RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
ÉCOLE NATIONALE SUPÉRIEURE DES TECHNOLOGIES AVANCÉES

Department of Industrial Engineering & Maintenance

Département Génie Industriel et Maintenance

Mémoire de fin d'étude en vue de l'obtention du diplôme

D'INGENIEUR d'État

Filière : Génie Industriel

Spécialité : Génie Industriel

Thème

Détection d'Intrusions dans les Systèmes de Contrôle Industriels

Une Approche Hybride Basée sur le Machine Learning

Réalisé par
Lebkara Haithem & Ouar Narimane

Soutenu le 22 juin 2025

Membres de Jury :

Prénom NOM	Université	Grade	Qualité
Mme. Belayadi Djahida	ENSTA	MCB	Encadrante académique
Mme. Djeghlouf Asma	ENSC	MCB	Co-encadrante académique
Mme. Ghoumari Leila	ENSTA	MCA	Examinateur
Mr. Boudhar Hamza	ENSTA	MCB	Président

2024/2025

Dédicace

Alhamdoulilah, toute gratitude revient à Dieu, Le Tout-Puissant,
qui m'a offert la force et la persévérance tout au long de ce parcours.

Je tiens également à m'adresser à moi-même, avec une sincère reconnaissance.
Merci pour ta résilience et ton courage.
Tu as su te rendre fière en découvrant, à chaque étape, de nouveaux potentiels.

Un hommage particulier à la femme de ma vie,
celle qui m'a forgée, guidée et aimée sans condition : ma chère maman.
Que ces mots ne suffisent jamais à exprimer l'immensité de ma reconnaissance.

À mon héro éternel, mon papa,
qui m'a toujours poussé à aller plus loin et à croire en mes capacités.
Merci d'avoir été ma boussole.

À Sara et Yousra, mes sœurs,
je vous remercie pour votre amour inconditionnel et votre soutien sans faille.

Ma gratitude va aussi à Magi, Fatma, Hajer, Ilhem, Besmala
Nour, Kawther et Amira , ma motivation et mon réconfort.

Merci d'avoir cru en moi.

Narimane

Dédicace

Je dédie ce travail à mes chers parents,
pour leur présence constante dans les jours difficiles comme dans les jours heureux.
Je vous dis, du fond du cœur :
je deviendrai cette personne dont vous avez toujours rêvé.

À mes frères et à mes amis — ils sont trop nombreux pour être tous cités ici —
avec qui j'ai partagé chaque étape, chaque leçon de la vie.
Grâce à vous, j'ai compris que les vraies richesses
sont les hommes qui nous entourent.

Je pense également à tous les enseignants
qui m'ont appris à être plus fort face aux épreuves.

Et je conclus par ces mots :
Faites confiance au plan d'Allah,
il sera toujours meilleur que ce que vous espérez.
Al hamdulilah .

Haithem

REMERCIEMENTS

Nous exprimons notre profonde gratitude à toutes celles et ceux qui, de près ou de loin, ont contribué à la réalisation de ce projet de fin d'études, marquant l'aboutissement de notre parcours académique.

Nos remerciements les plus chaleureux vont en premier lieu à notre encadrante académique, **Madame Belayadi Djahida**, pour sa disponibilité exemplaire, ses conseils éclairés et son accompagnement constant tout au long de ce travail.

Nous adressons également notre sincère reconnaissance à notre co-encadrante, **Madame Djeghlouf Asma**, pour son engagement, sa bienveillance et ses retours constructifs qui ont grandement enrichi notre réflexion.

Nos vifs remerciements s'étendent à **Monsieur Chelloufi Nabil** et **Monsieur Tamer Mohamed Saleh**, ingénieurs encadrants au sein de la centrale électrique Koudiet Draouèche, pour leur expertise technique, leur soutien indéfectible et les échanges stimulants qui ont jalonné notre collaboration.

Enfin, nous tenons à saluer l'ensemble du personnel du département de Génie Industriel et Maintenance de l'*École Nationale Supérieure des Technologies Avancées* pour la qualité exceptionnelle de la formation dispensée et pour leur accompagnement tout au long de notre cursus.

À toutes et à tous, un immense merci pour votre contribution inestimable.

الملخص

مع بزوغ عصر الصناعة 4.0، أصبحت أنظمة التحكم الصناعية, وخاصة أنظمة التحكم الموزعة (DCS)، أهدافًا رئيسية للهجمات السيبرانية، وذلك نتيجة لزيادة الترابط والاتصال واعتماد البروتوكولات الصناعية الضعيفة. يهدف هذا العمل إلى تعزيز أمن هذه البيئات الحيوية من خلال استكشاف دور تقنيات التعلم الآلي في كشف التسللات السيبرانية .قمنا بإجراء دراسة مقارنة شاملة لتسع خوارزميات من خوارزميات التعلم الآلي، مصنفة ضمن ثلاث فئات رئيسية: التجميع (Bagging)، والتعزيز (Boosting)، والتعلم العميق (Deep Learning)، وذلك باستخدام ثلاث مجموعات بيانات: CIC-IDS2018، وUNSW-NB15، بالإضافة إلى مجموعة بيانات حقيقية تم جمعها من نظام تحكم موزع لتوربينات الغاز. واستنادًا إلى نتائج هذا التقييم، قمنا بتطوير نموذج هجين أطلقنا عليه اسم X-RF Shield، يجمع بين خوارزميتي XGBoost وRandom Forest بهدف تحسين أداء الكشف .تم اختبار فعالية النموذج من خلال محاكاة في الوقت الحقيقي ضمن بيئة ICS افتراضية، وأظهرت النتائج التجريبية أن نموذج X-RF Shield يتفوق بوضوح على النماذج الفردية من حيث القدرة على اكتشاف التسللات.

الكلمات المفتاحية : الأمن السيبراني، التعلم الآلي، الكشف عن التسلل، أنظمة التحكم الصناعية، XRF-Shield، XGBoost، Random Forest، التحليلات الفورية، المرونة السيبرانية

Résumé

Avec l'émergence de l'industrie 4.0, les Systèmes de Contrôle Industriel (ICS), en particulier les Systèmes de Contrôle Distribuées (DCS), sont devenus des cibles privilégiées pour les cyberattaques en raison de leur connectivité accrue et de la vulnérabilité des protocoles industriels. Ce travail s'inscrit dans une démarche de sécurisation de ces environnements critiques en explorant l'apport de Machine Learning pour la détection d'intrusions. Nous avons mené une étude comparative rigoureuse de neuf algorithmes de machine learning répartis en trois catégories : bagging, boosting et deep leaning, sur trois jeux de données : CIC-IDS2018, UNSW-NB15 et un dataset issu d'un système DCS réel d'une turbine à gaz. Sur la base de cette évaluation, nous avons conçu un modèle hybride nommé X-RF Shield, combinant XGBoost et Random Forest, dans le but d'optimiser les performances de détection. Une simulation en temps réel a permis de valider son efficacité dans un environnement ICS simulé. Les résultats expérimentaux ont révélé que X-RF Shield surpasse les modèles individuels.

Mots-clés : Cybersécurité, apprentissage automatique, détection d'intrusions, systèmes de contrôle industriel, XRF-Shield, Random Forest, XGBoost, analytique en temps réel

Abstract

With the emergence of Industry 4.0, Industrial Control Systems (ICS), particularly Distributed Control System (DCS), have become prime targets for cyberattacks due to increased connectivity and the inherent vulnerabilities of industrial protocols. This work contributes to securing these critical environments by exploring the potential of Machine Learning for intrusion detection. We conducted a comparative study of nine machine learning algorithms, grouped into three categories : bagging, boosting, and deep learning across three datasets: CIC-IDS2018, UNSW-NB15, and a dataset collected from a real DCS system of a gas turbine. Based on this evaluation, we developed a hybrid model called **X-RF Shield**, combining XGBoost and Random Forest, with the objective of optimizing detection performance. A real-time simulation further validated its effectiveness in a simulated ICS environment, experimental results demonstrated that X-RF Shield outperforms individual models.

Keywords: Cybersecurity, machine learning, intrusion detection, industrial control systems, XRF-Shield, Random Forest, XGBoost, real-time analytics

Table des matières

Table des figures

Liste des tableaux

Liste des Abréviations

Abréviation	Description
CNN	Réseau de neurones convolutif
CPU	Unité centrale de traitement
DCS	Système de contrôle distribué
DL	Apprentissage profond
DNP3	Protocole de réseau distribué 3
DoS	Déni de service
FPR	Taux de faux positifs
HIDS	Système de détection d'intrusion basé sur l'hôte
HMI	Interface homme-machine
IA	Intelligence artificielle
ICS	Système de contrôle industriel
IDS	Système de détection d'intrusion
I/O	Entrée/Sortie
IoT	Internet des objets
KNN	K plus proches voisins
LSTM	Mémoire à long et court terme
ML	Apprentissage automatique
MTU	Unité terminale principale
NB	Bayes naïf
NIDS	Système de détection d'intrusion basé sur le réseau
NIST	Institut national des normes et de la technologie
PCA	Analyse en composantes principales
PLC	Contrôleur logique programmable
PRC	Courbe Précision-Rappel
RF	Forêt aléatoire
RNN	Réseau de neurones récurrent
ROC	Caractéristique de fonctionnement du récepteur
RTU	Unité terminale distante
SCADA	Supervision, contrôle et acquisition de données
SMOTE	Technique de suréchantillonnage de la minorité synthétique
SVM	Machine à vecteurs de support
TPR	Taux de vrais positifs
X-RF Shield	XGBoost Random Forest Shield

Introduction générale

Introduction générale

Avec l'essor de l'industrie 4.0, les environnements industriels connaissent une transformation majeure portée par l'intégration de technologies avancées. Au cœur de cette évolution, les Systèmes de Contrôle Industriel (ICS), en particulier les Systèmes de Contrôle Distribué (DCS), jouent un rôle clé en assurant la supervision et la gestion coordonnée des processus complexes. Conçus pour optimiser les performances, ces systèmes critiques sont exposés à de nouveaux défis, notamment en matière de cybersécurité, en raison de l'interconnexion croissante et des vulnérabilités inhérentes aux protocoles industriels .

Face à ces enjeux, l'intelligence artificielle (IA) s'impose comme une solution prometteuse pour renforcer les capacités des systèmes de détection d'intrusion (IDS) au sein des environnements DCS [11]. En appliquant des approches d'apprentissage automatique, il est possible d'identifier avec précision des anomalies et comportements suspects en temps réel, même dans des contextes complexes et fortement déséquilibrés [12].

De nombreux travaux récents ont exploré l'application des approches de deep learning, boosting et bagging pour améliorer la détection d'intrusions dans les ICS[32]-[35][37]. Cependant, la précision et la robustesse des modèles restent des défis majeurs, soulevant une problématique centrale : Quelle combinaison d'algorithmes de machine learning (notamment issus des familles de deep learning, boosting et bagging) permet d'atteindre le meilleur compromis entre précision, robustesse et détection en temps réel pour sécuriser efficacement les systèmes de contrôle industriel (ICS) ?

Notre travail s'articule autour de trois grandes phases. Dans un premier temps, nous menons un benchmark détaillé de neuf algorithmes répartis en trois catégories : bagging (Random Forest, Decision Tree, KNN), boosting (AdaBoost, Gradient Boosting, XGBoost) et deep learning (RNN, CNN, LSTM), sur trois jeux de données : CIC-IDS2018, UNSW-NB15 et un jeu de données dataset issu d'une turbine à gaz de la centrale hydroélectrique de Sonelgaz à Koudiet Eddraouech. Dans une deuxième étape, nous concevons une nouvelle approche, appelée X-RF

Shield, qui combine les deux meilleurs modèles identifiés : XGBoost et Random Forest pour maximiser l'efficacité de la détection. Enfin, nous mettons en œuvre une simulation en temps réel pour tester et valider la réactivité de X-RF Shield dans un environnement industriel simulé.

Ce rapport mémoire s'articule autour de quatre chapitres :

- **Chapitre 1 : Introduction aux ICS et DCS.** Ce chapitre introduit les systèmes de contrôle industriels (ICS), avec une attention particulière portée aux systèmes de contrôle distribués (DCS). On présente leur architecture, leurs vulnérabilités en matière de cybersécurité, ainsi que les solutions de détection d'intrusions (IDS) adaptées à ces systèmes.
- **Chapitre 2 : Revue de la littérature.** Ce chapitre est consacré à une analyse des travaux existants sur l'application de l'intelligence artificielle pour la détection d'intrusions dans les DCS. On met en évidence les approches fondées sur le deep learning, le boosting et le bagging, ainsi que leurs limites.
- **Chapitre 3 : Méthodologie.** Ce chapitre présente la méthodologie adoptée, déclinée en trois étapes : une évaluation comparative de neuf algorithmes, la proposition du modèle X-RF Shield, et la simulation en temps réel du modèle proposé.
- **Chapitre 4 : Résultats et discussion.** Ce dernier chapitre analyse les résultats expérimentaux obtenus sur les trois jeux de données, avec une attention particulière pour celui de Sonelgaz. On évalue la pertinence du modèle X-RF Shield dans un contexte de détection d'intrusion en conditions simulées.

Chapitre 01

La détection des intrusions dans les systémes de controle industriels

Chapitre 1

La détection des intrusions dans les systèmes de contrôle industriels

1.1 Introduction

Les ICS sont au cœur des infrastructures critiques tel que l'électricité. Avec l'intégration des technologies de l'industrie 4.0 et la numérisation des processus industriels les performances de ces systèmes ont été significativement renforcées. Cependant, ces avancées exposent les ICS à de nouvelles menaces en matière de cybersécurité. Parmi les architectures des ICS, les systèmes de contrôle distribués sont largement adoptés dans les environnements de production continue. Toutefois, ils deviennent vulnérables aux attaques ciblées.

Dans ce contexte, les nouveaux systèmes de détection d'intrusion qui intégrent des outils de l'intelligence artificielle (AI) jouent un rôle essentiel dans la protection des DCS. Notre approche se concentre sur les IDS basés sur la détection d'anomalies, soutenus par des techniques d'intelligence artificielle. Ce chapitre pose les bases nécessaires à l'étude en introduisant les ICS, avec un accent particulier sur les DCS, leurs vulnérabilités et les principes fondamentaux de la détection d'intrusion industrielle.

1.2 Les systèmes de contrôle industriels (ICS) : structure et technologies

Pour comprendre les enjeux de la cybersécurité dans les environnements industriels, il est essentiel de définir les systèmes de contrôle industriels, d'explorer leur conception et leur évolution.

1.2.1 Définition

Les ICS englobent un ensemble d'appareils, de réseaux et d'applications dédiés à l'automatisation et au contrôle des processus industriels [12]. Ces systèmes sont cruciaux pour le fonctionnement des infrastructures critiques, telles que les centrales électriques et les stations de traitement de l'eau . Un ICS intègre plusieurs composants : des capteurs pour la collecte de données en temps réel, des actionneurs pour ajuster les processus, des serveurs, ainsi que des interfaces homme-machine (HMI) et d'autres équipements spécifiques. Ces éléments interagissent de manière coordonnée pour garantir un contrôle efficace des installations industrielles.

1.2.2 Historique et évolution

L'évolution des systèmes de contrôle industriels est étroitement liée aux révolutions technologiques et industrielles. Chaque avancée a introduit de nouvelles exigences en matière de supervision des processus.

- **Avant l'automatisation :** Au début de l'industrialisation, les processus étaient gérés manuellement, avec des dispositifs électromécaniques comme des relais, des contacteurs, des minuteries et des régulateurs analogiques. Ces systèmes présentaient des limites, notamment une flexibilité réduite et une dépendance à l'intervention humaine.

- **Automates programmables industriels (PLC) – 1968 :** La fin des années 1960 a vu l'introduction des premiers automates programmables (PLC) par Modicon, remplaçant les tables de relais par des solutions électroniques reprogrammables. Cette innovation a réduit les coûts de maintenance et facilité la modification des équipements.

- **Systèmes de contrôle distribués (DCS) – Années 1980 :** Avec la complexité croissante des processus industriels, les DCS ont émergé, permettant la répartition des tâches de contrôle sur des unités intelligentes coordonnées par un centre de supervision. Cela a amélioré la fiabilité et réduit les points de défaillance.

- **Systèmes SCADA – Années 1990 :** Les systèmes SCADA ont introduit la supervision à distance, la collecte de données et le contrôle centralisé, facilitant l'intégration avec les systèmes d'information et préparant l'avènement de l'industrie 4.0.

- **Industrie 4.0 – Depuis 2000 :** L'industrie 4.0 repose sur l'intégration de technologies numériques avancées, telles que l'Internet des Objets (IoT), l'intelligence artificielle (IA), le cloud computing, le big data et les systèmes embarqués intelligents. Ces systèmes intelligents, contrôlables à distance, gèrent et analysent de grandes quantités de données, mais exposent les ICS à de nouvelles vulnérabilités, telles que les cyberattaques, accès non autorisés, les pannes logicielles ou la manipulation des processus.

1.3 Architecture des ICS

L'architecture des ICS est conçue pour assurer une interaction fluide entre les équipements de terrain et les systèmes de supervision, garantissant la sécurité et la fiabilité des opérations.

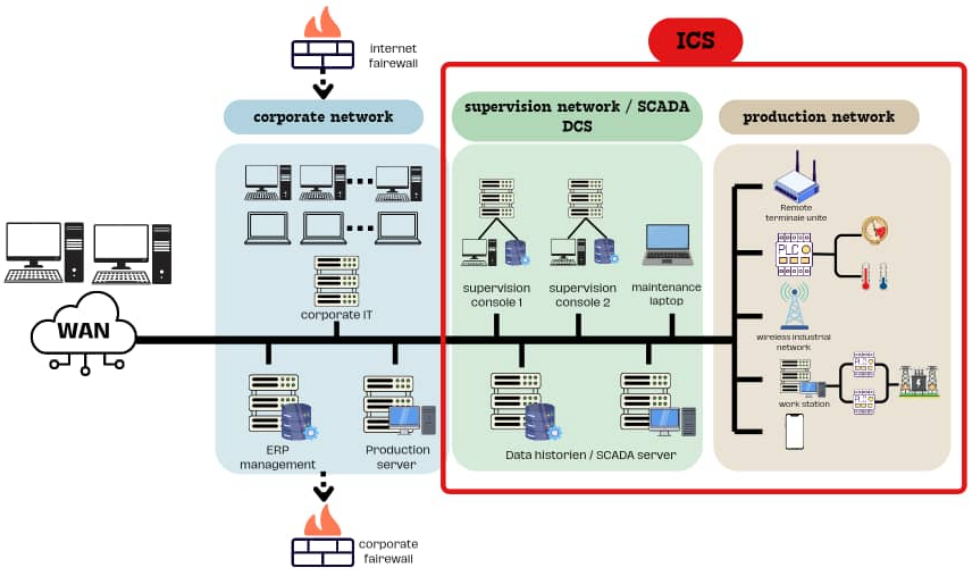


FIGURE 1.1 – Architecture matérielle des systèmes ICS
[40]

1.3.1 Types d'architectures

Les ICS peuvent être classés en deux grandes catégories architecturales selon leur organisation : les systèmes centralisés et les systèmes distribués [15].

- **Système centralisé :** Ce modèle repose sur un centre de traitement unique, situé généralement dans une salle de commande principale, qui exécute les algorithmes de contrôle et transmet les instructions aux actionneurs. Il est adapté aux installations de petite ou moyenne taille situées à proximité géographique.
- **Système distribué :** Ce modèle répartit les unités de commande sur différentes zones opérationnelles, chaque contrôleur local gérant une partie spécifique du processus. Les données sont traitées localement et synchronisées avec un système central. Cette architecture est idéale pour les environnements industriels complexes ou géographiquement dispersés, comme les centrales énergétiques.

1.3.2 Composants principaux

Les ICS comprennent plusieurs composants essentiels :

- **Système de traitement de l'information :** Inclut les systèmes de gestion des stocks, les serveurs, le matériel réseau et les postes de travail, qui permettent la gestion et l'analyse des données collectées.
- **Équipements spécifiques :** Comprend les PLC, RTU, capteurs et actionneurs qui interagissent directement avec le système physique.
- **Interfaces homme-machine (HMI) :** Permettent aux opérateurs de superviser et contrôler les processus en temps réel.

