Détection d'intrusion avec l'apprentissage automatique : analyse comparative des forêts aléatoires, XGBoost et des réseaux de neurones profonds

Mémoire présenté

dans le cadre du programme de maîtrise en informatique

en vue de l’obtention du grade de maître ès sciences

PAR

© THIERNO HAMIDOU SOW

NOVEMBRE 2025

Composition du jury :

Chan Wang Park, président du jury, UQAR
Mehdi Adda, directeur de recherche, UQAR
Hamid Mcheick, examinateur externe, UQAC

Dépôt initial le [30 Avril 2025] Dépôt final le [18 Novembre 2025]

DEDICACE

A ma famille, pour leur amour incommensurable et leur soutien, qui ont été une source de force et d'inspiration tout au long de ce parcours.

A Mon Directeur de recherche, pour ses précieux conseils et son accompagnement, sans lesquels ce travail n'aurait pu aboutir.

REMERCIEMENTS

Je tiens à remercier chaleureusement le professeur Mehdi Adda pour son accompagnement tout au long de ce travail. Son expertise, sa disponibilité et ses conseils ont été d'une grande aide et ont largement contribué à l'élaboration de ce projet. Grâce à son soutien, j'ai pu mener à bien cette recherche.

J’adresse mes sincères remerciements aux membres du jury pour leurs remarques constructives et leur temps précieux.

Je tiens également à exprimer ma gratitude envers ma famille pour son soutien inébranlable, ses encouragements, sa patience et sa compréhension tout au long de mon parcours.

AVANT-PROPOS

Ce mémoire marque la fin de mes études de maîtrise en informatique à l'Université du Québec à Rimouski. Tout au long de mes recherches, j'ai été animé par l'envie d'explorer comment les systèmes de détection d'intrusion (IDS) peuvent être optimisés au sein des réseaux informatiques, en m'appuyant sur des techniques d'apprentissage automatique. Ce domaine, en constante évolution, m'a offert de nombreuses possibilités et défis, me poussant à approfondir mes connaissances afin de contribuer à améliorer l'efficacité et la précision des IDS tout en réduisant les faux positifs.

Les chapitres qui suivent détaillent le cheminement de ma recherche, la méthodologie employée, ainsi que les résultats obtenus et les enseignements tirés de cette étude. J’aspire à ce que cette recherche enrichisse les réflexions actuelles autour des systèmes de détection d'intrusion et ouvre la voie à de nouvelles perspectives dans le domaine. C’est avec un réel enthousiasme que je partage ce travail, dans l’espoir qu’il suscite l’intérêt des chercheurs et stimule de futures initiatives en matière d’innovation et de cybersécurité.

RÉSUMÉ

Les systèmes de détection d'intrusion (IDS) sont essentiels pour protéger les infrastructures informatiques contre les cyberattaques. Toutefois, les IDS doivent relever plusieurs enjeux majeurs, notamment l'amélioration du taux de détection et la réduction du taux de fausses alarmes. Dans cette étude, nous proposons une analyse comparative de trois algorithmes d'apprentissage automatique, à savoir les forêts aléatoires (RF), XGBoost et les réseaux de neurones profonds (DNN), dans le but d'améliorer les performances des IDS.

La méthodologie mise en œuvre comprend la normalisation des données, la sélection des caractéristiques, l'équilibrage des classes à l'aide de la Synthetic Minority Oversampling Technique (SMOTE), ainsi que l'optimisation des hyperparamètres avec Optuna. Les expérimentations ont été menées sur le jeu de données NSL-KDD, en utilisant les bibliothèques Scikit-learn sous python pour l'implémentation. L'évaluation des modèles a été réalisée par validation croisée, suivie d'une comparaison avec les approches existantes rapportées dans la littérature. Les métriques retenues incluent l'exactitude, la précision, le rappel et le score F1.

Les résultats obtenus montrent que le RF atteint une exactitude de 99,80%, surpassant XGBoost (99.79%) et DNN (98,65 %). Par rapport aux travaux existants, notre étude apporte une contribution expérimentale en comparant trois algorithmes sous les mêmes conditions méthodologiques afin d'identifier celui offrant la meilleure précision.

Mots clés : Cybersécurité, IDS, apprentissage automatique, SMOTE, RF, XGBoost, DNN, Classification.

ABSTRACT

Intrusion Detection Systems (IDS) are essential for protecting IT infrastructures against cyberattacks. However, IDS face several major challenges, including improving the detection rate and reducing the false alarm rate. In this study, we present a comparative analysis of three machine learning algorithms, namely Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Deep Neural Networks (DNN), with the aim of improving IDS performance.

The implemented methodology includes data normalization, feature selection, class balancing using the Synthetic Minority Oversampling Technique (SMOTE), as well as hyperparameter optimization with Optuna. Experiments were conducted on the NSL-KDD dataset, using the Scikit-learn libraries in Python for implementation. Model evaluation was performed by cross-validation, followed by a comparison with existing approaches reported in the literature. The chosen metrics include accuracy, precision, recall, and F1-score.

The results show that Random Forest achieves an accuracy of 99.80%, surpassing XGBoost (99.79%) and DNN (98.65%). Compared to existing studies, this work provides an experimental contribution by comparing the three algorithms under the same methodological conditions to identify the one offering the highest accuracy.

Keywords: Cybersecurity, IDS, machine learning, SMOTE, RF, XGBoost, DNN, Classification.

Table des matières

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS

IDS	Intrusion Detection System
RFE	Recursive Feature Elimination
SMOTE	Synthetic Minority Over-sampling Technique
DNN	Deep Neural Networks
XGBoost	Extreme Gradient Boosting
ML	Machine Learning
ROC	Receiver Operating Characteristic
DoS	Denial of Service
R2L	Remote to Local
U2R	User to Root
AUC	Area Under the Curve
NSL-KDD	National Software Laboratory- Knowledge Discovery in Databases

INTRODUCTION GÉNÉRALE

1. CONTEXTE

L'évolution rapide des technologies de l'information au cours des dernières décennies a engendré une prolifération de menaces informatiques de plus en plus sophistiquées, souvent subtiles et ciblées, entraînant des pertes économiques considérables et compromettant la sécurité des organisations [1]. Face à ces enjeux, les techniques d'apprentissage automatique se sont imposées comme une solution prometteuse pour renforcer la détection des intrusions. Dans ce contexte, il devient impératif de développer des IDS performants pour protéger les réseaux informatiques. Les IDS ont pour objectif de détecter des activités malveillantes susceptibles de porter atteinte à l'intégrité, la confidentialité et la disponibilité des ressources réseau [1]. La détection d'intrusion peut être classée en deux grandes catégories : la détection basée sur les signatures et la détection basée sur les anomalies. La première approche repose sur la reconnaissance de schémas ou de comportements correspondant à des attaques préalablement identifiées, tandis que la seconde identifie les activités irrégulières en repérant les écarts dans le trafic réseau normal. Bien que la détection d'anomalies présente l'avantage d'identifier des attaques, qui sont des actions malveillantes menées dans le but de pertuber, d'accéder, ou de détruire les informations transitant par un réseau informatique, elle est néanmoins fréquemment associée à un taux élevé de faux positifs, ce qui constitue une limitation notable de cette approche [2].

