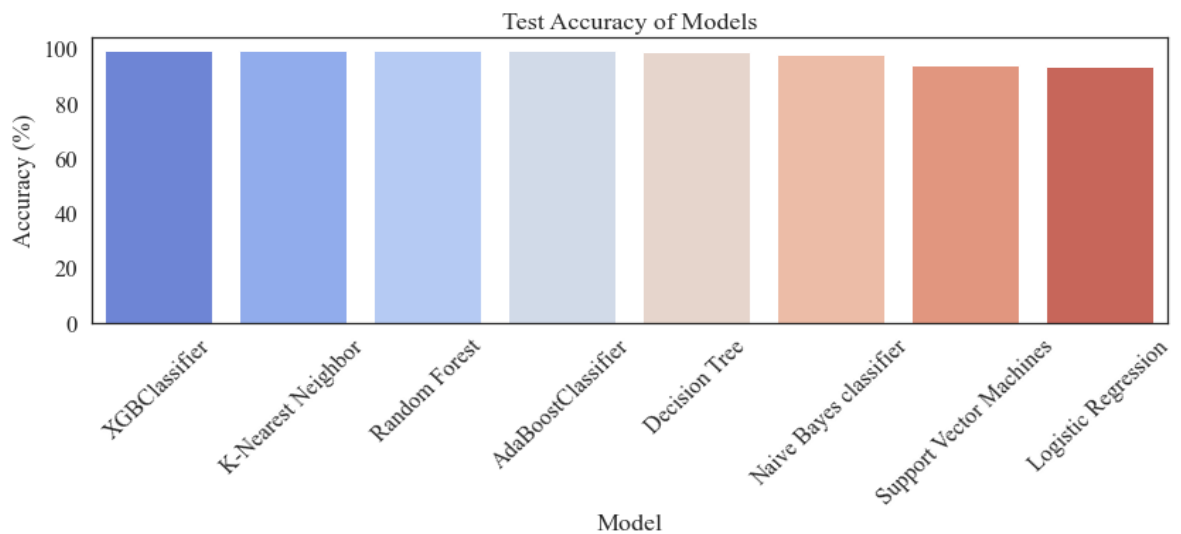
Moyenne pondérée : 0,99 pour la précision, le rappel et le score F1, prenant en compte le support de chaque classe, ce qui indique une performance élevée sur toutes les classes, pondérée par leur représentation dans les données.

Ce rapport indique que le modèle XGBoost a une performance exceptionnelle à la fois sur les données d'entraînement et de test, avec une capacité élevée à distinguer entre les classes de présence et d'absence de défaut.

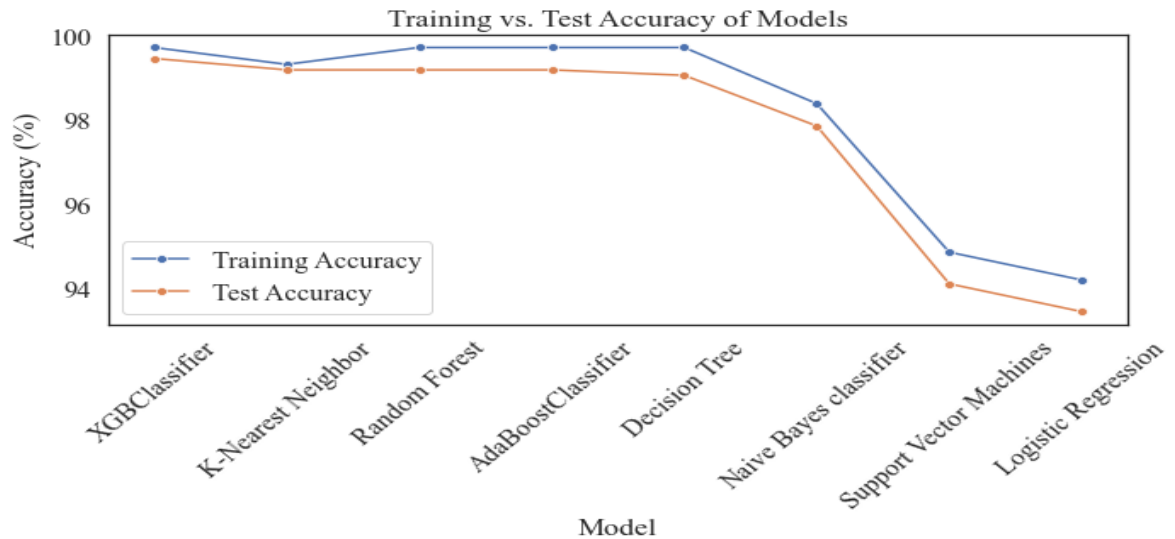
### **4.5.5. Évaluation des performances des algorithmes**

Model	Training Accuracy	Model Accuracy Score
XGBClassifier	99.73	99.47
Random Forest	99.73	99.33
K-Nearest Neighbor	99.33	99.2
AdaBoostClassifier	99.73	99.2
Decision Tree	99.73	99.07
Naive Bayes classifier	98.4	97.87
Support Vector Machines	94.88	94.13
Logistic Regression	94.22	93.47

**Figure 4.22 :** Mesures de performance de divers modèles d'apprentissage automatique sur des données de formation et de test.



**Figure 4.23 :** Comparaison de la précision des tests de modèles dans la classification par apprentissage automatique.



**Figure 4.24 :** Analyse comparative de la précision de l'entraînement et du test pour différents modèles d'apprentissage automatique.

Les figures montrent les taux de Précision d'entraînement (Training Accuracy) et de test pour divers modèles de classification. Pour interpréter et comparer les résultats :

- XGBClassifier : Il a la meilleure précision de test de tous les modèles avec 99,47 %, ce qui suggère qu'il est le meilleur modèle pour la prédiction sur cet ensemble de données spécifique.
- Random Forest et AdaBoostClassifier : Ces modèles ont également une précision de test élevée (99,33 % et 99,2 % respectivement), ce qui indique qu'ils sont très performants, bien que légèrement moins que XGBClassifier.
- K-Nearest Neighbor : Avec une précision de test de 99,2 %, ce modèle est presque aussi performant que les modèles Random Forest et AdaBoostClassifier.
- Decision Tree : Il montre une précision de test de 99,07 %, ce qui est un peu moins élevé que les modèles mentionnés précédemment, mais toujours très précis.



- Naive Bayes classifieur : Avec une précision de test de 97,87 %, ce modèle est moins précis que les modèles à base d'arbres et de forêts aléatoires. Cela peut être dû à la simplicité de l'algorithme Naive Bayes, qui suppose l'indépendance des caractéristiques.
- Support Vector Machines (SVM) : Il affiche une précision de test de 94,13 %, ce qui est nettement inférieur aux autres modèles. Cela pourrait indiquer que SVM peut ne pas être le meilleur choix pour cet ensemble de données ou pourrait nécessiter un réglage plus fin des hyperparamètres.
- Logistic Regression : Avec la précision de test la plus basse de 93,47 %, ce modèle est le moins performant du groupe. Cela peut indiquer que la régression logistique ne capte pas aussi bien la complexité des données ou les interactions entre les caractéristiques que les autres modèles.

En conclusion, les modèles à base d'arbres tels que XGBClassifier, Random Forest, et AdaBoostClassifier offrent les meilleures performances sur cet ensemble de données, avec le XGBClassifier en tête. Les modèles SVM et Logistic Regression sont les moins performants et pourraient bénéficier d'une optimisation supplémentaire ou ne sont peut-être pas adaptés à la nature des données traitées.

- **Évaluation par la validation croisée :**

Model	Training Accuracy ▼	Model Accuracy Score	Cross-Validation Mean Score
Decision Tree	99.73	99.2	98.7992
Random Forest	99.73	99.2	98.8658
AdaBoostClassifier	99.73	99.2	98.7991
XGBClassifier	99.73	99.47	98.8659
K-Nearest Neighbor	99.33	99.2	99.0658
Naive Bayes classifier	98.4	97.87	98.4988
Support Vector Machines	94.88	94.13	94.7624
Logistic Regression	94.22	93.47	94.5289

**Figure 4.25 :** Scores de la précision d'entraînement, de test et de validation croisée pour divers modèles de classification.

Voici l'interprétation et la comparaison des résultats :

- Decision Tree : La précision de test est très élevée à 99,2 %, mais la validation croisée à 98,7992 % suggère que le modèle pourrait légèrement surperformer sur les données d'entraînement.
- Random Forest : La précision de test est également élevée à 99,2 %, avec une validation croisée à 98,8658 %, ce qui indique une performance solide et une bonne généralisation.
- AdaBoostClassifier : La précision de test à 99,2 % est alignée avec la validation croisée à 98,7991 %, montrant une cohérence dans la performance du modèle.
- XGBClassifier : Bien qu'il ait la meilleure précision de test à 99,47 %, la validation croisée à 98,8659 % est légèrement inférieure à celle de Random Forest, ce qui peut indiquer une légère suradaptation par rapport à ce dernier.
- K-Nearest Neighbor : Avec une précision de test de 99,2 % et une validation croisée impressionnante à 99,0658 %, ce modèle montre la meilleure performance en validation croisée, suggérant une excellente généralisation.
- Naive Bayes classifier : Malgré une précision de test plus basse de 97,87 %, la validation croisée à 98,4988 % est relativement élevée, ce qui peut indiquer que le modèle est stable.
- Support Vector Machines : Ce modèle montre une performance moins compétitive avec une précision de test de 94,13 % et une validation croisée à 94,7624 %, ce qui est cohérent mais indique que le modèle est moins performant que les autres.
- Logistic Regression : Avec la précision de test la plus faible à 93,47 % et une validation croisée à 94,5289 %, ce modèle a la performance la plus basse en termes de validation croisée, ce qui pourrait indiquer une capacité limitée à généraliser à partir des données fournies.

En se basant sur la validation croisée, K-Nearest Neighbor et Random Forest semblent être les modèles les plus robustes avec la meilleure capacité de généralisation. XGBClassifier, bien que le plus précis sur le jeu de test, pourrait être légèrement surajusté en comparaison. SVM et Logistic Regression nécessitent potentiellement une amélioration ou une réévaluation pour cet ensemble de données.

## **4.6. Localisation des défauts des transformateurs**

La prédiction et la localisation des défauts revêtent une importance significative dans le domaine de la maintenance préventive des transformateurs électriques. En anticipant les défauts potentiels et en identifiant leur emplacement spécifique, il devient possible de planifier des interventions de maintenance de manière proactive. Cette approche permet d'éviter les arrêts imprévus, d'optimiser les opérations de maintenance et de réduire les coûts associés.

### **4.6.1. Données et stratégies d'apprentissage**

Dans cette étape, nous nous engageons dans la prédiction des défauts des transformateurs ainsi que de leurs localisations respectives. Nos données comprennent six entrées : les tensions triphasées ( $V_A, V_B, V_C$ ) et les courants triphasés ( $I_A, I_B, I_C$ ), ainsi que trois cibles ( $Y_d, Y_{t1}, Y_{t2}$ ). Ces informations serviront à former notre algorithme, lui permettant de prédire la présence de défauts et leurs emplacements.

Les cibles sont définies comme suit : ( $Y_d$ ) indique la présence ou l'absence du défaut, ( $Y_{t1}$ ) indique la présence ou l'absence du défaut dans le transformateur 1, et ( $Y_{t2}$ ) indique la présence ou l'absence du défaut dans le transformateur 2. Notre objectif est de prédire ces trois cibles simultanément, ce qui est réalisable en utilisant une approche de classification Multi-output plutôt qu'une classification traditionnelle.

"Multi-output," dans le contexte de l'apprentissage automatique et de l'analyse de données, fait référence à une situation où un modèle prédictif doit fournir plusieurs

valeurs de sortie ou prédictions pour une seule entrée. En d'autres termes, il s'agit d'un type de problème d'apprentissage supervisé où chaque entrée est associée à plusieurs variables cibles ou étiquettes, et l'objectif est de prédire toutes ces valeurs simultanément.

Cette méthode est spécialement pertinente pour notre situation, car elle vise à prédire non seulement la présence de défauts dans le transformateur, mais aussi à déterminer avec précision leur emplacement spécifique.

#### **4.6.2. Division et entraînement des algorithmes**

Les données ont également été divisées en 75% de données de formation et 25% de données de test, nous utilisons les classificateurs d'apprentissage automatique sélectionnés en utilisant la bibliothèque Scikit learn pour prédire les défauts des transformateurs et leur emplacement, dans notre cas nous avons deux transformateurs : "Transformateur 1" ou "Transformateur 2".

La prédiction et la localisation des défauts dans les transformateurs constituent un problème de classification, mais avec la particularité que cette classification est multi-output. Les algorithmes mentionnés précédemment seront appliqués pour résoudre cette tâche.

#### **4.6.3. Résultats obtenus et discussions**

La prédiction et la localisation des défauts dans les transformateurs électriques sont des éléments clés pour assurer la disponibilité continue de l'électricité, optimiser les coûts de maintenance, et renforcer la sécurité opérationnelle des installations électriques.

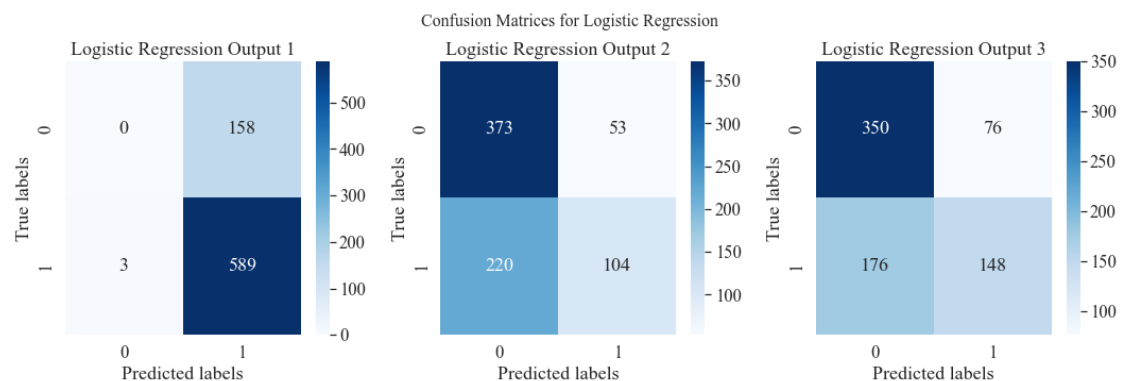
Afin d'anticiper et repérer les défauts des transformateurs électriques, nous avons exploré l'utilisation de l'apprentissage automatique supervisé adapté à un problème de classification Multi-output. Les algorithmes suivants ont été sélectionnés en raison de

leur robustesse, de leur efficacité et de leur aptitude à traiter des problèmes de classification similaires au nôtre.

#### - Régression Logistique (Logistic Regression) :

**Raison du choix:** La régression logistique est souvent utilisée comme référence de base. Elle est adaptée aux tâches de classification multilabel, et sa simplicité permet une interprétation facile des résultats.

#### Matrice de confusion et Évaluation :



**Figure 4.26 :** Ensemble de matrices de confusion affichant les résultats de l’algorithme régression logistique.

Logistic Regression - Training Accuracy: 18.42%, Test Accuracy: 18.67%

Classification Report:

	precision	recall	f1-score	support
0	0.79	0.99	0.88	592
1	0.66	0.32	0.43	324
2	0.66	0.46	0.54	324
micro avg	0.75	0.68	0.71	1240
macro avg	0.70	0.59	0.62	1240
weighted avg	0.72	0.68	0.67	1240
samples avg	0.72	0.54	0.59	1240

**Figure 4.27 :** Rapport de classification de l’algorithme régression logistique MultiOutput.

Le rapport de classification montre qu'une incohérence notable dans les précisions signalées pour l'entraînement et les tests (18,42 % et 18,67 % respectivement), qui sont extrêmement basses. Cela suggère que le modèle n'a pas du tout appris à partir des données. Voici l'interprétation des scores par classe, en supposant que les scores de précision soient corrects :

- Pour la classe 0 :

Précision : 0,79 indique que 79 % des prédictions du modèle étaient correctes.

Rappel : 0,99 suggère que le modèle a correctement identifié 99 % de toutes les instances réelles de la classe 0.

Score F1 : 0,88 est un score élevé, indiquant un bon équilibre entre la précision et le rappel pour la classe 0.

- Pour la classe 1 :

Précision : 0,66 indique que 66 % des prédictions de la classe 1 étaient correctes.

Rappel : 0,32 montre que seulement 32 % des instances réelles de la classe 1 ont été correctement identifiées.

Score F1 : 0,43 est relativement faible, reflétant la faible performance du modèle pour cette classe.

- Pour la classe 2 :

Précision : 0,66 indique que 66 % des prédictions de la classe 2 étaient correctes.

Rappel : 0,46 suggère que moins de la moitié des instances réelles de la classe 2 ont été correctement identifiées.

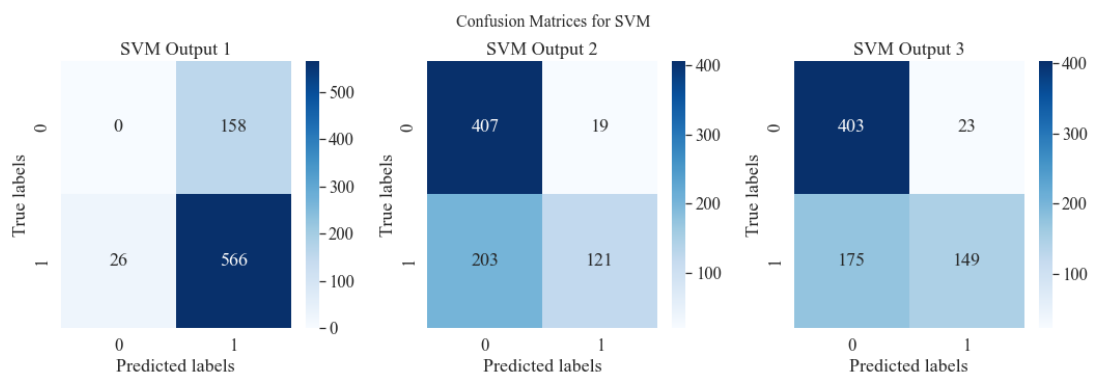
Score F1 : 0,54 indique une performance légèrement meilleure que celle de la classe 1 mais toujours inférieure à la moyenne.

Les moyennes de précision et de rappel (micro, macro et pondérée) montrent des performances générales assez basses, particulièrement pour les classes 1 et 2, ce qui est cohérent avec le faible score global de précision du modèle. Le modèle semble avoir une performance décente pour la classe 0, mais il a du mal à identifier correctement les classes 1 et 2, ce qui pourrait indiquer un déséquilibre des classes ou le besoin d'améliorer la représentativité des caractéristiques ou l'ajustement du modèle pour ces classes. On peut conclure que cet algorithme n'est pas adapté pour ce type de tâches.

#### - Machine à Vecteurs de Support (Support Vector Machine - SVM) :

**Raison du choix :** Les SVM sont efficaces dans la séparation de classes complexes. Leur capacité à gérer des espaces de grande dimension en fait un choix judicieux pour des problèmes multilabel complexes.

#### Matrice de confusion et Évaluation :



**Figure 4.28 :** Ensemble de matrices de confusion affichant les résultats de l'algorithme SVM MultiOutput.

SVM - Training Accuracy: 27.27%, Test Accuracy: 27.47%					
Classification Report:					
	precision	recall	f1-score	support	
0	0.78	0.96	0.86	592	
1	0.86	0.37	0.52	324	
2	0.87	0.46	0.60	324	
micro avg	0.81	0.67	0.73	1240	
macro avg	0.84	0.60	0.66	1240	
weighted avg	0.83	0.67	0.70	1240	
samples avg	0.73	0.54	0.60	1240	

**Figure 4.29 :** Rapport de classification de l'algorithme SVM MultiOutput.

La Précision d'entraînement (Training Accuracy) et de test sont très basses (27,28 % et 27,47 % respectivement), ce qui suggère un problème avec le modèle ou les données. Voici l'interprétation des scores par classe, en supposant que les chiffres sont corrects :

- Pour la classe 0 :

Précision : 0,78 indique que 78 % des instances que le modèle a prédit comme appartenant à la classe 0 étaient correctes.

Rappel : 0,96 suggère que le modèle a correctement identifié 96 % de toutes les instances réelles de la classe 0.

Score F1 : 0,86 est un score élevé, indiquant un bon équilibre entre la précision et le rappel pour la classe 0.