1.4 Typologie des systèmes ICS

Selon la complexité des processus et les variables physiques à contrôler, les systèmes ICS peuvent être classés en plusieurs types : les automates programmables industriels (PLC), les DCS et les systèmes de supervision et d'acquisition de données (SCADA).

1.4.1 Les automates programmables industriels (PLC)

Les contrôleurs logiques programmables sont des dispositifs électroniques conçus pour automatiser des processus industriels discrets ou séquentiels. Ils utilisent des programmes logiques, souvent en langage Ladder, pour activer les actionneurs. Les PLC sont largement utilisés dans la fabrication industrielle, comme l'assemblage automobile, où les actions sont précisément définies. Leur rapidité d'exécution les rendent adaptés aux environnements industriels, avec un entretien facile et une modularité accrue.

Le fonctionnement d'un PLC repose sur trois composantes principales :

- **CPU (Unité centrale de traitement) :** Cœur logique du PLC, la CPU exécute les commandes du programme en temps réel en fonction des signaux d'entrée. Elle inclut un microprocesseur, un système d'horloge et des interfaces de communication.
- **Modules d'entrée/sortie (I/O) :** Ces interfaces relient le PLC à l'environnement industriel. Les modules d'entrée capturent les signaux des capteurs qui peuvent être numériques ou analogiques. Les modules de sortie envoient des instructions aux actionneurs tels que les moteurs et les relais.
- **Mémoire :** Stocke les programmes, les données et les configurations nécessaires au fonctionnement du PLC.

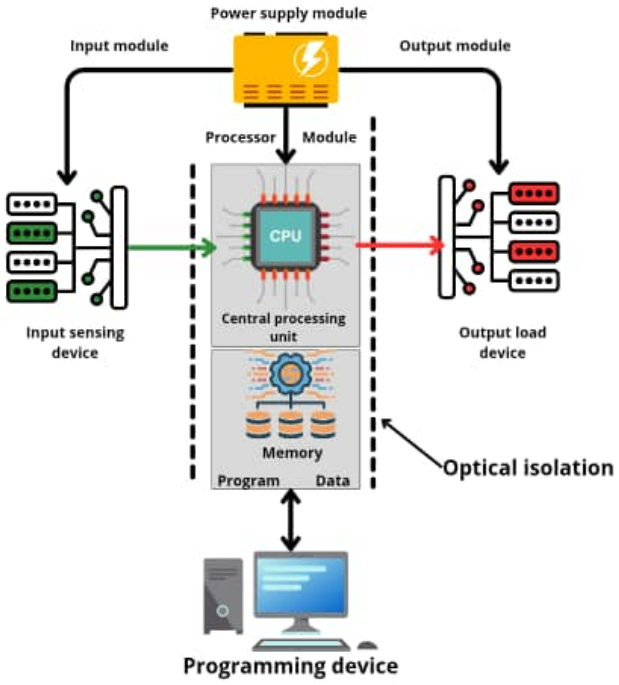


FIGURE 1.2 – Architecture d'un PLC

1.4.2 Les systèmes SCADA

Les systèmes SCADA collectent et analysent en temps réel les données provenant de sites industriels distants. Ils assurent une gestion décentralisée des installations via des connexions de communication, en intégrant l'acquisition de données via RTU ou PLC, la visualisation via HMI ou clients SCADA, l'exécution de commandes, l'enregistrement et l'analyse historique. Un système SCADA comprend :

- **RTU (Unités terminales distantes) :** Collectent les données des capteurs et les transmettent au système central SCADA.
- **MTU (Unité terminale principale) :** Collecte les données des RTU ou PLC, les traite et peut envoyer des commandes en retour.
- **SCADA computer center :** Permettent aux opérateurs de visualiser et contrôler les processus.
- **Serveurs :** Centralisent les flux de données entre les équipements distants et les postes opérateurs.
- **Historien :** Archive les données des processus pour une analyse à long terme.

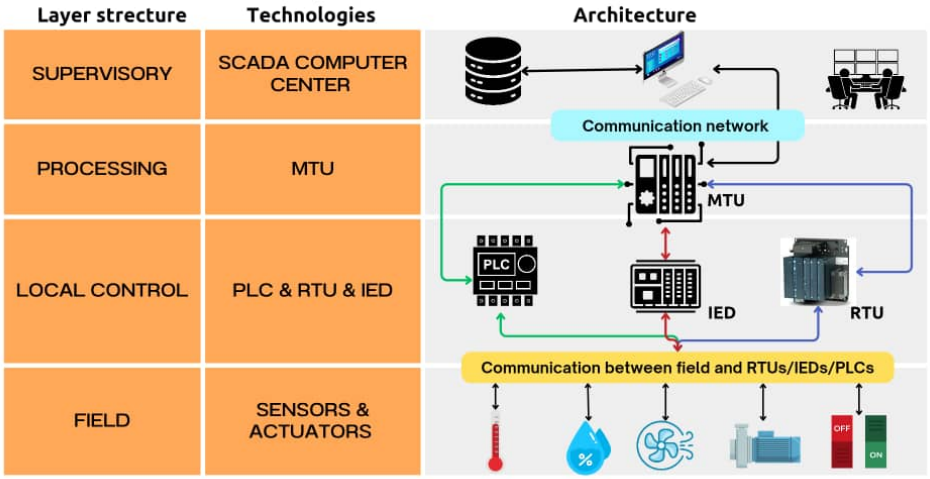


FIGURE 1.3 – Architecture de SCADA

1.4.3 Les systèmes de contrôle distribués 'DCS'

D'après le glossaire du National Institute of Standards and Technology (NIST) [12], un DCS est défini comme suit :

Un DCS représente un contrôle réalisé par une intelligence distribuée autour du processus à contrôler, plutôt que par une unité unique située de manière centralisée. Ces systèmes sont conçus pour superviser des processus industriels continus nécessitant un contrôle en boucle fermée et une coordination centralisée.

Les DCS trouvent leur application dans les secteurs de transformation tels que la pétrochimie, la cimenterie, la papeterie, l'industrie pharmaceutique, ainsi que dans les centrales thermiques et nucléaires. Grâce à leur structure décentralisée, ils sont capables de gérer plusieurs boucles de contrôle simultanément tout en assurant une supervision centralisée. Ils proposent une grande tolérance aux défaillances.

L'architecture typique d'un système DCS est composée de :

- **Unités de contrôle locales (Controller Units) :** C'est le cœur fonctionnel du DCS. Ce sont des unités de traitement indépendantes, généralement fondées sur des microprocesseurs industriels, qui mettent en œuvre les programmes de contrôle en temps réel.
- **Réseau de communication :** Il relie les diverses unités du DCS, garantissant un transfert de données synchronisé entre les contrôleurs, les postes d'opérateurs et les serveurs centraux.
- **Historien (Base de données des processus) :** Spécifiquement conçu pour l'archivage des données de production et de toutes les actions effectuées par les opérateurs.
- **Postes de travail opérateurs (Operator Workstations) :** Les interfaces homme-machine

offrent la possibilité de visualiser et interagir avec le système DCS. En proposant des tableaux de bord en temps réel, des alertes, des historiques d'événements.

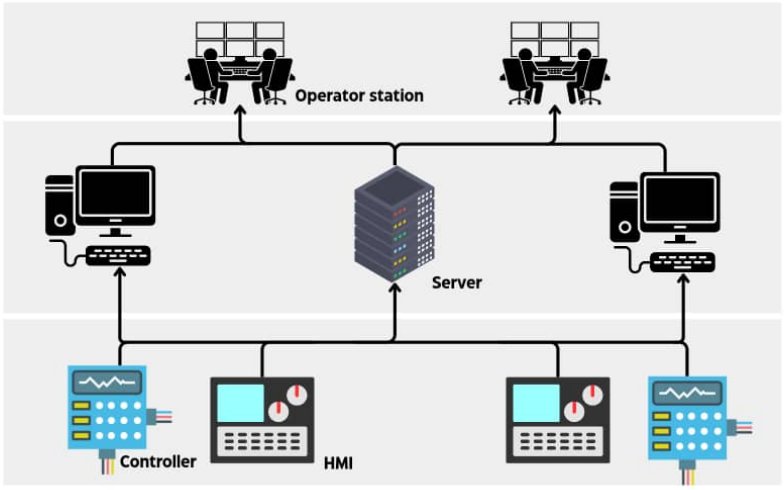


FIGURE 1.4 – Architecture de DCS

Étant donné l'architecture complexe des DCS, notre travail se concentre spécifiquement sur ces derniers. Toutefois, il convient d'abord de comprendre l'origine de leurs vulnérabilités.

1.5 Vulnérabilités des protocoles DCS

Les DCS reposent sur des protocoles de communication spécifiques pour l'échange d'informations qui peuvent être classés en protocoles ouverts élaborés par des organismes de normalisation et propriétaires qui sont affinés pour la communication entre dispositifs d'un seul fournisseur.

Parmi les protocoles les plus répandus, on peut citer :

- **Modbus :** Le MODBUS, acronyme de MODicon communication BUS, est probablement le protocole de communication le plus ancien et le plus utilisé dans les DCS en raison de sa facilité d'utilisation [9]. Cependant, il reste conçu sans mesures de sécurité inhérentes comme le chiffrement et l'authentification, ce qu'il le rend vulnérable aux interceptions, aux changements et aux injections de commandes [4].

- **PROFIBUS / PROFINET :** Siemens a développé les protocoles industriels PROFINET basé sur Ethernet et PROFIBUS basé sur une architecture série. Malgré leur efficacité dans le contrôle des processus, ces protocoles manquent de mesures de sécurité intégrées, ce qui les rend vulnérables aux attaques réseau [8] ;[7]

- **DNP3 :** Un protocole ouvert utilisé pour la surveillance des infrastructures critiques [3] ; [5]. Conçu à l'origine pour les communications série entre MTU, RTU et IED, il est basé sur une

architecture à trois couches (liaison, pseudo-transport et application). Cependant, la version standard de DNP3 ne permet pas l'authentification ou le chiffrement, exposant les systèmes au risque de manipulation des données ou d'interception [10]

Face à ces vulnérabilités, il faut adopter des protections adéquates à la criticité des environnements industriels.

Les systèmes de détection d'intrusion (IDS) offrent une réponse efficace, en identifiant en temps réel les activités suspectes sans perturber les opérations. Leur principe et leurs spécificités dans les systèmes industriels seront abordés dans la section suivante.

1.6 Systèmes de détection des intrusions dans les environnements DCS

1.6.1 Définition et rôle

On peut définir une intrusion comme toutes les actions qui violent la politique de sécurité du système. L'attaquant tente de trouver une méthode pour obtenir un accès non autorisé aux informations et causer des dommages .

Les systèmes de détection d'intrusion, également connus des IDS, sont des technologies de sécurité qui peuvent être matérielles comme les dispositifs ou logicielles comme les applications et les programmes [3]. Ils permettent de suivre le trafic d'un réseau ou d'un système informatique tout en contrôlant leurs activités pour rechercher des signes d'accès non autorisé ou de comportement malveillant .

1.6.2 Architecture d'un IDS

Un IDS repose sur une structure modulaire intégrant divers composants collaboratifs. Il est généralement composé des modules suivants :

- **Module de collecte des données :** Capture les données provenant du trafic réseau, des journaux système et des processus [8]. Ces données sont extraites soit en temps réel ou à un intervalle régulier.
- **Module de préparation des données :** Il standardise et trie les données recueillies pour les rendre utilisables par le moteur d'analyse.
- **Moteur d'analyse :** Il analyse les données de réseau afin d'identifier des comportements malveillants en examinant le trafic réseau ou les journaux de système pour repérer des modèles d'activité qui pourraient indiquer une intrusion.
- **Base de données :** C'est ici qu'on entrepose les signatures d'attaques identifiées, les modèles de comportements standards, et aussi les règles d'analyse.

- **Module de réponse :** C'est le responsable de produire des alertes lorsqu'une intrusion est détectée.
- **Console d'administration :** Il gère la liaison entre le système IDS et l'administrateur. Offrant la possibilité d'afficher les alertes, de gérer les journaux et d'établir les politiques de sécurité.

1.6.3 Classification des IDS

Les IDS peuvent être classés selon plusieurs critères, notamment leur position dans l'architecture réseau et leur approche de détection.

Selon leur position dans l'infrastructure, les IDS se divisent principalement en trois types :

- **IDS basé sur le réseau (NIDS) :** Agit au niveau du réseau [12]. Il fait la surveillance de tout le trafic circulant vers et depuis les équipements du réseau.
- **IDS basé sur l'hôte (HIDS) :** Il ne fait que surveiller l'appareil sur lequel il est déployé comme les serveurs ou les ordinateurs de bureau. En d'autres termes, il est installé sur un appareil spécifique qui le défend contre les dangers internes et externes [10].
- **IDS hybride :** Il intègre les méthodes des deux systèmes précédents, les données provenant du système ou de l'agent hôte sont combinées avec les renseignements du réseau afin d'obtenir une vision globale du système.

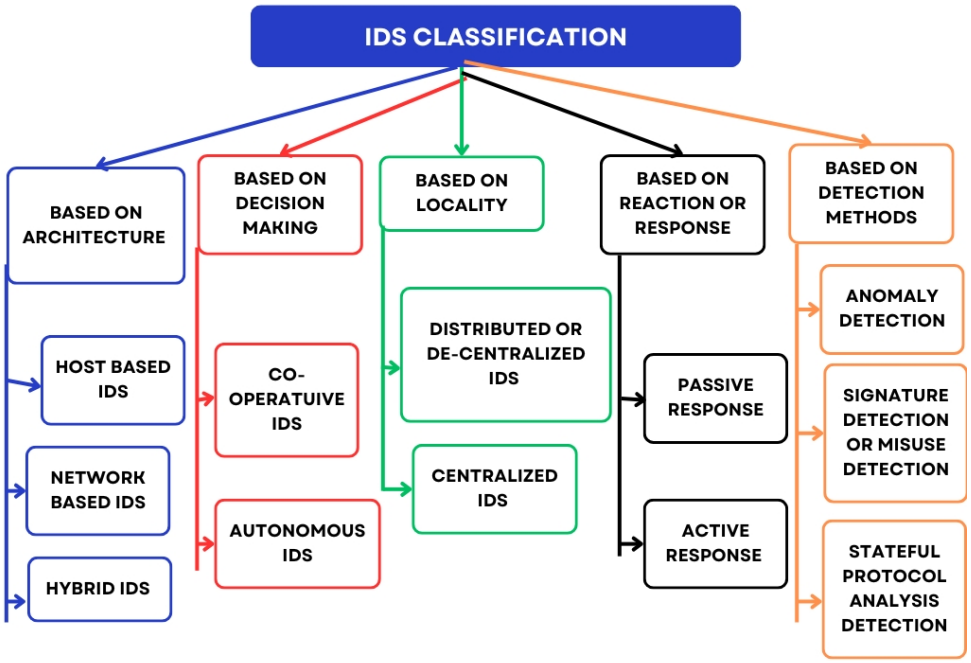
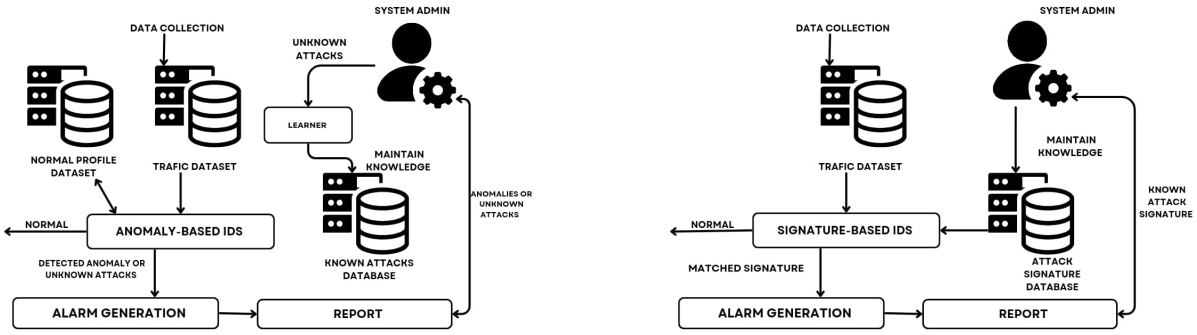


FIGURE 1.5 – Classification d'un IDS

En ce qui concerne la fonction fondamentale des IDS : la détection, on trouve deux grandes catégories prédominantes :

- **Basé sur les signatures :** Ils surveillent le réseau entrant pour des modèles et des séquences spécifiques qui correspondent aux signatures d'attaque connues. Bien que ce type d'IDS s'avère efficace contre les menaces identifiées, il demeure limité face aux attaques inédites.
- **Basé sur les anomalies :** La logique d'un IDS basé sur les anomalies part de l'hypothèse qu'un système industriel présente un comportement normal identifiable. L'objectif est de détecter tout écart qui signale une attaque ou un échec. La phase d'apprentissage collecte des données représentatives du fonctionnement normal (protocoles, trafic réseau, commandes, journaux) couvrant les communications internes et externes. À partir de ces données, des modèles comportementaux sont construits avec des algorithmes de machine learning [11].

  Ensuite, en phase de surveillance, l'IDS compare en temps réel ou périodiquement les événements et paquets observés au modèle établi. Si l'écart dépasse un seuil, une alerte est déclenchée [12]. Ce mécanisme assure une détection proactive, adaptée aux environnements industriels où chaque déviation peut compromettre sécurité, disponibilité ou intégrité.



(a) la logique d'un IDS basé sur anomalie (b) la logique d'un IDS basé sur signature

FIGURE 1.6 – la logique de fonctionnement d'un IDS

1.6.4 Évaluation des IDS

Les critères utilisés pour évaluer l'efficacité de n'importe quel système de détection d'intrusion sont :

- **Précision :** Le système est fiable dans la détection des attaques sans générer de fausses alertes.
- **La performance de traitement :** Mesurée par la rapidité de traitement des événements.
- **La complétude :** C'est la faculté d'un IDS à repérer toutes les attaques.
- **La résilience face aux défaillances :** La plupart des IDS s'exécutent sur des systèmes vulnérables aux attaques. Un bon IDS doit donc être capable de continuer à fonctionner même en cas d'attaques.

- **La rapidité :** L'IDS doit analyser les données et réagir rapidement pour limiter le temps de réponse. Cela permet d'éviter que l'attaquant ne modifie les traces ou ne perturbe le fonctionnement du système.

1.7 Conclusion

Ce premier chapitre a permis d'introduire les DCS. Nous avons examiné leur architecture, leurs fonctions principales, ainsi que les défis en matière de cybersécurité auxquels ils sont exposés en raison de leur complexité croissante.

Dans ce contexte, nous avons ensuite présenté les systèmes de détection d'intrusion comme une solution essentielle pour la protection des DCS. Les différents types d'IDS ainsi que leurs architectures ont été abordés.

Enfin, nous avons détaillé les principaux critères d'évaluation d'un IDS, tels que la rapidité et la précision. Cette analyse nous a préparé le terrain pour le prochain chapitre, qui traitera des contributions de l'intelligence artificielle à la sécurité de DCS en détaillant les algorithmes de machine learning.

Chapitre 02

Apprentissage automatique pour la détection des intrusions dans les DCS

Chapitre 2

Apprentissage automatique pour la détection des intrusions dans les DCS

2.1 Introduction

L'intelligence artificielle (IA), outil révolutionnaire du XXI siècle, est désormais omniprésente dans de nombreux secteurs. En santé, elle contribue au diagnostic médical par l'analyse d'images telles que les IRM ; en finance, elle permet de détecter les fraudes parmi des millions de transactions ; et dans la logistique, elle optimise la chaîne d'approvisionnement en temps réel. En cybersécurité, l'IA occupe une place centrale dans les systèmes modernes de détection d'intrusion (IDS), en renforçant leur capacité à identifier des comportements anormaux.

Ce chapitre présente les fondements de l'apprentissage automatique appliqué à la détection d'intrusions. Une attention particulière sera accordée aux approches de bagging, boosting et deep learning, fréquemment utilisées dans les travaux récents sur les IDS.