En parallèle, le déséquilibre des classes constitue un défi majeur dans la classification des données,, un phénomène particulièrement prononcé dans les IDS, où certaines classes, telles que les attaques, qui désignent des actions visant à exploiter une vulnérabilité pour compromettre la sécurité d'un système informatique, perturber son fonctionnement ou accéder illégalement à ses ressources, sont largement sous-représentées par rapport aux classes majoritaires, telles que les comportements normaux [3]. Dans un contexte similaire, ce déséquilibre se retrouve également dans les données des systèmes de détection de fraudes [4] ou dans les données de diagnostics médicaux [5]. Ce phénomène a été souligné par Narasimha Raju et al. [6] dans le cadre de la reconnaissance des émotions faciales et vocales, où ils ont démontré que le déséquilibre des données entraîne un biais des modèles en faveur des classes majoritaires, réduisant ainsi leur capacité à identifier correctement les émotions sous-représentées. Ce biais affecte la précision du modèle en augmentant le risque de faux négatifs, c'est-à-dire en ne détectant pas certaines émotions lorsqu'elles sont présentes.

En effet, cette disparité dans les données compromet la performance des modèles d'apprentissage automatique (ML) en altérant leur aptitude à généraliser correctement. Elle entraîne une augmentation du

taux de faux négatifs (cas où une attaque réelle n'est pas détectée), limitant ainsi l'identification des instances minoritaires, et de faux positifs (cas où une activité légitime est incorrectement signalée comme une attaque), générant une surcharge d'alertes inutiles qui nuit à l'efficacité des prédictions. Il devient donc crucial de traiter ce problème afin d'améliorer les performances des modèles ML.

Pour remédier à cette problématique, diverses techniques de rééquilibrage des données peuvent être envisagées. Ces techniques visent à atténuer l'impact de la nature déséquilibrée des données en ajustant la distribution des classes afin de permettre aux modèles de mieux apprendre les instances minoritaires [6]. Parmi les techniques de rééquilibrage des données, la technique SMOTE se distingue comme une solution largement utilisée pour rééquilibrer les classes et améliorer la détection des attaques. Certaines études [7],[8] ont montré que l'application de SMOTE a permis d'améliorer la capacité des modèles à mieux identifier les classes sous-représentées en générant de nouveaux échantillons synthétiques à partir des instances minoritaires existantes.

Leur approche permet d'équilibrer des classes sans dupliquer les données, ce qui réduit les biais d'apprentissage et résout ainsi les problèmes de surapprentissage. Dans le contexte des IDS, cela se traduit par une meilleure représentation des attaques dans l'ensemble des données, réduisant ainsi le risque que le modèle privilégie la classe majoritaire et néglige les intrusions.

En complément de cette technique de rééquilibrage, un réglage adéquat des valeurs des hyperparamètres permet de maximiser la précision des modèles, comme la profondeur des arbres ou le nombre d'échantillons, tout en réduisant le risque de surajustement lié à la complexité de la structure du modèle. Des cadres comme Optuna se distinguent par leur efficacité, car ils utilisent des algorithmes comme l'optimisation bayésienne pour explorer de manière plus ciblée l'espace des hyperparamètres. Contrairement aux méthodes de recherche aléatoire ou par grille, qui examinent l'espace de manière exhaustive ou aléatoire, Optuna ajuste sa recherche en fonction des résultats précédents, ce qui permet de réduire le temps de calcul et d'améliorer les choix des valeurs des hyperparamètres [9].

2. PROBLEMATIQUE

Les IDS basés sur les modèles de ML ont fait l'objet de nombreuses études visant à améliorer leur efficacité dans la détection des cybermenaces. Cependant, malgré les progrès réalisés, certains défis demeurent et continuent d'affecter leurs performances.

Parmi ces défis, le déséquilibre des données demeure l'un des principaux obstacles. La majorité des approches existantes obtiennent des performances élevées en termes de précision, comme l'approche basée sur XGBoost [10] avec une précision de 99,60 % et celle utilisant Random Forest [11] avec une

précision de 99,50 %. Cependant, ces résultats peuvent être biaisés en raison de l'absence d'un équilibre adéquat des données dans leur méthodologie, ce qui peut influencer la capacité réelle des modèles à détecter efficacement les intrusions, en particulier les classes minoritaires. De plus, certaines méthodes de sélection de caractéristiques, comme l'élimination basée sur la variance ou la corrélation, utilisées dans l'approche basée sur Random Forest, peuvent entraîner l'exclusion de variables importantes, ce qui peut affecter les performances du modèle. Il est donc essentiel d'adopter des méthodes de sélection de caractéristiques appropriées afin de ne pas compromettre ces performances.

En outre, l'ajustement des hyperparamètres des modèles représente un autre défi important. Par exemple, l'utilisation de DNN [12] sur le jeu de données NSL-KDD avec une sélection de caractéristiques basée sur la méthode RFE a permis d'atteindre une précision de 94 %, mais l'absence d'optimisation a limité les performances du modèle. Ainsi, il est évident qu'optimiser ces paramètres est essentiel pour améliorer l'efficacité des modèles.

Par ailleurs, une étude récente a exploré des combinaisons d'algorithmes, notamment SVM, Random Forest et XGBoost, associées à des techniques de rééquilibrage des données telles que SMOTE et RUS [7]. Cette approche a permis d'atteindre une précision de 99,62 % dans la détection des attaques DoS. Cependant, la portée de cette étude demeure restreinte, car elle se concentre uniquement sur un type d'attaque, limitant ainsi sa généralisation à d'autres formes d'intrusions telles que les attaques de type DOS, Probe, U2R et R2L, ce qui peut reduire leur taux de détection global et augmenter le nombre de faux négatifs.