- Pour la classe 1 :

Précision : 0,86 indique que 86 % des prédictions de la classe 1 étaient correctes.

Rappel : 0,37 montre que seulement 37 % des instances réelles de la classe 1 ont été correctement identifiées.

Score F1 : 0,52 est relativement faible, reflétant la faible performance du modèle pour cette classe.



- Pour la classe 2 :

Précision : 0,87 indique que 87 % des prédictions de la classe 2 étaient correctes.

Rappel : 0,46 suggère que moins de la moitié des instances réelles de la classe 2 ont été correctement identifiées.

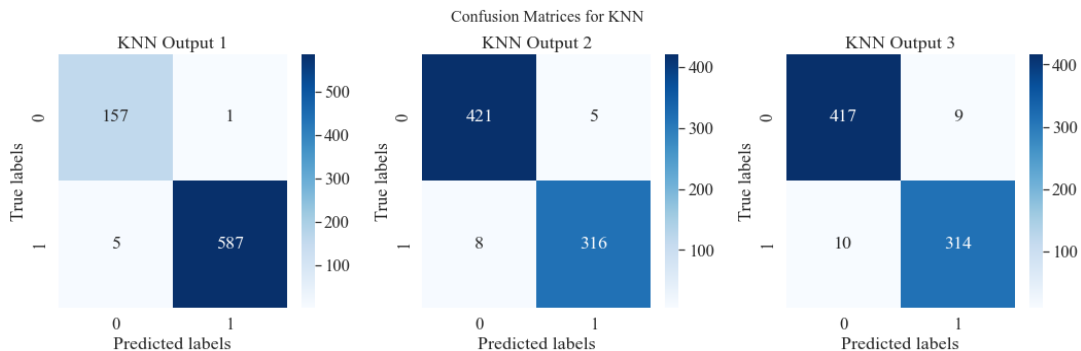
Score F1 : 0,60 indique une performance modérée pour la classe 2.

Les moyennes de précision et de rappel (micro, macro et pondérée) montrent des performances générales assez basses, particulièrement pour les classes 1 et 2, ce qui est cohérent avec les scores globalement bas de précision du modèle. Cela peut indiquer que le modèle SVM n'est pas bien adapté aux données. La différence entre la précision des classes indique également un déséquilibre possible dans la capacité du modèle à traiter chaque classe.

- **K-Nearest Neighbors (K-NN) :**

**Raison du choix :** K-NN est un choix approprié lorsque les relations de voisinage sont importantes. Il peut être robuste pour identifier des schémas locaux dans des données multilabel.

#### Matrice de confusion et Évaluation :



**Figure 4.30 :** Ensemble de matrices de confusion affichant les résultats de l'algorithme KNN MultiOutput.

KNN - Training Accuracy: 97.86%, Test Accuracy: 96.67%					
Classification Report:					
		precision	recall	f1-score	support
	0	1.00	0.99	0.99	592
	1	0.98	0.98	0.98	324
	2	0.97	0.97	0.97	324
	micro avg	0.99	0.98	0.98	1240
	macro avg	0.98	0.98	0.98	1240
	weighted avg	0.99	0.98	0.98	1240
	samples avg	0.78	0.78	0.78	1240

**Figure 4.31 :** Rapport de classification de l'algorithme KNN MultiOutput.

La précision d'entraînement (Training Accuracy) et la précision du test (test accuracy) sont respectivement de 97,86% et 96,67%, ce qui indique une performance élevée et une bonne généralisation du modèle. Voici l'interprétation des résultats par classe :

- Pour la classe 0 :

Précision : 1.00 signifie que 100% des prédictions du modèle étaient correctes.

Rappel : 0.99 indique que le modèle a correctement identifié 99% des instances réelles.

Score F1 : 0.99, reflétant un excellent équilibre entre la précision et le rappel.

- Pour la classe 1 :

Précision : 0.98 signifie que 98% des prédictions du modèle étaient correctes.

Rappel : 0.98 indique que le modèle a correctement identifié 98% des instances réelles.

Score F1 : 0.98, indiquant également un excellent équilibre entre la précision et le rappel.

- Pour la classe 2 :

Précision : 0.97 signifie que 97% des prédictions du modèle étaient correctes.

Rappel : 0.97 indique que le modèle a correctement identifié 97% des instances réelles.

Score F1 : 0.97, montrant une performance élevée pour la classe 2.

Les moyennes générales sont également élevées :

Moyenne micro : 0.99 pour la précision, le rappel et le score F1, reflétant la performance globale pour toutes les instances.

Moyenne macro : 0.98 pour la précision, le rappel et le score F1, indiquant une haute performance moyenne équilibrée sur toutes les classes sans tenir compte de leur fréquence.

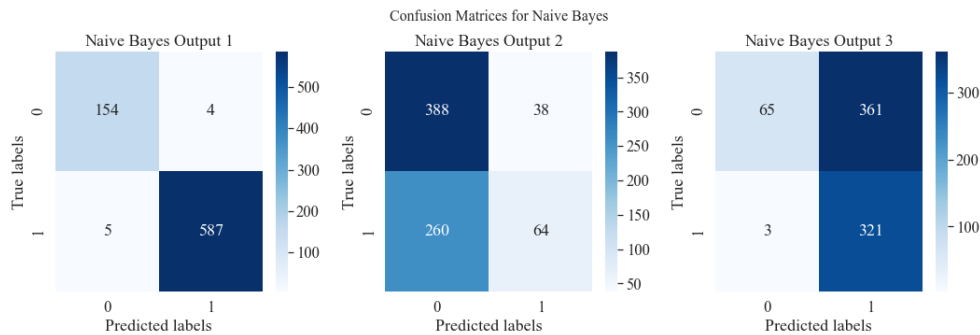
-Moyenne pondérée : 0.99 pour la précision, le rappel et le score F1, prenant en compte le support de chaque classe, ce qui confirme la bonne performance du modèle sur les classes proportionnellement représentées.

Ce rapport suggère que le modèle KNN a bien fonctionné sur cet ensemble de données, avec des performances élevées dans la classification des trois classes testées.

#### - **Naïve Bayes :**

**Raison du choix :** Naïve Bayes est souvent utilisé dans des contextes où les caractéristiques sont indépendantes. Si les différentes classes de défauts ont des caractéristiques relativement indépendantes, Naïve Bayes peut-être efficace.

#### **Matrice de confusion et Évaluation :**



**Figure 4.32 :** Ensemble de matrices de confusion affichant les résultats de l’algorithme Naïve Bayes MultiOutput.

```
Naive Bayes - Training Accuracy: 41.15%, Test Accuracy: 38.53%
Classification Report:
      precision    recall  f1-score   support

    0       0.99      0.99      0.99     592
    1       0.63      0.20      0.30     324
    2       0.47      0.99      0.64     324

 micro avg       0.71      0.78      0.74    1240
 macro avg       0.70      0.73      0.64    1240
weighted avg       0.76      0.78      0.72    1240
samples avg       0.64      0.62      0.62    1240
```

**Figure 4.33 :** Rapport de classification de l’algorithme Naïve Bayes MultiOutput.

La précision d’entraînement (Training Accuracy) et la précision du test (test accuracy) sont relativement basses (41,15 % et 38,53 % respectivement), ce qui suggère que le modèle ne s’adapte pas bien aux données. Voici l’interprétation des résultats par classe :

- Pour la classe 0 :

Précision : 0.99 indique que 99 % des prédictions du modèle étaient correctes.

Rappel : 0.99 suggère que le modèle a correctement identifié 99 % des instances réelles.

Score F1 : 0.99, reflétant une excellente performance pour la classe 0.

- Pour la classe 1 :

Précision : 0.63 indique que 63 % des prédictions de la classe 1 étaient correctes.

Rappel : 0.20 montre que seulement 20 % des instances réelles de la classe 1 ont été correctement identifiées.

Score F1 : 0.30, ce qui est très faible, indiquant une mauvaise performance du modèle pour cette classe.

- Pour la classe 2 :

Précision : 0.47 indique que 47 % des prédictions de la classe 2 étaient correctes.

Rappel : 0.99 indique que le modèle a correctement identifié 99 % des instances réelles.

Score F1 : 0.64, montrant une meilleure performance que la classe 1 mais toujours inférieure à la moyenne.

Moyenne micro : 0.71 pour la précision, 0.78 pour le rappel et 0.74 pour le score F1, reflétant la performance globale du modèle sur l'ensemble des instances.

Moyenne macro : 0.70 pour la précision, 0.73 pour le rappel et 0.64 pour le score F1, ce qui suggère des performances non uniformes à travers les classes.

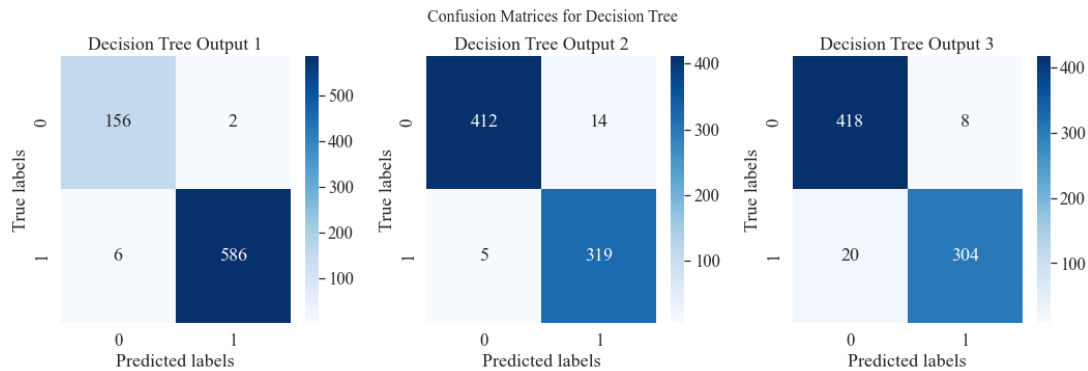
Moyenne pondérée : 0.76 pour la précision, 0.78 pour le rappel et 0.72 pour le score F1, ce qui tient compte de la fréquence de chaque classe.

Le modèle Naïve Bayes semble être performant pour les classes 0 et 2 mais a des difficultés considérables avec la classe 1, comme en témoigne le faible rappel et le score F1 pour cette classe. Cela pourrait être dû à des caractéristiques qui ne sont pas bien représentées par l'hypothèse d'indépendance ou à un déséquilibre dans les données qui n'est pas bien géré par le modèle Naïve Bayes.

- **Arbre de Décision (Decision Tree) :**

**Raison du choix :** Les arbres de décision sont bons pour modéliser des règles complexes. Ils peuvent être interprétables et permettre d'expliquer les décisions du modèle.

### Matrice de confusion et Évaluation :



**Figure 4.34 :** Ensemble de matrices de confusion affichant les résultats de l'algorithme arbres de décisions MultiOutput.

```
Decision Tree - Training Accuracy: 96.0%, Test Accuracy: 94.53%
Classification Report:

```

	precision	recall	f1-score	support
0	1.00	0.99	0.99	592
1	0.96	0.98	0.97	324
2	0.97	0.94	0.96	324
micro avg	0.98	0.97	0.98	1240
macro avg	0.98	0.97	0.97	1240
weighted avg	0.98	0.97	0.98	1240
samples avg	0.77	0.77	0.77	1240

**Figure 4.35 :** Rapport de classification de l'algorithme arbres de décisions MultiOutput.

Les résultats montrent une haute précision d'entraînement (training accuracy) (96,08 %) et de test (94,53 %), suggérant que le modèle généralise bien aux données inédites. Voici l'interprétation détaillée des résultats par classe :

- Pour la Classe 0 :

Précision : Parfaite à 1.00, ce qui signifie que toutes les prédictions du modèle pour la classe 0 étaient correctes.

Rappel : Très élevé à 0.99, indiquant que le modèle a identifié correctement 99 % des instances réelles de la classe 0.

Score F1 : 0.99, reflétant une excellente performance sur la classe 0.

- Pour la Classe 1 :

Précision : Élevée à 0.96, indiquant que 96 % des prédictions de la classe 1.

Rappel : Également élevé à 0.98, montrant que le modèle a identifié 98 % des instances réelles de la classe 1.

Score F1 : 0.97, indiquant une performance élevée sur la classe 1.

- Pour la Classe 2 :

Précision : Très bonne à 0.97, signifiant que 97 % des prédictions étaient correctes.

Rappel : Solide à 0.94, indiquant que le modèle a identifié 94 % des instances réelles.

Score F1 : 0.96, montrant une performance élevée pour la classe 2.

Moyenne micro : 0.98 pour la précision, 0.97 pour le rappel et 0.98 pour le score F1, reflétant la performance globale du modèle sur l'ensemble des instances.

Moyenne macro : 0.98 pour la précision, 0.97 pour le rappel et 0.97 pour le score F1, indiquant une haute performance moyenne équilibrée sur toutes les classes sans tenir compte de leur fréquence.

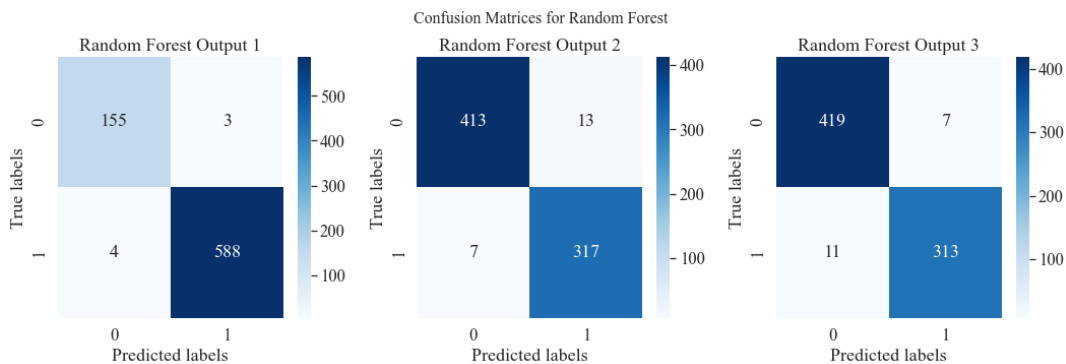
Moyenne pondérée : 0.98 pour la précision, 0.97 pour le rappel et 0.98 pour le score F1, ce qui tient compte de la fréquence de chaque classe.

Ces résultats suggèrent que le modèle d'arbre de décision est très performant pour cette tâche de classification, avec des scores élevés dans toutes les classes et une bonne capacité à généraliser des données d'entraînement aux données de test.

#### - Forêt Aléatoire (Random Forest) :

**Raison du choix :** Random Forest est une extension des arbres de décision. Il peut gérer la complexité et apporter une amélioration de la précision en agrégeant plusieurs arbres.