Une revue critique de la littérature sera menée afin d'évaluer l'efficacité de ces approches, d'identifier leurs limites, et de proposer des recommandations pour orienter le choix des méthodes les plus adaptées.

2.2 Apprentissage automatique : définition et application

Le machine learning (ML), ou apprentissage automatique, est l'une des branches fondamentales de l'intelligence artificielle qui permet aux systèmes comme les systèmes de détection d'apprendre à partir des données et de s'améliorer automatiquement sans avoir besoin d'être programmés pour chaque tâche.

De nombreuses utilisations de l'apprentissage automatique peuvent être intégrées pour la cybersécurité et comprennent :

- **Détection et classification des menaces** : En analysant de grands ensembles de données pour identifier les modèles d'activité malveillante.
- **Sécurisation des points de terminaison mobiles** : L'apprentissage automatique est utilisé pour se défendre contre les attaques qui utilisent des commandes vocales en entraînant des modèles qui distinguent entre les voix du propriétaire et celles des pirates.
- **Automatisation des tâches répétitives** : Appliquer le ML à des tâches spécifiques peut aider les équipes de sécurité à gérer des tâches répétitives en agissant comme un multiplicateur de force qui leur permet de répondre plus rapidement aux alertes entrantes.

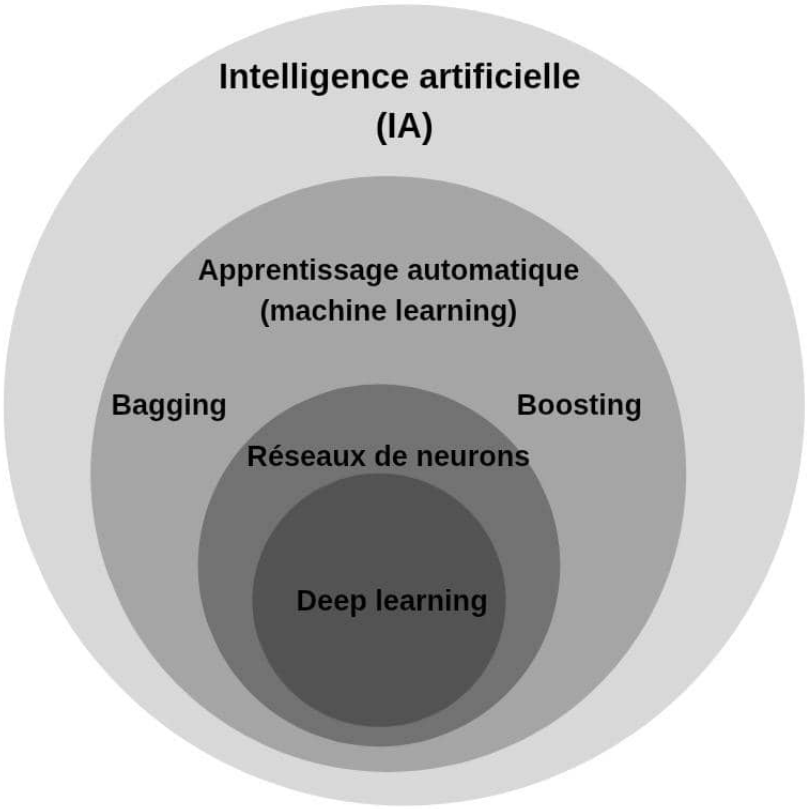


FIGURE 2.1 – Diagramme d'Euler de l'intelligence artificielle
[41]

2.3 Fondamentaux du Machine Learning

Les racines de l'apprentissage automatique sont les statistiques, qui peuvent également être considérées comme l'art d'extraire des connaissances des données. En particulier, des méthodes telles que la régression linéaire et les statistiques bayésiennes, qui datent déjà de plus de deux siècles, sont aujourd'hui encore au cœur de l'apprentissage automatique.

Un système ML typique utilise des algorithmes qui analysent un ensemble de données d'entrée pour extraire des modèles standard ou des régularités. Par la suite, ces modèles sont utilisés pour faire des prédictions ou prendre des décisions en fonction de nouvelles données.

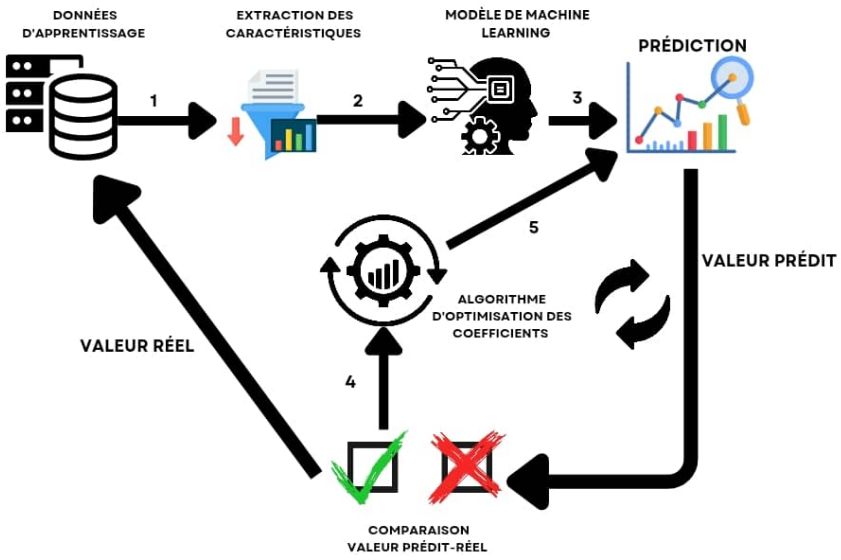


FIGURE 2.2 – Processus d'apprentissage pour le Machine Learning

Les domaines de l'apprentissage automatique sont souvent divisés en sous-domaines selon les types de problèmes qu'ils abordent. On peut les classer comme suit :

- **Apprentissage supervisé** : Utilise des données étiquetées pour entraîner un modèle à prédire une sortie.
- **Apprentissage non supervisé** : Analyse des données non étiquetées pour identifier des structures cachées.
- **Apprentissage par renforcement** : Un agent apprend par essais-erreurs en recevant des récompenses ou pénalités selon ses actions dans un environnement donné.

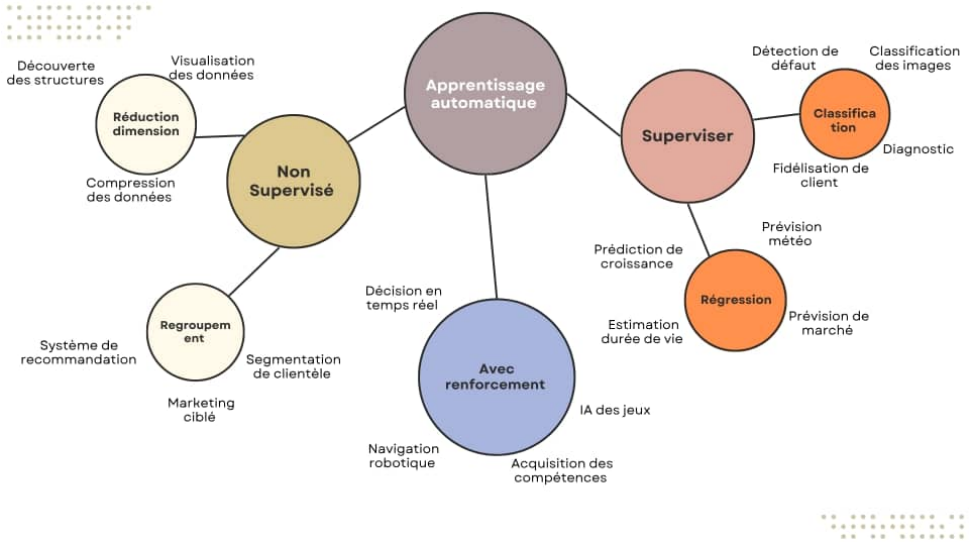


FIGURE 2.3 – Utilisations des types d'apprentissage automatique
[42]

2.4 Catégories d'algorithmes de Machine Learning

Il existe plusieurs catégories de machine learning dont trois catégories majeures sont : les algorithmes de 'Bagging', 'Boosting', et le 'Deep Learning' :

2.4.1 Deep Learning

Le deep learning (DL) est une branche du machine learning utilisant des réseaux de neurones artificiels. Ils sont formés d'une couche d'entrée, de plusieurs couches cachées et d'une couche de sortie. Chaque couche est reliée à la suivante par des connexions pondérées. Ce qui distingue les modèles profonds, c'est leur architecture complexe et leur capacité à extraire automatiquement des caractéristiques à partir de données brutes.

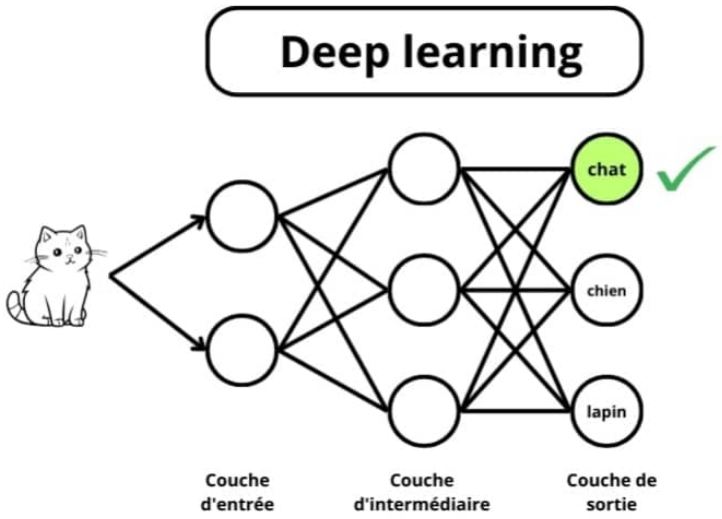


FIGURE 2.4 – Architecture d'un modèle de réseau de neurones profond

Parmi de nombreuses implémentations de modèles d'apprentissage profond, on sélectionne :

- **Réseaux de neurones convolutionnels (CNN)** : Les CNN sont des réseaux à propagation directe utilisés principalement pour l'analyse d'images, de signaux et de textes. Leur architecture repose sur des couches convolutionnelles pour extraire les caractéristiques, suivies de couches de regroupement et de classification. Ils préservent la structure spatiale des données, ce qui les rend efficaces pour la reconnaissance d'images et le traitement du langage.
- **Long Short-Term Memory (LSTM)** : Les LSTMs [37], une variante des RNN, intègrent des mécanismes de mémoire à long terme via des portes spécialisées. Ils surmontent les limites des RNN classiques et sont utilisés pour les séries temporelles, la modélisation de séquences et la reconnaissance vocale.
- **Réseaux neuronaux récurrents (RNN)** : Les RNN [37] traitent des données séquentielles en intégrant une mémoire des entrées précédentes grâce à des connexions récurrentes. Ils sont adaptés aux tâches temporelles comme la reconnaissance vocale ou la traduction, mais souffrent du problème de disparition du gradient pour les longues séquences.

2.4.2 Algorithmes de Boosting

Le boosting est une méthode d'apprentissage en ensemble qui combine plusieurs modèles faibles appelés weak learners pour former un modèle prédictif plus performant. Il améliore la précision en corrigeant les erreurs successives des modèles précédents. Utilisé en classification, régression ou classement, les algorithmes de boosting nécessitent un modèle de base, souvent un arbre de décision, qu'ils améliorent itérativement pour renforcer les performances globales.

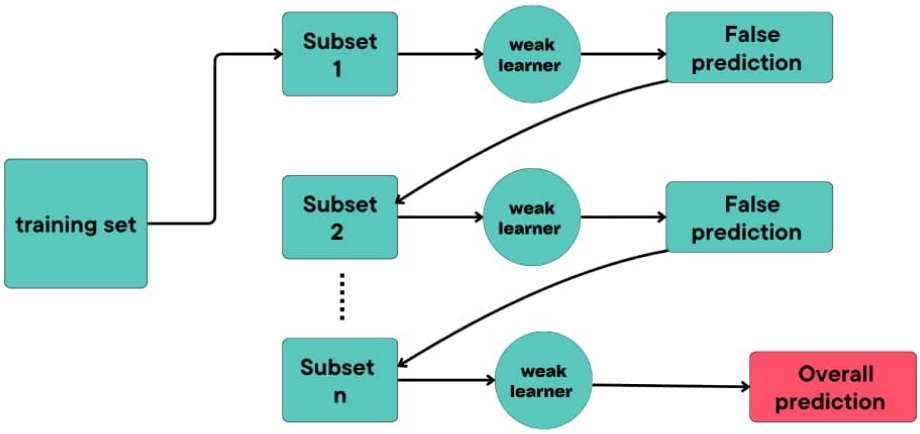


FIGURE 2.5 – Vue d'ensemble des algorithmes de Boosting

Parmi les algorithmes de Boosting les plus performants, on sélectionne :

- **Adaptive Boosting (AdaBoost)** : Proposé en 1995 [39], AdaBoost combine plusieurs apprenants faibles en leur attribuant des poids en fonction de leurs performances. Chaque nouveau classificateur est entraîné pour corriger les erreurs des précédents, ce qui rend le modèle adaptatif face aux erreurs successives.
- **Gradient Boosting (GBoost)** : Méthode d'ensemble construisant des modèles séquentiellement, [38]chaque nouvel arbre de décision corrigeant les erreurs des modèles précédents. Il minimise une fonction de perte via la descente de gradient, ce qui le rend efficace pour les tâches de classification et de régression.
- **Extreme Gradient Boosting (XGBoost)** : Amélioration du Gradient Boosting introduite en 2014, intégrant des techniques de régularisation pour éviter le surapprentissage [39]. Il attribue des poids aux données afin d'ajuster les prédictions à chaque itération, tout en optimisant les performances grâce à une implémentation efficace.

2.4.3 Les Algorithmes de Bagging (Les algorithmes classiques)

Le bagging, également connu sous le nom d'agrégation bootstrap, est une méthode d'apprentissage utilisée pour réduire la variance dans un ensemble de données bruyantes. En 1996, Leo Breiman a introduit l'algorithme d'ensachage qui comporte trois étapes de base :

- **Bootstrapping** : Création de plusieurs sous-ensembles d'entraînement par échantillonnage aléatoire avec remplacement, permettant d'obtenir des jeux de données diversifiés.
- **Entraînement en parallèle** : Chaque sous-ensemble est utilisé pour entraîner un modèle indépendant, ce qui permet un apprentissage parallèle efficace.

- **Agrégation** : Pour la classification, la prédiction finale est déterminée par vote majoritaire ; pour la régression, on calcule la moyenne des sorties de tous les modèles.

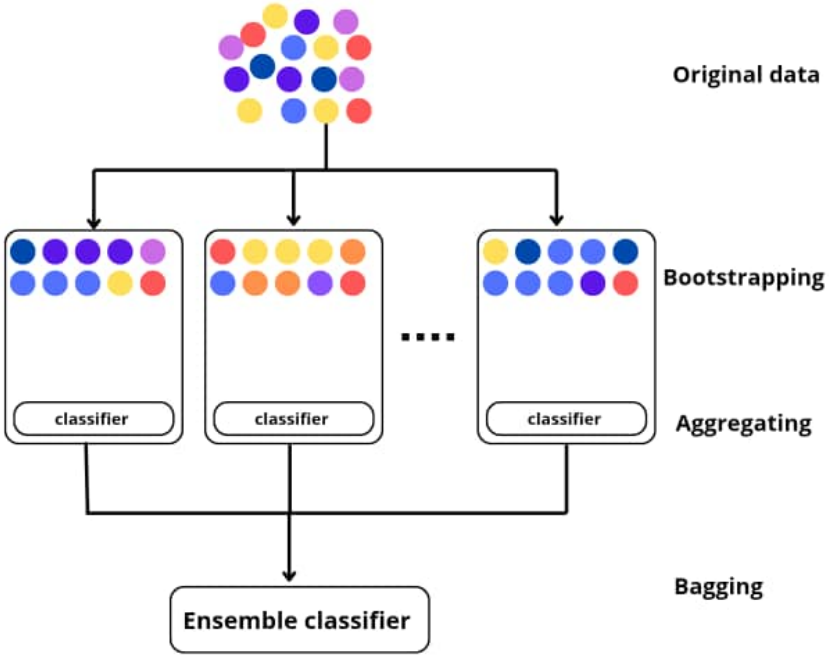


FIGURE 2.6 – Structure de bagging

- **Random Forest (RF)** : Est un algorithme d'ensemble basé sur plusieurs arbres de décision, construits sur des échantillons aléatoires avec sélection aléatoire de variables. Il combine leurs prédictions pour améliorer la précision et réduire le surajustement.
- **Arbre décisionnel (DT)** : Est un modèle supervisé structuré en arbre, où chaque nœud représente un test sur une variable. Il est interprétable mais sensible au surapprentissage sans élagage.
- **k-Nearest Neighbor (kNN)** : kNN classe une observation selon les k exemples les plus proches selon une distance choisie. Sa performance dépend du choix de k et de la normalisation des données.

2.5 Détection d'intrusion basée sur le Machine Learning

La détection d'intrusions basée sur l'IA est un domaine en pleine expansion. Elle repose sur l'utilisation de jeux de données publics ou privés.

2.5.1 Jeux de données utilisés

Les ensembles de données sont essentiels pour concevoir et évaluer les IDS, ils permettent l'entraînement, le test et la comparaison des modèles dans un cadre contrôlé. Comme le soulignent Khraisat [20] la qualité, la représentativité et l'accessibilité des jeux de données utilisés sont des facteurs clés pour déterminer l'efficacité des IDS.

- **Datasets publics** : En raison de leur accessibilité et de leur capacité à fournir des comparaisons reproductibles, les jeux de données publics sont les plus souvent utilisés dans la recherche.Le tableau des jeux données les plus utilisés est dans l'annexe : 5

Ces ensembles de données sont largement utilisés comme benchmarks pour évaluer les approches basées sur l'IA [21] ; [19]. Tandis que des jeux historiques comme KDDCup99 et NSL-KDD sont critiqués pour leur manque de réalisme, des datasets récents tels que UNSW-NB15 et CIC-IDS2018 sont mieux adaptés aux méthodes modernes, bien qu'ils requièrent un prétraitement avancé pour optimiser la performance des modèles.

- **Ensembles de données en temps réel et privés** : Les jeux de données publics, bien que précieux, présentent souvent des limites telles que le manque de diversité des scénarios ou le déséquilibre des classes. Pour y remédier, certains chercheurs génèrent leurs propres ensembles à partir d'environnements industriels ou de laboratoires spécialisés. Toutefois, comme l'indiquent Soman[21], ces ensembles privés sont rarement accessibles en raison de considérations liées à la confidentialité, à la sécurité ou à la propriété intellectuelle.

2.5.2 Attaques ciblant les infrastructures de contrôle industriel

Les systèmes DCS sont vulnérables à diverses cybermenaces selon les segments visés. Les unités centrales et les contrôleurs locaux peuvent être la cible d'attaques internes exploitant des failles organisationnelles comme les malwares. Les segments de communication sont exposés à des attaques externes comme le déni de service, compromettant l'intégrité et l'authenticité du système. Le tableau 2.1 regroupe les principaux types des attaques possibles.

TABLE 2.1 – Quelques types d'attaques ciblant les systèmes DCS

Type d'attaque	**Description**
Déni de service (DoS)	Perturbe la disponibilité des systèmes DCS en les saturant de trafic.
Hameçonnage (Phishing)	Attaques d'ingénierie sociale visant à obtenir des informations sensibles.
Injection SQL	Insertion de requêtes SQL malveillantes pour manipuler les bases de données des DCS.
Attaque de l'homme du milieu (MitM)	Interception de la communication entre les composants des DCS pour obtenir un accès non autorisé.
Reconnaissance	Collecte d'informations sur le système DCS avant une attaque.
Attaques par mot de passe	Tentatives de craquage de mots de passe pour accéder aux systèmes DCS.
Escalade de privilèges	Obtention de droits d'accès élevés pour exécuter des actions critiques sur le système DCS.

2.5.3 Travaux connexes

Dans cette section, on présente les travaux récents qui ont travaillé sur le ML pour la détection des intrusions. Ils sont regroupés dans le tableau 7.