3. OBJECTIF

L'objectif principal de cette étude est de concevoir des modèles de détection d'intrusion basés sur trois algorithmes d'apprentissage automatique : RF, XGBoost et DNN, appliqués à l'ensemble de données NSL-KDD, capables de maintenir une haute précision tout en réduisant les taux de faux positifs et de faux négatifs. Le choix de ces algorithmes repose sur leur efficacité démontrée dans des travaux antérieurs [7],10].

Afin d'atteindre cet objectif, plusieurs objectifs spécifiques sont définis :

L'étude vise dans un premier temps, à améliorer la précision globale des modèles de détection d'intrusion, tout en réduisant les risques de surapprentissage ainsi que les taux de faux positifs et de faux négatifs, afin de renforcer la fiabilité des résultats obtenus.

Ensuite, il s'agira de comparer les performances des trois algorithmes d'apprentissage automatique

(RF, XGBoost et DNN) afin d'évaluer leurs capacités respectives dans la détection des intrusions.

Enfin, l'objectif est d'identifier l'algorithme qui présente le meilleur compromis entre précision, et capacité de généralisation, de manière à proposer un modèle plus performant.

Dans ce cas, la question principale qui guide cette recherche est la suivante : *Comment concevoir des modèles de détection d'intrusion précis et capable de gérer le déséquilibre des classes tout en généralisant efficacement à différents types d'attaques ?*

4. METHODOLOGIE

L'approche proposée dans cette étude pour la détection d'intrusion repose sur six étapes fondamentales. Le processus commence par la sélection du jeu de données NSL-KDD, suivie du prétraitement, qui englobe la gestion des variables catégorielles, la sélection des caractéristiques pertinentes et la normalisation des données.

Ensuite, une validation croisée stratifiée est appliquée afin de garantir une division équilibrée des données en plis pour l'entraînement et le test. L'équilibrage des classes au niveau des données d'apprentissage est assuré par SMOTE. Cette méthode a été choisie suite à des expériences ablatives réalisées dans cette étude, montrant qu'elle améliorait le taux de rappel et réduisant les faux négatifs, ce qui justifie son adoption comme méthode principale. La phase suivante concerne la classification, où les modèles sont entraînés et testés. Pour l'optimisation des hyperparamètres, Optuna a été utilisé, car les expériences ablatives comparatives ont confirmé que cette approche améliorait la précision globale des modèles par rapport à GridSearchCV, justifiant ainsi son choix pour cette étude.

Enfin, la dernière étape consiste en une évaluation des performances, permettant de mesurer l'efficacité des modèles à l'aide de plusieurs indicateurs de performance.

4.1 Description du jeu de données NSL-KDD :

Dans la littérature actuelle, le jeu de données NSL-KDD [13,14] est l'un des plus utilisés pour l'évaluation des IDS, notamment dans le cadre d'études expérimentales impliquant diverses techniques ML. C'est pourquoi nous avons opté pour ce jeu de données afin de mener nos propres évaluations. Il s'agit d'une version améliorée du jeu de données KDD Cup 99 [15], dans laquelle les enregistrements redondants ont été éliminés.

Le jeu de données NSL-KDD est constitué de données simulées représentant des connexions de réseaux informatiques. Il comprend un total de 25 192 instances, réparties sur 42 caractéristiques. Parmi ces caractéristiques figurent le type de protocole, la durée de la connexion, le type de service réseau et le

nombre de connexions échouées, dont trois sont de nature qualitative. En plus, une variable cible est présente pour indiquer si une connexion est considérée comme normale ou anormale. Les attaques sont classifiées en quatre catégories : DoS, Probe, R2L et U2R, et sont regroupées sous une unique classe intitulée anormal.

4.2 L'approche proposée pour le système de détection d'intrusion réseau

Dans cette étude, nous avons suivi une méthodologie visant à améliorer la détection des intrusions réseau par le biais de techniques ML. L'approche a débuté par l'exploitation et le prétraitement des données, comprenant notamment la vérification des anomalies telles que les valeurs manquantes, les doublons et les valeurs aberrantes.

Les variables catégorielles ont ensuite été encodées à l'aide de LabelEncoder, et La réduction de la dimensionnalité a été réalisée via RFE, une méthode qui permet de sélectionner les variables les plus informatives, réduisant ainsi le risque de surajustement en éliminant les caractéristiques redondantes ou peu pertinentes [16].

La mise à l'échelle des caractéristiques a été effectuée avec RobustScaler afin d'assurer une uniformité des données tout en minimisant l'impact des valeurs aberrantes [17].

La validation croisée k-fold, dans sa version stratifiée, a été appliquée pour diviser les données en ensembles d'entraînement et de test, assurant ainsi une évaluation fiable et représentative des performances du modèle tout en préservant la distribution des classes dans chaque pli [18].

En raison de la nature déséquilibrée des données, la technique SMOTE a été utilisée sur les données d'entraînement, permettant de traiter les déséquilibres de classe et d'améliorer ainsi la capacité du modèle à identifier précisément les attaques [8],[19].

L'étape de classification a intégré trois algorithmes distincts : RF, XGBoost et DNN. Ces modèles ont été optimisés à l'aide du cadre Optuna [9], afin de déterminer la combinaison optimale de paramètres pour chaque modèle, garantissant ainsi des performances maximales dans le cadre de la détection des intrusions.

Les modèles ont été évalués à l'aide de plusieurs métriques de performance, incluant l'exactitude, la précision, le rappel, le score F1. Pour visualiser et synthétiser ces métriques, des outils d'évaluation tels que la matrice de confusion et le rapport de classification ont également été utilisés. Des analyses graphiques supplémentaires, notamment les courbes d'apprentissage et les courbes ROC, ont également été réalisées pour évaluer la robustesse et la performance des modèles. Toutes ces mesures ont été moyennées sur les différents plis de validation croisée, permettant ainsi d'obtenir une évaluation globale et fiable de l'efficacité des modèles.

L’approche méthodologique suivie pour la conception des classificateurs RF, XGBoost et DNN dans le cadre du système de détection d’intrusions réseau est représentée dans la Figure 1.