#### Matrice de confusion et Évaluation :



**Figure 4.36 :** Ensemble de matrices de confusion affichant les résultats de l'algorithme forêt aléatoire MultiOutput.

```
Random Forest - Training Accuracy: 96.31%, Test Accuracy: 95.2%
Classification Report:
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	592
1	0.96	0.98	0.97	324
2	0.98	0.97	0.97	324
micro avg	0.98	0.98	0.98	1240
macro avg	0.98	0.98	0.98	1240
weighted avg	0.98	0.98	0.98	1240
samples avg	0.77	0.78	0.77	1240

**Figure 4.37 :** Rapport de classification de l'algorithme foret aléatoire MultiOutput.



Les résultats montrent une précision d'entraînement (training accuracy) de 96,31% et de test de 95,2%, indiquant que le modèle généralise bien aux données de test. Voici l'interprétation des résultats par classe :

- Pour la Classe 0 :

Précision : 0,99, signifiant que 99% des prédictions du modèle étaient correctes.

Rappel : 0,99, indiquant que le modèle a correctement identifié 99% des instances réelles.

Score F1 : 0,99, montrant une performance exceptionnelle pour la classe 0.

- Pour la Classe 1 :

Précision : 0,96, indiquant que 96% des prédictions de la classe 1 étaient correctes.

Rappel : 0,98, montrant que le modèle a correctement identifié 98% des instances réelles de la classe 1.

Score F1 : 0,97, indiquant une performance très élevée pour la classe 1.

- Pour la Classe 2 :

Précision : 0,98, signifiant que 98% des prédictions de la classe 2 étaient correctes.

Rappel : 0,97, indiquant que le modèle a correctement identifié 97% des instances réelles.

Score F1 : 0,97, montrant également une performance élevée pour la classe 2.

Les moyennes générales sont toutes élevées, indiquant une performance consistante à travers les différentes classes :

Moyenne micro : 0,98 pour la précision, le rappel et le score F1, reflétant la performance globale du modèle sur l'ensemble des instances.

Moyenne macro : 0,98 pour la précision, le rappel et le score F1, indiquant une haute performance moyenne équilibrée sur toutes les classes sans tenir compte de leur fréquence.

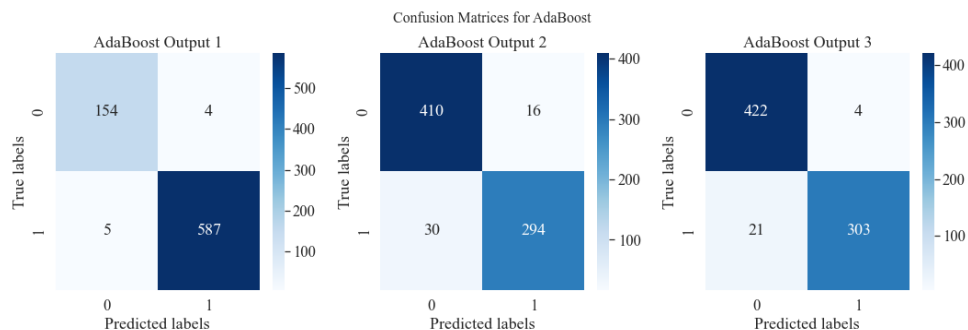
Moyenne pondérée : 0,98 pour la précision, le rappel et le score F1, prenant en compte la fréquence de chaque classe dans l'ensemble de données.

En somme, le modèle de Forêt Aléatoire démontre une excellente capacité de classification sur cet ensemble de données, avec une performance particulièrement élevée pour toutes les classes testées.

#### - AdaBoost :

**Raison du choix :** AdaBoost peut être utile pour améliorer la performance des modèles faibles. Il peut être efficace pour améliorer la précision globale de la classification multilabel.

#### Matrice de confusion et Évaluation :



**Figure 4.38 :** Ensemble de matrices de confusion affichant les résultats de l'algorithme AdaBoost MultiOutput.

AdaBoost - Training Accuracy: 91.95%, Test Accuracy: 90.93%					
Classification Report:					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	592	
1	0.95	0.91	0.93	324	
2	0.99	0.94	0.96	324	
micro avg	0.98	0.95	0.97	1240	
macro avg	0.98	0.94	0.96	1240	
weighted avg	0.98	0.95	0.97	1240	
samples avg	0.78	0.75	0.76	1240	

**Figure 4.39 :** Rapport de classification de l'algorithme AdaBoost MultiOutput.

Les résultats montrent une précision d'entraînement (training accuracy) de 91,95% et de test de 90,93%, indiquant une bonne généralisation du modèle aux données de test. Voici l'interprétation des résultats par classe :

- Pour la Classe 0 :

Précision : 0,99, indiquant que 99% des prédictions du modèle étaient correctes.

Rappel : 0,99, suggérant que le modèle a correctement identifié 99% des instances réelles.

Score F1 : 0,99, montrant une performance exceptionnelle pour la classe 0.

- Pour la Classe 1 :

Précision : 0,95, signifiant que 95% des prédictions de la classe 1 étaient correctes.

Rappel : 0,91, montrant que le modèle a correctement identifié 91% des instances réelles.

Score F1 : 0,93, indiquant une performance très élevée pour la classe 1.

- Pour la Classe 2 :

Précision : 0,99, indiquant que 99% des prédictions de la classe 2 étaient correctes.

Rappel : 0,94, suggérant que le modèle a correctement identifié 94% des instances réelles.

Score F1 : 0,96, reflétant également une haute performance pour la classe 2.

Les moyennes générales sont élevées, illustrant une performance constante à travers les différentes classes :

Moyenne micro : 0,98 pour la précision et 0,97 pour le score F1, reflétant la performance globale du modèle sur l'ensemble des instances.

Moyenne macro : 0,98 pour la précision et 0,96 pour le score F1, indiquant une haute performance moyenne équilibrée sur toutes les classes sans tenir compte de leur fréquence.

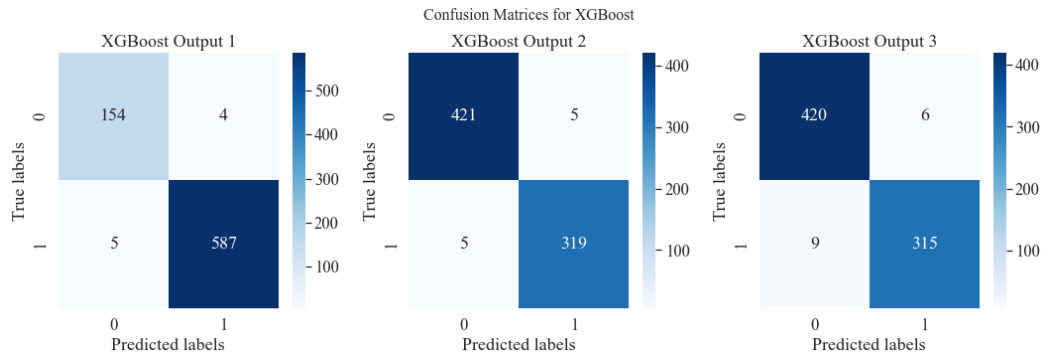
Moyenne pondérée : 0,98 pour la précision et 0,97 pour le score F1, prenant en compte la fréquence de chaque classe dans l'ensemble de données.

En somme, le modèle AdaBoost démontre une excellente capacité de classification sur cet ensemble de données, avec des performances particulièrement élevées pour toutes les classes testées.

#### - **Extreme Gradient Boosting (XGBoost) :**

**Raison du choix :** XGBoost est une version avancée de l'AdaBoost. Il offre une meilleure précision, une régularisation efficace et peut être adapté aux tâches multilabel.

#### **Matrice de confusion et Évaluation :**



**Figure 4.40 :** Ensemble de matrices de confusion affichant les résultats de l’algorithme XGBoost MultiOutput.

XGBoost - Training Accuracy: 98.8%, Test Accuracy: 96.8%					
Classification Report:					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	592	
1	0.98	0.98	0.98	324	
2	0.98	0.97	0.98	324	
micro avg	0.99	0.98	0.99	1240	
macro avg	0.99	0.98	0.98	1240	
weighted avg	0.99	0.98	0.99	1240	
samples avg	0.78	0.78	0.78	1240	

**Figure 4.41 :** Rapport de classification de l’algorithme XGBoost MultiOutput.

Les résultats montrent une excellente précision d'entraînement (training accuracy) de 98,8% et une précision de test de 96,8%, indiquant une haute performance et une bonne généralisation du modèle. Voici l’interprétation des résultats par classe :

- Pour la Classe 0 :

Précision : 0,99, ce qui signifie que 99% des prédictions du modèle étaient correctes.

Rappel : 0,99, indiquant que le modèle a correctement identifié 99% des instances réelles.

Score F1 : 0,99, montrant une performance presque parfaite pour la classe 0.

- Pour la Classe 1 :

Précision : 0,98, signifiant que 98% des prédictions de la classe 1 étaient correctes.

Rappel : 0,98, montrant que le modèle a correctement identifié 98% des instances réelles.

Score F1 : 0,98, indiquant une performance très élevée pour la classe 1.

- Pour la Classe 2 :

Précision : 0,98, indiquant que 98% des prédictions de la classe 2 étaient correctes.

Rappel : 0,97, suggérant que le modèle a correctement identifié 97% des instances réelles.

Score F1 : 0,98, reflétant également une haute performance pour la classe 2.

Moyenne micro : 0,99 pour la précision, 0,98 pour le rappel et 0,99 pour le score F1, reflétant la performance globale du modèle sur l'ensemble des instances.

Moyenne macro : 0,99 pour la précision, 0,98 pour le rappel et 0,98 pour le score F1, indiquant une performance uniforme et élevée sur toutes les classes.

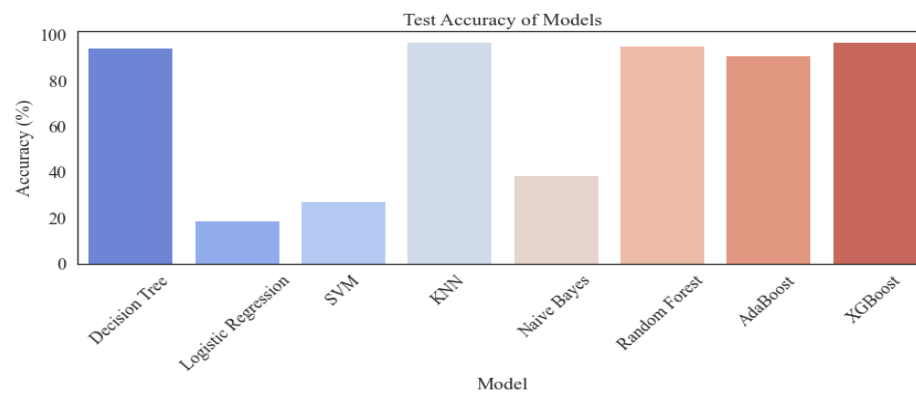
Moyenne pondérée : 0,99 pour la précision, 0,98 pour le rappel et 0,99 pour le score F1, prenant en compte la fréquence de chaque classe dans l'ensemble de données.

Ces résultats suggèrent que le modèle XGBoost est extrêmement performant pour cette tâche de classification, avec des performances élevées dans toutes les classes testées et une excellente capacité à généraliser de l'entraînement aux données de test.

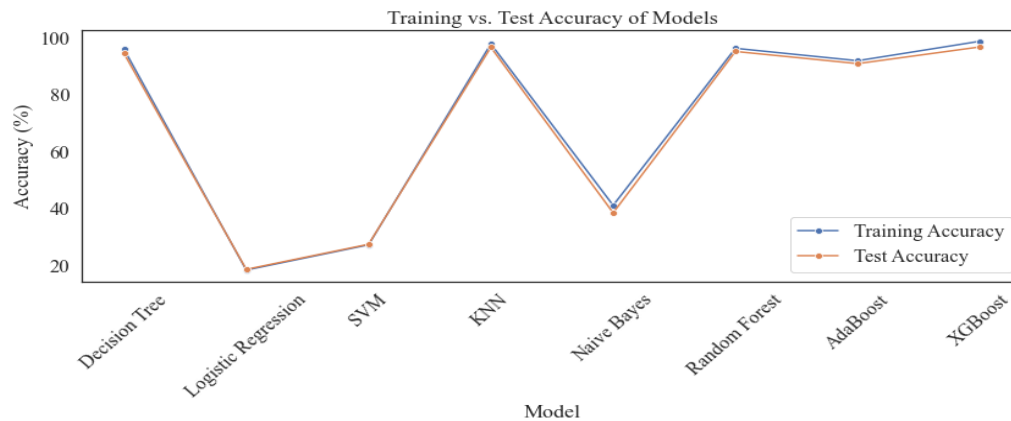
#### **4.6.4. Évaluation des performances des algorithmes**

Model	Training Accuracy	Test Accuracy
XGBoost	98.8	96.8
KNN	97.86	96.67
Random Forest	96.31	95.2
Decision Tree	96	94.53
AdaBoost	91.95	90.93
Naive Bayes	41.15	38.53
SVM	27.27	27.47
Logistic Regression	18.42	18.67

**Figure 4.42 :** Mesures de performance de divers modèles d'apprentissage automatique sur des données de formation et de test.



**Figure 4.43 :** Comparaison de la précision des tests de modèles dans la classification par apprentissage automatique.



**Figure 4.44 :** Analyse comparative de la précision de l'entraînement et du test pour différents modèles d'apprentissage automatique.

Voici l'interprétation et la comparaison des performances de ces modèles :

- **XGBoost** : Présente les meilleures performances avec une précision d'entraînement de 98,8% et de test de 96,8%, ce qui indique une très bonne généralisation.
- **KNN (K-Nearest Neighbors)** : Affiche également de hautes performances avec une précision d'entraînement de 97,86% et de test de 96,67%. C'est le deuxième meilleur modèle en termes de précision de test.
- **Random Forest** : Montre une précision d'entraînement de 96,31% et de test de 95,2%. Il reste compétitif mais légèrement derrière KNN et XGBoost.
- **Decision Tree** : A une précision d'entraînement de 96% et de test de 94,53%. Bien que la performance soit solide, elle est inférieure à celle des modèles basés sur des ensembles tels que Random Forest et XGBoost.
- **AdaBoost** : Affiche une précision d'entraînement de 91,95% et de test de 90,93%, ce qui est inférieur aux modèles mentionnés précédemment mais reste respectable.
- **Naive Bayes** : A des performances nettement inférieures avec une précision d'entraînement de 41,15% et de test de 38,53%. Cela suggère que le modèle n'est pas bien adapté aux données ou que les caractéristiques ne suivent pas l'hypothèse d'indépendance de Naive Bayes.
- **SVM (Support Vector Machine)** : Présente une précision d'entraînement de 27,27% et de test de 27,47%. Ces performances sont très basses par rapport aux autres modèles, ce qui peut indiquer un mauvais choix de paramètres du noyau ou d'autres paramètres du modèle.
- **Logistic Regression** : A les performances les plus basses avec une précision d'entraînement de 18,42% et de test de 18,67%. Cela pourrait être dû à une



séparation linéaire insuffisante des classes dans l'espace des caractéristiques ou à une sous-adaptation sévère.

En conclusion, XGBoost, KNN et Random Forest sont les modèles les plus performants pour cet ensemble de données spécifique, tandis que Naïve Bayes, SVM et Logistic Regression semblent être inappropriés et nécessiteraient des ajustements significatifs ou une refonte pour améliorer leur performance.

#### **4.7. Conclusion**

En synthèse, le présent rapport se divise en deux segments principaux. La première section a été dédiée à la détection de défauts et a révélé que les modèles basés sur des structures arborescentes, spécifiquement le XGBClassifier, le Random Forest et l'AdaBoostClassifier, ont excellé dans l'analyse de nos données. Le XGBClassifier s'est démarqué, démontrant une supériorité en termes de performance. En contraste, les modèles comme le SVM et la régression logistique ont enregistré des performances plus modestes. Il en découle que de futures recherches pourraient être bénéfiques pour parfaire ces modèles.

Dans la deuxième section, axée sur la prédiction et la localisation des anomalies au sein des transformateurs électriques, l'analyse a permis d'identifier le XGBoost, le KNN et le Random Forest comme étant les méthodes les plus adéquates et les plus performantes pour notre jeu de données. À l'inverse, les techniques telles que le Naïve Bayes, le SVM et la régression logistique se sont avérées moins efficaces.

## **Chapitre 5**

# **Prédiction des défauts des transformateurs électriques par analyse prédictive des données de surveillance**

### **5.1. Introduction**

La fabrication d'un transformateur de distribution suit un processus similaire à celui utilisé pour les transformateurs de moindre envergure. Les composants essentiels de ce type de transformateur incluent le réservoir contenant l'huile, un dispositif conservateur, le relais Buchholz, un dispositif d'aération, un indicateur du niveau d'huile, un capteur de température, une valve de soulagement de pression, un relais de température, un système de refroidissement par radiateur et une connexion externe par l'intermédiaire d'un manchon.

L'huile dans le transformateur est distribuée en utilisant un réservoir qui permet de saturer les bobines en étant placé à l'intérieur de celles-ci. Au-dessus de ce réservoir d'huile et hors du cadre principal du transformateur, on installe un dispositif de conservation. Ce dernier est connecté au réservoir principal par un conduit en métal, facilitant ainsi le contact et l'expansion de l'huile en fonction des variations de charge, ce qui aide à réguler la température de l'huile.

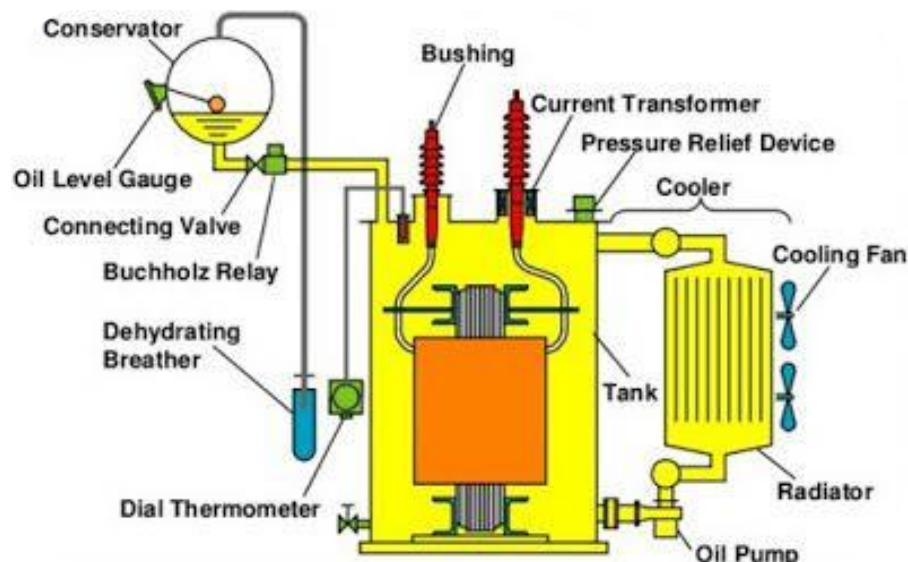
Pour surveiller et signaler des anomalies telles que la baisse du niveau d'huile ou des problèmes dans la circulation de l'huile entre le réservoir et le transformateur, on utilise le relais Buchholz, particulièrement quand un dispositif de conservation est présent.

Le dessiccateur, contenant du gel de silice, joue un rôle crucial en éliminant l'humidité présente dans l'huile. Ce gel change de couleur, passant du bleu au rose, lorsqu'il n'est plus en mesure d'absorber l'humidité.

Un indicateur de niveau d'huile est présent pour montrer la quantité d'huile dans le dispositif de conservation, tandis qu'un capteur de température surveille la chaleur de l'huile. En cas de surchauffe, le transformateur est automatiquement mis hors service pour éviter les dommages.

Pour prévenir une éventuelle explosion due à une pression interne excessive, un mécanisme de soulagement de la pression est intégré. Un relais thermique est employé pour indiquer la température des bobines.

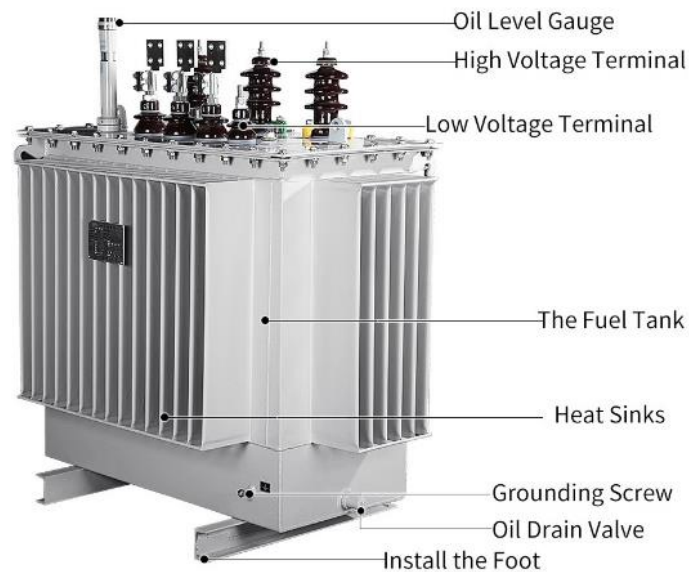
Le radiateur, quant à lui, améliore l'efficacité du système de refroidissement du transformateur. Enfin, une douille assure la connexion entre les bobinages internes du transformateur et le réseau électrique externe.



**Figure 5.1 :** Diagramme d'un Transformateur à Huile avec Système de Refroidissement et Protection [58].

- **Conservator :** Conservateur.

- **Oil Level Gauge** : Jauge de Niveau d'Huile
- **Bushing** : Traversée
- **Current Transformer** : Transformateur de Courant
- **Pressure Relief Device** : Dispositif de Détente de Pression
- **Cooler** : Refroidisseur
- **Cooling Fan** : Ventilateur de Refroidissement
- **Tank** : Cuve
- **Radiator** : Radiateur
- **Oil Pump** : Pompe à Huile
- **Connecting Valve** : Vanne de Connexion
- **Buchholz Relay** : Relais Buchholz
- **Dehydrating Breather** : Déshydrateur d'Air
- **Dial Thermometer** : Thermomètre à Cadran



**Figure 5.2** : Transformateur Électrique avec Système de Refroidissement et Terminaux de Tension [59].

- Oil Level Gauge : Jauge de Niveau d'Huile
- High Voltage Terminal : Borne Haute Tension

- Low Voltage Terminal : Borne Basse Tension
- The Fuel Tank : Le Réservoir d'Huile
- Heat Sinks : Dissipateurs Thermiques
- Grounding Screw : Vis de Terre
- Oil Drain Valve : Vanne de Vidange d'Huile
- Install the Foot : Installer le Pied

Dans cette section, nous nous concentrerons sur la prédiction des défauts et des pannes potentielles des transformateurs électriques en exploitant divers indicateurs de surveillance. Ces indicateurs fournissent des mesures critiques qui peuvent être analysées pour anticiper et identifier les problèmes avant qu'ils ne conduisent à des défaillances majeures.

## **5.2. Base de données**

Dans le cadre de cette étude de cas, le jeu de données utilisé pour l'analyse a été acquis auprès d'une plateforme en ligne renommée, spécialisée dans la mise à disposition de bases de données dédiées à la recherche.

Ce jeu de données exhaustif comprend un total de 20 316 instances, chacun représentant des instances distinctes liées au sujet de l'étude. Les données sont soigneusement organisées dans un fichier format CSV, permettant une utilisation facile des données. Originellement, ces données sont collectées via des dispositifs IoT du 25 juin 2019 au 14 avril 2020 et ont été mises à jour toutes les 15 minutes.

Le volume important et la diversité du jeu de données sont cruciaux pour améliorer la précision et la fiabilité des résultats de l'étude.

Cette base de données contient sept indicateurs. Parmi ces indicateurs, six d'entre eux (OTI, WTI, ATI, OLI, OTI\_A, OTI\_T) seront utilisés comme entrées pour nos algorithmes de Machine Learning. L'indicateur restant (MOG\_A) sera la variable que

nous chercherons à prédire ou à cibler. Voici les indicateurs clés qui sont décrits ci-dessous :

- **OTI (Oil Temperature Indicator)** : En français, Indicateur de température de l'huile. Cet indicateur mesure la température de l'huile du transformateur, offrant des insights sur le fonctionnement thermique global et la dissipation de chaleur.
- **WTI (Winding Temperature Indicator)** : En français, Indicateur de température de bobinage. Il surveille la température des bobinages, un facteur crucial pour évaluer la santé du transformateur, car une surchauffe des bobinages peut indiquer une surcharge ou des défauts internes.
- **ATI (Ambient Temperature Indicator)** : En français, Indicateur de température ambiante. Cet indicateur mesure la température ambiante autour du transformateur. Les variations significatives entre la température ambiante et les températures de l'huile ou des bobinages peuvent signaler des inefficacités ou des anomalies.
- **OLI (Oil Level Indicator)** : En français, Indicateur de niveau d'huile. Il affiche le niveau d'huile dans le transformateur, essentiel pour assurer une isolation et un refroidissement adéquats. Un niveau d'huile bas peut indiquer des fuites ou une consommation excessive due à une surchauffe.
- **OTIA (Oil Temperature Indicator Alarm)** : En français, Alarme de l'indicateur de température d'huile. Cette alarme se déclenche lorsqu'une température d'huile prédéterminée est dépassée, signalant potentiellement une surcharge ou un défaut de refroidissement.