Approches classiques dans les IDS industriels

On trouve plusieurs études qui ont évalué des algorithmes classiques comme les SVM, DT, RF, KNN et Naive Bayes pour la détection d'intrusions. Tamy et al. [36] ont testé ces modèles sur le dataset KDD'99, avec le Random Forest obtenant la meilleure précision (94%). De même, Banda et al. [29] ont comparé ces algorithmes sur NSL-KDD, confirmant la supériorité du RF avec 98,2% de précision. Par ailleurs, Aha Konye et al. [27] ont amélioré la détection en appliquant un test Chi² pour sélectionner les caractéristiques pertinentes, simplifiant le modèle et réduisant le bruit, avec une précision de 96,4%.

Approches de boosting pour la détection des intrusions

Des études ont évalué des algorithmes comme XGBoost, AdaBoost et Gradient Boosting pour la détection d'intrusions. Kundap et al. [33]ont proposé XG-ADICS pour un banc d'essai hydraulique ICS (données HAI), avec un F1 de 0,9986 et une précision de 99,96%. Upadhyay et al. [38]ont utilisé GBFS pour sélectionner 15 caractéristiques clés sur un dataset SCADA électrique appelé ORNL, obtenant 97,3% de précision et un taux de détection de 98,5% avec un faible FPR : 3,7. Okur et Dener et al. [34] ont testé XGBoost sur des données IIoT SCADA (WUSTL-2018-IIoT) et atteint 97,82% de précision en classification binaire, F1=96,86%.

Approches profondes pour la détection d'intrusion

Plusieurs études récentes ont exploré l'usage du deep learning pour améliorer les IDS. Salama [35] ont combiné DBN et SVM on NSL-KDD, atteignant plus de 90% de précision avec seulement 5 caractéristiques. Diaba et al.[32] ont utilisé un modèle hybride CNN-RNN sur UNSW-NB15 avec un taux de détection de 99,6% et un FAR < 1,5. Torres et al. ont appliqué LSTM sur CTU13-42/47, obtenant 97% de précision et un FAR très bas : 0,018. Enfin, Soman et al. [21]ont montré, dans une revue, que les approches DL : CNN, AE, LSTM, DNN dépassaient souvent les 98% sur divers jeux de données comme NSL-KDD et CICIDS2017.

Synthése :

L'analyse des travaux existants dans le domaine de la détection d'intrusions appliquée aux systèmes industriels montre une large diversité d'approches appartenant à trois grandes familles d'algorithmes : le bagging, le boosting et le deep learning. Chacune de ces catégories présente des atouts distincts, mais également des limites qui restreignent leur efficacité lorsqu'elles sont utilisées isolément.

Les méthodes de bagging, telles que le Random Forest et les arbres de décision, sont reconnues pour leur robustesse face au surapprentissage, leur simplicité d'interprétation et leur efficacité sur des données bruitées. Elles permettent des temps d'entraînement relativement courts, ce qui est avantageux dans des contextes industriels contraints. Toutefois, elles montrent rapidement leurs limites face à des attaques sophistiquées ou à des données hautement non linéaires, et nécessitent souvent une phase préalable d'ingénierie des caractéristiques pour optimiser les résultats.

Les approches de boosting, notamment XGBoost, AdaBoost et Gradient Boosting, ont démontré des performances exceptionnelles en termes de précision, notamment sur des jeux de données déséquilibrés ou complexes. Elles sont capables de modéliser finement des relations subtiles entre les caractéristiques, ce qui leur confère une forte capacité de discrimination. Cependant, ces modèles peuvent être sensibles au bruit si les paramètres ne sont pas soigneusement réglés, et leur entraînement peut devenir coûteux sur de très grands volumes de données.

Les techniques issues du deep learning se distinguent par leur capacité à apprendre automatiquement des représentations complexes, particulièrement utiles pour les données séquentielles des systèmes industriels . Leur performance sur des jeux de données volumineux est généralement supérieure à celle des modèles classiques. Néanmoins, elles requièrent des ressources computationnelles importantes, sont moins interprétables, et peuvent souffrir de surapprentissage en l'absence de régularisation ou de validation rigoureuse.

À la lumière de ces constats, il apparaît que l'intégration des points forts de ces différentes familles constitue une voie prometteuse. Plusieurs études récentes, Al-Abassi et al [28] ont combiné CNN, RNN et Autoencodeurs en ensemble sur NSL-KDD, atteignant plus de 99% de précision. Zhang et al. [39] ont associé XGBoost et AdaBoost avec une sélection rigou-

reuse des caractéristiques, obtenant 99,2% sur CICIDS2018 et UNSW-NB15. Vale Dalarmelina et al. [31]ont proposé TENNER, un modèle combinant XGBoost, RF, DT et KNN avec undersampling, atteignant 99,93% d'accuracy avec un faible taux de fausses alertes et un temps d'entraînement réduit (1h34 vs 156h). Enfin, Chih-Ta Lin et al. [30]ont développé une détection dynamique en ligne intégrant des modèles ML dans un réseau de boîtes de réaction pour une réponse rapide aux menaces.

Dans cette optique, notre contribution se distingue par le développement d'une solution hybride , afin de bénéficier à la fois de la puissance de modélisation et de la robustesse structurelle. Cette solution a été évaluée sur deux jeux de données de référence, ainsi que sur un jeu de données industriel réel provenant d'un système DCS de l'entreprise Sonelgaz.

2.6 Conclusion

Ce chapitre a permis d'examiner les principaux fondements théoriques de l'apprentissage machine et ses applications à la cybersécurité industrielle. En passant par la classification des algorithmes, le benchmark des ensembles de données et des attaques ciblant les systèmes de contrôle ainsi qu'une revue des travaux connexes, il a montré qu'aucune approche n'est strictement supérieure aux autres et que pour trouver la meilleure approche, il faut tester les trois catégories dans les mêmes conditions, évaluer et combiner les points forts des meilleurs modèles.

Dans le chapitre suivant, nous mettons en pratique notre approche proposée dans un contexte expérimental concret en évaluant la performance de nombreux algorithmes supervisés de ces différentes familles sur trois ensembles de données.

L'objectif est de fournir un système de détection d'intrusion adapté aux besoins du DCS tout en tenant compte des contraintes industrielles et des exigences de sécurité.

Chapitre 03

Evaluation des algorithms pour le dévelopement de la solution X-RF Shield

Chapitre 3

Evaluation des algorithms et dévelopement de la solution X-RF Shield

3.1 Introduction

Dans le deuxième chapitre, nous avons examiné en profondeur les principales approches existantes pour la détection d'intrusion en utilisant le machine learning. Ce chapitre expose la premiére partie de notre approche proposée pour la détection d'intrusions dans les DCS. Nous procédons à un benchmark comparatif de neuf algorithmes d'apprentissage automatique.

Cette évaluation repose sur des métriques de performance telles que la précision, le taux de détection, le taux de faux positifs et le F1-score sur les jeux de données : CIC-IDS2018, UNSW-NB15 et un jeu de données réel collecté lors de notre stage à Sonelgaz.

À l'issue de cette analyse, les deux algorithmes les plus performants sont retenus pour construire une solution visant à améliorer l'efficacité de la détection.

3.2 Méthodologie

Notre approche proposée pour la détection d'intrusions dans les systèmes industriels s'articule autour de trois phases complémentaires : l'évaluation comparative, la construction de la solution combinée, puis la simulation en temps réel. Dans ce premier chapitre nous allons présenter la premiére phase .

- **Phase 1 : Évaluation comparative de neuf algorithmes** : La première étape de notre méthodologie consiste à sélectionner et évaluer neuf algorithmes d'intelligence artificielle répartis en trois catégories : CNN, RNN, LSTM, XGBoost, Gradient Boosting, AdaBoost, Random Forest, Decision Tree et k-Nearest Neighbors. Ces algorithmes ont été choisis en se basant sur leur performance démontrée dans les études récentes. Pour évaluer leur efficacité, nous les avons entraînés et testés sur trois jeux de données représentatifs : CIC-IDS2018, UNSW-

NB15 et un jeu de données réel collecté dans le cadre de notre stage chez Sonelgaz, il contient des variables d'une turbine à gaz. Avant l'entraînement, un prétraitement rigoureux a été appliqué : nettoyage des données, sélection des caractéristiques importantes et normalisation via l'analyse de la plage de valeurs. Chaque modèle a ensuite été évalué à l'aide de métriques standard : précision, exactitude, taux de détection, taux de faux positifs, et F1-score.

- **Phase 2 : Construction de la solution hybride XRF-Shield** : Suite aux résultats obtenus dans la première phase, nous avons retenu les deux algorithmes les plus performants. Ces deux modèles ont été combinés pour construire une solution nommée XRF-Shield, qui vise à exploiter les avantages de chacun .

- **Phase 3 : Simulation en temps réel de XRF-Shield** : Dans cette phase finale, nous avons conçu un système de simulation pour évaluer la solution XRF-Shield dans un environnement ICS virtuel. Les données utilisées proviennent d'une turbine à gaz, avec des attaques injectées simulant des cybermenaces réalistes. Afin d'approcher les conditions d'un système industriel réel, les données ont été prétraitées en ajoutant du bruit. Cette simulation permet d'évaluer XRF-Shield sur sa capacité de détection en temps réel et sa réactivité dans un contexte proche du terrain.

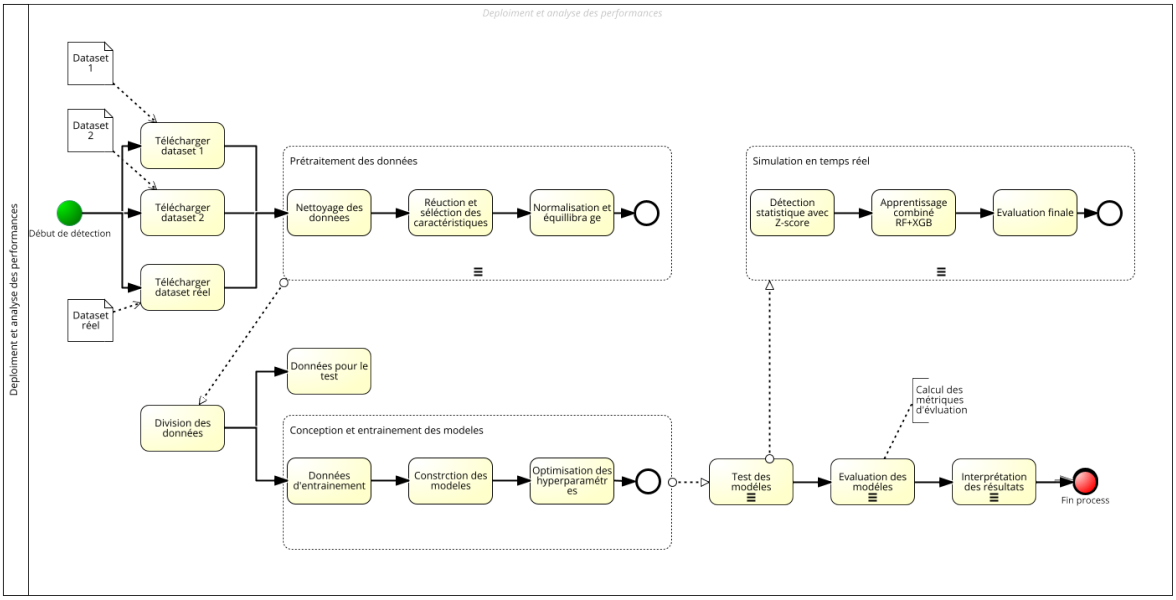


FIGURE 3.1 – Processus de notre approche

Dans le suivant , nous allons présenter la premiére phase de notre processus .

3.3 Phase 1 : Évaluation comparative de neuf algorithmes

3.3.1 Description des jeux de données utilisées

Afin d'assurer une évaluation approfondie et précise de la performance des modèles de détection d'intrusion, nous avons utilisé trois ensembles de données différents pour cette expérience : un ensemble de données réel collecté lors d'un stage, et deux ensembles téléchargés.

Premier ensemble : UNSW-NB15

L'ensemble de données UNSW-NB15, publié en 2015 par le Centre australien pour la cybersécurité (ACCS) de l'Université de la Nouvelle-Galles du Sud, a été créé pour répondre aux limitations des jeux de données précédents en incluant des attaques plus récentes et en améliorant la distribution des classes.

Le trafic réseau a été collecté dans un environnement de test simulant un réseau d'entreprise, comprenant à la fois le trafic régulier et neuf formes différentes d'attaques : vers, shellcode, reconnaissance, analyse, backdoor, DoS, exploits, générique et fuzzers. UNSW-NB15 contient 49 attributs et 2 540 044 enregistrements, répartis en plusieurs catégories : caractéristiques de base, statistiques temporelles, caractéristiques de contenu, attributs dérivés et statistiques de session. Chaque instance est associée à un label indiquant le type de trafic et, le cas échéant, le type d'attaque. les caractéristiques de cet ensemble de données sont présentés sur dans le tableau 3.1

TABLE 3.1 – Extrait des attributs de l'ensemble UNSW-NB15

Variable	Désignation	Type
srcip	Adresse IP source	adresse IP
sport	Port source	entier
dstip	Adresse IP destination	adresse IP
dport	Port destination	entier
proto	Protocole	texte
state	État de la connexion	texte
dur	Durée du flux	secondes
sbytes	Octets envoyés	octets
dbytes	Octets reçus	octets

Deuxième ensemble : CIC-IDS2018

Nous avons utilisé l'ensemble de données CIC-IDS2018, publié par le Canadian Institute for Cybersecurity (CIC). Cet ensemble a été créé pour simuler des activités incluant à la fois le trafic normal et une variété d'attaques réalistes sur un réseau d'entreprise réel. L'ensemble

de données comprend sept scénarios d'attaque : déni de service distribué (DDoS), force brute, heartbleed, botnet, attaques sur le web et infiltration.

Chaque type d'attaque a été observé dans un environnement réel attaqué par une infrastructure malveillante composée de cinquante machines, et chaque élément est identifié par son type de trafic (normal ou attaque) ainsi que le type d'attaque spécifique. Les mesures ont été obtenues par le CICFlowMeter. les caractéristiques de cet ensemble de données sont présentés sur dans le tableau : 3.2

TABLE 3.2 – Caractéristiques typiques de l'ensemble CIC-IDS2018

Variable	**Désignation**	**Unité**
Flow Duration	Durée du flux	millisecondes (ms)
Total Fwd Packets	Nombre total de paquets sortants	unité
Total Backward Packets	Nombre total de paquets entrants	unité
Flow Bytes/s	Taux de débit du flux	octets/seconde
Fwd Packet Length Mean	Longueur moyenne des paquets sortants	octets
Bwd Packet Length Mean	Longueur moyenne des paquets entrants	octets
Flow IAT Mean	Temps inter-arrivée moyen des paquets	millisecondes (ms)
Label	Étiquette (attaque ou normal)	classe (texte)

Troisième ensemble : TG Sonelgaz

Le dernier jeu de données a été collecté lors de notre stage à Sonelgaz : le système DCS MARK VI, développé par General Electric et utilisé dans la centrale électrique de Koudiet Eddraouech.

La turbine à gaz TG étant l'un des équipements de base dans le processus de production d'électricité, nous avons choisi d'axer notre analyse sur cet équipement, car la turbine à gaz combine plusieurs paramètres critiques, y compris la température, la pression, la consommation de carburant et les indicateurs de commande et d'état.

Les données ont été recueillies à une fréquence d'une milliseconde à partir de l'historique du système MARK VI, offrant une résolution temporelle extrêmement élevée et permettant un suivi précis des fluctuations dynamiques du système. L'ensemble de données se compose de 20 000 lignes avec 20 variables mesurables.Le tableau de jeu donnée TG Sonelgaz est dans l'annexe : 6

3.3.2 Prétraitement des données

Nous décrivons dans cette section les procédures de prétraitement appliquées aux jeux de données :

Nettoyage des données

Le nettoyage vise à éliminer les anomalies qui peuvent perturber les modèles. Cela inclut la suppression des valeurs manquantes (`NaN`), infinies (`inf`, `-inf`) ou non numériques, ainsi que la correction des doublons. Les colonnes entièrement vides ont été supprimées, et les valeurs manquantes restantes ont été remplacées par zéro pour éviter les erreurs lors de l'entraînement.

Sélection des caractéristiques

Afin de réduire la dimension du jeu de données tout en conservant les variables les plus importantes pour la modélisation, la sélection des caractéristiques est une étape cruciale. Une analyse basée sur la matrice de corrélation 3.2 a été effectuée pour évaluer la relation entre les différentes variables de l'ensemble de données.

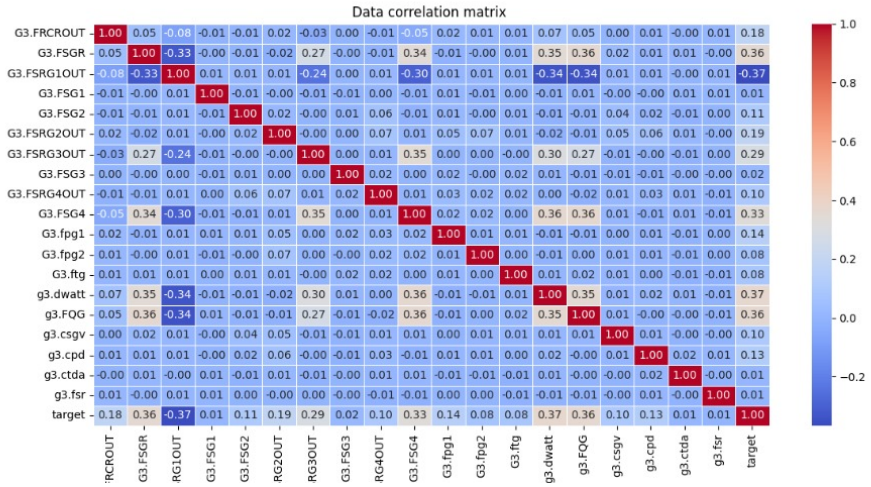


FIGURE 3.2 – Matrice de corrélation pour le jeu de données TG Sonelgaz

D'après les informations fournies par cette matrice, une sélection des caractéristiques a été réalisée à l'aide de la méthode `VarianceThreshold` pour réduire le bruit et la redondance dans le dataset, en éliminant les colonnes à variance nulle, qui n'apportent aucune information discriminative.

```
from sklearn.feature_selection import VarianceThreshold
selector = VarianceThreshold(threshold=0.0)
X_var = selector.fit_transform(X_scaled_df)
var_columns = X_scaled_df.columns[selector.get_support()]
```

FIGURE 3.3 – Utilisation de VarianceThreshold

Cette analyse de corrélation a permis de supprimer les colonnes fortement corrélées (corrélation $> 0,8$).

Normalisation et équilibrage des données

Les systèmes de contrôle industriels (ICS) génèrent des mesures hétérogènes sur de grandes quantités physiques comme la pression, la température et la position, caractérisées par des unités et des valeurs distinctes. Cette hétérogénéité peut introduire des biais importants dans les modèles d'apprentissage automatique, en particulier pour ceux sensibles aux amplitudes variables. Pour assurer l'homogénéité et la convergence stable de l'algorithme, les données sont normalisées à l'aide de `StandardScaler`, qui projette chaque variable dans un intervalle entre $[0, 1]$, tout en maintenant une distribution des valeurs.

```
from sklearn.preprocessing import StandardScaler
# 🔹 Normalize
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# 🔹 Apply SMOTE
smote = SMOTE(random_state=42)
X_train_sm, y_train_sm = smote.fit_resample(X_train, y_train)
```

FIGURE 3.4 – Normalisation et équilibrage des données

De plus, un déséquilibre entre les classes a été observé dans les jeux de données, le trafic normal étant majoritaire par rapport aux attaques. La méthode SMOTE (*Synthetic Minority Oversampling Technique*) a été appliquée sur l'ensemble d'entraînement pour générer des exemples synthétiques de la classe minoritaire (attaques). Cela permet d'améliorer la capacité des modèles à détecter les attaques sans biaiser l'ensemble de test.