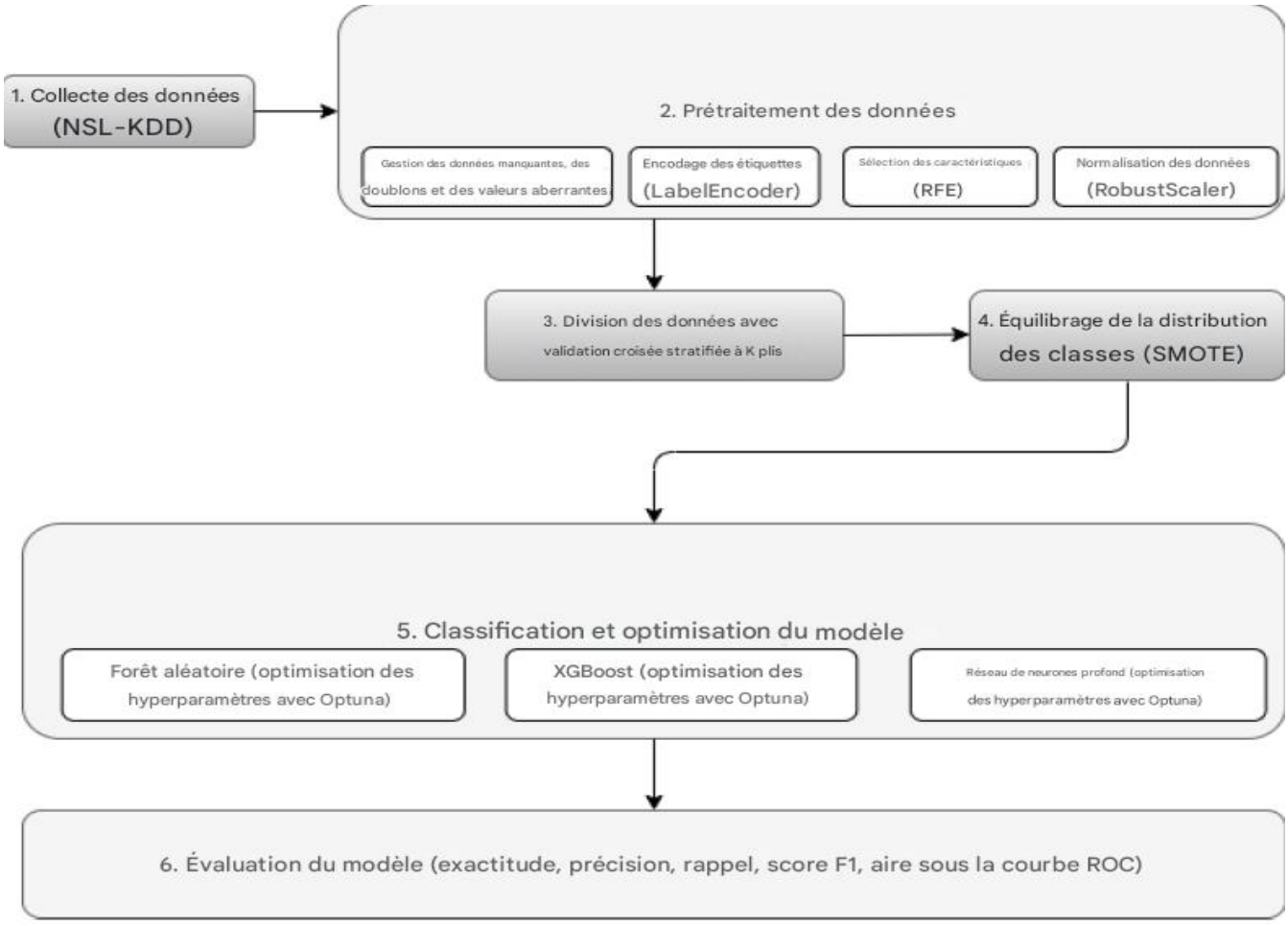

Figure 1Méthodologie proposée pour un système de détection

5. CONTRIBUTION

Dans cette étude, nous apportons des contributions visant à améliorer la conception et les performances des IDS. Nous démontrons que le réglage optimal des hyperparamètres via Optuna, appliqué aux modèles IDS sur l'ensemble de données NSL-KDD, améliore considérablement la précision de détection par rapport aux études précédentes, établissant notre approche comme une référence pour des modèles à haute précision.

Nous avons développé un cadre expérimental comparatif permettant d’évaluer trois algorithmes

d'apprentissage automatique RF, XGBoost et DNN dans des conditions expérimentales identiques. Ce cadre isole les effets du rééquilibrage des classes avec SMOTE et de l'optimisation des hyperparamètres, garantissant que les gains observés proviennent directement de ces techniques, ce qui constitue une contribution originale par rapport aux travaux existants.

Nos expériences montrent que même sur un ensemble de données relativement équilibré comme NSL-KDD, SMOTE améliore légèrement la détection des classes minoritaires, réduisant les faux négatifs et renforçant la fiabilité globale des modèles.

Notre comparaison détaillée des trois algorithmes, prenant en compte à la fois les performances de classification et les coûts de calcul, fournit une base expérimentale pratique pour la sélection des techniques IDS.

Enfin, ce travail a donné lieu à la rédaction de deux articles, dont un article de recherche publié et un autre article de synthèse finalisé.

6. ORGANISATION

Ce mémoire est présenté sous forme d'articles et est structuré en deux chapitres, encadrés par une introduction générale et une conclusion générale. L'introduction générale situe le contexte de la cybersécurité et la place des systèmes de détection d'intrusion (IDS) comme mécanismes essentiels pour renforcer la protection des réseaux. Elle met en évidence les limites des approches traditionnelles et justifie l'intérêt de recourir à des techniques d'apprentissage automatique. Elle définit également la problématique centrale de ce mémoire, à savoir l'amélioration de la précision et de l'efficacité des IDS, tout en formulant les objectifs de recherche.

Tandis que la conclusion générale résume les résultats obtenus en démontrant que les techniques proposées améliorent la performance des IDS, avec Random Forest atteignant la précision et l'AUC les plus élevées par rapport à XGBoost et DNN. Elle souligne aussi les limites liées à l'utilisation exclusive du jeu de données NSL-KDD. Enfin, elle ouvre des perspectives de recherche, notamment l'expérimentation sur des données réelles ou récentes, la prise en compte des attaques adversariales, et l'exploration d'approches hybrides pour mieux détecter les menaces émergentes.

Le premier chapitre présente une revue de littérature sur les techniques d'apprentissage automatique appliquées aux IDS. Il explore diverses techniques, telles que les réseaux de neurones récurrents (RNN), les réseaux de neurones convolutifs (CNN), les autoencodeurs, les ensembles d classificateurs, visant à améliorer la détection des activités malveillantes et la sécurité des réseaux. L'article met en lumière l'efficacité des IDS basés sur l'apprentissage automatique pour identifier des attaques complexes, en comparaison avec les approches traditionnelles. L'objectif principal de ce chapitre est d'offrir une vue

d'ensemble des approches récentes, afin de renforcer la pertinence de notre sujet de recherche et de justifier les choix des méthodes employées dans notre étude. Statut de l'article : finalisé.