- **OTIT (Oil Temperature Indicator Trip)** : En français, Déclenchement de l'indicateur de température d'huile. Si la température de l'huile atteint un seuil critique, ce dispositif peut déconnecter automatiquement le transformateur du réseau pour prévenir des dommages majeurs.
- **MOG\_A (Magnetic Oil Gauge Indicator)** : En français, Indicateur de jauge d'huile magnétique. Cet indicateur magnétique de niveau d'huile offre une

visualisation précise du niveau d'huile, permettant de détecter rapidement les problèmes de fuite ou les variations anormales du niveau d'huile.

### **5.3. Prétraitement des données**

Pour garantir une analyse précise et robuste, il est essentiel d'effectuer certaines opérations de nettoyage sur les données brutes collectées à partir des capteurs. Ces opérations de nettoyage représentent la première étape de notre processus d'apprentissage.

Avant de commencer le prétraitement des données, il est important de déterminer quelles tâches spécifiques de prétraitement sont requises pour nos données. Dans notre cas, après avoir évalué les données, nous avons identifié deux étapes cruciales : la normalisation et la gestion des données manquantes ou invalides lorsque cela est nécessaire.

La Normalisation des données est nécessaire quand l'incompatibilité des unités de mesures entre les variables peut affecter les résultats sans apporter d'interprétations claires, et recommandée chaque fois que les rapports finaux expriment les résultats dans des unités spécifiques significatives/compatibles. En ce qui concerne la gestion des données manquantes ou invalides, cela implique de traiter les valeurs manquantes en les remplaçant par des valeurs appropriées ou en supprimant les entrées invalides. Ces étapes sont essentielles pour garantir que notre ensemble de données soit propre, fiable et prêt à être utilisé dans nos analyses ultérieures.

### **5.4. Division des données**

Cette étape consiste à séparer les données en deux groupes : un pour l'entraînement (ensemble d'entraînement) et un pour les tests (ensemble de test). Cette séparation est cruciale en apprentissage automatique, car elle vise à mesurer avec précision la performance du modèle sur des données non vues auparavant, fournissant ainsi une évaluation fiable de sa capacité à appliquer ce qu'il a appris à de nouvelles

informations. Ce processus est essentiel pour vérifier la robustesse et l'efficacité du modèle au-delà de l'environnement contrôlé de l'ensemble de données initial utilisé pour son développement.

Pour cette étude spécifique, les données sont divisées de manière que 75 % servent à l'apprentissage du modèle et les 25 % restants sont utilisés pour évaluer sa précision.

## **5.5. Entraînement des algorithmes du Machine Learning**

Comme évoqué dans le chapitre précédent, notre étude s'attaque à une problématique de classification. En réponse, nous avons choisi d'appliquer une méthode d'apprentissage supervisé, en utilisant des algorithmes spécifiquement conçus pour la classification. Cette décision s'appuie sur la spécificité de notre problème : nous souhaitons répartir les données en classes distinctes, qui sont ici la présence ou l'absence de défaut.

Dans le cadre de l'apprentissage supervisé, l'algorithme est formé à partir d'un jeu de données annotées, lequel inclut des variables et des indicateurs (OTI, WTI, ATI, OLI, OTI\_A, OTI\_T).

Dans le cadre de l'apprentissage supervisé, l'algorithme assimile des connaissances à partir d'un ensemble de données étiqueté, composé de mesures et des indicateurs (OTI, WTI, ATI, OLI, OTI\_A, OTI\_T). Dans notre cas, la classe 0 est attribuée lorsque le défaut est absent, et la classe 1 lorsque le défaut est présent. L'algorithme est ainsi entraîné à reconnaître et à associer des patterns spécifiques, manifestés par les données aux classes de défauts appropriées.

L'indicateur MOG\_A joue un rôle déterminant dans ce processus. Il est spécifiquement responsable de signaler l'état de défaut, en affichant un '1' si un défaut est détecté, et un '0' en son absence. Grâce à cet indicateur, l'algorithme est capable de déduire et de prédire la présence d'un défaut, en s'appuyant sur l'apprentissage effectué.



L'indicateur MOG\_A est considéré comme la variable dépendante, tandis que les indicateurs OTI, WTI, ATI, OLI, OTI\_A, et OTI\_T sont considérés comme les variables indépendantes.

Cette phase d'entraînement permet à l'algorithme de généraliser ces relations, lui permettant de classer de manière autonome de nouvelles données non étiquetées en fonction de leurs caractéristiques.

Les algorithmes de Machine Learning choisis, pour la prédiction des défauts des transformateurs électriques, comprennent :

- Régression Logistique (Logistic Regression)
- Machine à Vecteurs de Support (Support Vector Machine)
- K-Nearest Neighbors (K-NN)
- Naive Bayes
- Arbre de Décision (Decision Tree)
- Forêt Aléatoire (Random Forest)
- AdaBoost
- Extreme Gradient Boosting (XGBoost)

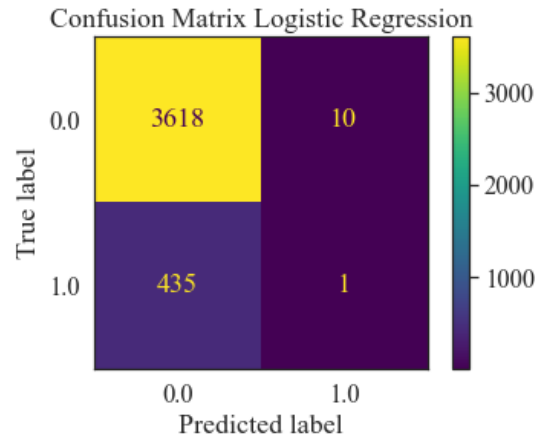
## **5.6. Résultats obtenus et discussion**

Les algorithmes de classification supervisé ont été sélectionnés en tenant compte de leurs capacités à traiter des données complexes, à fournir des prédictions précises et à gérer des dimensions importantes. Chaque algorithme a été entraîné sur 75% des données collectées et testé sur 25% des données et reçoit six entrées OTI, WTI, ATI, OLI, OTI\_A, et donne une sortie. La sortie du modèle prend une valeur soit 0 indiquant l'absence du défaut ou 1 sa présence.

Dans ce qui suit nous allons évaluer les résultats obtenus par les algorithmes après leurs entraînements ;

- **Régression Logistique (Logistic Regression)**

### Matrice de confusion :



**Figure 5.3 :** Matrice de confusion du classificateur de régression logistique.

La matrice de confusion pour le modèle de régression logistique affiche les résultats de classification pour deux classes, où la classe 0 est l'absence de défaut et la classe 1 est la présence de défaut. Voici l'interprétation des résultats :

- Vrais Négatifs (VN) : 3618 instances ont été correctement prédites comme ne présentant pas de défaut.
- Faux Positifs (FP) : 10 instances ont été incorrectement prédites comme présentant un défaut alors qu'elles n'en avaient pas.
- Faux Négatifs (FN) : 435 instances ont été incorrectement prédites comme ne présentant pas de défaut alors qu'elles en avaient un.
- Vrais Positifs (VP) : 1 instance a été correctement prédite comme présentant un défaut.

Ce modèle a une forte tendance à prédire l'absence de défaut, comme en témoignent les VN élevés et les VP très faibles. Le modèle a du mal à détecter les défauts (classe 1), ce qui est indiqué par un nombre élevé de FN et un très faible nombre de VP. Cela peut indiquer que le modèle est biaisé vers la classe la plus fréquente (classe 0) et qu'il n'est pas suffisamment sensible pour identifier la classe de défauts, ce qui est

crucial pour la prédiction des défaillances de transformateurs. Une telle performance n'est pas idéale pour un système de détection de défauts où le coût de ne pas détecter un défaut (FN) est généralement très élevé.

#### Évaluation des résultats obtenus :

Logistic Regression - Training Accuracy: 89.18%, Test Accuracy: 89.05%					
Classification Report:					
	precision	recall	f1-score	support	
0.0	0.89	1.00	0.94	3628	
1.0	0.09	0.00	0.00	436	
accuracy			0.89	4064	
macro avg	0.49	0.50	0.47	4064	
weighted avg	0.81	0.89	0.84	4064	

**Figure 5.4 :** Rapport de classification de l'algorithme Logistic Regression.

Le rapport de classification pour le modèle de régression logistique montre les performances suivantes :

- Pour la classe 0 (absence de défaut) :

Précision : 0.89, ce qui signifie que 89% des instances prédites comme n'ayant pas de défaut étaient correctement classées.

Rappel : 1.00, indiquant que le modèle a identifié correctement toutes les instances réelles.

Score F1 : 0.94, qui est la moyenne harmonique de la précision et du rappel, montrant une performance élevée pour cette classe.

- Pour la classe 1 (présence de défaut) :

Précision : 0.09, ce qui indique que seulement 9% des instances prédites comme ayant un défaut étaient correctement classées.

Rappel : 0.00, signifiant que le modèle n'a correctement identifié aucune des instances réelles présentant un défaut.

Score F1 : 0.00, indiquant une très faible performance pour cette classe.

Les mesures globales sont les suivantes :

Précision globale : 89.05%, cependant, cette mesure est trompeuse car elle est dominée par la performance sur la classe la plus fréquente (classe 0).

Moyenne macro :

Précision : 0.49, Rappel : 0.50 et Score F1 : 0.47, qui sont des moyennes non pondérées. Ces scores sont bas, reflétant la mauvaise performance sur la classe 1.

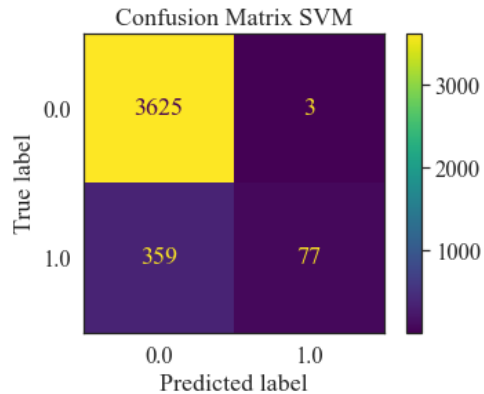
Moyenne pondérée :

Précision : 0.81, Rappel : 0.89 et Score F1 : 0.84, qui tiennent compte de la fréquence des classes. Ces chiffres sont plus élevés en raison de la forte représentation de la classe 0 dans les données.

Ce rapport indique que le modèle est très biaisé en faveur de la classe 0 et ne parvient pas à identifier correctement la classe 1. Dans le contexte de la prédiction de défauts de transformateurs, cette tendance est problématique car elle signifie que le modèle échoue à détecter la majorité des défauts réels, ce qui pourrait entraîner des conséquences graves dans une application réelle.

- **Machine à Vecteurs de Support (Support Vector Machine)**

**Matrice de confusion :**



**Figure 5.5 :** Matrice de confusion du classificateur de SVM.

La matrice de confusion présentée dans la figure illustre les résultats d'un modèle de classification utilisant une Machine à Vecteurs de Support (SVM) pour deux classes, où la classe 0 est l'absence de défaut et la classe 1 la présence de défaut. Voici l'interprétation des résultats :

- Vrais Négatifs (VN) : Le modèle a correctement prédit 3625 instances comme n'ayant pas de défaut (classe 0).
- Faux Positifs (FP) : 3 instances ont été incorrectement prédites comme ayant un défaut (classe 1) alors qu'elles n'en avaient pas (classe 0).
- Faux Négatifs (FN) : 359 instances ont été incorrectement prédites comme n'ayant pas de défaut (classe 0) alors qu'elles en avaient un (classe 1).
- Vrais Positifs (VP) : 77 instances ont été correctement prédites comme ayant un défaut (classe 1).

Cette matrice indique que le modèle SVM est très performant pour identifier les instances sans défaut, comme le montrent les VN élevés et les FP très faibles. Cependant, il a une performance relativement faible pour identifier les défauts, ce qui est illustré par un nombre élevé de FN. Bien que le nombre de VP soit faible, le modèle a quand même réussi à identifier correctement certains cas de défauts.

Dans un contexte où il est crucial de détecter les défauts (comme pour les transformateurs électriques), le nombre élevé de FN serait préoccupant car cela signifie que de nombreux défauts réels ne sont pas détectés, ce qui pourrait entraîner des échecs ou des dommages si ces défauts n'étaient pas identifiés et traités en temps opportun.

#### Évaluation des résultats obtenus :

SVM - Training Accuracy: 91.08%, Test Accuracy: 91.09%					
Classification Report:					
	precision	recall	f1-score	support	
0.0	0.91	1.00	0.95	3628	
1.0	0.96	0.18	0.30	436	
accuracy			0.91	4064	
macro avg	0.94	0.59	0.63	4064	
weighted avg	0.92	0.91	0.88	4064	

**Figure 5.6 :** Rapport de classification de l'algorithme SVM.

Le rapport de classification fourni montre les performances d'un modèle SVM (Machine à Vecteurs de Support) sur un ensemble de données de test. Voici l'interprétation des résultats :

Précision d'entraînement (training accuracy) et de test est d'environ 91 %, ce qui suggère que le modèle est stable et généralise bien.

Pour chaque classe :

- Classe 0 (absence de défaut) :

Précision : 0,91 signifie que 91 % des instances prédites étaient correctes.

Rappel : 1,00 indique que le modèle a identifié 100 % des instances réelles.

Score F1 : 0,95 est un score élevé qui indique une excellente performance pour la classe 0.

- Classe 1 (présence de défaut) :

Précision : 0,96 indique que 96 % des instances prédites étaient correctes.

Rappel : 0,18 montre que seulement 18 % des instances réelles défectueuses ont été identifiées par le modèle.

Score F1 : 0,30 est un score faible, reflétant une mauvaise performance pour la classe 1, surtout en termes de rappel.

Les moyennes globales :

Précision micro et F1-score micro : 0,91 indique la performance globale sur toutes les instances.

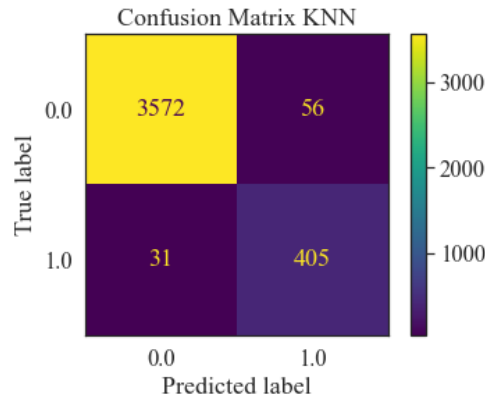
Moyenne macro de la précision et du rappel : 0,94 et 0,59 respectivement, ce qui indique que le modèle est plus performant en précision qu'en rappel, en particulier pour la classe 1.

Moyenne pondérée de la précision et du rappel : 0,92 et 0,91 respectivement, ce qui montre que la classe majoritaire (classe 0) influence significativement ces scores en raison de son grand support.

En résumé, le modèle SVM est très précis pour prédire les instances non défectueuses mais a du mal à détecter correctement les instances défectueuses, comme le démontre le faible rappel pour la classe 1. Cela peut être problématique dans des applications pratiques où la détection des défauts est cruciale.

- **K-Nearest Neighbors (K-NN)**

**Matrice de confusion :**



**Figure 5.7 :** Matrice de confusion du classificateur de KNN.

La matrice de confusion pour le modèle K-Nearest Neighbors (KNN) présente les résultats suivants :

- Vrais Négatifs (VN) : Le modèle a correctement identifié 3572 instances comme étant de la classe 0, ce qui signifie qu'elles ne présentent pas de défaut.
- Faux Positifs (FP) : 56 instances ont été incorrectement classées par le modèle comme étant de la classe 1 (ayant un défaut) alors qu'en réalité, elles sont de la classe 0.
- Faux Négatifs (FN) : 31 instances qui appartiennent réellement à la classe 1 ont été incorrectement classées comme appartenant à la classe 0.
- Vrais Positifs (VP) : 405 instances ont été correctement identifiées comme appartenant à la classe 1.

Cette matrice de confusion indique que le modèle KNN est très performant pour identifier les instances sans défaut et présente également une bonne capacité à détecter les instances avec défaut, comme en témoignent les vrais positifs élevés. Le nombre de faux positifs et de faux négatifs est relativement faible, ce qui signifie que le modèle a une bonne précision et un bon rappel pour les deux classes.

Dans un contexte pratique, le nombre réduit de faux négatifs est particulièrement important car il indique que le modèle est capable de détecter la majorité des défauts



réels, ce qui est essentiel pour éviter les conséquences potentiellement graves des défauts non détectés dans les transformateurs électriques.

#### Évaluation des résultats obtenus :

KNN - Training Accuracy: 97.58%, Test Accuracy: 97.86%					
Classification Report:					
	precision	recall	f1-score	support	
0.0	0.99	0.98	0.99	3628	
1.0	0.88	0.93	0.90	436	
accuracy			0.98	4064	
macro avg	0.93	0.96	0.95	4064	
weighted avg	0.98	0.98	0.98	4064	

**Figure 5.8 :** Rapport de classification de l'algorithme KNN.

Le rapport de classification pour le modèle K-Nearest Neighbors (KNN) montre les performances suivantes :

La précision d'entraînement (Training Accuracy) est d'environ 97.58 %, et celle de test est d'environ 97.86 %, ce qui suggère que le modèle est stable et généralise bien.

Pour chaque classe :

- Classe 0 (absence de défaut) :

Précision : 0,99, ce qui signifie que 99% des prédictions étaient correctes.

Rappel : 0,98, indiquant que le modèle a correctement identifié 98% des instances réelles.

Score F1 : 0,99, montrant une performance quasi parfaite pour la classe 0.

- Classe 1 (présence de défaut) :

Précision : 0,88, indiquant que 88% des prédictions faites par le modèle étaient correctes.

Rappel : 0,93, montrant que le modèle a correctement identifié 93% des instances réelles.

Score F1 : 0,90, reflétant une très bonne performance pour la classe 1.

Moyenne micro :

Précision : 0,98, Rappel : 0,98 et Score F1 : 0,98, reflétant la performance globale du modèle sur toutes les instances.

Moyenne macro :

Précision : 0,93, Rappel : 0,96 et Score F1 : 0,95, indiquant une performance moyenne équilibrée sur toutes les classes.

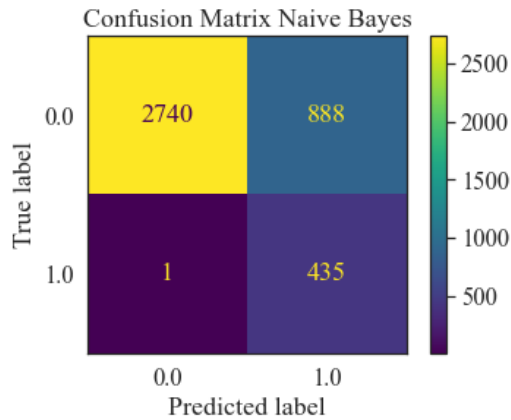
Moyenne pondérée :

Précision : 0,98, Rappel : 0,98 et Score F1 : 0,98, tenant compte de la fréquence de chaque classe.

Ce rapport indique que le modèle KNN a très bien performé, non seulement sur l'ensemble d'entraînement mais également sur l'ensemble de test, ce qui démontre sa capacité à généraliser à de nouvelles données. Il est particulièrement efficace pour la classification des instances sans défaut et montre également une bonne capacité à identifier les défauts.

- **Naïve Bayes**

**Matrice de confusion :**



**Figure 5.9 :** Matrice de confusion du classificateur de Naïve Bayes.

La matrice de confusion pour le modèle Naïve Bayes affiche les résultats suivants pour deux classes :

- Vrais Négatifs (VN) : 2740 instances ont été correctement prédites comme n'ayant pas de défaut (classe 0).
- Faux Positifs (FP) : 888 instances ont été incorrectement prédites comme ayant un défaut (classe 1) alors qu'elles n'en avaient pas.
- Faux Négatifs (FN) : 1 instance a été incorrectement prédite comme n'ayant pas de défaut alors qu'elle en avait un.
- Vrais Positifs (VP) : 435 instances ont été correctement prédites comme ayant un défaut.

Cette matrice de confusion indique que le modèle a une bonne capacité à détecter les vrais positifs, c'est-à-dire à identifier correctement les instances présentant un défaut. Cependant, le modèle présente un nombre élevé de faux positifs, ce qui peut être problématique car cela signifie que de nombreuses instances sont faussement identifiées comme défectueuses. Cela peut conduire à des coûts d'inspection et de maintenance inutiles si ces résultats sont utilisés dans un contexte pratique. Le très faible nombre de faux négatifs est positif car il montre que le modèle manque

rarement les défauts réels, ce qui est crucial pour éviter les conséquences graves d'un défaut non détecté.

#### Évaluation des résultats obtenus :