3.3.3 Division des données

Afin de procéder à l'entraînement et à l'évaluation des modèles d'apprentissage, chaque ensemble de données a été divisé en deux sous-ensembles : 70 % des données ont été affectées à la phase d'entraînement, permettant au modèle d'apprendre des exemples donnés, et 30 % restants ont été réservés pour la phase de test, offrant une évaluation impartiale de la capacité du modèle à généraliser.

```
7  from sklearn.model_selection import train_test_split
8  # 🔹 Split
9  X_train, X_test, y_train, y_test = train_test_split(X_final_df, y, test_size=0.3, random_state=42, stratify=y)
```

FIGURE 3.5 – Division des données

3.3.4 Conception et entraînement des modèles

Cette section décrit l'approche expérimentale utilisée pour développer les modèles. Elle s'articule autour de trois axes : la sélection des algorithmes divisés en trois catégories pour

couvrir un large éventail de stratégies d'apprentissage, la configuration d'hyperparamètres optimisés, et l'évaluation des modèles à l'aide de plusieurs métriques adaptées au contexte de la cybersécurité industrielle.

Construction des modèles

Après le prétraitement des données, nous avons construit les modèles dans trois catégories :

Modèles de bagging

Les algorithmes de cette catégorie, à savoir les arbres de décision (Decision Tree, DT), les forêts aléatoires (Random Forest, RF) et les k-plus proches voisins (k-NN), ont été implémentés à l'aide de la bibliothèque Scikit-learn. Pour les modèles DT et RF, les hyperparamètres tels que la profondeur maximale et le nombre d'estimateurs ont été définis de manière empirique afin d'optimiser la performance tout en limitant le surapprentissage. Le modèle k-NN a été configuré avec un nombre optimal de voisins (k=5), sélectionné après expérimentation. Chaque modèle a été entraîné avec la méthode `fit()` sur les données normalisées et équilibrées par SMOTE, puis évalué par `predict()` à l'aide des métriques standards.

Modèles de boosting

Les modèles de cette catégorie comprennent AdaBoost, Gradient Boosting et XGBoost, tous implémentés via les bibliothèques Scikit-learn et XGBoost. Ces techniques reposent sur le principe de l'apprentissage séquentiel, où plusieurs estimateurs faibles sont combinés pour former un modèle robuste. Chaque modèle a été construit avec 50 estimateurs et des hyperparamètres standards optimisés. L'entraînement a été réalisé à l'aide de la méthode `fit()` sur les données équilibrées par SMOTE, et les prédictions obtenues via `predict()` ont permis d'évaluer les performances à travers des métriques classiques. Cette approche vise à exploiter la puissance des algorithmes d'agrégation pour améliorer la précision et la généralisation du modèle.

Modèles de deep learning

Dans cette catégorie, trois architectures ont été explorées : CNN, LSTM et RNN, chacune implémentée à l'aide de la bibliothèque Keras. Les données ont d'abord été restructurées sous forme de tenseurs 3D pour correspondre aux entrées attendues par les réseaux séquentiels (samples, features, 1). Les modèles sont construits selon une architecture typique pour les tâches de classification binaire : une couche d'entrée, suivie d'une couche principale spécifique à l'architecture convolutionnelle pour CNN, mémoire courte pour LSTM, ou récursive pour RNN, puis de couches Dense avec Dropout pour la régularisation, et une couche finale à activation sigmoïde. Chaque modèle est compilé avec l'optimiseur Adam, une fonction de perte binary crossentropy, et la métrique accuracy. L'entraînement s'effectue sur 10 époques avec un batch size de 64, via la méthode `fit()`.

3.3.5 Les hyperparamètres

Un ensemble d'hyperparamètres a été soigneusement sélectionné pour une performance optimale. Un processus d'ajustement approfondi a permis d'équilibrer l'efficacité de la formation, la vitesse de convergence et la généralisation des modèles.Le tableau 3.3 présente les hyperparamétres selectionné :

TABLE 3.3 – Résumé des hyperparamètres

Catégorie	**Paramètre**	**Valeur**
Modèles de Deep Learning		
DL Models	Optimizer	Adam
	Loss Function	Binary Cross-Entropy
	Batch Size	64
	Epochs	10
	Hidden Activation	ReLU
	Output Activation	Sigmoid
Modèles de Boosting		
AdaBoost	Estimators	50
	Random State	42
	Weak Learner	DecisionTreeClassifier (max_depth=1)
XGBoost	Estimators	50
	Random State	42
	Objective/Evaluation	Binary Logistic / Log-Loss
Gradient Boosting	Estimators	50
	Learning Rate	0.1
	Loss Function	Log-Loss
	Max Depth	3
Modèles traditionnels		
Random Forest	n_estimators	200
	Criterion	Gini
	Max Depth	None
	Min Samples Split	10
Decision Tree	Criterion	Entropy
	Max Depth	None
k-NN	n_neighbors	5

3.3.6 Métriques d'évaluation

L'évaluation quantitative de notre système repose sur l'analyse de quatre paramètres fondamentaux, qui constituent les éléments de base de la matrice de confusion 3.6 :

- **True Positive (TP)** : Nombre d'attaques qui ont été correctement identifiées.
- **True Negative (TN)** : Total de comportements normaux correctement détectés.
- **False Positive (FP)** : Quantité de faux positifs, c'est-à-dire d'alertes erronées.
- **False Negative (FN)** : Nombre d'attaques non identifiées.

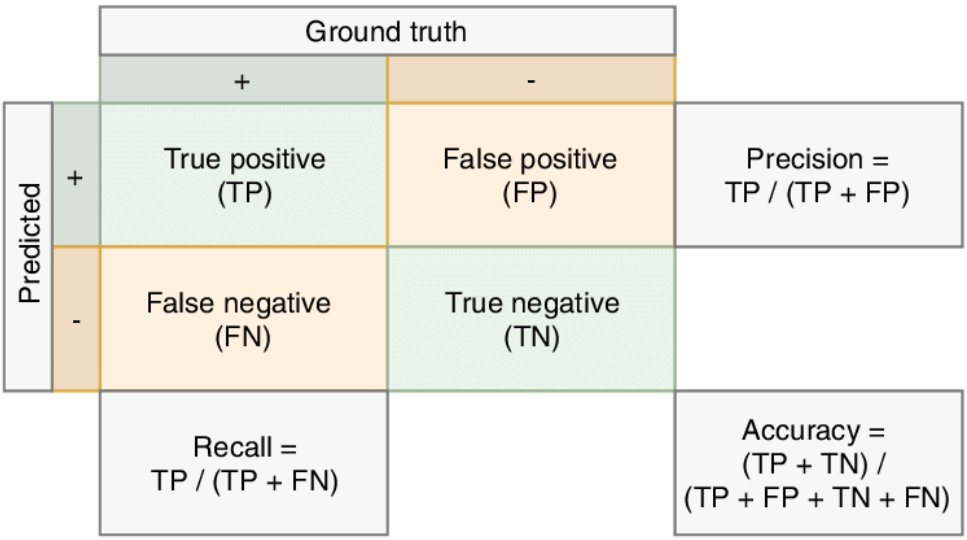


FIGURE 3.6 – Matrice de confusion

À partir de cette matrice, plusieurs métriques d'évaluation essentielles peuvent être dérivées pour mesurer la performance du modèle :

- **Accuracy (Exactitude)** : Calcule le pourcentage global de prédictions précises, en tenant compte des comportements normaux correctement identifiés (TN) et des attaques détectées (TP).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.1}$$

- **Precision (Précision)** : Démontre la capacité du modèle à prédire une intrusion seulement quand elle se produit réellement. Une précision élevée indique un faible taux de faux positifs (FP).

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.2}$$

- **Recall (Rappel ou Sensibilité)** : Évalue la capacité du modèle à identifier toutes les attaques réelles. Un faible rappel indique que certaines intrusions passent inaperçues (FN), ce qui constitue un risque sérieux.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.3}$$

- **F1-Score** : Équilibre la précision et le rappel, particulièrement utile lorsque les classes sont déséquilibrées.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

- **False Positive Rate (FPR)** : Quantifie le pourcentage de comportements normaux classés par erreur comme des attaques. Un FPR élevé peut augmenter les alertes et impacter la productivité.

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} \tag{3.5}$$

En plus des métriques de la matrice de confusion, des représentations graphiques sont utilisées :

- **Courbe ROC (Receiver Operating Characteristic)** : Trace le taux de vrais positifs (TPR) en fonction du taux de faux positifs (FPR). L'aire sous la courbe (AUC) proche de 1 indique une bonne distinction des classes, tandis qu'une AUC proche de 0,5 indique un rendement moyen.

- **Courbe PRC (Precision-Recall Curve)** : Indique l'évolution de la précision en fonction du rappel. Elle est plus instructive que la courbe ROC dans les cas de déséquilibre de classes. L'aire sous la courbe (AUPR) est utilisée pour résumer les performances.

3.3.7 Outils et environnements de développement

L'expérimentation a été réalisée dans un environnement de développement basé sur Python, en utilisant l'IDE PyCharm, sur un PC HP i3 10ème génération avec SSD 256 Go et 8 Go de RAM.

3.3.8 Bibliothèques utilisées

Les bibliothèques suivantes ont été utilisées pour différentes parties de la construction des modèles :

- **Pandas** : Manipulation et chargement des datasets.

- **NumPy** : Opérations mathématiques sur les tableaux.

- **Scikit-learn** : Implémentation des modèles classiques , boosting , normalisation (`StandardScaler`), sélection des caractéristiques (`VarianceThreshold`), et calcul des métriques (accuracy, precision, recall, F1, ROC-AUC).

- **XGBoost** : Implémentation du modèle XGBoost.

- **TensorFlow/Keras** : Construction et entraînement des modèles de deep learning.

- **Matplotlib/Seaborn** : Visualisation des résultats (matrices de confusion, courbes ROC, courbes Precision-Recall, histogrammes FPR).
- **Imblearn (SMOTE)** : Équilibrage des classes via suréchantillonnage de la classe minoritaire.
- **Scikit-learn.decomposition.PCA** : Réduction dimensionnelle pour la visualisation.

3.4 Analyse de Phase 1 : Évaluation comparative de neuf algorithmes

Dans cette partie nous allons présenter les résultats obtenus pour chaque dataset :

3.4.1 Jeu de données CIC-IDS 2018

Le tableau 3.4 présente les résultats détaillés des métriques d'évaluation des différents modèles appliqués au dataset CIC-IDS2018, classés selon trois catégories : bagging, boosting et deep learning. Un histogramme comparatif des différents modèles (figure 3.7) offre une visualisation synthétique des résultats obtenus.

TABLE 3.4 – Résultats expérimentaux sur le dataset CIC-IDS2018

Catégorie	Modèle	Train Acc (%)	Test Acc (%)	Précision (%)	Rappel (%)	F1-Score (%)
Bagging	Random Forest	100.00	99.99	99.99	99.97	99.98
	Decision Tree	100.00	99.99	99.98	99.97	99.97
	KNN	99.98	99.94	99.49	99.83	99.66
Boosting	AdaBoost	99.98	99.98	99.85	99.94	99.89
	Gradient Boosting	99.94	99.97	99.82	99.87	99.85
	XGBoost	99.99	99.99	99.98	99.96	99.97
Deep Learning	CNN	99.92	99.98	99.91	99.84	99.87
	RNN	99.66	99.63	95.91	99.70	97.77
	LSTM	64.39	96.95	77.63	87.56	82.29

Les résultats montrent que la majorité des modèles atteignent des performances remarquablement élevées sur ce dataset, avec des accuracies dépassant souvent les 99 %. Parmi eux, les algorithmes d'ensemble comme `Random Forest` et `XGBoost` se démarquent par leur capacité à maintenir un équilibre optimal entre précision et rappel, traduisant une détection efficace des attaques avec un taux minimal de fausses alertes. Les approches basées sur le deep learning `CNN`,

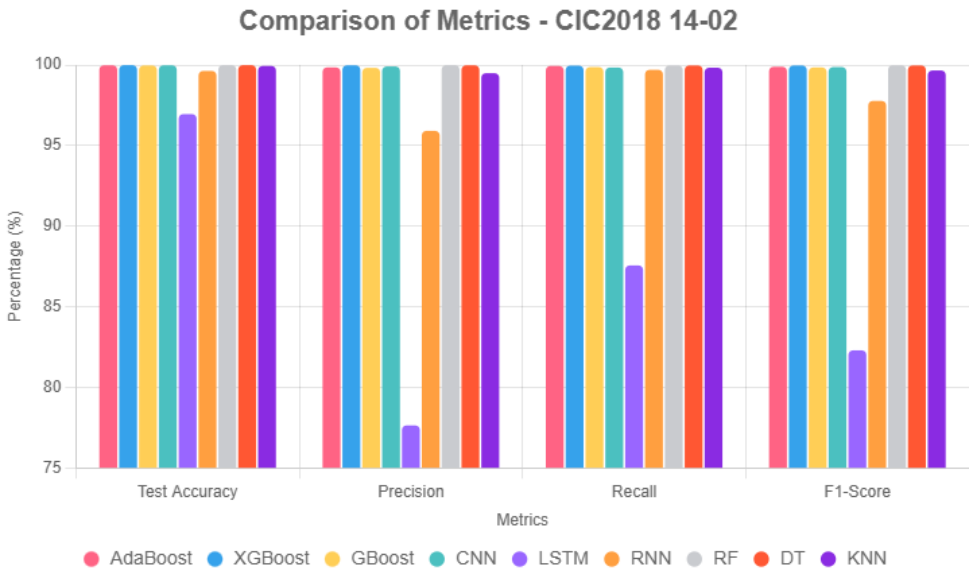


FIGURE 3.7 – Histogramme comparatif des performances des modèles sur CIC-IDS2018

`RNN`, `LSTM` affichent également de bons résultats, bien que le modèle `LSTM` présente des performances moindres, possiblement en raison de la complexité temporelle de ses architectures.

Le tableau 3.5 représente nos résultats obtenus sur CIC-IDS 2018 comparés avec ceux des travaux récents .

TABLE 3.5 – Comparaison des performances sur le dataset CIC-IDS2018

Reference	Model	Acc (%)	P (%)	R (%)	F1 (%)
[40]	RF	99.97	–	98.88	–
	XGBoost	99.77	–	–	94.94
[41]	DT	99.94	–	–	–
	Extra Trees	99.94	–	–	–
	RF	99.93	–	–	–
	XGB	98.87	–	–	–
[42]	RF-PCA (Z-score)	99.61	–	–	–
	XGBoost-PCA (Z-score)	99.77	–	–	–
Cette étude	RF	100.00	99.99	99.99	99.98
	DT	100.00	99.99	99.98	99.97
	kNN	99.98	99.94	99.49	99.66
	AdaBoost	99.98	99.98	99.85	99.89
	GBoost	99.94	99.97	99.82	99.85
	XGBoost	99.99	99.99	99.98	99.97
	CNN	99.92	99.98	99.91	99.87
	RNN	99.66	99.63	95.91	97.77
	LSTM	64.39	96.95	77.63	82.29

Nos résultats dépassent ceux des travaux récents, notamment [41] pour RF , DT et XGB, en atteignant une accuracy parfaite de 100%. Cela confirme la performance remarquable de notre solution, aussi bien en précision, rappel et F1-score, sur un jeu de données aussi varié que CIC-IDS2018.

3.4.2 Jeu de données : UNSW-NB15

Après avoir appliqué nos modèles au dataset UNSW-NB15, nous avons regroupé les résultats dans le tableau 3.6. Un histogramme comparatif des performances est présenté dans la figure 3.8.

On remarque que l'ensemble des modèles atteint une accuracy très élevée (> 99 %). En ce qui concerne la précision, les modèles basés sur le boosting tels que `XGBoost` et `Gradient Boosting`, ainsi que le bagging, notamment `Random Forest`, affichent des scores supérieurs à 90 %. Pour le rappel, presque tous les modèles dépassent les 99 %, ce qui témoigne de leur capacité à capturer presque toutes les attaques présentes. Concernant le F1-score, qui combine pré-

TABLE 3.6 – Résultats des modèles sur le dataset UNSW-NB15

Catégorie	**Modèle**	**Train Acc (%)**	**Test Acc (%)**	**Précision (%)**	**Rappel (%)**	**F1-Score (%)**
Bagging	Random Forest	100.00	99.80	94.87	99.24	97.01
	Decision Tree	100.00	99.80	96.26	97.73	96.99
	KNN	99.78	99.50	86.97	99.32	92.74
Boosting	AdaBoost	99.65	99.32	82.39	100.00	90.34
	XGBoost	99.88	99.75	93.32	99.36	96.25
	Gradient Boosting	99.70	99.40	84.32	99.96	91.48
Deep Learning	CNN	99.72	99.46	85.48	99.98	92.17
	LSTM	99.66	99.41	84.45	99.91	91.53
	RNN	99.69	99.32	82.46	99.95	90.37

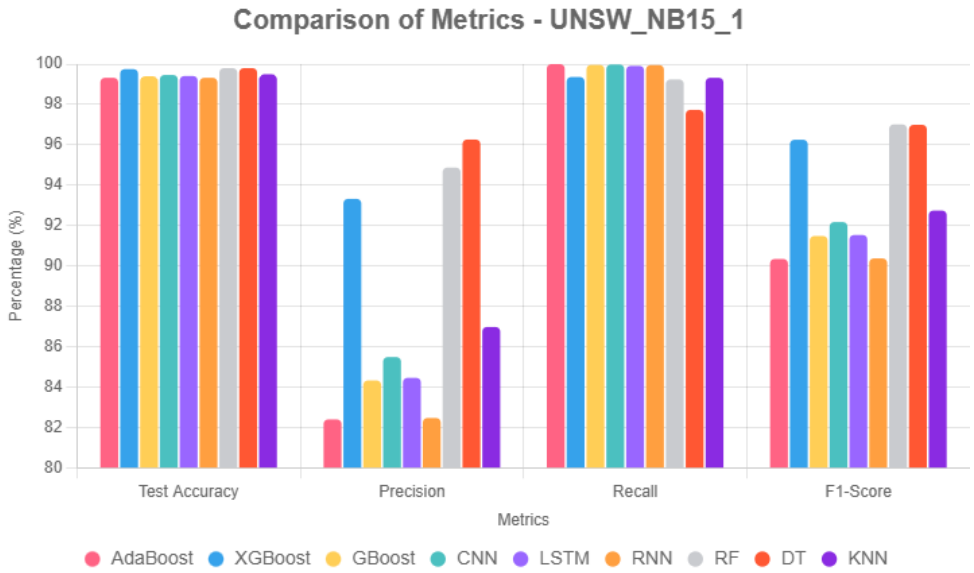


FIGURE 3.8 – Histogramme comparatif des performances des modèles sur UNSW-NB15

cision et rappel, on observe des performances équilibrées, notamment pour `XGBoost`, `Random Forest` et `CNN`, qui dépassent les 97 %. Ces résultats confirment que les trois familles testées boosting, bagging, deep learning apportent chacune des atouts spécifiques, et que globalement, elles atteignent un haut niveau de performance. Le tableau 3.7 présente une comparaison de notre étude avec des travaux récents sur le UNSW-NB15.

TABLE 3.7 – Comparaison des performances sur le dataset UNSW-NB15

Référence	**Modèle**	**Acc (%)**	**P (%)**	**R (%)**	**F1 (%)**
[43]	RF + Extra Trees + Oversampling	99.95	–	–	–
[44]	RF (Spark + PCA)	99.94	–	–	–
Cette étude	Random Forest	100.00	99.80	94.87	99.24
	Decision Tree	100.00	99.80	96.26	97.73
	KNN	99.78	99.50	86.97	99.32
	AdaBoost	99.65	99.32	82.39	100.00
	XGBoost	99.88	99.75	93.32	99.36
	Gradient Boosting	99.70	99.40	84.32	99.96
	CNN	99.72	99.46	85.48	99.98
	LSTM	99.66	99.41	84.45	99.91
	RNN	99.69	99.32	82.46	99.95

Notre modèle Random Forest a atteint 100% d'accuracy sur UNSW-NB15, dépassant les résultats récents de Talukder [43] (99.95%) et Bagui [44] (99.94%). Cela confirme l'efficacité et la compétitivité de notre approche, sans recourir à des techniques complexes comme le sur-échantillonnage ou la réduction de dimension.