Le deuxième chapitre présente le deuxième article, qui est l'article de recherche, publié dans le journal *Machine Learning with Applications (Hamidou, S. T., & Mehdi, A. (2025). Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks. Machine Learning with Applications, 100738) [22*].

Ce chapitre analyse les enjeux des systèmes de détection d'intrusion (IDS), en particulier la nécessité de maintenir un taux de détection élevé malgré le déséquilibre des données. Trois algorithmes Random Forest, XGBoost et DNN ont été comparés sur le jeu de données NSL-KDD, avec des techniques d'optimisation et de généralisation telles que RFE, SMOTE, Optuna et la validation croisée. Statut de l'article : Publié

CHAPITRE 1

ARTICLE 1 : UNE ENQUETE APPROFONDIE SUR LES SYSTEMES DE DETECTION D'INTRUSION RESEAU BASES SUR L'APPRENTISSAGE AUTOMATIQUE.

1.1 RESUME EN FRANÇAIS DU PREMIER ARTICLE

La détection d'intrusion joue un rôle crucial dans la protection des organisations en facilitant l'identification, l'analyse et la réponse rapide aux menaces informatiques, contribuant ainsi à la protection des systèmes critiques, des données sensibles et de la continuité opérationnelle.

Cet article propose une revue systématique des approches basées sur l'apprentissage automatique appliquées aux IDS, en analysant un large éventail d'études pour identifier les tendances dominantes et les avancées dans le domaine. L'étude met en évidence l'efficacité de techniques telles que les RNN, CNN, autoencodeurs et ensembles de classificateurs pour améliorer la précision et la robustesse des IDS.

Notre contribution dans ce chapitre consiste à synthétiser ces travaux pour fournir une vue d'ensemble claire des méthodes les plus performantes en matière de détection d'intrusion, constituant ainsi une base solide pour le développement et l'évaluation des modèles proposés dans ce mémoire.

Mots-clés : Cybersécurité, Cyberattaques, Systèmes de détection d'intrusion (IDS), Apprentissage automatique.

.

1.2 A COMPREHENSIVE SURVEY ON MACHINE LEARNING-BASED NETWORK INTRUSION DETECTION SYSTEMS

A Comprehensive Survey on Machine Learning-Based Network Intrusion Detection Systems

[1er] T.Hamidou Sow [2ème] Adda Mehdi
Department of Mathematics, Computer Science, and Engineering, University of Quebec at Rimouski
300, allée des Ursulines, Rimouski and G5L 3A1, Canada
thiernohamidou.sow@uqar.ca, mehdi_adda@uqar.ca

***Abstract*—Intrusion detection plays a crucial role in safeguarding organizations by facilitating the identification, analysis, and prompt response to cyber threats, thereby contributing to the protection of critical systems, sensitive data, and operational continuity. This paper offers a comprehensive review of machine learning-based approaches applied to Intrusion Detection Systems (IDS). The primary objective of this study is to provide an in-depth understanding of the techniques and methodologies employed within the field of intrusion detection. To achieve this, a systematic and exhaustive review of the existing scientific literature was conducted, covering a broad spectrum of studies to identify prevailing trends and advancements in IDS. The analysis underscores the effectiveness of various machine learning techniques, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), autoencoders, and classifier ensembles, in enhancing the accuracy and robustness of IDS. The reviewed literature demonstrates that these methods enable high detection accuracy, positioning them as essential tools for strengthening cybersecurity mechanisms.**

***Index Terms*—Cybersecurity, Cyberattacks, Intrusion Detection Systems (IDS), Machine Learning.**

I. Introduction

Cybersecurity represents a major challenge for organizations, which are confronted with an increasing number of cyber threats that can compromise their systems, data, and reputation. With the rise of the internet, the volume of sensitive data exchanged over networks has significantly increased, making digital infrastructures more vulnerable to cyberattacks. These attacks exploit security vulnerabilities to access confidential information, thus compromising the integrity, availability, and confidentiality of information systems.

In this context, intrusion detection systems (IDS) play a fundamental role within security infrastructures. These systems continuously monitor network traffic to detect suspicious or malicious activities, as well as any violations of security policies. Thanks to these capabilities, IDS provide network administrators with enhanced visibility of threats, allowing them to anticipate and respond effectively to attacks [1].

Intrusion detection systems (IDS) are divided into two types: host-based IDS (HIDS), which monitor the activity of a specific device, and network-based IDS (NIDS), which analyze overall network traffic [2]. HIDS offer detailed monitoring but require installation on each device, which can affect performance. In contrast, NIDS detect threats at the network level without direct intervention on the hosts. Attack detection relies on two methods: signature-based analysis, which identifies known threats, and anomaly-based analysis, which detects unusual behaviors [3].

Increasingly, IDS presented in the literature have favored the anomaly-based approach, using techniques such as machine learning and deep learning to detect intrusions in networks [4,5].

This study explores machine learning-based intrusion detection systems (IDS), an approach that uses artificial intelligence to identify intrusions and abnormal behaviors in computer networks. Unlike traditional IDS, which rely on predefined rules and signatures to detect threats, machine learning-based IDS can identify more complex patterns and have the ability to detect new (zero-day) attacks, thus providing more robust protection against cyberattacks.

The problem addressed in this research lies in the evaluation of the effectiveness and limitations of the techniques used in machine learning-based intrusion detection. This study is particularly relevant in a context where cyberattacks are multiplying, jeopardizing the security of organizations and their sensitive data.

This article is divided into several key sections. First, we will present the fundamentals of intrusion detection systems (IDS), detailing the different types of these systems as well as the techniques used for intrusion detection. Next, we will provide a literature review on IDS using machine learning, highlighting the various methods applied in this field. Finally, we will analyze the main limitations of these systems.

II. Fundamentals of Intrusion Detection Systems (IDS):

A. Definition

An Intrusion Detection System (IDS) is a device or software designed to monitor network or system activities in search of abnormal, potentially malicious activities, or security breaches. The IDS analyzes network traffic, system logs to detect signs of intrusion or suspicious activity[6], while generating alerts upon detection, although some of them may be false alarms.

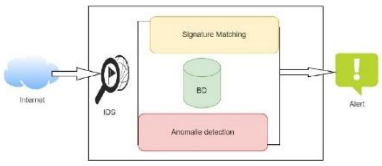

Figure 1. Intrusion Detection System (IDS).