```
Naive Bayes - Training Accuracy: 78.31%, Test Accuracy: 78.12%
Classification Report:
```

	precision	recall	f1-score	support
0.0	1.00	0.76	0.86	3628
1.0	0.33	1.00	0.49	436
accuracy			0.78	4064
macro avg	0.66	0.88	0.68	4064
weighted avg	0.93	0.78	0.82	4064

**Figure 5.10 :** Rapport de classification de l'algorithme Naïve Bayes.

Le rapport de classification pour le modèle Naïve Bayes affiche les performances suivantes :

Précision d'entraînement (Training Accuracy) : 78.31%

Précision de test (Test accuracy) : 78.12%

Pour chaque classe :

- Classe 0 (absence de défaut) :

Précision : 1.00, signifiant que toutes les instances prédites étaient correctes.

Rappel : 0.76, indiquant que le modèle a correctement identifié 76% des instances réelles.

Score F1 : 0.86, reflétant une performance assez élevée pour la classe 0.

- Classe 1 (présence de défaut) :

Précision : 0.33, indiquant que seulement 33% des instances prédites étaient correctes.

Rappel : 1.00, montrant que le modèle a identifié toutes les instances réelles défectueuses.

Score F1 : 0.49, reflétant une performance médiocre pour la classe 1, due à la faible précision.

Les mesures globales :

Précision globale : 0.78, reflétant la proportion d'instances correctement prédites par le modèle sur l'ensemble de données de test.

Moyenne macro :

Précision : 0.66, Rappel : 0.88 et Score F1 : 0.68, des scores moyens qui n'accordent pas d'importance à la répartition des classes dans l'ensemble des données. Ces scores sont biaisés positivement par le rappel parfait de la classe 1 mais affectés négativement par la faible précision pour cette classe.

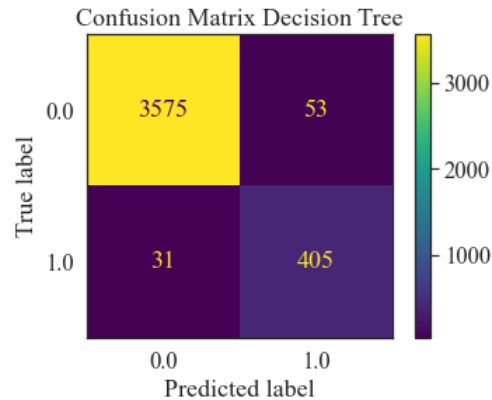
Moyenne pondérée :

Précision : 0.93, Rappel : 0.78 et Score F1 : 0.82, pondérés par le support de chaque classe, donnant plus de poids à la performance du modèle sur la classe 0, qui est plus fréquente.

Le modèle montre une tendance à bien détecter les instances de la classe 1 (présence de défaut) mais avec un taux élevé de fausses alarmes, comme le montre la faible précision pour cette classe. Cela indique que, bien que le modèle identifie correctement tous les cas positifs, il produit également un grand nombre de fausses alertes, ce qui pourrait entraîner des coûts d'inspection et de maintenance inutiles dans une application pratique.

- **Arbre de Décision (Decision Tree)**

**Matrice de confusion :**



**Figure 5.11 :** Matrice de confusion du classificateur des arbres de décisions.

La matrice de confusion pour le modèle d'arbre de décision présente les résultats suivants :

- Vrais Négatifs (VN) : 3575 instances ont été correctement identifiées comme n'ayant pas de défaut (classe 0).
- Faux Positifs (FP) : 53 instances ont été incorrectement classées comme ayant un défaut (classe 1) alors qu'elles n'en avaient pas.
- Faux Négatifs (FN) : 31 instances ayant un défaut (classe 1) ont été incorrectement classées comme n'ayant pas de défaut.
- Vrais Positifs (VP) : 405 instances ont été correctement identifiées comme ayant un défaut.

Cette matrice indique que le modèle d'arbre de décision est assez efficace pour identifier les instances sans défaut et a également une bonne capacité à détecter les instances avec défaut, comme en témoignent les vrais positifs élevés. Le nombre de faux positifs et de faux négatifs est relativement faible, ce qui indique que le modèle a une bonne précision et un bon rappel pour les deux classes.

**Évaluation des résultats obtenus :**