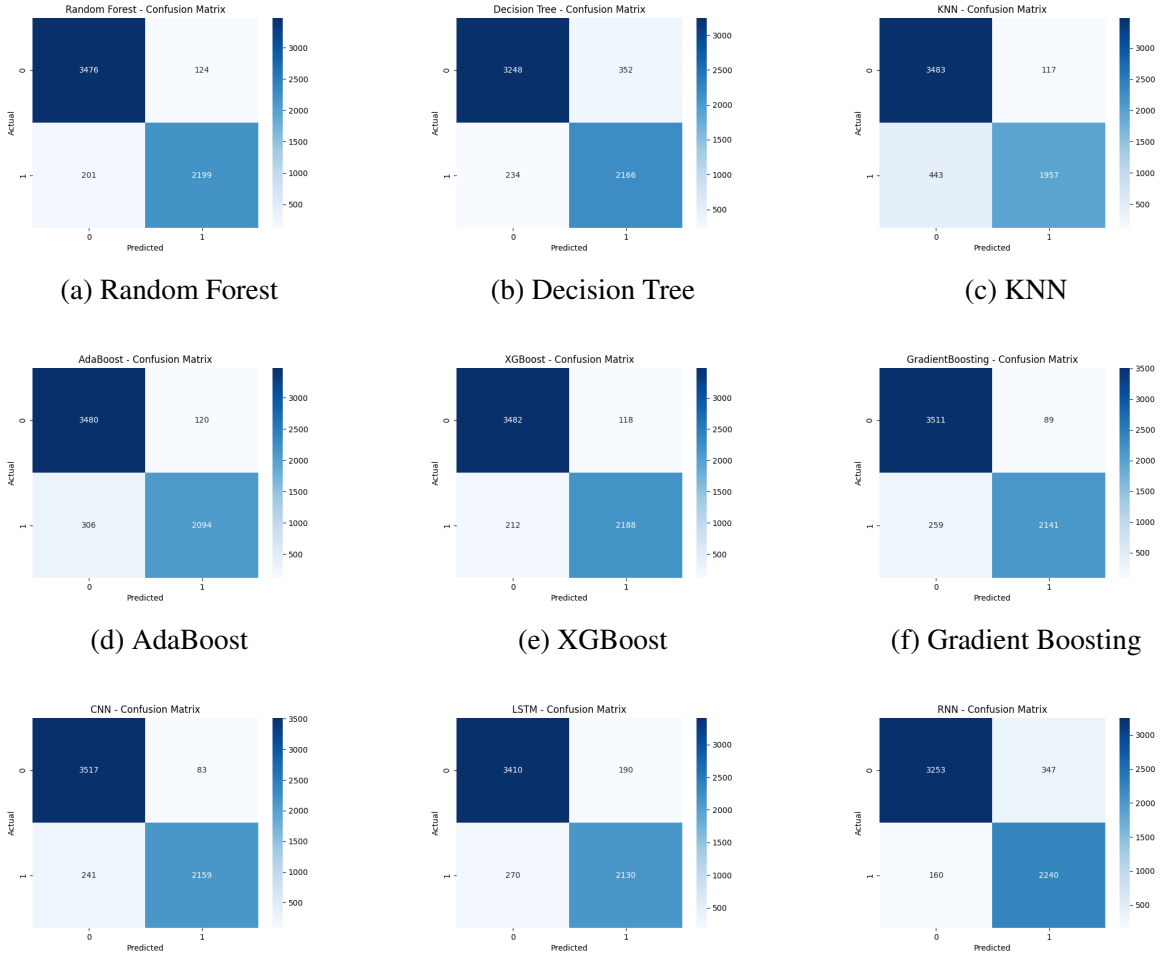
Pourquoi notre solution dépasse l'existant ?

Bien que plusieurs travaux antérieurs aient obtenu de bons résultats sur les jeux de données CIC-IDS2018 et UNSW-NB15, notre étude se distingue par des performances supérieures, notamment en termes d'accuracy, de F1-score et de rappel. Cette amélioration s'explique par un prétraitement rigoureux des données, une sélection fine des caractéristiques, et une optimisation méthodique des hyperparamètres. Ces choix méthodologiques ont permis de maximiser la capacité de détection tout en minimisant les erreurs, confirmant ainsi la robustesse de notre approche.

3.4.3 Jeu de données de Sonelgaz

En examinant les matrices de confusion des neuf modèles , nous pouvons analyser comment chaque modèle se comporte en termes de prédictions précises et d'erreurs de classification pour

chaque classe. Ces matrices, présentées dans la figure 3.9, fournissent une vue détaillée des résultats obtenus, permettant de comparer les performances des modèles en déterminant pour quelles classes ils sont plus ou moins spécifiques.



(a) Random Forest (b) Decision Tree (c) KNN

(d) AdaBoost (e) XGBoost (f) Gradient Boosting

(g) CNN (h) LSTM (i) RNN

FIGURE 3.9 – Matrices de confusion des modèles sur le dataset de Sonelgaz

Les matrices de confusion pour les neuf modèles appliqués au dataset industriel sont présentées dans la figure 3.9. Chaque matrice illustre les performances de classification pour les classes normale et anomale, permettant d'évaluer le nombre de vrais positifs (TP), faux positifs (FP), vrais négatifs (TN) et faux négatifs (FN). Ces visualisations complètent les métriques du tableau 3.8 en offrant une perspective détaillée sur les erreurs de classification.

Tous les modèles ont atteint des performances élevées, avec des valeurs de précision, de rappel et de score F1 souvent supérieures à 95 %. Cependant, les modèles `Random Forest` et `XGBoost` se distinguent avec des valeurs de précision, de rappel et de score F1 élevées.

TABLE 3.8 – Résultats des modèles sur le dataset Sonelgaz

Catégorie	Modèle	Train Acc (%)	Test Acc (%)	Précision (%)	Rappel (%)	F1-Score (%)	FPR
Bagging	Random Forest	100.00	94.58	94.66	91.62	93.11	0.1197
	Decision Tree	100.00	90.23	86.02	90.25	88.08	0.3659
	KNN	95.30	90.66	94.35	81.54	87.48	0.2001
Boosting	AdaBoost	91.98	92.90	94.57	87.25	90.76	0.0299
	XGBoost	97.25	94.50	94.88	91.16	92.98	0.1144
	Gradient Boosting	94.02	94.20	96.01	89.21	92.48	0.2013
Deep Learning	CNN	93.75	94.58	96.97	89.25	92.95	0.1779
	LSTM	90.80	91.73	90.72	88.38	89.53	0.2192
	RNN	93.59	93.82	97.38	86.88	91.83	0.1445

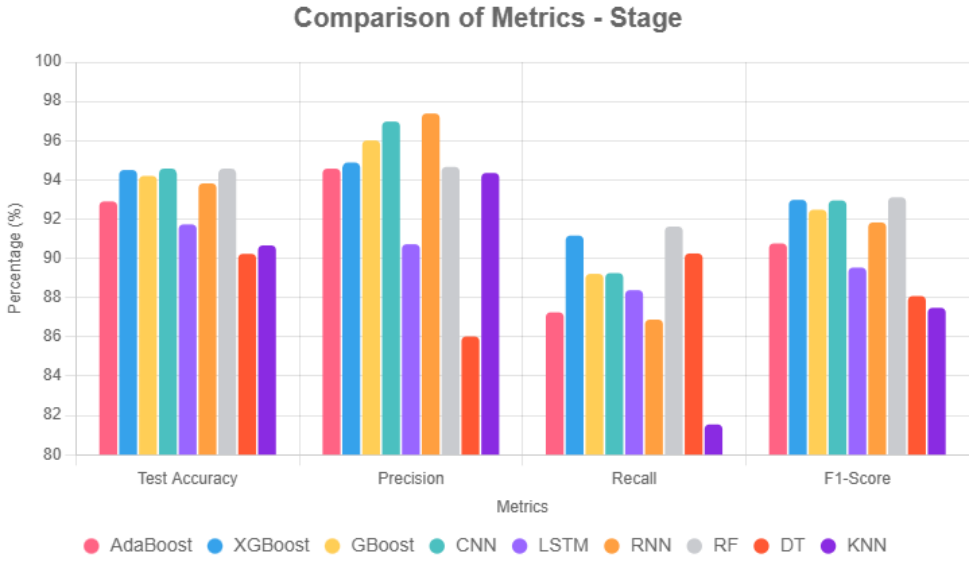
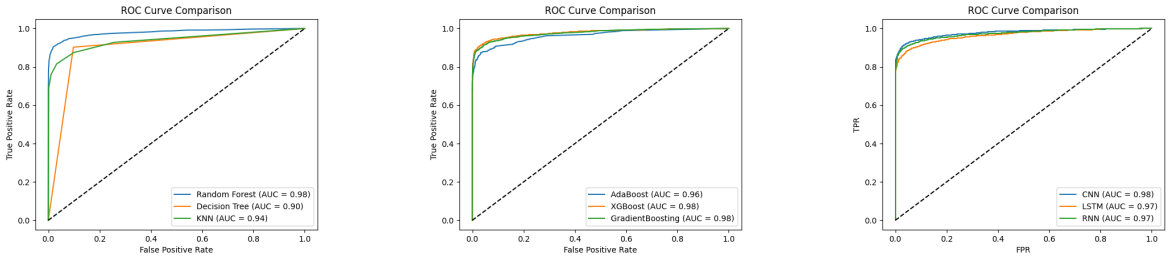


FIGURE 3.10 – Histogramme comparatif des performances des modèles sur le dataset industriel

Analyse de courbes ROC et PRC

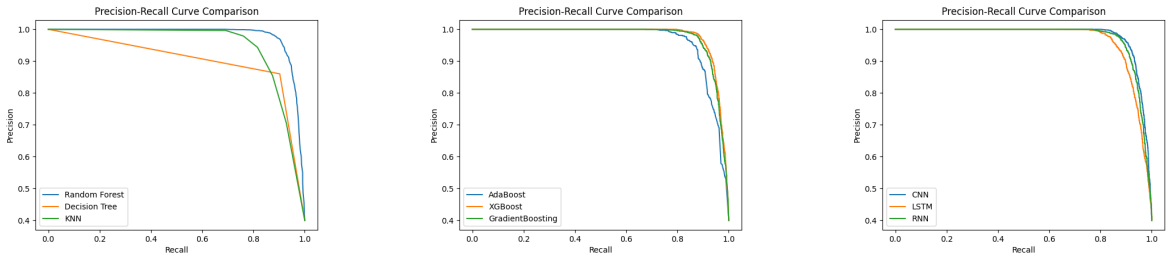
Les courbes ROC, illustrées dans la figure 3.11, montrent que les algorithmes des trois catégories présentent des tracés proches du coin supérieur gauche, indiquant un bon équilibre entre le taux de vrais positifs (TPR) et le taux de faux positifs (FPR), ce qui traduit une performance satisfaisante.



(a) Bagging (b) Boosting (c) Deep Learning

FIGURE 3.11 – Courbes ROC des modèles sur le dataset industriel

L'analyse des courbes Precision-Recall (PRC), présentée dans la figure 3.12, révèle que les valeurs de l'aire sous la courbe (AUC) pour les trois catégories sont relativement proches, bien que `Random Forest`, `XGBoost` et `CNN` affichent une légère supériorité. Ces résultats confirment la capacité de notre approche à apprendre efficacement et à s'adapter aux données pour effectuer des prédictions précises.



(a) Bagging (b) Boosting (c) Deep Learning

FIGURE 3.12 – Courbes Precision-Recall des modèles sur le dataset industriel

3.5 Conclusion

Cette première phase a permis d'évaluer comparativement neuf algorithmes issus des familles bagging, boosting et deep learning sur trois jeux de données : CIC-IDS2018, UNSW-NB15 et un dataset industriel réel provenant d'un système DCS. Les résultats montrent que Random Forest, XGBoost et CNN se démarquent, en atteignant des performances élevées en précision, rappel et F1-score, tout en maintenant de faibles taux de faux positifs.

Chaque famille présente des atouts spécifiques : le bagging est robuste et interprétable, le boosting offre une excellente capacité de généralisation, et le deep learning permet d'extraire des caractéristiques complexes. Toutefois, aucun modèle ne s'impose comme universellement optimal. Ces constats justifient le choix d'une approche hybride, combinant les forces des meilleurs algorithms pour concevoir notre solution X-RF Shield.

Dans le chapitre suivant , nous allons présentrer la phase 2 et 3 , dans la phase 2 nous détaillerons la sélection finale des algorithmes constituant X-RF Shield, en nous appuyant sur les résultats expérimentaux de cette premiére phase. Enfin, la troisième phase sera consacrée à la simulation temps réel de la solution proposée, afin d'évaluer sa robustesse et sa réactivité dans un environnement industriel simulé.

Chapitre 04

La nouvelle approche X-RF Shield : Résultats, Analyse Comparative et Simulation en Temps Réel

Chapitre 4

La nouvelle approche X-RF Shield : Résultats, Analyse Comparative et Simulation en Temps Réel

4.1 Introduction

Ce chapitre prolonge l'analyse entamée au chapitre précédent, en s'appuyant sur les résultats expérimentaux de la phase 1, où plusieurs algorithmes ont été comparés sur trois jeux de données : CIC-IDS2018, UNSW-NB15 et un dataset industriel de Sonelgaz.

Sur cette base, la phase 2 est consacrée à la construction de notre solution hybride X-RF Shield, combinant les deux modèles les plus performants dans une architecture d'ensemble afin de tirer parti de leurs forces complémentaires. Une évaluation multicritère est réalisée à partir d'indicateurs clés tels que l'accuracy, la précision, le rappel, le F1-score et le FPR, permettant de justifier leur sélection finale.

La phase 3 présente une simulation en temps réel de la solution X-RF Shield, appliquée à un environnement industriel simulé basé sur une turbine à gaz. L'objectif est d'évaluer les performances de la solution en conditions proches du terrain, en analysant sa capacité à détecter des anomalies injectées dans un flux de données simulées, tout en assurant une faible latence et un taux de faux positifs minimal.

Ce chapitre vise ainsi à démontrer la pertinence et la faisabilité de la solution X-RF Shield comme outil fiable de cybersécurité pour les ICS.

4.2 Phase 2 : Construction de notre solution XRF-Shield

Cette étape est basée sur les résultats de la premiére phase d'ou, a effectué une évaluation de ces résultats obtenus .

Dans cette section, nous présentons une évaluation qualitative des performances des algorithmes testés, sur la base des principales métriques de classification : l'accuracy, la précision, le rappel, le F1-score et le taux de faux positifs (FPR). Ces critères ont été choisis pour refléter à la fois la capacité globale de classification et la sensibilité aux attaques. Afin de faciliter l'interprétation et la comparaison, chaque métrique a été classée selon des intervales définis comme suit :

- **Accuracy** : Excellent (> 98 %) ; Très Bon (96–98 %) ; Acceptable (94–96 %) ; Faible ($<$ 94 %)
- **F1-Score** : Excellent (> 95 %) ; Très Bon (90–95 %) ; Acceptable (85–90 %) ; Faible (< 85 %)
- **FPR** : Excellent (< 0.1) ; Très Bon (0.1–0.2) ; Acceptable (0.2–0.3) ; Faible (> 0.3)
- **Précision** : Excellent (> 95 %) ; Très Bon (90–95 %) ; Acceptable (85–90 %) ; Faible ($<$ 85 %)
- **Rappel** : Excellent (> 99 %) ; Très Bon (95–99 %) ; Acceptable (90–95 %) ; Faible (< 90 %)

La décision globale pour un algorithme est déterminée par la majorité des évaluations obtenues. Le tableau 4.1 présente les évaluations pour les modèles testés sur le dataset de turbine a gaz de Sonelgaz.

TABLE 4.1 – Grille d'évaluation des modèles selon les critères clés

Modèle	**Accuracy**	**Précision**	**Rappel**	**F1-Score**	**FPR**	**Décision globale**
AdaBoost	Bien	Très Bien	Bien	Bien	Bien	Acceptable
XGBoost	Excellent	Excellent	Excellent	Excellent	Bien	Excellent
GradientBoost	Très Bien	Très Bien	Très Bien	Très Bien	Bien	Très Bon
CNN	Excellent	Tres Bien	Excellent	Très Bien	Excellent	Très Bon
LSTM	Bien	Bien	Très Bien	Bien	Faible	Acceptable
RNN	Bien	Très Bien	Très Bien	Bien	Faible	Acceptable
Random Forest	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent
Decision Tree	Excellent	Excellent	Très Bien	Excellent	Faible	Acceptable
KNN	Très Bien	Bien	Bien	Bien	Bien	Acceptable

À l'issue de cette analyse comparative, XGBoost et Random Forest se sont distingués par leur stabilité et leurs performances globales. Ces deux modèles ont été retenus pour construire la solution hybride X-RF Shield, exploitant leurs forces en matière de précision .

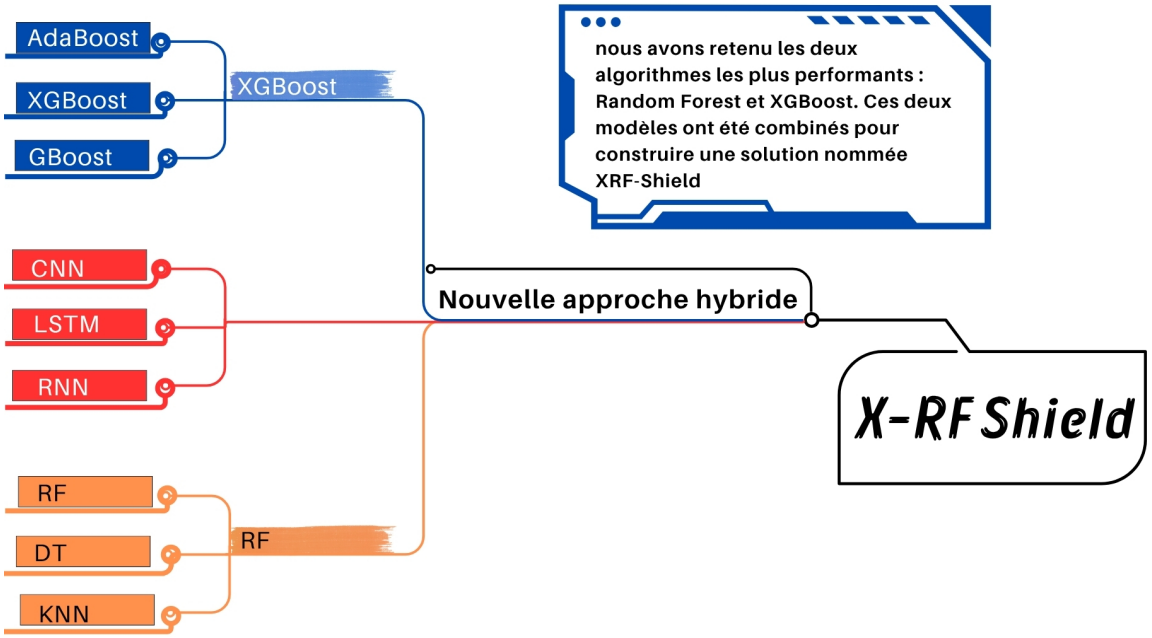


FIGURE 4.1 – Sélection des algorithmes pour la solution XRF-Shield

4.3 Analyse de Phase 2 : Construction de notre solution XRF-Shield

Cette phase consiste à sélectionner deux modèles performants, Random Forest et XG- Boost , en raison de leurs bonnes performances lors de la phase initiale pour construire une solution combinée nommée XRF-Shield. De plus :

- Random Forest : Robuste face au bruit, évite le surapprentissage grâce à l'agrégation d'arbres, et facilite l'interprétation via l'importance des variables.
- XGBoost : Boosting optimisé avec régularisation, très efficace sur des données volumineuses,améliore la précision et la généralisation.

4.3.1 Objectif de notre solution

L'objectif est de combiner les forces des deux modèles via une approche d'ensemble hybride appelée XRF-Shield, permettant :

- Augmenter la fiabilité de la détection en compensant les forces individuelles des modèles.
- Améliorer les taux de détection et de précision, tout en réduisant les FPR.
- Fournir un système fiable pour la cybersécurité des ICS.