B. Types of Intrusion Detection Systems:

1) Network Intrusion Detection Systems (NIDS): A Network Intrusion Detection System (NIDS) is a software solution designed to detect potential security issues within a network. It continuously monitors network traffic, looking for behaviors that may indicate malicious intent or abnormal activity. NIDS can analyze both incoming and outgoing network traffic, identifying attack patterns such as Denial of Service (DoS) attacks, unauthorized access attempts, and other suspicious activities. By detecting these threats in real-time, NIDS helps administrators take immediate action to prevent or mitigate security vulnerabilities[6].

2) Host-based Intrusion Detection Systems (HIDS): HIDS (Host-Based Intrusion Detection System) is a type of IDS designed to monitor and analyze the activities of a machine or a single host, such as a computer, server, or IoT (Internet of Things) device. Unlike network-based intrusion detection systems (NIDS), which focus on network traffic, HIDS focuses on analyzing system files, logs, and running processes to detect abnormal behavior or malicious intrusion attempts[6].

C. Intrusion Detection Techniques:

Intrusion detection serves to spot vulnerabilities in network traffic as well as on hosts. Generally, two methods are employed for detecting intrusions: anomaly-based detection and signature-based detection.

1) Anomaly-based Intrusion Detection: Anomaly-based intrusion detection systems, also known as behavior-based IDS, build a reference model of normal system behavior and identify potential threats by detecting deviations from this established model. This approach is particularly effective in identifying previously unknown attacks. However, its effectiveness is limited by a typically higher rate of false positives, which can hinder its practical deployment in real-world environments [7].

2) Signature-based Intrusion Detection: Signature-based intrusion detection, also known as misuse detection or knowledge-based detection, identifies attacks by comparing network traffic to stored signatures and patterns in a database. When a match is found, an alert is triggered. This approach is easy to implement and highly effective at detecting known threats, with a low false positive rate. However, it cannot detect new or zero-day attacks and requires significant resources for continuous updating and management of its signature database [7].

III. Machine Learning Techniques in Intrusion Detection

Intrusion detection is a constantly evolving field, characterized by a variety of approaches. Recent studies have demonstrated that the use of machine learning methods and the exploitation of various data sources can enhance the prediction and detection of cyberattacks.

A. Recurrent Neural Networks (RNNs)

Ansari, Bartoš, and Lee[8] preferred Gated Recurrent Units (GRU) over LSTM to predict specific features of cyberattacks, including severity and target. Their approach effectively handles sequential data, making it suitable for detecting patterns of network traffic indicating potential intrusions. On the other hand, Akwetey Henry Matey, Paul Danquah, and Godfred Yaw Koi-Akrofi[9] evaluated cybersecurity vulnerabilities among Independent Power Producers (IPPs) based on a quantitative approach (PLS-SEM). Their study provided better insight into the human role in cyberattacks and validated the effectiveness of security measures.

Yeboah-Ofori and al.[10] combined Adversarial Machine Learning (AML) with ontology-based approaches for in-depth analysis of cyber threats. Their research on Cyber Threat Ontology (CTO) and AML focused on modeling APT attacks and predicting responses to attacks after the insertion of adversarial attacks.

Martin Husak and Jaroslav Kaspar[11] adopted a data-driven approach using real security alert data to predict attacks. Their study illustrates how data mining and sequential rule extraction can be used to build effective prediction models, validating their results in real operational settings. Jun Zhao and al.[12] predicted cyberattack preferences using attributed heterogeneous information networks (AHIN). Their HinAp framework demonstrated its ability to model attack preferences and apply transductive learning with meta-graphs.

The effectiveness of neural networks in predicting cyberattacks has been demonstrated by Bohdan Bebeshko and al.[13] They explored improving prediction model performance using techniques such as MLP, SVM, and KNN, and managing missing data to increase model reliability.

Additionally, Ouissem Ben Fredj and al.[14] explored the use of deep learning techniques, including LSTM, RNN, and MLP, to predict cyberattacks. Their study reveals that LSTM and RNN models outperform for time series data, indicating an evolution in attack prediction effectiveness.

Sydney Mambwe Kasongo[15] also utilized recurrent neural networks for IDS, finding that RNNs outperformed LSTM and GRU methods, consistent with the findings of TONGTONG SU and al.

Congyuan Xu and al.[16] enhanced IDS using recurrent neural networks with GRU, demonstrating their effectiveness against various attacks such as DOS and U2R. This complements Kasongo's results on the effectiveness of RNNs.

Yang Jia and al.[17] proposed an intrusion detection method based on a deep neural network model (NDNN), achieving

high accuracy, highlighting the effectiveness of deep learning approaches.

B. Convolutional Neural Networks (CNNs))

Researchers Moshe Kravchik and Asaf Shabtai [18] focused on identifying cyberattacks in industrial control systems, proposing an anomaly detection method based on measuring the statistical deviation between predicted and observed values. They applied this method using different architectures of deep neural networks, including variants of convolutional and recurrent networks. The SWaT test dataset included 36 different cyberattacks. The proposed method successfully detects the vast majority of attacks with a low false positive rate.

At the same time, P. Shanmuga Prabha and S. Magesh Kumar [19] developed a system to predict cyberattacks on Internet of Things (IoT) devices, using a Recurrent Convolutional Neural Network (R2CNN), which reduced the time for detecting malware and unauthorized intrusions.

C. Deep Neural Networks (DNNs)

Research by Wen Xu and al.[20] highlighted the performance of multilayer autoencoders in network anomaly detection, distinguishing itself from traditional methods in terms of accuracy and F1 score. Similarly, the study conducted by Bilal Mohammeda and Ekhlas K. Gbash[21] demonstrated the ability of deep and recursive neural networks to detect intrusions, as evidenced by their high precision on the NSL-KDD database.

Finally, Shisrut Rawat, Aishwarya Srinivasan, Vinayakumar Ravi, and Uttam Ghosh [22] conducted a study focused on network intrusion detection. Their work stands out for an approach that integrates both classical machine learning techniques and deep neural networks (DNNs). This method proved effective in identifying various types of attacks, including DOS, R2L, U2R, and Probe, while leveraging the NSL-KDD dataset.

D. Autoencoders

Sandeep Gurung and al.[23] designed an IDS using the "sparse auto-encoder" deep learning approach, demonstrating its effectiveness against DOS, R2L, U2R, and Probing attacks, reinforcing the notion of the effectiveness of deep learning techniques. Akalanka Bandara Mailewa and Shehram Sikander Khan[24] detected attacks by combining deep autoencoders with Support Vector Machines (SVMs), as well as Principal Component Analysis (PCA) and Support Vector Machines (SVMs), to accurately and reliably detect cyberattacks.