```

Decision Tree - Training Accuracy: 97.76%, Test Accuracy: 97.93%
Classification Report:

```

	precision	recall	f1-score	support
0.0	0.99	0.99	0.99	3628
1.0	0.88	0.93	0.91	436
accuracy			0.98	4064
macro avg	0.94	0.96	0.95	4064
weighted avg	0.98	0.98	0.98	4064

**Figure 5.12 :** Rapport de classification de l'algorithme arbres de décisions.

Le rapport de classification pour le modèle d'arbre de décision montre les performances suivantes :

Précision d'entraînement (Training Accuracy) 97.76%

Précision du test (test accuracy): 97.93%

Pour chaque classe :

- Classe 0 (absence de défaut) :

Précision : 0.99, ce qui signifie que 99% des prédictions de la classe 0 étaient correctes.

Rappel : 0.99, indiquant que le modèle a identifié 99% des instances réelles de la classe 0.

Score F1 : 0.99, reflétant une excellente performance pour cette classe.

- Classe 1 (présence de défaut) :

Précision : 0.88, signifiant que 88% des prédictions de la classe 1 étaient correctes.

Rappel : 0.93, montrant que le modèle a identifié 93% des instances réelles de la classe 1.

Score F1 : 0.91, reflétant une très bonne performance pour cette classe.

Moyenne macro :

Précision : 0.94, Rappel : 0.96 et Score F1 : 0.95, qui sont des moyennes non pondérées sur toutes les classes, indiquant une performance uniforme et élevée sur les deux classes.

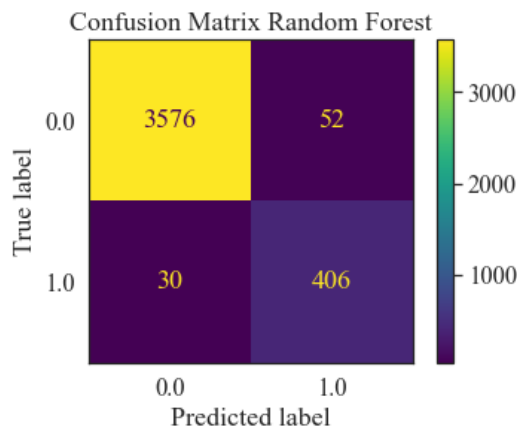
Moyenne pondérée :

Précision : 0.98, Rappel : 0.98 et Score F1 : 0.98, qui tiennent compte de la fréquence de chaque classe, indiquant une performance élevée globale, pondérée en fonction du support de chaque classe.

Ces résultats indiquent que le modèle d'arbre de décision est très performant sur l'ensemble de données, avec une capacité particulièrement forte à identifier les instances non défectueuses (classe 0) et également une très bonne capacité à détecter les instances défectueuses (classe 1). La haute précision et le rappel pour la classe 1 sont notables, car cela signifie que le modèle est à la fois précis et sensible dans la détection des défauts.

#### - Forêt Aléatoire (Random Forest)

**Matrice de confusion :**



**Figure 5.13 :** Matrice de confusion du classificateur de Forêt aléatoire.



La matrice de confusion pour le modèle de forêt aléatoire (Random Forest) présente les résultats suivants :

- Vrais Négatifs (VN) : 3576 instances ont été correctement identifiées comme n'ayant pas de défaut (classe 0).
- Faux Positifs (FP) : 52 instances ont été incorrectement classées comme ayant un défaut (classe 1) alors qu'elles n'en avaient pas.
- Faux Négatifs (FN) : 30 instances ayant un défaut (classe 1) ont été incorrectement classées comme n'ayant pas de défaut.
- Vrais Positifs (VP) : 406 instances ont été correctement identifiées comme ayant un défaut.

Cette matrice de confusion indique que le modèle de forêt aléatoire est très efficace pour identifier les instances sans défaut (classe 0) et possède également une bonne capacité de détection des instances avec défaut (classe 1), comme en témoigne le nombre élevé de vrais positifs. Le nombre relativement bas de faux positifs et de faux négatifs signifie que le modèle a une bonne précision et un bon rappel pour les deux classes.

La capacité du modèle à minimiser les faux négatifs est particulièrement importante dans des situations où il est crucial de détecter tous les défauts réels, comme dans le contrôle de qualité des équipements industriels ou la maintenance préventive, pour éviter les pannes ou les défaillances. Un nombre faible de faux positifs est également avantageux pour réduire les coûts inutiles et les désagréments liés à des inspections supplémentaires inutiles.

**Évaluation des résultats obtenus :**

Random Forest - Training Accuracy: 97.75%, Test Accuracy: 97.98%				
Classification Report:				
	precision	recall	f1-score	support
0.0	0.99	0.99	0.99	3628
1.0	0.89	0.93	0.91	436
accuracy			0.98	4064
macro avg	0.94	0.96	0.95	4064
weighted avg	0.98	0.98	0.98	4064

**Figure 5.14 :** Rapport de classification de l'algorithme Forêt aléatoire.

Le rapport de classification pour le modèle de forêt aléatoire (Random Forest) montre les performances suivantes :

Précision d'entraînement (training accuracy): 97.75%

Précision du test (test accuracy): 97.98%

Pour chaque classe :

- Classe 0 (absence de défaut) :

Précision : 0.99, signifiant que 99% des instances prédites comme non défectueuses étaient correctes.

Rappel : 0.99, indiquant que le modèle a identifié 99% des instances réelles de la classe 0.

Score F1 : 0.99, reflétant une performance presque parfaite pour cette classe.

- Classe 1 (présence de défaut) :

Précision : 0.89, signifiant que 89% des prédictions de la classe 1 étaient correctes.

Rappel : 0.93, montrant que le modèle a identifié 93% des instances réelles de la classe 1.

Score F1 : 0.91, indiquant une très bonne performance pour la classe 1.

Moyenne macro :

Précision : 0.94, Rappel : 0.96 et Score F1 : 0.95, qui sont des moyennes non pondérées sur toutes les classes, indiquant une performance uniforme et élevée sur les deux classes.

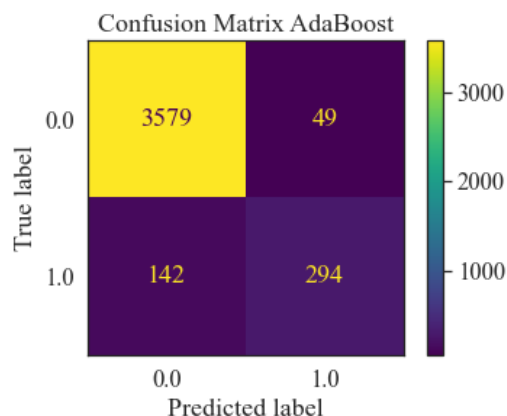
Moyenne pondérée :

Précision : 0.98, Rappel : 0.98 et Score F1 : 0.98, qui tiennent compte de la fréquence de chaque classe, indiquant une performance élevée globale, pondérée en fonction du support de chaque classe.

Ces résultats indiquent que le modèle de forêt aléatoire est très performant sur l'ensemble de données, avec une capacité particulièrement forte à identifier les instances non défectueuses (classe 0) et également une très bonne capacité à détecter les instances défectueuses (classe 1). La haute précision et le rappel pour la classe 1 sont notables, car cela signifie que le modèle est à la fois précis et sensible dans la détection des défauts.

#### - AdaBoost

**Matrice de confusion :**



**Figure 5.15 :** Matrice de confusion du classificateur de AdaBoost.

La matrice de confusion pour le modèle AdaBoost présente les résultats suivants pour la classification :

- Vrais Négatifs (VN) : 3579 instances ont été correctement identifiées comme n'ayant pas de défaut (classe 0).
- Faux Positifs (FP) : 49 instances ont été incorrectement prédites comme ayant un défaut (classe 1) alors qu'elles n'en avaient pas.
- Faux Négatifs (FN) : 142 instances ayant un défaut (classe 1) ont été incorrectement classées comme n'ayant pas de défaut.
- Vrais Positifs (VP) : 294 instances ont été correctement identifiées comme ayant un défaut.

Cette matrice de confusion indique que le modèle AdaBoost est très performant pour identifier les instances de la classe 0 mais montre des limitations dans la détection de la classe 1. Le nombre de vrais positifs est inférieur à celui des faux négatifs, ce qui suggère que le modèle peut manquer une proportion significative de cas positifs réels (défauts). Cette tendance pourrait être préoccupante dans des situations où il est essentiel de détecter tous les défauts, tels que la maintenance préventive dans l'industrie, car elle pourrait entraîner des défaillances non détectées et donc des coûts ou des conséquences plus graves.

#### Évaluation des résultats obtenus :

AdaBoost - Training Accuracy: 95.23%, Test Accuracy: 95.3%				
Classification Report:				
	precision	recall	f1-score	support
0.0	0.96	0.99	0.97	3628
1.0	0.86	0.67	0.75	436
accuracy			0.95	4064
macro avg	0.91	0.83	0.86	4064
weighted avg	0.95	0.95	0.95	4064

**Figure 5.16 :** Rapport de classification de l'algorithme AdaBoost.

Le rapport de classification pour le modèle AdaBoost montre les performances suivantes :

Précision d'entraînement (training accuracy) : 95.23%

Précision de test (test accuracy) : 95.3%

Pour chaque classe :

- Classe 0 (absence de défaut) :

Précision : 0.96, ce qui signifie que 96% des prédictions de la classe 0 par le modèle étaient correctes.

Rappel : 0.99, indiquant que le modèle a identifié 99% des instances réelles de la classe 0.

Score F1 : 0.97, montrant une performance exceptionnelle pour la classe 0.

- Classe 1 (probablement présence de défaut) :

Précision : 0.86, signifiant que 86% des prédictions de la classe 1 par le modèle étaient correctes.

Rappel : 0.67, montrant que le modèle a identifié 67% des instances réelles de la classe 1.

Score F1 : 0.75, reflétant une bonne performance pour la classe 1 mais indiquant une marge d'amélioration, notamment en termes de rappel.

Moyenne macro :

Précision : 0.91, Rappel : 0.83 et Score F1 : 0.86, qui sont des moyennes non pondérées sur toutes les classes. Cela indique une performance globale solide mais suggère que le modèle peut être amélioré, en particulier pour la classe 1.

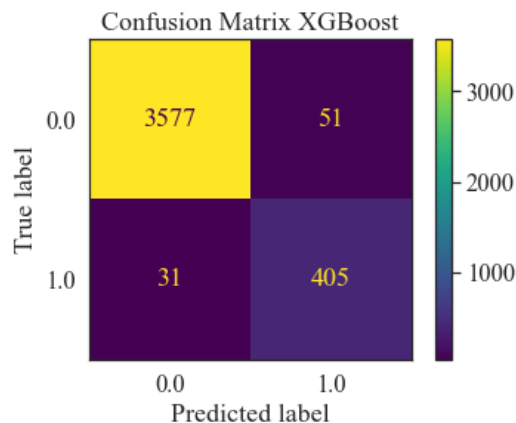
Moyenne pondérée :

Précision : 0.95, Rappel : 0.95 et Score F1 : 0.95, tenant compte de la fréquence de chaque classe et montrant une performance globale élevée.

En résumé, le modèle AdaBoost montre une précision globale élevée et est particulièrement efficace pour identifier les instances de la classe 0. Cependant, bien que la précision pour la classe 1 soit élevée, le rappel et le score F1 indiquent que le modèle pourrait être amélioré pour mieux détecter les instances réelles de la classe 1, ce qui est essentiel pour éviter de manquer des défauts potentiels dans les applications pratiques.

#### - Extreme Gradient Boosting (XGBoost)

##### Matrice de confusion :



**Figure 5.17 :** Matrice de confusion du classificateur de XGBoost.

La matrice de confusion pour le modèle XGBoost affiche les résultats suivants pour la classification binaire :

- Vrais Négatifs (VN) : 3577 instances ont été correctement prédites comme n'appartenant pas à la classe positive (classe 0 - absence de défaut).
- Faux Positifs (FP) : 51 instances ont été incorrectement prédites comme appartenant à la classe positive (classe 1 - présence de défaut) alors qu'elles n'en faisaient pas partie.

- Faux Négatifs (FN) : 31 instances appartenant à la classe positive ont été incorrectement prédites comme appartenant à la classe négative.
- Vrais Positifs (VP) : 405 instances ont été correctement prédites comme appartenant à la classe positive.

Cette matrice indique que le modèle XGBoost est très performant pour détecter les instances négatives (absence de défaut) avec une très haute valeur de vrais négatifs et une faible valeur de faux positifs. Pour la détection de la classe positive (présence de défaut), le modèle a également bien performé, avec un nombre significatif de vrais positifs et un nombre relativement faible de faux négatifs.

En termes d'application pratique, surtout dans des domaines où il est crucial de détecter tous les cas positifs, comme la maintenance prédictive ou la sécurité, le modèle XGBoost montre une bonne sensibilité à la classe positive tout en conservant une spécificité élevée pour la classe négative. Cela suggère qu'il pourrait être un outil fiable pour la prédiction de défauts, minimisant le risque de manquer des défauts réels tout en évitant un trop grand nombre de fausses alarmes.

#### Évaluation des résultats obtenus :

XGBoost - Training Accuracy: 97.67%, Test Accuracy: 97.98%				
Classification Report:				
	precision	recall	f1-score	support
0.0	0.99	0.99	0.99	3628
1.0	0.89	0.93	0.91	436
accuracy			0.98	4064
macro avg	0.94	0.96	0.95	4064
weighted avg	0.98	0.98	0.98	4064

**Figure 5.18 :** Rapport de classification de l'algorithme XGBoost.

Le rapport de classification pour le modèle XGBoost montre les performances suivantes :

Précision d'entraînement (training accuracy): 97.67%

Précision du test (test accuracy): 97.98%

Pour chaque classe :

- Classe 0 (absence de défaut) :

Précision : 0.99, ce qui signifie que 99% des instances prédites comme non défectueuses étaient correctes.

Rappel : 0.99, indiquant que le modèle a identifié 99% des instances réelles de la classe 0.

Score F1 : 0.99, reflétant une performance exceptionnelle pour cette classe.

- Classe 1 (présence de défaut) :

Précision : 0.89, signifiant que 89% des prédictions de la classe 1 étaient correctes.

Rappel : 0.93, montrant que le modèle a identifié 93% des instances réelles de la classe 1.

Score F1 : 0.91, reflétant une très bonne performance pour la classe 1.

Moyenne macro :

Précision : 0.94, Rappel : 0.96 et Score F1 : 0.95, qui sont des moyennes non pondérées sur toutes les classes. Cela indique une performance uniforme et élevée sur les deux classes.

Moyenne pondérée :

Précision : 0.98, Rappel : 0.98 et Score F1 : 0.98, qui tiennent compte de la fréquence de chaque classe, indiquant une performance élevée globale.

Ces résultats indiquent que le modèle XGBoost est très performant sur l'ensemble de données, avec une capacité particulièrement forte à identifier les instances non

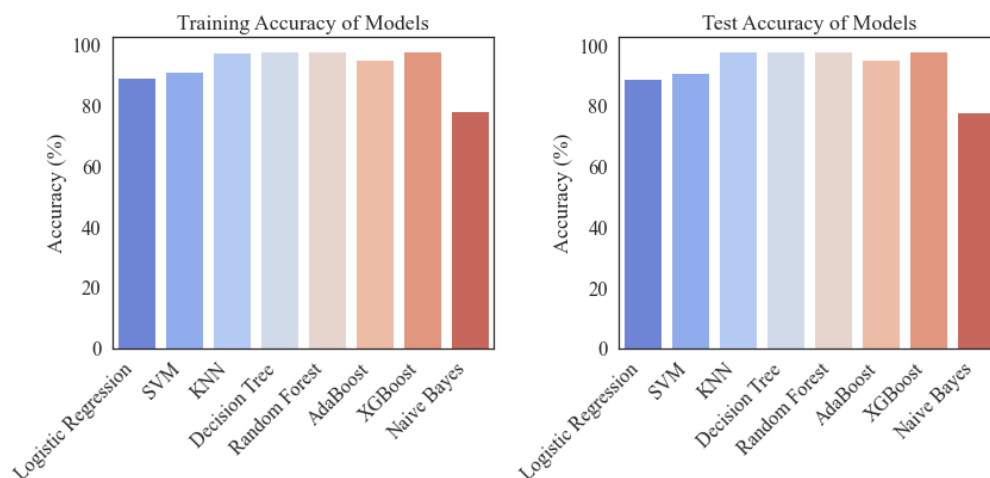


défectueuses (classe 0) et également une très bonne capacité à détecter les instances défectueuses (classe 1). La haute précision et le rappel pour la classe 1 sont notables, car cela signifie que le modèle est à la fois précis et sensible dans la détection des défauts.

## 5.7. Évaluation globale des algorithmes

Model	Training Accuracy	Test Accuracy ▼
Random Forest	97.75	97.98
XGBoost	97.67	97.98
Decision Tree	97.76	97.93
KNN	97.58	97.86
AdaBoost	95.23	95.3
SVM	91.08	91.09
Logistic Regression	89.18	89.05
Naïve Bayes	78.31	78.12

**Figure 5.19 :** Comparaison de l'accuracy des tests de modèles dans la classification par apprentissage automatique.



**Figure 5.20 :** Comparaison des modèles de Machine Learning.

Le tableau fourni récapitule la précision d'entraînement et de test de différents modèles de machine learning. Voici une interprétation et une comparaison des résultats :

- **Random Forest :**

Accuracy d'entraînement : 97.75%

Accuracy de test : 97.98%

Random Forest montre un bon ajustement du modèle et une excellente généralisation avec des scores presque identiques pour l'entraînement et le test.

- **XGBoost**

Accuracy d'entraînement : 97.67%

Accuracy de test : 97.98%

XGBoost, tout comme Random Forest, offre d'excellentes performances avec une accuracy de test très élevé.

- **Decision Tree**

Accuracy d'entraînement : 97.76%

Accuracy de test : 97.93%

L'arbre de décision affiche également des scores de précision élevés, légèrement en deçà de Random Forest et XGBoost.

- **KNN (K-Nearest Neighbors)**

Accuracy d'entraînement : 97.58%

Accuracy de test : 97.86%

KNN performe très bien, avec une légère différence entre l'entraînement et le test.

- **AdaBoost**

Accuracy d'entraînement : 95.23%

Accuracy de test : 95.3%

AdaBoost montre une bonne performance, bien que légèrement inférieure aux modèles précédents.

- **SVM (Support Vector Machine)**

Accuracy d'entraînement : 91.08%

Accuracy de test : 91.09%

SVM présente des performances correctes mais notablement inférieures par rapport aux modèles basés sur des arbres.

- **Logistic Regression**

Accuracy d'entraînement : 89.18%

Accuracy de test : 89.05%

La régression logistique affiche des performances modérées, suggérant des difficultés potentielles à capturer la complexité des données.

- **Naïve Bayes**

Accuracy d'entraînement : 78.31%

Accuracy de test : 78.12%

Naive Bayes montre la plus faible performance parmi tous les modèles présentés, ce qui peut refléter l'impact négatif de l'hypothèse d'indépendance sur cet ensemble de données spécifique.

En comparaison, les modèles basés sur des arbres (Random Forest, XGBoost et Decision Tree) offrent les meilleures performances en termes de test Accuracy, ce qui suggère qu'ils sont mieux adaptés pour cet ensemble de données spécifique. Ils parviennent à capturer la complexité des données et à généraliser à de nouvelles données non vues lors de l'entraînement. Les modèles tels que Naïve Bayes et Logistic Regression, qui font des hypothèses plus simplistes, semblent être moins performants sur cet ensemble de données.

## **5.8. Conclusion**

En conclusion de ce chapitre, il ressort que les modèles fondés sur les arbres de décision, à savoir Random Forest, XGBoost et Decision Tree, se distinguent par leur supériorité en matière de précision des tests. Ces modèles ont montré une aptitude remarquable à généraliser leurs prédictions aux données inédites, ce qui reflète leur adaptabilité et leur robustesse. À l'opposé, des modèles tels que Naïve Bayes et la régression logistique, reposant sur des postulats plus élémentaires, n'ont pas atteint le même niveau de performance. Cela souligne l'importance de choisir une méthode d'apprentissage adaptée à la nature et à la complexité des données traitées pour optimiser l'exactitude des prévisions.

## **Chapitre 6**

# **Prédiction des défauts des transformateurs électriques en utilisant l'apprentissage profond**

### **6.1. Introduction**

Cette section se focalisera sur la prévision des pannes de transformateurs électriques en exploitant les techniques d'apprentissage en profondeur (Deep Learning), et en particulier les réseaux de neurones artificiels (RNA). Pour ce faire, nous réutiliserons les ensembles de données examinés dans les sections antérieures de l'étude. L'objectif principal ici sera de mettre en évidence les différences de performance entre les modèles d'apprentissage automatique (Machine Learning), l'apprentissage ensembliste (Ensemble learning) et les modèles avancés d'apprentissage profond, en établissant une comparaison approfondie entre ces deux approches en termes de précision, de rapidité et de fiabilité dans la détection des défaillances potentielles.

### **6.2. Prédiction des défauts des transformateurs en utilisant des données de courant et de tension**

#### **6.2.1. Base de données**

Après la construction du modèle électrique (Figure 4.2), ce dernier a été employé pour générer notre base de données. Cette base inclut des mesures triphasées de tension ( $V_A, V_B, V_C$ ) et de courant ( $I_A, I_B, I_C$ ), recueillies sur un bus de mesure triphasé, sous des conditions normales et en cas de défaut. La collecte de données a permis de rassembler 3000 instances, couvrant une large gamme de types de défauts pour améliorer l'efficacité de l'apprentissage des modèles en intégrant diverses

situations et scénarios qui reflètent le fonctionnement réel des systèmes électriques. Ces informations sont compilées dans un fichier Excel nommé « Dataset », facilitant ainsi leur manipulation.

### **6.2.2. Division de la base de données**

La base de données créée est répartie en trois segments distincts : 70% des données sont utilisées pour l'entraînement des réseaux de neurones artificiels (ANN), désignées sous le terme de données d'entraînement (training set), tandis que 15% sont attribuées à la validation (validation set) et les 15% restants sont consacrés aux tests (test set). Il est à noter que ces proportions sont susceptibles de varier en fonction des spécificités de chaque projet.

Les modèles de réseaux de neurones artificiels (ANN) sont effectivement susceptibles de surapprentissage (overfitting), surtout lorsqu'ils sont entraînés sur un grand nombre d'époques ou avec un ensemble de données d'entraînement très détaillé. Pour contrer ce phénomène, l'utilisation d'un ensemble de validation devient cruciale.

**L'ensemble de validation** permet d'évaluer la performance du modèle sur des données qu'il n'a pas utilisées pendant l'entraînement, sans pour autant toucher à l'ensemble de test, qui doit rester indépendant et servir uniquement à l'évaluation finale du modèle. Cette pratique aide à ajuster les hyperparamètres du modèle et à détecter le surapprentissage à un stade précoce.

#### **Objectif de l'ensemble de validation**

- Mise au point et sélection du modèle : L'ensemble de validation est utilisé pour évaluer un modèle pendant la phase d'entraînement, ce qui permet d'affiner les hyperparamètres (par exemple, le taux d'apprentissage, l'architecture d'un réseau neuronal) et de prendre des décisions sur le modèle ou la configuration de modèle le plus performant sans exposer le modèle à l'ensemble de test.

- Prévention du surajustement : Il permet de détecter les ajustements excessifs. En évaluant les performances du modèle sur un ensemble de données qui n'est pas utilisé pour l'entraînement, nous pouvons obtenir une indication de la manière dont le modèle se généralise aux données non vues.

### 6.2.3. Prétraitement des données

Dans l'apprentissage profond, la normalisation des données est une pratique fortement recommandée. La normalisation des données implique le processus de mise à l'échelle des données d'entrée pour qu'elles aient une plage de valeurs standard, souvent avec une moyenne de zéro et une variance d'un. Ce processus aide à améliorer la convergence du modèle pendant l'entraînement et à augmenter la stabilité globale du processus d'apprentissage.

### 6.2.4. Architecture du RNA

Dans cette étude, le réseau de neurones artificiels (RNA) est conçu avec une architecture spécifique qui comprend plusieurs couches, chacune ayant un nombre défini de neurones ou d'unités. Voici une description détaillée de la structure du réseau :

**Couche d'entrée :** Elle contient 6 neurones. Cette couche est la première étape du réseau, où chaque neurone correspond à une caractéristique différente de l'ensemble de données d'entrée. L'objectif de cette couche est de recevoir les signaux d'entrée, dans notre cas le courant et la tension, et de les transmettre aux couches suivantes pour un traitement ultérieur.

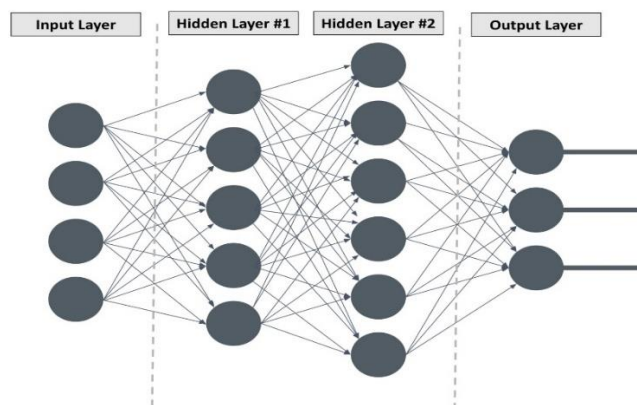