4.3.2 Mise en œuvre

Pour atteindre ces objectifs, la démarche suivante sera suivie :

- Sélection des modèles RF et XGB suite à leur performance lors de la phase initiale.
- Entraînement des deux modèles sur les données prétraitées, normalisées et équilibrées.
- Évaluation par les métriques classiques : Precision, Recall et FPR.

X-RF Shield est principalement composéed de deux méthodes de détection :

Filtrage initial par la méthode du Z-score

On a appliqué la méthode du Z-score pour une première phase d'élimination des données suspectes. Cette technique statistique permet d'identifier les valeurs suspectes comme les pics et les chutes en mesurant l'écart de chaque point par rapport à la moyenne, en le normalisant par l'écart-type. Elle allège ainsi la charge des modèles de classification en écartant les anomalies évidentes avant une analyse plus approfondie.

$$Z = \frac{x - \mu}{\sigma} \tag{4.1}$$

Une valeur est considérée comme suspecte si son $|Z| > 3$, ce qui indique qu'elle est située à plus de trois écarts-types de la moyenne.

Détection par les modèles XGBoost et Random Forest

Dans cette dernière étape, les données prétraitées décrites à la section 3.3.2 sont soumises aux algorithmes d'apprentissage automatique sélectionnés pour la solution XRF-Shield : XGBoost et Random Forest.

XGBoost, basé sur le principe du gradient boosting, se distingue par sa capacité à gérer des données bruitées et à optimiser les performances en minimisant les erreurs résiduelles à chaque itération.

Random Forest, quant à lui, repose sur une agrégation d'arbres de décision construits sur des sous-échantillons aléatoires, apportant une stabilité et une résistance accrue à l'overfitting. L'association de ces deux modèles vise à tirer parti de leurs forces complémentaires. Les données utilisées sont issues de fichiers CSV contenant les valeurs des capteurs normalisées et avec bruits.

Résultats attendus

À l'issue de l'implémentation de notre solution, les objectifs suivants sont censés être atteints :

- Amélioration de la capacité de détection des attaques.
- Robustesse accrue grâce à la complémentarité des modèles.
- Solution adaptable à différents types de données ICS.

4.4 Phase 3 : Simulation en temps réel de X-RF Shield

L'objectif principal de cette simulation est d'évaluer les performances de la solution XRF-Shield dans un environnement proche du réel. Cette simulation vise à valider l'efficacité opérationnelle du système dans des conditions similaires à celles rencontrées sur le terrain industriel, en s'appuyant sur des données d'une turbine à gaz contenant des attaques injectées.

La simulation s'est déroulée en trois étapes principales :

Sélection des variables et génération du dataset simulé

- Dans cette première étape, deux variables critiques ont été sélectionnées pour la supervision : **la température ambiante** de la turbine et **la puissance** générée. Les valeurs ont été calculées à l'aide de lois de la thermodynamique spécifiques aux turbines.
- Afin d'imiter les conditions d'acquisition dans un environnement industriel, un bruit aléatoire de ±0.02 a été injecté dans les données. Cette perturbation simule les imperfections de mesure typiques des capteurs industriels.
- Les données ont ensuite été diffusées ligne par ligne avec une fréquence temporelle de 50 millisecondes, reproduisant un traitement en quasi temps réel.
- Les modèles **XGBoost et Random Forest** ont été chargés via `joblib`.
- Une alerte peut être déclenchée en cas d'*incohérences* détectées, même si les modèles statistiques ne réagissent pas.

4.5 Analyse de phase 3 : Simulation en temps réel de X-RF Shield

Afin d'évaluer la solution XRF-Shield dans des conditions réalistes et de manière interactive, une interface graphique de simulation en temps réel a été développée. Cette interface,

illustrée par la Figure 4.2, a été conçue à l'aide des bibliothèques Tkinter et Matplotlib. Elle est composée de quatre visualisations principales :

1. **Taux d'anomalies détectées** : Visualisation du taux d'alertes cumulatif au fil du temps, permettant de suivre la fréquence des anomalies signalées par les modèles.

2. **Paramètres physiques** : Affichage de la température post-compression (T) et de la puissance générée (P), calculées à partir des mesures `mass_flow`, `fpg1`, `fpg2` et `TIT`. Ces graphiques permettent de surveiller les variations thermiques et énergétiques.

3. **Suivi des métriques IA** : Courbes de précision, rappel et score F1, mises à jour dynamiquement à chaque itération de la simulation.

4. **Journal d'alertes** : Chaque détection d'anomalie déclenche une alerte visuelle sous forme de popup, contenant :

   - Le timestamp local associé à chaque ligne de données simulée
   - Les prédictions des modèles XGBoost et Random Forest pour chaque échantillon
   - La température post-compression T calculée dynamiquement
   - La puissance estimée générée par la turbine simulée.

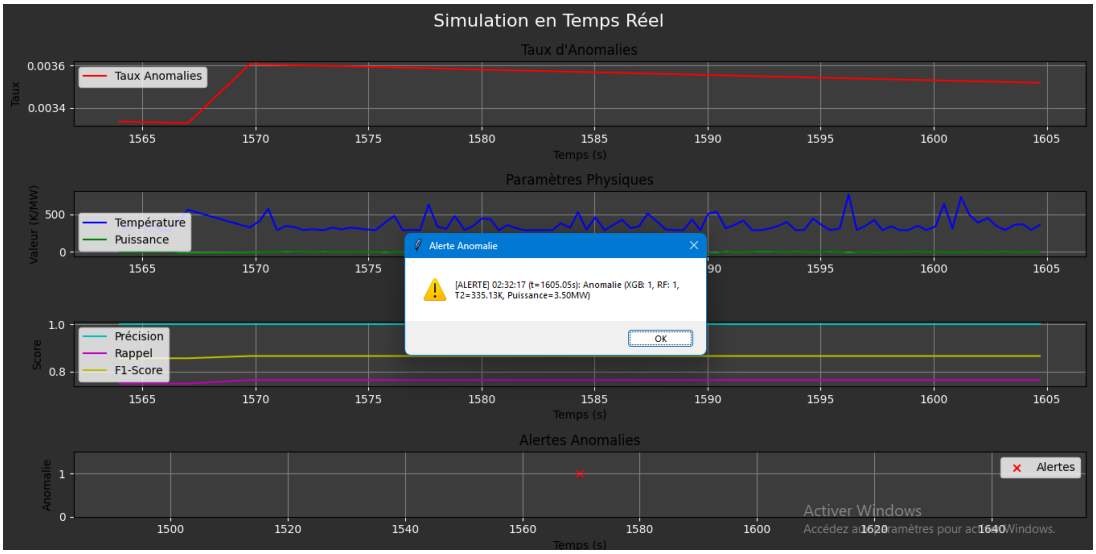


FIGURE 4.2 – Interface de simulation de X-RF Shield

La simulation en temps réel a été construite à l'aide de divers composants techniques, dont les principaux sont résumés dans le tableau 4.2. L'objectif de cette simulation était de tester l'efficacité de notre solution *XRF-Shield* dans un environnement réaliste en injectant volontairement différentes anomalies dans un jeu de données issu d'un système industriel réel.

TABLE 4.2 – Composants techniques utilisés pour la simulation

Élément	**Description**
Jeu de données	Données extraites d'un DCS industriel (MARK VI - General Electric), utilisé dans une turbine à gaz à la centrale Sonelgaz de Koudiet Eddraouech
Prétraitement	Normalisation avec `StandardScaler` appliquée à 20 variables physiques
Bibliothèques Python	`pandas`, `numpy`, `joblib`, `scikit-learn`, `matplotlib`, `tkinter`
Modèle physique	Calcul simplifié de la température T et de la puissance P générée à partir des variables `mass_flow`, `fpg1`, `fpg2` et `TIT`
Structure de simulation	Défilement ligne par ligne avec mise à jour des prédictions et des visualisations à chaque itération (pseudo temps réel)

Pour évaluer notre solution, un total de **8000 anomalies** de types variés ont été injectées manuellement dans le dataset d'origine. Ces anomalies simulaient plusieurs scénarios de dysfonctionnement comme les valeurs hors plage, comportements suspects et les transitions anormales. À l'issue de la simulation, la solution *XRF-Shield* a détecté **8037 anomalies**, ce qui indique :

- **8000 anomalies correctement détectées** (*True Positives*)
- **37 fausse alerte** (*False Positive*)

Ces résultats montrent la grande sensibilité de la solution avec un rappel proche de 98 % tout en maintenant un taux de faux positifs très faible : FPR ≈ 0.003. Le tableau 4.3 présente les métriques finales calculées à l'issue de cette simulation.

TABLE 4.3 – Résultats de la simulation temps réel sur le dataset industriel

Indicateur	**Valeur**
Nombre d'échantillons traités	20 000
Nombre total d'anomalies injectées	8000
Nombre d'anomalies détectées	8037
Vrais positifs (TP)	8000
Faux positifs (FP)	37
Rappel (Recall) (%)	97.76
Précision (%)	97.31
F1-Score (%)	97.54
Taux de faux positifs (FPR)	0.003
Temps moyen de traitement (ms)	50

Les résultats de cette phase de simulation en temps réel démontrent l'efficacité de notre solution *XRF-Shield* dans un contexte industriel réaliste. La détection quasi-parfaite des anomalies injectées, associée à un très faible taux de faux positifs, confirme le potentiel de cette approche hybride pour renforcer la cybersécurité des systèmes de contrôle industriels en assurant une surveillance fiable.

Le tableau 4.4 présente une comparaison des résultats de X-RF Shield avec les résultats de chaque catégories .

TABLE 4.4 – Comparaison des performances des modèles et de X-RF Shield

Catégorie	**Modèle**	**Précision (%)**	**Rappel (%)**	**F1-score (%)**	**FPR**
Bagging	Random Forest	94.66	91.62	93.11	0.1197
	Decision Tree	86.02	90.25	88.08	0.3659
	k-NN	94.35	81.54	87.48	0.2001
Boosting	AdaBoost	94.57	87.25	90.76	0.0299
	XGBoost	94.88	91.16	92.98	0.1144
	Gradient Boost	96.01	89.21	92.48	0.2013
Deep Learning	CNN	96.97	89.25	92.95	0.1779
	LSTM	90.72	88.38	89.53	0.2192
	RNN	97.38	86.88	91.83	0.1445
Proposition	**X-RF Shield**	**97.31**	**97.76**	**97.54**	**0.0030**

Ces résultats montrent que les modèles de Boosting XGBoost et ceux de Deep Learning comme CNN et RNN offrent de bonnes performances globales, avec des F1-scores oscillant entre 90–93%. Cependant, la solution proposée XRF-Shield surpasse l'ensemble des modèles testés, atteignant une précision de 97.31%, un rappel de 97.76%, un F1-score de 97.54% et un taux de faux positifs remarquablement bas 0.003.

4.6 Discussion et interprétation

Les résultats obtenus sur les trois jeux de données CIC-IDS2018, UNSW-NB15 et le dataset industriel de Sonelgaz dans la prmeiére phase montrent que les modèles , en particulier Random Forest et XGBoost, offrent les meilleures performances globales, avec un rappel élevé et un faible taux de FPR. Ces deux indicateurs sont prioritaires en milieu industriel, où la détection rapide d'intrusions et la réduction des fausses alertes sont cruciales pour garantir la continuité des opérations.

Les modèles de deep learning, notamment CNN, ont également montré de bons résultats, mais leur performance sur le dataset industriel reste légèrement inférieure, probablement à cause du volume limité de données . Le LSTM, bien qu'adapté aux données séquentielles, a été moins performant, notamment en raison de sa complexité et du tuning limité effectué dans cette étude.

La solution hybride proposée, XRF-Shield, tirant parti des complémentarités entre XGBoost et Random Forest, a démontré une efficacité accrue et une meilleure capacité de généralisation sur l'ensemble des jeux de données. Cette approche offre un compromis pertinent entre précision, temps d'inférence et stabilité, rendant la solution adaptée à une intégration en environnement industriel réel.

Comme tout travail, notre étude présente certaines limites qu'il convient de souligner :

Tout d'abord, bien que l'intégration d'un modèle de type CNN ait été initialement envisagée pour enrichir la solution proposée, cette piste a été abandonnée en raison de la complexité computationnelle du modèle, nécessitant un temps d'apprentissage élevé et des ressources matérielles avancées dont nous ne disposions pas.

Ensuite, bien que nous ayons utilisé un jeu de données simulé pour entraîner et évaluer notre solution, une validation dans un environnement réel constituerait une étape essentielle. Tester X-RF Shield en temps réel, connectée à un système ICS opérationnel, permettrait d'évaluer sa réactivité face à des attaques concrètes et son adaptabilité aux contraintes industrielles.

4.7 Conclusion

Ce chapitre a présenté et analysé les résultats expérimentaux obtenus à partir de 9 algorithmes de détection d'intrusion appliqués aux jeux de données CIC-IDS2018, UNSW-NB15 et à un jeu de données fourni par Sonelgaz. Les modèles Random Forest et XGBoost se sont distingués par leur capacité à détecter efficacement les intrusions avec des performances élevées en termes d'exactitude, de précision et de F1-score.

Le modèle proposé, XRF-Shield, combinant la robustesse de Random Forest et la puissance de XGBoost, a démontré une performance satisfaisante, tout en respectant les contraintes liées aux systèmes industriels, notamment en matière de réactivité. Ces résultats confirment la pertinence des approches hybrides basées sur l'apprentissage automatique pour renforcer la cybersécurité des environnements critiques tels que les DCS.

Conclusion générale

Conclusion générale

La montée en puissance de l'Industrie 4.0 a induit une transformation profonde des systèmes de production, marquée par l'intégration massive de technologies de communication et de contrôle. Cette évolution a entraîné une interconnexion croissante des Systèmes de Contrôle Industriel , plus particulièrement des Systèmes de Contrôle Distribué , exposant ces derniers à de nouvelles menaces en matière de cybersécurité. La problématique centrale ayant motivé ce travail peut ainsi être formulée comme suit : quelle approche d'intelligence artificielle permet de détecter efficacement et en temps réel les intrusions dans les DCS, tout en assurant robustesse, fiabilité et adaptabilité aux spécificités industrielles ?

Afin d'apporter des éléments de réponse à cette problématique, ce projet s'est articulé autour de trois contributions majeures. Tout d'abord, une étude comparative approfondie de neuf algorithmes de classification, répartis en trois familles (bagging, boosting et deep learning), a été réalisée sur trois jeux de données pertinents : CIC-IDS2018, UNSW-NB15 et un dataset industriel réel de Sonelgaz. Ensuite, une nouvelle approche hybride, dénommée X-RF Shield, combinant les atouts de Random Forest et XGBoost, a été proposée afin d'optimiser la détection d'anomalies. Enfin, une évaluation en temps réel du modèle a été effectuée dans un environnement industriel simulé, permettant de valider sa capacité de généralisation et sa réactivité dans des conditions proches du terrain.

Notre solution a affiché des performances remarquables en termes de précision, de rappel et de F1-score, y compris dans des contextes réels simulés. Ces résultats confirment l'hypothèse selon laquelle une hybridation de modèles de type bagging et boosting peut dépasser les approches individuelles classiques.

Perspectives

Plusieurs pistes peuvent être envisagées pour approfondir ce travail : Les résultats obtenus sont encourageants, mais ouvrent également la voie à plusieurs perspectives de recherche. Une première direction consiste à intégrer des techniques avancées de sélection de caractéristiques ou de réduction de dimension telles que PCA, ou autoencodeurs afin d'améliorer les performances computationnelles du modèle dans un contexte de déploiement à grande échelle. Par ailleurs, il serait pertinent de tester l'approche X-RF Shield dans un environnement industriel

réel avec des données en flux continu, pour en l'évaluer face aux contraintes opérationnelles et aux scénarios d'attaques variés. Enfin, l'intégration de mécanismes d'apprentissage en ligne et de défense adversariale pourrait renforcer la résilience du système face à des attaques évolutives , s'inscrivant ainsi dans une démarche proactive de cybersécurité industrielle.

Ce travail nous a permis d'acquérir une vision approfondie des enjeux de la cybersécurité industrielle et de développer des compétences solides en traitement de données, modélisation et analyse critique.

Bibliographie

[1] Ahakonye, L. A. C., Nwakanma, C. I., Lee, J. M., Kim, D. S. Schéma de détection d'intrusion SCADA exploitant la fusion d'arbres de décision modifiés et la sélection de caractéristiques par chi-deux [en ligne]. Article académique : Internet des Objets. Amsterdam : Elsevier, 2023. Publié en 2023. [27]

[2] Al-Abassi, A., Karimipour, H., Dehghantanha, A., Parizi, R. M. Détection de cyberattaques basée sur l'apprentissage profond dans les systèmes de contrôle industriels [en ligne]. Actes de conférence : Cybersécurité. Piscataway : IEEE, 2020. Publié en 2020. [28]

[3] Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., Wahab, A. Une revue des systèmes de détection d'intrusion utilisant l'apprentissage automatique et l'apprentissage profond dans l'Internet des Objets : Défis, solutions et directions futures [en ligne]. Article académique : Internet des Objets. Basel : MDPI, 2020. Publié en 2020. [14]

[4] Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., Wahab, A. Une revue des systèmes de détection d'intrusion utilisant l'apprentissage automatique et l'apprentissage profond dans l'Internet des Objets : Défis, solutions et directions futures [en ligne]. Article académique : Internet des Objets. Basel : MDPI, 2020. Publié en 2020. [40]

[5] Bagui, S., Bagui, S., Kebede, E. A Scalable Random Forest Algorithm Based on Spark for Predicting Network Intrusions. IGI Global, 2021. [44]

[6] Benabid, Y., Aissani, D., Laouid, A. Comparative Study of Decision Trees, Random Forest, Extra Trees and XGBoost for Intrusion Detection Using Feature Selection [en ligne]. Article académique : Preprints, 2023. Publié en 2023. [41]

[7] Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J. C. Exploration de données pour la fraude par carte de crédit : Une étude comparative [en ligne]. Article académique : Science des Données. Amsterdam : Elsevier, 2011. Publié en 2011. [15]

[8] Banda, T. V., Blaauw, D., Watson, B. Vers un modèle de cybersécurité SCADA avec des algorithmes d'apprentissage automatique [en ligne]. Actes de conférence : Cybersécurité. Inconnu : International Conference, 2023. Publié en 2023.

[9] Chih-Ta Lin, Wu, S.-L., Lee, M.-L. Attaque et défense cybernétiques sur les systèmes de contrôle industriels [en ligne]. Rapport : Cybersécurité. Taipei : CyberTrust Technology Institute, 2017. Publié en 2017. [30]

[10] Clarke, J., Larsen, R., Reves, A. Introduction au protocole DNP3 [en ligne]. Rapport technique : Ingénierie Électrique. Calgary : GE Harris Canada Inc., 2004. Publié en 2004. [3]

[11] Dalarmelina, V. D., et al. TENNER : Un modèle hybride pour la sécurité des systèmes d'eau basés sur SCADA [en ligne]. Article académique : Cybersécurité. Inconnu : Éditeur non spécifié, 2024. Publié en 2024.