E. Classifier Ensembles

In the field of Software-Defined Networks (SDN), Saurav Nanda and al. [25] bolstered security by leveraging machine learning. They utilized various algorithms to predict and identify SSH brute force attacks, illustrating the effectiveness of using historical data in training predictive models.

Studies conducted by researchers such as Md. Badiuzzaman Pranto and al. [26] have demonstrated the importance of feature selection and the use of classifier ensembles, including Random Forest, for detecting various attacks such as DDoS and phishing. In a similar study, Mouhammad Alkasassbeh and Mohammad Almseidin [27] explored the effectiveness of different machine learning classifiers, including J48, to detect attacks such as DOS and PROBE, highlighting the versatility of classifiers in IDS systems.

Similarly, Chongzhen Zhang and al. [28] proposed a framework for intrusion detection, combining Stacked Autoencoder, Random Forest, and CART, revealing the effectiveness of Random Forest in various attack contexts. This approach complements the work of AHEDI AZAM and al. [29], who highlighted the effectiveness of decision trees, especially C4.5 and XGBOOST models, in intrusion detection, aiding in categorizing IDS systems and selecting optimal models. Azam Rashid and al. [6], in their comparative analysis of IDS, employed techniques such as SVM, Naive Bayes, and k-NN, demonstrating the effectiveness of KNN against various attacks, highlighting the importance of a hybrid approach and feature selection. The work of Fekadu Yihunie and al. [30] showed superior performance of Random Forests compared to other classifiers, consistent with previous studies.

Gaurav Meena and Ravi Raj Choudhary [31] compared J48 Graft Decision Tree and Naive Bayes, revealing the effectiveness of J48 in intrusion detection, a commonality with the work of Alkasassbeh and Almseidin.

Furthermore, the approach by Prakash Chandra and al. [32] using MRFA, presented as an enhanced version of Random Forest, illustrates the importance of customizing and optimizing traditional techniques to strengthen intrusion detection.

F. Statistical and Probabilistic Models

Continuing the exploration of predicting cyberattacks in specific contexts, Hidden Markov Models and Bayesian Networks have been applied by researchers like Ghafir et al.[33] to predict Advanced Persistent Threats (APT), while Jemili and al.[34] used hybrid Bayesian approaches that represent probabilistic relationships between different variables to predict complex multi-stage attacks. This use of statistical and probabilistic models indicates an evolution in cyberattack prediction methodologies.

IV. Taxonomy of Machine Learning Approaches for IDS

To provide a clearer structure to the reviewed studies, we propose a taxonomy of machine learning approaches applied to Intrusion Detection Systems (IDS). This taxonomy organizes existing works according to four main dimensions: (i) the type of model employed, (ii) the IDS deployment scenario, (iii) the detection strategy, and (iv) the targeted application domain.

- **Type of model:** traditional machine learning methods (e.g., Random Forest, SVM, k-NN) versus deep learning approaches (e.g., CNN, RNN, DNN, Autoencoders, GANs).
- **IDS deployment:** Host-based IDS (HIDS) versus Network-based IDS (NIDS).

Table I
DIFFERENT MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION

Article	Dataset	Feature selection	Detection techniques	Attack detected	Result
[6]	NSL-KDD and CIDDS-001	CFS, IGR, Gini Index	SVM, Naïve Bayes, k-NN, NN, DNN and AE.	DoS, Probe, U2R, R2L	KNN has 99.80 accuracy.
[8]	KDD 99 and NSL-KDD	-	GRU	DoS, R2L, U2L	Over 76% accuracy.
[9]	Producer Independent Power Producer (PIE) survey method.	-	Quantitative approach (PLS-SEM)	Phishing, Malware, Privacy Breach, DDoS	C15 (0.018), indicating a small effect. AK2 (0.264), medium impact. C6 (0.543), medium effect. C5 (0.617) large impact.
[10]	Microsoft Windows Defender public data.	-	CTO and AML.	Ransomware, Spyware	GBoost achieves 78% precision, while RF achieves 73%.
[11]	CTU: 13 different scenarios and DARPA: DDoS attack	-	Metastability theory and statistical analysis	DDoS	The return rate (Kendall tau) being positively correlated at 0.590.
[12]	Hacker forums, security blogs	-	Heterogeneous Attributed Information Networks (AHIN)	Exploitation preference, platform preference, domain preference	HinAp has a precision respectively of 0.8912, 0.8149, and 0.8689 for attack preferences.
[13]	KDD Cup 99	-	MLP, SVM, KNN.	Endpoint Malware, Malicious Destination, Malicious Email.	AUC values for KNN EM(0.88), MD(0.91), ME(0.95).
[14]	CTF'17 Dataset provided by Defcon platform	-	LSTM, RNN, and MLP	DoS, Probe, R2L, U2L	LSTM with an F-measure above 93%.
[15]	UNSW-NB15 and NSL-KDD	XGboost	RNN, LSTM and GRU	Exploits, Fuzzers, DoS, Reconnaissance, Backdoor, Shellcode	LSTM-XGBoost (85.93%), GRU-XGBoost (85.65%) accuracy.
[16]	KDD 99 and NSL-KDD	-	RNN, BGRU + MLP, LSTM, GRU	DoS, R2L, U2R	BGRU + MLP (KDD 99 and NSL-KDD) has 99.8% and 99.24% accuracy.
[17]	KDD99 and NSL-KDD	-	NDNN	DoS, R2L, U2R, Probe	Accuracy is over 98%, and F-measure can reach 98.84%.
[18]	SWaT (Secure Water Treatment) Dataset	-	Unsupervised Convolutional Neural Networks	Saturation, Injection, Identity Spoofing, Denial of Service	AUC of 0.967 for the eight-layer convolutional network.