**Première couche cachée :** Cette couche suit immédiatement la couche d'entrée et comprend 26 neurones. Les couches cachées sont au cœur des réseaux de neurones, permettant au modèle d'apprendre des représentations complexes des données d'entrée grâce à des poids et des biais ajustables. Les neurones de cette couche vont

traiter les informations reçues de la couche d'entrée, appliquer une fonction d'activation (comme ReLU = Rectifier Linear Unit), et transmettre le résultat à la couche suivante.

**Deuxième couche cachée :** Elle contient 16 neurones et fonctionne de manière similaire à la première couche cachée, en prenant les sorties de celle-ci comme entrées et en appliquant une fonction d'activation, dans notre cas, ReLU = Rectifier Linear Unit. Cette couche permet au réseau de capturer encore plus de complexité et de nuances dans les données, facilitant l'apprentissage de relations plus abstraites et profondes.

**Couche de sortie :** La dernière couche du réseau a 1 neurone et est utilisée pour produire le résultat final du modèle. Pour une tâche de classification binaire, dans notre cas, une fonction d'activation sigmoid est appliquée à cette couche pour produire une probabilité entre 0 et 1, indiquant l'appartenance à une classe.

Cette architecture de réseau permet de traiter efficacement les données d'entrée à travers les différentes couches et neurones, chaque couche capturant des niveaux de complexité croissants, pour finalement aboutir à une prédiction ou classification précise à la couche de sortie. La présence de couches cachées multiples (dans ce cas, deux couches de 26 et 16 neurones respectivement) est cruciale pour permettre au réseau d'apprendre des caractéristiques non linéaires et des interactions complexes entre les variables d'entrée.



**Figure 6.1 :** Architecture d'un réseau de neurones artificiels.



### 6.2.5. Entraînement du RNA

Pendant l'entraînement d'un réseau neuronal artificiel (ANN), plusieurs étapes clés se produisent. Cette étape est très importante car l'ajustement des poids du modèle en fonction des données d'apprentissage est appliqué, dans le but de minimiser la fonction de perte (Loss Function). Voici un aperçu de ce qui se passe exactement pendant la phase de formation :

#### **Compilation du modèle**

Durant la phase de la préparation d'un modèle de réseau de neurones artificiels pour l'apprentissage, il est crucial de faire la configuration nécessaire. Cela se fait lors d'une étape appelée compilation du modèle, où l'on définit comment le modèle doit apprendre. Voici les éléments clés à spécifier :

- L'optimiseur 'Adam' : Une méthode avancée pour ajuster les poids du modèle, permettant un apprentissage efficace en modifiant intelligemment le taux d'apprentissage au fil du temps pour une meilleure et plus rapide convergence.
- La fonction de perte 'binary\_crossentropy' : Pour les problèmes de classification binaire, cette fonction aide à mesurer à quel point les prédictions du modèle sont éloignées des vraies valeurs. L'objectif est de réduire cette différence au minimum.
- Les métriques d'évaluation (précision et erreur quadratique moyenne) : Permettent de suivre et d'évaluer la performance du modèle.

#### **Entraînement du modèle**

Dans le processus d'entraînement d'un modèle de réseau de neurones artificiels, les données d'apprentissage sont examinées en petits groupes nommés "lots". À chaque traitement de lot, le modèle effectue des prédictions basées sur ses connaissances actuelles, qu'on pourrait assimiler à des "suppositions". Ces prédictions sont ensuite comparées aux réponses réelles pour mesurer l'exactitude du modèle. Si des erreurs sont détectées, le modèle les utilise pour s'améliorer grâce à un mécanisme connu sous le nom de "rétropropagation". Cette étape lui permet de s'ajuster et d'améliorer

ses performances pour les prédictions futures. Pour s'assurer que le modèle ne se contente pas de mémoriser les données d'apprentissage mais est également capable de généraliser son apprentissage à de nouvelles données, il est périodiquement évalué sur un ensemble de validation distinct (15 % dans notre cas). Ce processus de vérification, d'apprentissage, et d'ajustement se répète à travers plusieurs itérations, appelées "époques". Dans cette étude, nous avons choisi de réaliser 100 époques. L'entraînement se conclut idéalement lorsque le modèle démontre une capacité à bien prédire non seulement sur les données d'entraînement mais aussi sur les données de validation, indiquant ainsi qu'il a appris de manière efficace sans simplement mémoriser les réponses.

#### **6.2.6. Évaluation du RNA**

Dans le cadre de la classification, le choix des réseaux de neurones artificiels a été motivé par leur excellente capacité à analyser des données complexes, à générer des prédictions fiables et à gérer des ensembles de données de grande dimension.

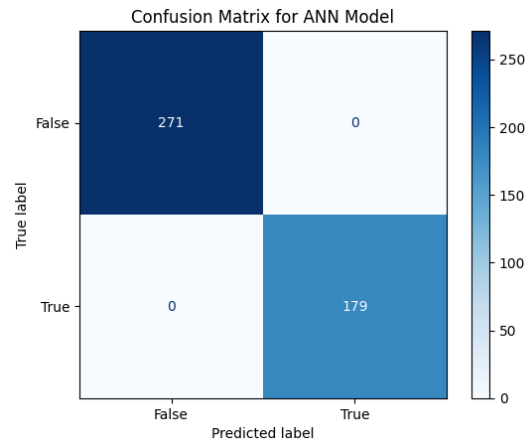
L'architecture de l'ANN a été spécifiquement conçue pour traiter six entrées des mesures triphasées de tension ( $V_A, V_B, V_C$ ) et de courant ( $I_A, I_B, I_C$ ).

Le modèle a été entraîné en utilisant 70% de l'ensemble des données, 15% ont été utilisés pour le test et les 15% restants pour la validation.

L'objectif d'entraînement et du test du réseau est de classer chaque cas en prévoyant un résultat binaire : 0 indiquant l'absence de défaut et 1 indiquant un défaut. Cette méthode permet non seulement d'évaluer la précision de l'ANN dans l'identification des défauts, mais aussi d'examiner sa capacité à généraliser à de nouvelles données qu'il n'a pas rencontrées pendant la phase d'entraînement.

Pour présenter les résultats d'un réseau de neurones artificiels (ANN) après l'entraînement, le test, et la validation, on pourrait structurer l'information de la manière suivante :

- **Matrice de confusion :**



**Figure 6.2 :** Matrice de confusion du réseau de neurones artificiels.

La matrice de confusion pour le modèle du réseau de neurones artificiels (ANN) présente les résultats suivants :

- Vrais Négatifs (VN): Le nombre 271 dans le coin supérieur gauche indique que le modèle a correctement prédit 271 instances où le défaut était absent (label 0).
- Faux Négatifs (FN): Le nombre 0 dans le coin inférieur gauche montre que le modèle n'a pas fait d'erreurs en prédisant qu'un défaut était absent alors qu'il était en réalité présent. Autrement dit, il n'y a pas eu de cas où le modèle a indiqué 'aucun défaut' de façon incorrecte.
- Faux Positifs (FP): Le nombre 0 dans le coin supérieur droit indique que le modèle n'a pas fait d'erreurs en prédisant la présence d'un défaut là où il n'y en avait pas. Le modèle n'a pas donné d'alerte de défaut fausse.
- Vrais Positifs (VP): Le nombre 179 dans le coin inférieur droit signifie que le modèle a correctement prédit 179 instances où un défaut était présent (label 1).

Cette matrice de confusion montre que le modèle a réalisé des prédictions parfaites sur l'ensemble des données testé, sans aucune erreur de classification, ce qui suggère une performance idéale.

Classification Report:					
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	271	
1	1.00	1.00	1.00	179	
accuracy			1.00	450	
macro avg	1.00	1.00	1.00	450	
weighted avg	1.00	1.00	1.00	450	

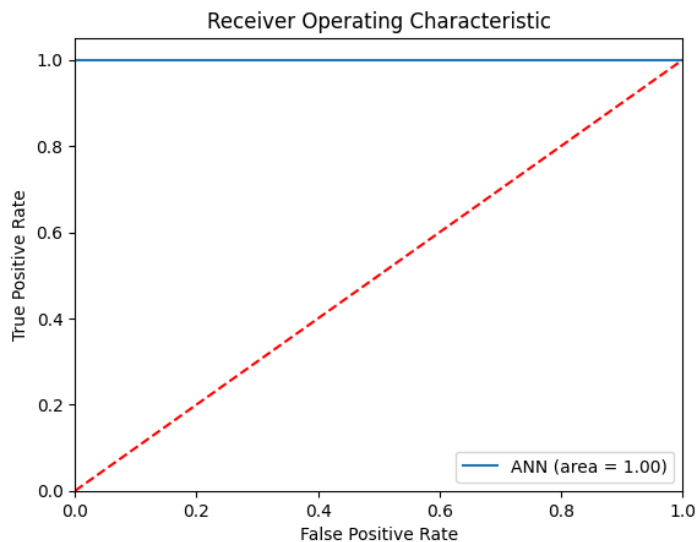
**Figure 6.3 :** Rapport de classification du réseau de neurones artificiels.

Le rapport de classification fourni montre les mesures de performance pour un modèle de classification binaire :

Précision d'entraînement (training accuracy) : 100%, Précision globale (model accuracy score) : 100% ce qui signifie que 100% des prédictions étaient correctes.

- Pour les deux classes (0 et 1), le modèle a atteint une précision et un rappel (recall) de 1.00, ce qui signifie que toutes les prédictions positives et négatives étaient correctes.
- Le score F1 est également de 1.00 pour les deux classes, indiquant une précision et un rappel parfaits. Le score F1 est la moyenne harmonique de la précision et du rappel et est particulièrement utile pour évaluer les modèles sur des ensembles de données déséquilibrés.
- Le support fait référence au nombre d'occurrences réelles de chaque classe dans l'ensemble de données spécifié. Ici, 271 instances de la classe 0 (absence de défaut) et 179 de la classe 1 (présence de défaut) ont été testées.
- La moyenne macro et la moyenne pondérée des scores pour les deux classes sont également parfaites (1.00).

Ces résultats indiquent que le modèle a performé parfaitement sur l'ensemble de test, sans commettre une seule erreur dans la classification des instances testées.



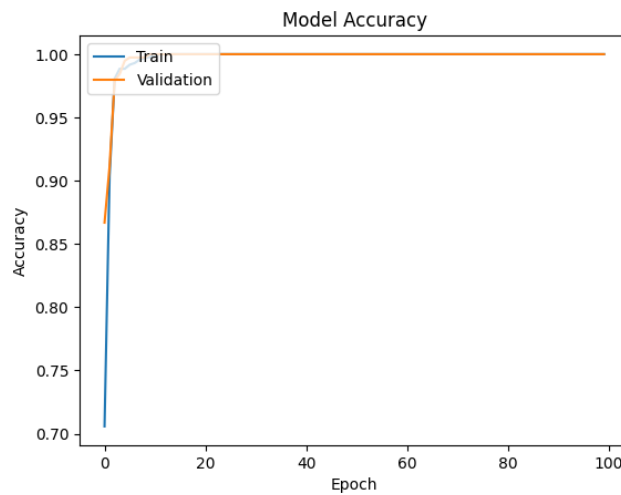
**Figure 6.4** : une courbe ROC (Receiver Operating Characteristic).

La figure ci-dessus montre une courbe ROC (Receiver Operating Characteristic), qui est un graphique illustrant la capacité de diagnostic d'un système de classification binaire en fonction de la variation de son seuil de discrimination.

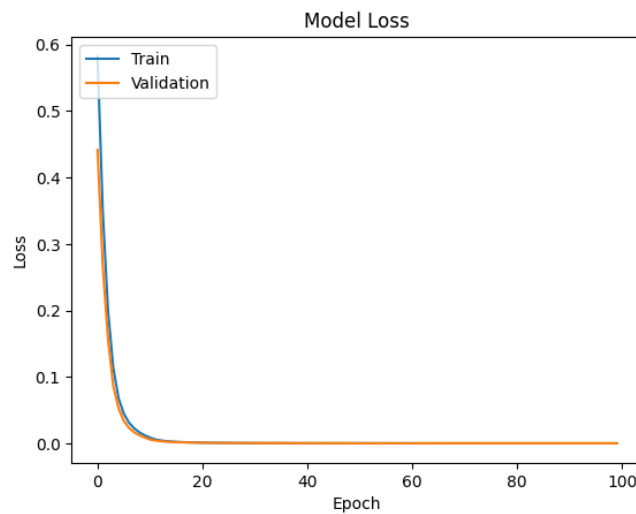
- Le Taux de vrais positifs (TPR), également appelé rappel ou sensibilité, se trouve sur l'axe des ordonnées. Il représente la proportion de vrais positifs correctement identifiés par le modèle.
- Le taux de faux positifs (FPR) se trouve sur l'axe des x. Il représente la proportion de négatifs réels qui sont incorrectement identifiés comme positifs par le modèle.
- La ligne rouge en pointillés représente la performance d'une supposition aléatoire, avec une aire sous la courbe (AUC) de 0,5.
- La ligne bleue, qui représente les performances de l'ANN, a une aire sous la courbe (AUC) de 1,00, ce qui correspond à une performance parfaite. Cela signifie que le modèle a correctement classé tous les cas positifs et tous les cas négatifs sans aucune erreur.

Cependant, des résultats aussi parfaits peuvent être suspects et devraient être examinés avec prudence, car ils peuvent être le signe d'un surajustement.

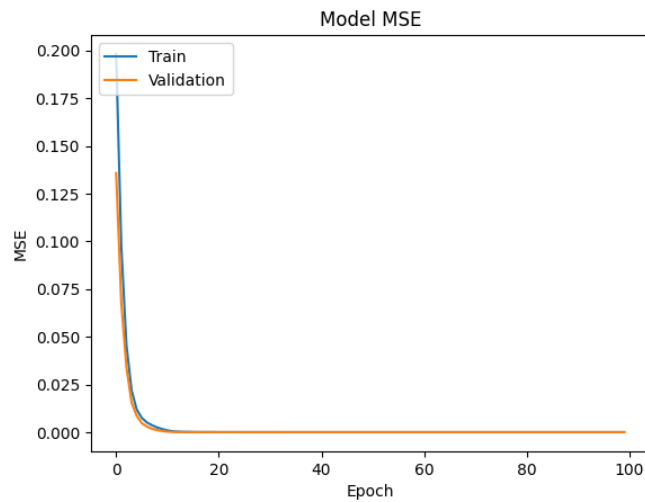
Dans notre cas, nous avons utilisé l'ensemble de validation pour prévenir le surajustement, en permettant d'évaluer si le modèle peut bien généraliser ses prédictions au-delà des données sur lesquelles il a été entraîné. Cela nous aidera à affiner les réglages du modèle et offre une mesure de performance plus réaliste, servant également à décider quand arrêter l'entraînement pour éviter d'apprendre par cœur les données d'entraînement.



**Figure 6.5 :** Tracés de la précision du modèle (Model Accuracy).



**Figure 6.6 :** Tracés de la perte du modèle.



**Figure 6.7** : Tracés de l'erreur quadratique moyenne du modèle (MSE).

Les figures plus présentent des tracés de l'erreur quadratique moyenne du modèle (MSE), de la perte du modèle et de la précision du modèle sur un certain nombre d'époques au cours de la formation d'un réseau neuronal artificiel.

**La précision du modèle (Model Accuracy)** : Ce graphique montre la précision du modèle sur les ensembles d'apprentissage et de validation au fil des époques. Une tendance à l'augmentation de la précision au fil du temps est une bonne chose, et une précision élevée qui plafonne indique que le modèle a appris à classer correctement les données d'entrée. Une fois encore, le fait que la précision de formation et de validation soit très proche l'une de l'autre et élevée suggère que le modèle s'est bien généralisé.

**Perte du modèle (Loss model)** : Ce graphique montre la perte d'apprentissage et de validation. Une perte décroissante au fil des époques est souhaitable, car elle indique que le modèle apprend effectivement à partir des données d'apprentissage. La convergence étroite des lignes de perte d'entraînement et de validation suggère que le modèle s'est bien généralisé et qu'il n'est pas surajusté.

**MSE du modèle** : Le graphique illustre l'évolution de l'Erreur Quadratique Moyenne (EQM) pour les données d'entraînement et de validation. Idéalement, on s'attend à ce

que les deux courbes diminuent à mesure que le nombre d'époques progresse, signalant que le modèle apprend et affine ses prédictions. Le fait que les EQM pour l'entraînement et la validation soient basses et suivent une trajectoire similaire suggère que le modèle a une bonne performance aussi bien sur les données connues de l'entraînement que sur les données inconnues de la validation, tout en évitant le piège du surajustement.

<b>L'Erreur Quadratique Moyenne (MSE)</b>	$3.0907876258368105 \times e^{-10}$
<b>Perte (Loss)</b>	$7.0653 \times e^{-6}$
<b>Précision globale (Accuracy)</b>	100%

**Tableau 6.1 : Métriques d'évaluation.**

D'après le tableau ci-dessus, les résultats montrent que le modèle a une Erreur Quadratique Moyenne (MSE) de  $3.0907876258368105 \times e^{-10}$ , ce qui est extrêmement bas et suggère que les prédictions du modèle sont très proches des valeurs réelles. La perte (Loss) affichée est également très faible, avec une valeur de  $7.0653 \times e^{-6}$ , indiquant que le modèle s'ajuste bien aux données pendant l'apprentissage. Enfin, une précision globale (Accuracy) de 100% indique que le modèle a correctement classé toutes les instances dans l'ensemble de données testé.

### **6.3. Prédiction des défauts des transformateurs en utilisant des données de Surveillance**

#### **6.3.1. Base de données**

Dans cette analyse de cas, la collection de données se compose de 20 316 observations uniques, chacune correspondant à un cas spécifique lié à l'objet de l'étude. Ces informations ont été initialement recueillies à l'aide de capteurs IoT entre le 25 juin 2019 et le 14 avril 2020, avec des enregistrements actualisés toutes les quinze minutes.

La base de données en question comprend sept variables de mesure. Six de ces



variables (OTI, WTI, ATI, OLI, OTI\_A, OTI\_T) serviront de variables d'entrée pour les modèles d'apprentissage automatique. La variable restante (MOG\_A) sera celle que nous tenterons de prédire ou d'utiliser comme variable cible.

### **6.3.2. Base de données et prétraitement des données**

La base de données est divisée en deux ensembles distincts, où 75% des données sont allouées à l'entraînement et les 25% restants sont destinés au test. Par ailleurs, il est essentiel de souligner l'importance du prétraitement, et en particulier la normalisation des données, qui est une étape cruciale lorsqu'on travaille avec des réseaux de neurones artificiels.

### **6.3.3. Structure du RNA**

Dans cette étude, le réseau de neurones artificiels (RNA) est conçu avec une architecture spécifique (6-16-5-1) qui comprend plusieurs couches, chacune ayant un nombre défini de neurones ou d'unités. Voici une description détaillée de la structure du réseau :

**Couche d'entrée :** Elle contient 6 neurones. Cette couche est la première étape du réseau, où chaque neurone correspond à une variable (OTI, WTI, ATI, OLI, OTI\_A, OTI\_T) de l'ensemble de données d'entrée.

**Première couche cachée :** Cette couche comprend 16 neurones. Les couches cachées sont au cœur des réseaux de neurones. Les neurones de cette couche vont traiter les informations reçues de la couche d'entrée, appliquer une fonction d'activation (comme ReLU = Rectifier Linear Unit), et transmettre le résultat à la couche suivante.

**Deuxième couche cachée :** Elle contient 5 neurones et fonctionne de manière similaire à la première couche cachée, en prenant les sorties de celle-ci comme entrées et en appliquant une fonction d'activation, dans notre cas, ReLU = Rectifier Linear Unit. Cette couche permet au réseau de capturer encore plus de complexité et

de nuances dans les données, facilitant l'apprentissage de relations plus abstraites et profondes.

**Couche de sortie :** La dernière couche du réseau a 1 neurone et est utilisée pour produire le résultat final du modèle. Dans le contexte d'une tâche de classification binaire, la couche de sortie de notre réseau utilise une fonction d'activation de type sigmoïd, qui génère une probabilité variant de 0 à 1 pour déterminer l'appartenance de l'entrée à l'une des classes.

#### **6.3.4. Entraînement du RNA**

Le réseau de neurones artificiels (ANN) est configuré avec l'optimiseur Adam. La fonction de perte choisie est l'entropie croisée binaire. En outre, le modèle évalue deux métriques clés : la précision, qui calcule le pourcentage de prédictions correctes, et l'erreur quadratique moyenne (MSE), qui quantifie la différence moyenne au carré entre les prédictions et les valeurs réelles.

Afin d'assurer une évaluation robuste, la technique de validation croisée K-fold est employée. Elle divise l'ensemble de données d'entraînement en K sous-ensembles, ou "plis", et utilise successivement chaque pli comme un ensemble de validation tout en entraînant le modèle sur les K-1 plis restants. Cela permet de minimiser le biais de validation et d'obtenir une estimation plus fiable de la capacité du modèle à généraliser. Dans notre cas, nous avons choisi  $k=5$ .

L'arrêt précoce (Early Stopping), une méthode qui arrête l'entraînement si la performance sur l'ensemble de validation ne s'améliore pas pendant un nombre défini d'époques consécutives, et la réduction du taux d'apprentissage sont utilisés comme des rappels pour éviter le surajustement et pour ajuster le taux d'apprentissage si aucune amélioration n'est observée.

Tout au long de l'entraînement avec validation croisée, les performances de validation et les valeurs de l'EQM sont enregistrées et moyennées pour chaque pli, fournissant

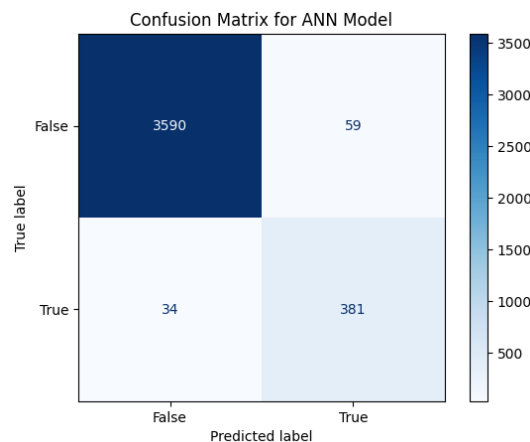
une vue d'ensemble cohérente de la performance du modèle avant l'évaluation finale sur l'ensemble de test.

### 6.3.5. Évaluation du RNA

Le réseau de neurones artificiels est un algorithme très puissant, il est capable de traiter des données très complexes et de générer des résultats fiables et robustes.

L'architecture de l'ANN a été spécifiquement conçue pour traiter six entrées des indicateurs de surveillance (OTI, WTI, ATI, OLI, OTI\_A, OTI\_T).

#### - Matrice de confusion



**Figure 6.8 :** Matrice de confusion du RNA.

La matrice de confusion présentée résume les performances d'un modèle de réseau de neurones artificiels (ANN) pour une tâche de classification binaire. Voici l'interprétation des chiffres indiqués :

- Vrais Négatifs (VN) : Le nombre 3590 indique que le modèle a correctement prédit l'absence de défaut dans 3590 cas.
- Faux Positifs (FP) : Le nombre 59 signifie que le modèle a incorrectement prédit la présence d'un défaut dans 59 cas où il n'y en avait pas.
- Faux Négatifs (FN) : Le nombre 34 représente les cas où le modèle a faussement prédit l'absence d'un défaut alors qu'il y en avait un.

- Vrais Positifs (VP) : Le nombre 381 montre que le modèle a correctement identifié la présence d'un défaut dans 381 cas.

Le modèle semble avoir une performance solide, avec un nombre élevé de VN et VP et un nombre relativement bas de FN et FP, ce qui pourrait indiquer une bonne capacité de prédiction.

- **Rapport de classification :**

Classification Report:					
	precision	recall	f1-score	support	
0.0	0.99	0.98	0.99	3649	
1.0	0.87	0.92	0.89	415	
accuracy			0.98	4064	
macro avg	0.93	0.95	0.94	4064	
weighted avg	0.98	0.98	0.98	4064	

**Figure 6.9 :** Rapport de classification du RNA.

Le rapport de classification présente les performances d'un modèle pour une tâche de classification binaire.

Précision : Le modèle a une précision de 0.99 pour la classe 0 et de 0.87 pour la classe 1.

Rappel: Ici, le modèle a un rappel de 0.98 pour la classe 0 et de 0.92 pour la classe 1.

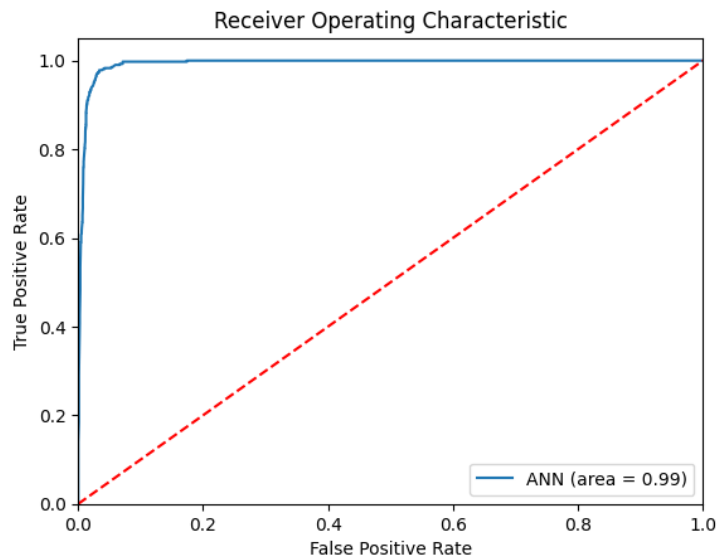
Score F1 : Le score F1 est de 0.98 pour la classe 0 et de 0.89 pour la classe 1, indiquant une performance équilibrée entre la précision et le rappel pour les deux classes.

Exactitude (Accuracy) : C'est la proportion totale de prédictions correctes, avec une valeur globale de 0.98.

Moyenne macro : Elle calcule la moyenne des scores sans prendre en compte la proportion pour chaque classe. Ici, la moyenne macro est de 0.93 pour la précision et de 0.95 pour le score F1.

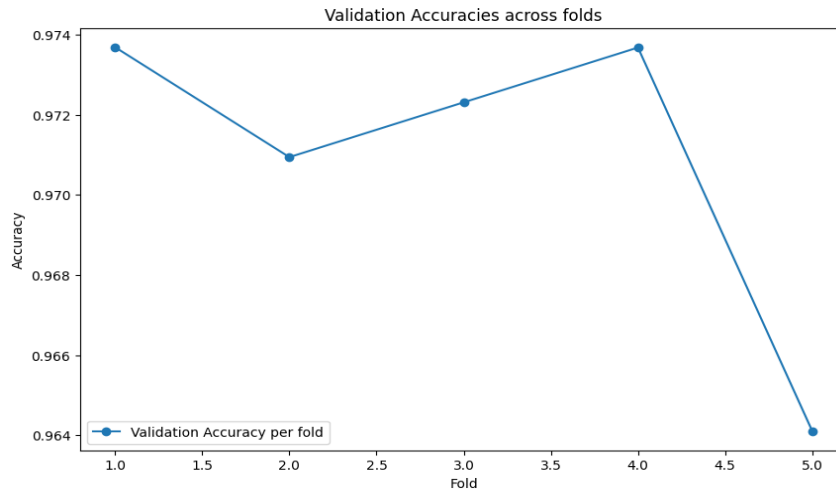
Moyenne pondérée : Elle tient compte du support de chaque classe. La moyenne pondérée est de 0.98 pour la précision et le score F1, ce qui est cohérent avec une haute précision et un fort rappel pour la classe majoritaire (classe 0).

En résumé, ces résultats indiquent que le modèle est très performant dans l'identification de la classe 0, avec une légère baisse de performance pour la classe 1. Cependant, les scores sont globalement élevés, ce qui suggère que le modèle fonctionne bien pour les deux classes.

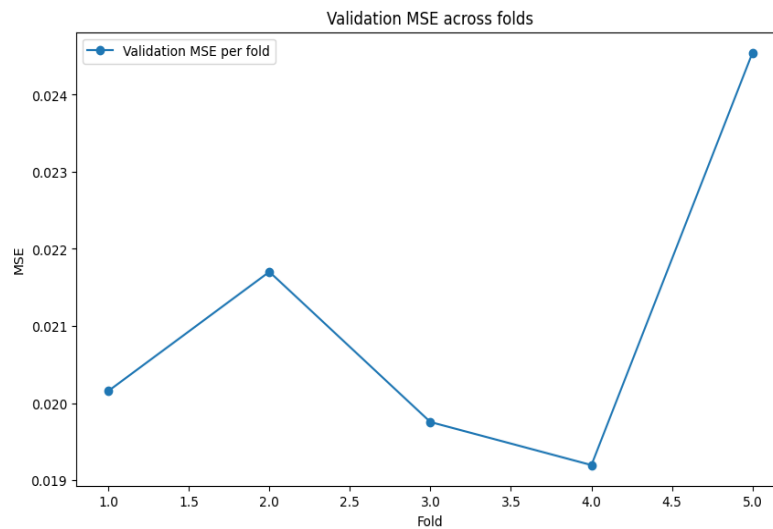


**Figure 6.10** : Courbe ROC (Receiver Operating Characteristic).

Cette courbe ROC suggère que l'ANN a une très haute performance pour la tâche de classification en question, avec une capacité presque parfaite à distinguer entre les classes positives et négatives. Une AUC de 0.99 est considérée comme exceptionnellement élevée pour la plupart des applications pratiques.