[12] DataScienceDojo. Algorithmes de boosting en apprentissage automatique [en ligne]. Article technique : Machine Learning. Seattle : Data Science Dojo, 2023. Publié en 2023. [18]

[13] Diaba, S. Y., et al. Système de sécurisation SCADA utilisant l'apprentissage profond pour prévenir l'infiltration cybernétique [en ligne]. Article académique : Cybersécurité. Amsterdam : Elsevier, 2023. Publié en 2023. [32]

[14] Erez, E., Wool, A. Intégrité du flux de contrôle dans les systèmes de contrôle industriels [en ligne]. Article académique : Sécurité Informatique. New York : ACM, 2015. Publié en 2015. [4]

[15] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., Thrun, S. Classification au niveau dermatologique du cancer de la peau avec des réseaux neuronaux profonds [en ligne]. Article académique : Intelligence Artificielle. Londres : Nature Publishing Group, 2017. Publié en 2017. [16]

[16] Ferrag, A., Maglaras, L., Moschoyiannis, S., Janicke, H. Apprentissage profond pour la détection d'intrusion en cybersécurité : Approches, ensembles de données et étude comparative [en ligne]. Article académique : Cybersécurité. Amsterdam : Elsevier, 2020. Publié en 2020. [19]

[17] Gao, W., Morris, T. H., Reaves, B., Richey, D. Injection de commandes et réponses dans les systèmes SCADA et détection d'intrusion [en ligne]. Actes de conférence : Cybersécurité. Birmingham : IEEE, 2014. Publié en 2014. [5]

[18] Ghaleb, Y., Alzahrani, B. A., Eissa, A. M., Alabdulatif, A. Apprentissage par transfert profond pour la détection d'intrusion dans les réseaux de contrôle industriels [en ligne]. Article académique : Cybersécurité. Riyad : Éditeur non spécifié, 2023. Publié en 2023. [25]

[19] Huitsing, P., Chandia, R., Papa, M., Shenoi, S. Conception et architecture d'un système de détection d'intrusion SCADA [en ligne]. Article académique : Protection des Infrastructures Critiques. Amsterdam : Elsevier, 2008. Publié en 2008. [6]

[20] IntechOpen. Techniques de détection d'anomalies en cybersécurité [en ligne]. Chapitre de livre : Cybersécurité. Londres : IntechOpen, 2023. Publié en 2023. [13]

[21] iTrust Centre. Ensemble de données Secure Water Treatment (SWaT) et Water Distribution (WADI) [en ligne]. Ensemble de données : Cybersécurité Industrielle. Singapour : Singapore University of Technology and Design, s.d. Publié en s.d. [26]

[22] Ivanov, D., Dolgui, A., Sokolov, B. L'impact de la technologie numérique et de l'Industrie 4.0 sur l'effet d'ondulation et l'analyse des risques de la chaîne d'approvisionnement [en ligne]. Article académique : Gestion de la Chaîne d'Approvisionnement. Londres : Taylor & Francis, 2019. Publié en 2019. [17]

[23] Kaspersky ICS CERT. Paysage des menaces pour les systèmes d'automatisation industrielle : 2024–2025 [en ligne]. Rapport : Cybersécurité Industrielle. Moscou : Kaspersky ICS CERT, 2025. [7]

[24] Khraisat, A., Gondal, I., Vamplew,, P., Kamruzzaman, J. Enquête sur les systèmes de détection d'intrusion : Techniques, ensembles de données et défis [en ligne]. Article académique : Cybersécurité. Berlin : Springer, 2019. Publié en 2019. [20]

[25] Kundap, D. A., et al. XG-ADICS : Un système de détection d'intrusion efficace pour ICS utilisant XGBoost [en ligne]. Article académique : Cybersécurité. Inconnu : Éditeur non spécifié, 2022. Publié en 2022.

[26] Makrakis, G. M., Kolias, C., Kambourakis, G., Rieger, C., Benjamin, J. Vulnérabilités et attaques contre les systèmes de contrôle industriels et les infrastructures critiques [en ligne]. Article académique : Cybersécurité. Idaho Falls : ResearchGate, 2021. Publié en 2021. [1]

[27] Majmuah. Défis de sécurité dans les systèmes SCADA [en ligne]. Article académique : Ingénierie Appliquée. Dubaï : Majmuah, 2024. Publié en 2024.

[28] ModbusIDA. Spécification du protocole d'application Modbus v1.1b [en ligne]. Document technique : Ingénierie des Communications. Hopkinton : ModbusIDA, 2004. Publié en 2004. [9]

[29] Moustafa, N., Slay, J. UNSW-NB15 : Un ensemble de données complet pour les systèmes de détection d'intrusion réseau [en ligne]. Actes de conférence : Cybersécurité. Canberra : IEEE, 2015. Publié en 2015. [23]

[30] NIST. Détection d'intrusion basée sur les anomalies à l'aide de grands modèles de langage [en ligne]. Rapport : Intelligence Artificielle et Cybersécurité. Gaithersburg : NIST, 2024. Publié en 2024. [12]

[31] Okur, M. C., Dener, M. Évaluation de XGBoost pour les ensembles de données IIoT basés sur SCADA [en ligne]. Article académique : Internet des Objets. Inconnu : Éditeur non spécifié, 2025.

[32] Rahmouni, F., Tighilt, M., Zemmouri, R. Intrusion Detection in Industrial Networks Using Z-Score Normalization and PCA with Ensemble Learning [en ligne]. Article académique : SpringerLink, 2023. Publié en 2023. [42]

[33] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A. Une enquête sur les ensembles de données pour la détection d'intrusion basée sur le réseau [en ligne]. Article académique : Cybersécurité. Amsterdam : Elsevier, 2019. Publié en 2019. [24]

[34] Salama, M. A., Eid, H. F., Ramadan, R. A., Darwish, A., Hassanien, A. E. Schéma de détection d'intrusion intelligent hybride [en ligne]. Article académique : Cybersécurité. Berlin : Springer, 2011. Publié en 2011. [35]

[35] Shahzad, A., Lee, M., Kim, H. D., Woo, S.-m., Xiong, N. Un modèle d'authentification multi-facteurs pour les services bancaires en ligne [en ligne]. Article académique : Informatique. Basel : MDPI, 2015. Publié en 2015. [2]

[36] Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A. Vers la génération d'un nouvel ensemble de données pour la détection d'intrusion et la caractérisation du trafic d'intrusion [en ligne]. Actes de conférence : Sécurité Informatique. Funchal : SCITEPRESS, 2018. Publié en 2018. [22]

[37] Soman, K. P., Alazab, M., Venkatraman, S., Al-Nemrat, A. Tutoriel complet et enquête sur les applications de l'apprentissage profond pour la cybersécurité [en ligne]. Article académique : Cybersécurité. Piscataway : IEEE, 2020. Publié en 2020. [21]

[38] Stouffer, K., Falco, J., Scarfone, K. Guide de sécurité des systèmes de contrôle industriels (ICS) (NIST SP 800-82 Rev. 2) [en ligne]. Rapport technique : Cybersécurité. Gaithersburg : NIST, 2014. Publié en 2014. [10]

[39] Talukder, N., Acharjee, S., Paul, M., Biswas, S. Deep Learning Based Ensemble Model for Intrusion Detection in IoT-Enabled Smart Grid. arXiv preprint, arXiv :2401.12262, 2024. [43]

[40] Tamy, S., Belhadoui, H., et al. Évaluation des algorithmes d'apprentissage automatique pour détecter les attaques dans le réseau SCADA [en ligne]. Actes de conférence : Cybersécurité. Piscataway : IEEE, 2019. Publié en 2019. [36]

[41] TechMagic. IA dans la détection d'anomalies : Techniques et applications [en ligne]. Article technique : Intelligence Artificielle. Tokyo : TechMagic, 2023. Publié en 2023. [11]

[42] Torres, R., et al. Détection de botnets utilisant des RNN basés sur LSTM sur les ensembles de données CTU13 [en ligne]. Article académique : Cybersécurité. Inconnu : Éditeur non spécifié, 2016. Publié en 2016.

[43] Upadhyay, V., Joshi, R. Détection des cyberattaques dans les systèmes SCADA utilisant la sélection de caractéristiques et le boosting de gradient [en ligne]. Article académique : Cybersécurité. Inconnu : Éditeur non spécifié, 2021. Publié en 2021.

[44] Zhang, Y., et al. Détection d'intrusion utilisant XGBoost et AdaBoost avec sélection de caractéristiques [en ligne]. Article académique : Cybersécurité. Inconnu : Éditeur non spécifié, 2022. Publié en 2022.

Annexe A

Jeux de données publiques

Dataset	Année	Taille	Nbre d'attributs	Types d'attaques	Utilisation principale
KDDCup-99	1999	$\approx$ 5M entrées	41	DoS, Probe, U2R, R2L	Benchmark historique pour IDS, apprentissage supervisé
NSL-KDD	2009	$\approx$ 150k entrées	41	DoS, Probe, U2R, R2L, Normal	Évaluation équitable, version améliorée de KDD
UNSW-NB15	2015	$\approx$ 2,5M entrées	49	Exploits, Fuzzers, DoS, Reconnaissance, Shellcode, etc.	IDS modernes, trafic réaliste, ML supervisé
DEFCON-10	2002	N/A	N/A	FTP/Telnet, Port scans, Sweep, exploit	Test de corrélation d'alertes, compétition CTF
CIC-IDS2018	2018	$>$ 20 Go	80	Botnet, Heartbleed, Brute-force, Web, Infiltration, DoS, DDoS	Scénarios réalistes pour ML/DL
CIC-DDoS2019	2019	$\approx$ 16 Go	Variable	DDoS sur HTTP, HTTPS, SSH, FTP, Email	Analyse spécifique des attaques DDoS modernes
Edge-IIoTset	2021	Variable (IoT-based)	61 (sur 1176)	DoS, DDoS, MITM, Infiltration, Malware	IDS dans les environnements IIoT et edge computing
BoT-IoT	2018	Variable	$>$ 25	DoS, DDoS, Scan, Keylogging	Environnements IoT, détection des botnets

Dataset	Année	Taille	Nbre d'attributs	Types d'attaques	Utilisation principale
SWaT	2015	$\approx$ 11 jours	Capteurs industriels	Altération capteurs, sabotage, falsification	ICS réels – usine traitement des eaux
WADI	2017	16 jours	Capteurs industriels	Diverses attaques sur réseaux hydrauliques	Sécurité des systèmes ICS à long terme
ADFA-LD	2013	$\approx$ 6000 traces	Appels système	Brute force, Webshell, escalade de privilèges	Détection d'intrusion basée sur Hôte (HIDS)

TABLE 5 – Datasets publics pour IDS

Jeu de donnée TG Sonelgaz

Nom de la variable	Désignation	Unité
G3.FRCROUT	Fuel Gas Speed Ratio Servo Command	%
G3.FSGR	Position Feedback Servo (high value selected)	%
G3.FSRG1OUT	G1 Control Valve Servo Output Command	%
G3.FSG1	Gas Control Valve #1 Position Feedback (high value selected)	%
G3.FSG2	Gas Control Valve #2 Position Feedback (high value selected)	%
G3.FSRG2OUT	G2 Control Valve Servo Output Command	%
G3.FSRG3OUT	G3 Control Valve Servo Output Command	%
G3.FSG3	Gas Control Valve #3 Position Feedback (high value selected)	%
G3.FSRG4OUT	G4 Control Valve Servo Output Command	%
G3.FSG4	Gas Control Valve #4 Position Feedback (high value selected)	%
G3.fpg1	Fuel Gas Inlet Pressure Transducer	psi/bar
G3.fpg2	Interstage Fuel Gas Pressure	psi/bar
G3.ftg	Fuel Gas Temperature	°C
G3.TTXM	Exhaust Temperature Median Corrected by Average	°C
g3.dwatt	Generator Watts Max Selected	MW
g3.FQG	Gas Fuel Flow	kg/h ou Nm^3/h
g3.csgv	IGV Inlet Guide Vane Feedback Angle	Degrees (°)
g3.cpd	Compressor Discharge Pressure Max Selected	bar/psi
g3.ctda	Compressor Discharge Temperature	°C
g3.fsr	Fuel Stroke Reference	%

TABLE 6 – Variables du jeu de données TG Sonelgaz (turbine à gaz)

Annexe B

Travaux connexes

Catégorie	Auteurs (Année)	Méthode / Approche	Dataset utilisé	Performances principales
Algorithmes classiques	Tamy et al. (2019)	RF, DT, SVM, KNN, NB	KDD'99 (SCADA)	RF $\approx$ 94%
	Banda et al. (2023)	RF, SVM, KNN, NB	NSL-KDD	RF = 98.2%
	Ahakonye et al. (2023)	RF + Chi² (sélection de caractéristiques)	NSL-KDD	Accuracy = 96.4%
Apprentissage profond (DL)	Salama et al. (2011)	DBN + SVM	NSL-KDD	Réduction à 5 features, ACC $>$ 90%
	Diaba et al. (2023)	CNN + RNN (hybride)	UNSW-NB15	DR = 99.6%, FAR $<$ 1.5%
	Torres et al. (2016)	LSTM (RNN)	CTU13-42, CTU13-47	ADR = 97.0%, FAR = 0.018%
	Soman et al. (2020)	Revue : CNN, AE, LSTM, DNN	NSL-KDD, CICIDS2017, etc.	DL $>$ 98% (selon études analysées)
Boosting (XGBoost, AdaBoost, GBM)	Kundap et al. (2022)	XGBoost (XG-ADICS)	HAI (testbed hydraulique)	ACC = 99.96%, F1 = 0.9986
	Upadhyay et al. (2021)	Gradient Boosting + DT	ORNL Power Grid	ACC $\approx$ 97.3%, DR $\approx$ 98.5%, FPR $\approx$ 3.7%
	Okur & Dener (2025)	XGBoost	WUSTL-2018-IIoT	ACC = 97.82%, F1 = 96.86%
Approches hybrides et dynamique	Al-Abassi et al. (2020)	CNN + RNN + AE (ensemble)	NSL-KDD	ACC $>$ 99%
	Zhang et al. (2022)	XGBoost + AdaBoost + FS	CICIDS2018, UNSW-NB15	ACC = 99.2%
	Dalarmelina et al. (2024)	Stacking : XGBoost + RF + DT + KNN	SCADA IIoT (eau)	ACC = 99.93%, FAR $\approx$ 0, entraînement réduit
	Chih-Ta Lin et al. (2023)	RF, SVM, KNN + moteur de prévention dynamique	Données SCADA simulées	ACC = 97.3%, DR = 96%

TABLE 7 – Comparaison des approches pour la détection d'intrusions (IDS)

Annexe C

Présentation du lieu de stage : Centrale électrique de Koudiet Eddraouech-Sonelgaz

La centrale électrique de Koudiet Eddraouch, située dans la wilaya d'El Tarf (Algérie) détenue par les groupes Sonelgaz, est l'une des plus importantes infrastructures de production d'électricité du pays. Mise en service en 2012, elle dispose d'une capacité installée de 1200 MW, assurée par trois turbines à gaz couplées à trois générateurs à vapeur dans une configuration à cycle combiné.

La centrale de Koudiet Eddraouch contribuera à coup sûr à répondre à la demande croissante en énergie électrique en Algérie et à assurer la sécurité de l'alimentation électrique dans la région. Le coût de cette centrale est évalué à 2,7 milliards de dollars, soit 179 milliards de DA.

Cette centrale à cycle combiné comporte essentiellement une turbine à gaz, une chaudière de récupération à trois niveaux de pression, une turbine à vapeur, un condenseur refroidi avec l'eau de mer en circuit fermé et un alternateur de 463 MVA refroidi avec de l'hydrogène. L'énergie électrique produite est évacuée à travers un poste blindé 400 kV d'évacuation d'énergie.

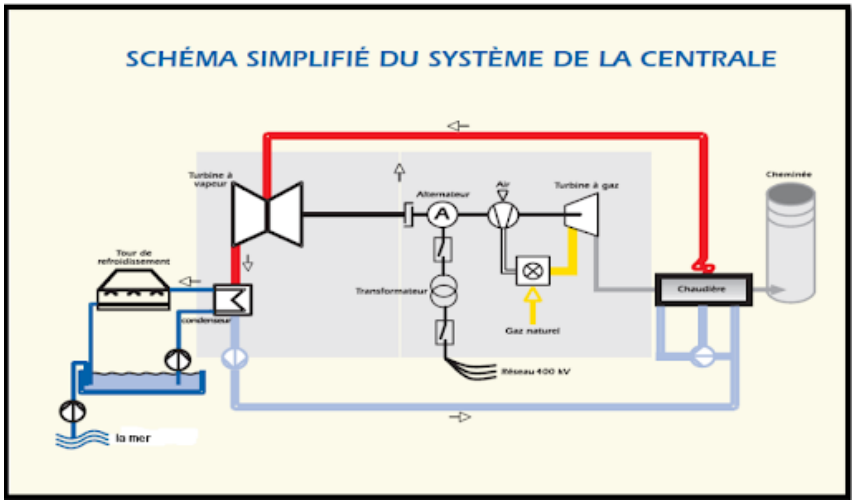


FIGURE .1 – Schéma du système de la centrale (Source)

Contrôle et supervision dans la centrale

L'ensemble des équipements de la centrale comme les turbines, générateurs, ainsi que le système de contrôle-commande sont fournis par General Electric (GE). Le contrôle opérationnel est assuré par le système MARK VI, un Distributed Control System (DCS) propriétaire de GE.

Ce système assure la supervision en temps réel de l'ensemble des composants de la centrale. Il repose sur un réseau câblé industriel utilisant principalement le protocole de communication Ethernet/IP, adapté aux environnements critiques.

MARK VI permet la collecte des données de capteurs à une fréquence élevée (1 ms), le contrôle des actionneurs, la gestion des alarmes et l'interface avec les opérateurs via des interfaces homme-machine (HMI). Il constitue ainsi le cerveau numérique de la centrale,

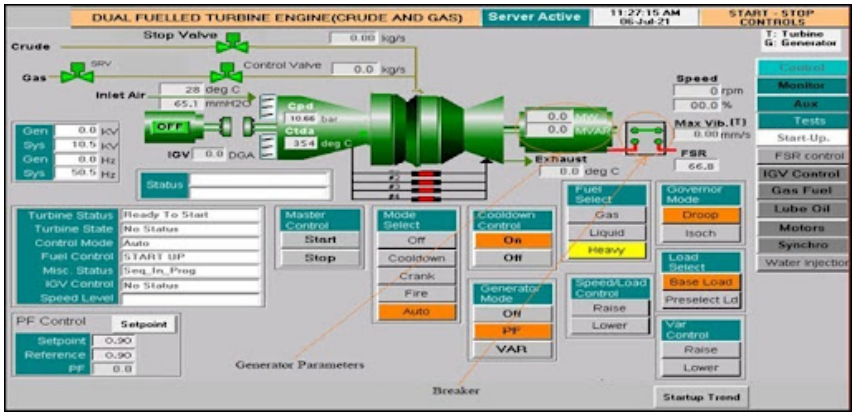


FIGURE .2 – Interface engineering station de système MARK VI (Source)

La turbine à gaz 9FB

La turbine à gaz 9FB développée par General Electric (GE) est une technologie de dernière génération, elle est indispensable dans les centrales à cycle combiné à cause de son rendement thermique élevé et sa fiabilité. À Koudiet Eddraouech, chaque unité de production est équipée d'une turbine 9FB, intégrée au système global de production.

Principe de fonctionnement

La 9FB fonctionne selon un cycle thermodynamique de Brayton, dans lequel l'air ambiant est comprimé, mélangé avec du combustible dans notre cas c'est du gaz naturel, puis enflammé dans la chambre de combustion. Les gaz chauds produits sous haute pression sont ensuite dirigés vers les aubes de la turbine, provoquant leur rotation. Cette rotation entraîne à la fois un alternateur pour la production d'électricité et le compresseur en amont.

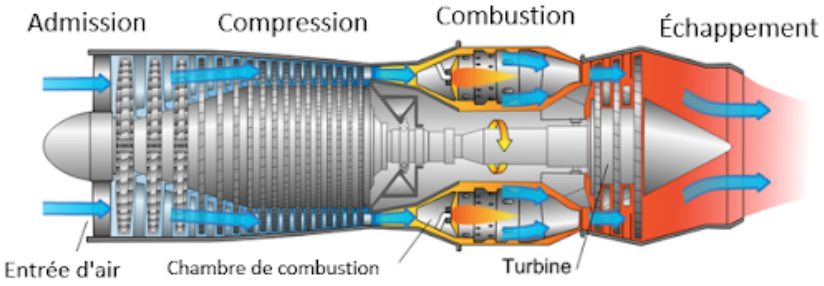


FIGURE .3 – Étape de fonctionnement d'une turbine à gaz (Source)

Le fonctionnement de la turbine repose sur la conversion harmonieuse entres des grandeurs physiques critiques telles que la température, le débit, la vitesse de rotation. Toute perturbation

dans l'équilibre de ces variables peut entraîner un arrêt immédiat de la production, des dommages matériels graves, voire des risques pour la sécurité du personnel. Ce qui rend son contrôle rigoureux et sa protection contre toutes intrusions non seulement souhaitables, mais essentiels.