**Figure 6.11** : Graphique des précisions de validation pour un modèle sur cinq plis de validation croisée K-fold.



**Figure 6.12** : Graphique des erreurs quadratiques moyennes (MSE) pour un modèle sur cinq plis de validation croisée K-fold.

Les graphiques montrent les précisions de validation et les erreurs quadratiques moyennes (MSE) pour un modèle sur cinq plis de validation croisée K-fold.

Dans le premier graphique (précisions de validation), on observe que les précisions sur les quatre premiers plis sont relativement stables et élevées, mais il y a une chute

mineure de la précision dans le cinquième pli.

Le deuxième graphique (MSE de validation) montre des valeurs assez basses et stables de l'erreur pour les trois premiers plis, mais une petite augmentation pour le quatrième et surtout pour le cinquième pli.

Ces résultats indiquent que, bien que le modèle semble performant dans la plupart des cas. Il serait utile d'examiner les données du cinquième pli pour comprendre pourquoi le modèle ne performe pas aussi bien et d'explorer si des ajustements de paramètres ou une méthode de prétraitement des données pourraient améliorer la constance de la performance du modèle sur tous les plis.

<b>L'Erreur Quadratique Moyenne (MSE)</b>	0.0185
<b>Perte (Loss)</b>	0.0636
<b>Précision globale (Accuracy)</b>	98%

**Tableau 6.2 : Métriques d'évaluation du RNA.**

Le tableau présente trois métriques d'évaluation pour un réseau de neurones artificiels (RNA) :

- L'Erreur Quadratique Moyenne (MSE) est de 0.0185, ce qui est relativement bas et suggère que les prédictions du modèle sont en moyenne proches des valeurs réelles.
- La Perte (Loss) est de 0.0636, une mesure de l'inadéquation entre les prédictions du modèle et les valeurs réelles pendant l'entraînement. Un chiffre bas indique que le modèle a bien appris à partir des données d'entraînement.
- La Précision globale (Accuracy) est de 98%, ce qui signifie que le modèle a correctement prédit 98% des cas dans l'ensemble des données évalué, ce qui indique une haute performance.

Ces résultats indiquent que le RNA performe bien sur les données utilisées pour l'évaluation, avec une faible erreur moyenne et une haute précision.

## **6.4. Conclusion**

En somme, ce chapitre a exposé deux cas d'étude. Dans le premier, l'emploi d'un réseau de neurones artificiels pour l'analyse de données de courant et de tension a révélé une capacité de généralisation remarquable dans la prédiction des défaillances de transformateurs, avec une précision totale du modèle atteignant les 100%. Cette performance dépasse celle des modèles de machine learning décrits au chapitre 4.

Dans le second cas d'étude, l'utilisation du réseau de neurones artificiels avec des données de monitoring a également démontré une efficacité significative dans la prédiction des anomalies des transformateurs, avec une précision globale du modèle s'élevant à 98%. Cette performance est comparable à celle obtenue avec d'autres modèles de machine learning.



## CONCLUSION

Les transformateurs électriques constituent des éléments cruciaux au sein des infrastructures des réseaux électriques, assurant une fonction vitale dans la transmission et la distribution de l'électricité. Bien que leur fiabilité soit généralement élevée, ils ne sont pas à l'abri de pannes potentielles. Ces incidents, bien que rares, peuvent entraîner des conséquences significatives, notamment en termes de coûts élevés de réparation et de maintenance. Plus préoccupant encore, ils peuvent causer des interruptions prolongées du service électrique, impactant la continuité et la qualité de l'approvisionnement en énergie.

Ce rapport a exploré l'utilisation de méthodes d'apprentissage automatique pour prédire les défaillances et identifier les emplacements des défauts dans les transformateurs électriques. L'étude s'est déroulée en deux phases principales.

Initialement, la prédiction et la localisation des défaillances se sont appuyées sur des données simulées de courant et de tension, issues d'un modèle électrique spécifiquement développé à cet effet. Huit algorithmes d'apprentissage automatique ont été soigneusement sélectionnés, entraînés, testés et évalués sur cette base de données simulée afin de déterminer la méthode la plus performante.

Dans une seconde phase, l'accent a été mis sur la prédiction des défaillances des transformateurs en utilisant des données réelles, tirées d'indicateurs de surveillance de l'état obtenus d'une base de données accessible en ligne. Ces indicateurs ont servi à entraîner, tester et évaluer les mêmes méthodes d'apprentissage automatique, dans le but de distinguer l'approche la plus efficace.

Dans une phase subséquente, l'application des réseaux de neurones artificiels a été étendue aux ensembles de données traités dans les chapitres 4 et 5. Les réseaux ont subi des phases d'entraînement, de test et d'évaluation sur chacun de ces ensembles.

L'objectif était d'observer la réactivité des données face à une méthodologie fondée sur l'apprentissage profond.

Tout au long de cette recherche, une attention particulière a été portée à l'affinement des modèles pour assurer une fiabilité et une précision optimales, essentielles pour la maintenance préventive et la gestion des infrastructures électriques.

Pour réaliser cette étude, nous avons utilisé Matlab/Simulink comme plateforme de simulation, en nous appuyant sur les bibliothèques "SimPowerSystems" pour modéliser avec précision le système électrique. Pour les besoins de l'apprentissage automatique, le logiciel Spyder a été choisi en tant qu'environnement de développement intégré (IDE) pour sa compatibilité et son efficacité avec le langage de programmation Python, qui est largement reconnu pour sa puissance et sa flexibilité dans le traitement des données et la mise en œuvre des algorithmes de machine learning.

En considérant les futurs développements de cette recherche, il est envisageable d'intégrer des analyses plus approfondies concernant la localisation précise et la classification des types de défauts.

En outre, la mise en place d'une architecture de traitement de données en temps réel, en intégrant des approches de traitement par flux, pourrait permettre une surveillance en continu et une réactivité accrue face aux anomalies détectées, aboutissant à une maintenance prédictive plus efficace et à une diminution des temps d'arrêt imprévus.

## BIBLIOGRAPHIE

- [1] A. Hmich, A. Sahel, A. Badri, K. Loutfi, Y. El Kouari, and N. E. Belbounaguia, *surveillance et diagnostique du processus industriel complexe par réseaux de neurones*. 2008.
- [2] T. Kempowsky-Hamon, "Surveillance de procédés à base de méthodes de classification : conception d'un outil d'aide pour la détection et le diagnostic des défaillances," INSA de Toulouse, 2004. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-00010247>
- [3] N. Saber, "Protection et maintenance d'un transformateur industrielle," master, 2020.
- [4] R. Zemouri, "Contribution to the production system monitoring using dynamic neural networks : Application to the e-maintenance.," Université de Franche-Comté, 2003. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-00006003>
- [5] N. Fauzi *et al.*, "Fault Prediction for Power Transformer Using Optical Spectrum of Transformer Oil and Data Mining Analysis," *IEEE Access*, vol. PP, pp. 1-1, 07/23 2020, doi: 10.1109/ACCESS.2020.3011504.
- [6] V. V. Tejasvi roop singh, shivam, shivam chawla, sargam ghorela, sukirat singh mann, 7prashant singh rana, "fault detection in three phase transformer using ensemble model," presented at the Proceedings of 45th Research World International Conference, Los Angeles, USA, 2018.
- [7] B. Sudha, L. S. Praveen, and A. Vadde, "Classification of faults in distribution transformer using machine learning," *Materials Today: Proceedings*, vol. 58, pp. 616-622, 2022/01/01/ 2022, doi: <https://doi.org/10.1016/j.matpr.2022.04.514>.
- [8] U. A. Bakshi and M. V. Bakshi, *Electrical Machines-I*. Technical Publications, 2020.
- [9] D. A. Ravi Shankar Chauhan, "Fault Detection in Three Phase Transformer Using Decision Tree," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2015.
- [10] J. Sanchez, "Faults diagnostic aid on power transformers. Aide au diagnostic de défauts des transformateurs de puissance," Université de Grenoble, 2011GRENT031, 2011. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-01017179>
- [11] "transformateur électrique." <https://images.app.goo.gl/JJ5Girz8W3DubHgD8> (accessed).

- [12] "Types of Electric Transformers."  
<https://www.iqsdirectory.com/articles/electric-transformer/types-of-electric-transformers.html> (accessed).
- [13] R. Afzal, *Transformer Failures, Causes & Impact*. 2015.
- [14] D. B. N. C. Pankaj S. Kamble, "application of infrared camera for monitoring of transformer bushing," 2017.
- [15] J. H. Harlow, *Electric Power Transformer Engineering*. 2004.
- [16] T. Nasser, *Power Systems Modelling and Fault Analysis*. 2019.
- [17] C. Å, "Power Transformer Design Fundamentals. ABB Transformers," *ABB, Ludvika*, august 2000.
- [18] P. Wolff, "Predictive maintenance : a stochastic approach Maintenance prédictive : une approche stochastique," Université Paul Verlaine - Metz, 1996METZ025S, 1996. [Online]. Available: <https://hal.univ-lorraine.fr/tel-01777137>
- [19] M. de Castro-Cros, M. Velasco, and C. Angulo, "Machine-Learning-Based Condition Assessment of Gas Turbines&mdash;A Review," *Energies*, vol. 14, no. 24, p. 8468, 2021. [Online]. Available: <https://www.mdpi.com/1996-1073/14/24/8468>.
- [20] Mathworks. "Overcoming Four Common Obstacles to Predictive Maintenance with MATLAB and Simulink."  
<https://www.mathworks.com/campaigns/offers/predictive-maintenance-challenges.html> (accessed).
- [21] A. D. Scaife, "Improve predictive maintenance through the application of artificial intelligence: A systematic review," *Results in Engineering*, vol. 21, p. 101645, 2024/03/01/ 2024, doi: <https://doi.org/10.1016/j.rineng.2023.101645>.
- [22] Mathworks. "Predictive Maintenance with MATLAB." (accessed).
- [23] S. Geng and X. Wang, "Predictive maintenance scheduling for multiple power equipment based on data-driven fault prediction," *Computers & Industrial Engineering*, vol. 164, p. 107898, 2022/02/01/ 2022, doi: <https://doi.org/10.1016/j.cie.2021.107898>.
- [24] B. Taşcı, A. Omar, and S. Ayvaz, "Remaining useful lifetime prediction for predictive maintenance in manufacturing," *Computers & Industrial Engineering*, vol. 184, p. 109566, 2023/10/01/ 2023, doi: <https://doi.org/10.1016/j.cie.2023.109566>.
- [25] S. Schramm, C. Wehner, and U. Schmid, "Comprehensible Artificial Intelligence on Knowledge Graphs: A survey," *Journal of Web Semantics*, vol. 79, p. 100806, 2023/12/01/ 2023, doi: <https://doi.org/10.1016/j.websem.2023.100806>.
- [26] (2021). *regulation of the european parliament and of the council. laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts*. [Online] Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206>

- [27] N. Yüksel, H. R. Börklü, H. K. Sezer, and O. E. Canyurt, "Review of artificial intelligence applications in engineering design perspective," *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105697, 2023/02/01/ 2023, doi: <https://doi.org/10.1016/j.engappai.2022.105697>.
- [28] X. Li, M. Zhu, B. Zhang, X. Wang, Z. Liu, and L. Han, "A review of artificial intelligence applications in high-speed railway systems," *High-speed Railway*, 2024/01/29/ 2024, doi: <https://doi.org/10.1016/j.hspr.2024.01.002>.
- [29] F. Sami, "Optimize electric automation control using artificial intelligence (AI)," *Optik*, vol. 271, p. 170085, 2022/12/01/ 2022, doi: <https://doi.org/10.1016/j.ijleo.2022.170085>.
- [30] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine Learning for industrial applications: A comprehensive literature review," *Expert Systems with Applications*, vol. 175, p. 114820, 2021/08/01/ 2021, doi: <https://doi.org/10.1016/j.eswa.2021.114820>.
- [31] C.-H. Chien, A. J. C. Trappey, and C.-C. Wang, "ARIMA-AdaBoost hybrid approach for product quality prediction in advanced transformer manufacturing," *Advanced Engineering Informatics*, vol. 57, p. 102055, 2023/08/01/ 2023, doi: <https://doi.org/10.1016/j.aei.2023.102055>.
- [32] A. Kumbhar, P. G. Dhawale, S. Kumbhar, U. Patil, and P. Magdum, "A comprehensive review: Machine learning and its application in integrated power system," *Energy Reports*, vol. 7, pp. 5467-5474, 2021/11/01/ 2021, doi: <https://doi.org/10.1016/j.egy.2021.08.133>.
- [33] C. E. Lawson *et al.*, "Machine learning for metabolic engineering: A review," *Metabolic Engineering*, vol. 63, pp. 34-60, 2021/01/01/ 2021, doi: <https://doi.org/10.1016/j.ymben.2020.10.005>.
- [34] M. A. Hariri-Ardebili, G. Mahdavi, L. K. Nuss, and U. Lall, "The role of artificial intelligence and digital technologies in dam engineering: Narrative review and outlook," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106813, 2023/11/01/ 2023, doi: <https://doi.org/10.1016/j.engappai.2023.106813>.
- [35] S. R. Vadyala, S. N. Betgeri, J. C. Matthews, and E. Matthews, "A review of physics-based machine learning in civil engineering," *Results in Engineering*, vol. 13, p. 100316, 2022/03/01/ 2022, doi: <https://doi.org/10.1016/j.rineng.2021.100316>.
- [36] F. Wedel and S. Marx, "Application of machine learning methods on real bridge monitoring data," *Engineering Structures*, vol. 250, p. 113365, 2022/01/01/ 2022, doi: <https://doi.org/10.1016/j.engstruct.2021.113365>.
- [37] M. Kozan. "Supervised and Unsupervised Learning (an Intuitive Approach)." <https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f64b644> (accessed).
- [38] P. Santos, L. F. Villa, A. Reñones, A. Bustillo, and J. Maudes, "An SVM-Based Solution for Fault Detection in Wind Turbines," *Sensors*, vol. 15, no. 3,

- pp. 5627-5648, 2015. [Online]. Available: <https://www.mdpi.com/1424-8220/15/3/5627>.
- [39] S. Jain and A. Saha, "Improving performance with hybrid feature selection and ensemble machine learning techniques for code smell detection," *Science of Computer Programming*, vol. 212, p. 102713, 2021/12/01/ 2021, doi: <https://doi.org/10.1016/j.scico.2021.102713>.
  - [40] Z. Mian *et al.*, "A literature review of fault diagnosis based on ensemble learning," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107357, 2024/01/01/ 2024, doi: <https://doi.org/10.1016/j.engappai.2023.107357>.
  - [41] S. Rao, G. Zou, S. Yang, and S. Barmada, "A feature selection and ensemble learning based methodology for transformer fault diagnosis," *Applied Soft Computing*, vol. 150, p. 111072, 2024/01/01/ 2024, doi: <https://doi.org/10.1016/j.asoc.2023.111072>.
  - [42] X. Dong, Y. Liang, S. Miyamoto, and S. Yamaguchi, "Ensemble learning based software defect prediction," *Journal of Engineering Research*, vol. 11, no. 4, pp. 377-391, 2023/12/01/ 2023, doi: <https://doi.org/10.1016/j.jer.2023.10.038>.
  - [43] "classification." <https://scikit-learn.org/stable/modules/svm.html#classification> (accessed).
  - [44] J. Xiao, "SVM and KNN ensemble learning for traffic incident detection," *Physica A: Statistical Mechanics and its Applications*, vol. 517, pp. 29-35, 2019/03/01/ 2019, doi: <https://doi.org/10.1016/j.physa.2018.10.060>.
  - [45] T. Emmanuel, D. Mpoeleng, and T. Maupong, "Power plant induced-draft fan fault prediction using machine learning stacking ensemble," *Journal of Engineering Research*, 2023/10/04/ 2023, doi: <https://doi.org/10.1016/j.jer.2023.10.001>.
  - [46] S. D. Permai and K. Herdianto, "Prediction of Health Insurance Claims Using Logistic Regression and XGBoost Methods," *Procedia Computer Science*, vol. 227, pp. 1012-1019, 2023/01/01/ 2023, doi: <https://doi.org/10.1016/j.procs.2023.10.610>.
  - [47] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, "A novel kNN algorithm with data-driven k parameter computation," *Pattern Recognition Letters*, vol. 109, pp. 44-54, 2018/07/15/ 2018, doi: <https://doi.org/10.1016/j.patrec.2017.09.036>.
  - [48] "K-nearest neighbor. <https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4> (accessed).
  - [49] C. Catal, U. Sevim, and B. Diri, "Practical development of an Eclipse-based software fault prediction tool using Naive Bayes algorithm," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2347-2353, 2011/03/01/ 2011, doi: <https://doi.org/10.1016/j.eswa.2010.08.022>.
  - [50] Q. Tan *et al.*, "A new sensor fault diagnosis method for gas leakage monitoring based on the naive Bayes and probabilistic neural network

- classifier," *Measurement*, vol. 194, p. 111037, 2022/05/15/ 2022, doi: <https://doi.org/10.1016/j.measurement.2022.111037>.
- [51] H. Zhang, "The optimality of Naive Bayes.," *Proc. FLAIRS.*, 2004. [Online]. Available: <https://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf>.
- [52] S. Mishra, S. C. Swain, T. Biswal, and A. Routray, "Tree Based Fault Classification in Underground Cable," *Procedia Computer Science*, vol. 218, pp. 524-531, 2023/01/01/ 2023, doi: <https://doi.org/10.1016/j.procs.2023.01.034>.
- [53] "Decision tree algorithm." <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html> (accessed).
- [54] "Random forest algorithm." <https://www.spotfire.com/glossary/what-is-a-random-forest> (accessed).
- [55] A. A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation," *Expert Systems with Applications*, vol. 244, p. 122778, 2024/06/15/ 2024, doi: <https://doi.org/10.1016/j.eswa.2023.122778>.
- [56] A. Wan, Z. Gong, T. Chen, and K. Al-Bukhaiti, "Mass flow characteristics prediction of refrigerants through electronic expansion valve based on XGBoost," *International Journal of Refrigeration*, vol. 158, pp. 345-352, 2024/02/01/ 2024, doi: <https://doi.org/10.1016/j.ijrefrig.2023.12.011>.
- [57] "Machine learning end to end process." <https://www.analyticsvidhya.com/blog/2020/12/understand-machine-learning-and-its-end-to-end-process/> (accessed).

## ANNEXE A

### EXPLOITATION DE LA BASE DES DONNÉES

La base de données, créée à partir des simulations, est stockée dans un fichier Excel. La capture d'écran ci-dessous présente une section de la base de données utilisé dans le chapitre 4.

	A	B	C	D	E	F	G	H	I
1	Va	Vb	Vc	Ia	Ib	Ic	Yd	Yt1	Yt2
2	0,7192546	0,7192391	0,7192397	0,70191	0,7019412	0,7019842	0	0	0
3	0,7192535	0,7192362	0,7192433	0,7019028	0,70194	0,7019762	0	0	0
4	0,7192519	0,7192346	0,7192471	0,7018949	0,7019387	0,7019669	0	0	0
5	0,6983193	0,7192489	0,7162277	0,7635835	0,7017864	0,7690024	1	1	0
6	0,692995	0,7192458	0,713936	0,8132225	0,7017772	0,8062464	1	1	0
7	0,6491301	0,6662052	0,7192364	1,134491	0,5971828	0,7018604	1	0	1
8	0,6484199	0,6658923	0,7192386	1,129142	0,5909018	0,7018722	1	0	1
9	0,6482053	0,665807	0,7192406	1,1245086	0,5859378	0,7018878	1	0	1
10	0,6484661	0,6657961	0,7192427	1,1196601	0,5811141	0,7019082	1	0	1
11	0,6487147	0,6657631	0,7192463	1,1141177	0,5759063	0,7019311	1	0	1

Cette base de données est utilisée dans le chapitre 5. Voici une capture d'écran montrant une section de cette base de données.

1	DeviceTimeStamp	OTI	WTI	ATI	OLI	OTI_A	OTI_T	MOG_A
2	2019-07-08T14:45	31.0	0.0	31.0	39.0	0.0	0.0	0.0
3	2019-07-08T14:59	31.0	0.0	32.0	39.0	0.0	0.0	0.0
4	2019-07-08T14:59	31.0	0.0	32.0	39.0	0.0	0.0	0.0
5	2019-07-08T15:15	31.0	0.0	32.0	39.0	0.0	0.0	0.0
6	2019-07-10T00:01	0.0	0.0	0.0	39.0	0.0	0.0	0.0
7	2019-07-10T00:02	0.0	0.0	0.0	39.0	0.0	0.0	0.0
8	2019-07-10T00:05	0.0	0.0	0.0	39.0	0.0	0.0	1.0
9	2019-07-13T11:51	42.0	0.0	35.0	38.0	0.0	0.0	1.0
10	2019-07-13T11:51	42.0	0.0	35.0	38.0	0.0	0.0	1.0
11	2019-07-13T12:05	43.0	0.0	35.0	38.0	0.0	0.0	0.0
12	2019-07-13T12:15	42.0	0.0	34.0	39.0	0.0	0.0	0.0
13	2019-07-13T12:28	41.0	0.0	34.0	39.0	0.0	0.0	1.0
14	2019-07-13T12:30	41.0	0.0	34.0	39.0	0.0	0.0	1.0
15	2019-07-13T12:45	39.0	0.0	33.0	39.0	0.0	0.0	1.0
16	2019-07-13T13:00	39.0	0.0	33.0	39.0	0.0	0.0	1.0
17	2019-07-13T13:15	38.0	0.0	33.0	39.0	0.0	0.0	1.0



## ANNEXE B

### ALGORITHME KNN

**Données :** Considérons un ensemble de données comportant "n" caractéristiques (variables) et une cible (Y). Chaque point de données est représenté par (X1, X2, ..., Xn, Y).

**Choisir "K" :** Décidez du nombre de voisins, "k". Il s'agit d'une étape cruciale, car "k" détermine le nombre de voisins les plus proches qui influenceront la prédiction. Le choix de "k" peut affecter de manière significative les performances de l'algorithme.

**Calcul de la distance :** Pour un nouveau point de données, calculez la distance entre ce point et tous les autres points de l'ensemble de données. Les mesures de distance couramment utilisées sont les suivantes :

- **La distance euclidienne :** La distance en ligne droite entre deux points dans l'espace euclidien.  $\sqrt{\sum (X_i - Y_i)^2}$

- **Distance de Manhattan :** Somme des différences absolues de leurs coordonnées cartésiennes.  $\sum |X_i - Y_i|$

- **Distance de Minkowski :** Une métrique généralisée qui inclut d'autres cas particuliers.

Trouver les plus proches voisins : Identifier les "k" points les plus proches (voisins) du nouveau point de données sur la base du calcul de la distance.

**Prédiction :**

- Pour la classification : La classe prédite est la classe la plus fréquente parmi les "k" voisins les plus proches.

**Évaluation du modèle :** Utilisez des mesures telles que la précision (pour la classification) ou l'erreur quadratique moyenne (pour la régression) sur un ensemble de validation pour évaluer les performances du modèle.