Table II
DIFFERENT MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION

Article	Dataset	Feature selection	Detection techniques	Attack detected	Result
[19]	Network traffic information, IoT sensor data	-	Recurrent Convolutional Neural Network (R2CNN)	DoS, Malware, Unauthorized Intrusions, Man-in-the-Middle	CR2CNN achieved an accuracy of 99.2%.
[20]	NSL KDD	PCA	Autoencoder (AE)	DoS, R2L, U2R, Probe	AE achieved an accuracy of over 90% and F1 score of 92.26%.
[21]	NSL KDD	RFC-RFE	RNN and DNN	DoS, R2L, U2R, Probe	DNN (binary) has an accuracy of 94%, and RNN (multiclass) has 94% accuracy.
[22]	NSL-KDD	PCA	DNN	DoS, R2L, U2R, Probe	PCA + Deep Neural Network has an accuracy of 98.2%, and DNN 97.2% accuracy.
[23]	NSL KDD	-	Sparse auto-encoder	DoS, R2L, U2R, Probing	The model's accuracy was calculated at 87.2%.
[24]	NSL KDD	PCA + SVM	Deep Autoencoders (DAE) and SVM	DoS, R2L, U2R, Probing	AE + SVM has a precision of 95% for DOS attacks and 65% for Probe attacks.
[25]	Marist College "LongTail" project data	-	C4.5, BayesNet, DT and Naive-Bayes.	Brute Force, SSH.	Bayesian networks (BayesNet) achieve 91.68% accuracy.
[26]	NSL-KDD	-	RF, K-NN, SVM	DDoS, SQL injection, phishing, brute-force	RF achieved an accuracy of 99.5% and a precision of 99.3%.
[27]	NSL-KDD	-	J48, MLP, and Bayesian network.	DoS, R2L, U2R, and PROBE	J48 has an accuracy of 93.1083% and a precision of 0.989.
[28]	CICIDS2017	AE, SAE, PCA	SAE, RF, Regression Tree	Port scan, BF, DDoS, XSS, SQL injection	RF has an accuracy of 0.99 and SAE has an accuracy of 91.83 and 91.65 for binary and multiclass class.
[29]	NSL-KDD and KDD Cup99	PCA, DVS, RFE	C4.5, Naïve Bayes, NB-Tree, Multilayer Perceptron, DT	DoS, DDoS, Source Port Manipulation, Spoofing	RFE for DT has 0.99 accuracy.
[30]	NSL-KDD	-	Stochastic Gradient Descent (SGD), LogReg, SVM, RF	DoS, R2L, U2R, and PROBE	RF has a precision of 0.99.

Table III
DIFFERENT MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION

Article	Dataset	Feature selection	Detection techniques	Attack detected	Result
[31]	NSL-KDD	-	J48 Graft Decision Tree and Naive Bayes	DoS, SYN Flood, R2L, U2R	J48 Graft Decision Tree 99.435% accuracy.
[32]	KDD-Cup 99, NSL KDD	-	Modified Random Forest Algorithm (MRFA)	DoS, R2L, U2R, Probe	MRFA achieved 97.9 accuracy with KDD-Cup 99 and 99.25 with NSL KDD.
[33]	Network traffic data, security sensor.	-	Hidden Markov Model (HMM), Forward-Backward Algorithm, Viterbi Algorithm, Baum-Welch Algorithm	Targeted Intrusions, Advanced Phishing, Persistent Malware, Lateral Movement	At least 91.80% accuracy in APT sequence estimation.
[34]	Grand Challenge Problem (GCP).	-	K2 algorithm, Junction Tree Inference Algorithm, Hybrid Propagation	Buffer Overflow Attack, Suspicious Incoming Connection, Suspicious Outbound Connection, Suspicious Data Export	94.5% correlation.
[35]	Social media (Twitter), GDELT open source project.	-	Bayesian network	Malware, Defacement, DOS and MEU	Study achieves AUC values of at least 0.70 for all attack types.
[36]	D2web, Ground Truth Database.	-	Probabilistic relational reasoning (PRR), probabilistic logic programming (APT-LOGIC)	M-E, MD, E-M	F1 score around 45% and 57%.
[37]	Dark/deep web, Twitter, blogs, NVD.	-	ARIMA and ARIMAX and LSTM, Phased LSTM and GRU	Malicious destination attacks, malicious URLs, malware attacks on endpoints, malicious email attacks	GRU F1 score 65.30 and ARIMAX F1 score 65.22.
[38]	Honeypot Programs: Dionaea3, Mwcollector4, Amun5, and Nepenthes	-	BRNN-LSTM	TCP or UDP-based attacks, Unfinished TCP, Unassigned ports, Honeypots	PMAD and MAPE (.01243741 and .01387808), prediction errors are less than 5%.
[39]	NSL-KDD	-	BAT-MC (BLSTM)	DoS, R2L, U2R, Probe	BAT-MC accuracy reaches 84.25%.
[40]	NSL-KDD, Kyoto Honeypot, MAWILab	-	CNN	DoS, R2L, U2R, and PROBE	99% binary classification accuracy on both datasets.
[41]	NSL-KDD	-	NB, K-means, SVM, DT, RF, AE	DoS, FB	Layer Autoencoder (AE) has 90.61% accuracy.

- **Detection strategy:** signature-based detection, anomaly-based detection, and hybrid approaches.
- **Application domain:** general-purpose networks, Internet of Things (IoT), Cloud computing, Industrial Control Systems (ICS/SCADA), and Software-Defined Networks (SDN).

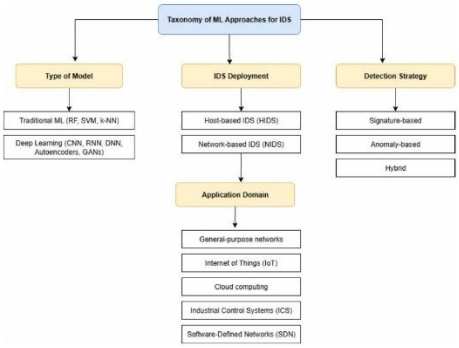

Figure 2. Taxonomy of Machine Learning Approaches for IDS

V. Discussions and Limitations

Despite the advantages of machine learning for intrusion detection, several limitations remain. The performance of models heavily depends on the quality and quantity of available data. Imbalanced, incomplete, or outdated datasets can lead to biases, false positives, and poor generalization to unseen attacks. Moreover, most publicly available datasets do not fully reflect modern network environments or emerging attack types, limiting the applicability of the models.

Attackers can also bypass IDS using sophisticated evasion techniques, such as adversarial attacks, polymorphic malware, or slow-and-stealthy attacks. These threats emphasize the need to develop robust and resilient detection methods capable of resisting manipulations and novel attack strategies.

Additionally, the computational complexity and execution time of models pose challenges for real-time detection. Complex architectures, high-dimensional feature spaces, and ensemble methods can generate significant latency, which is critical in industrial networks or cloud infrastructures. Optimizing efficiency while maintaining high accuracy remains a major challenge.

Finally, challenges related to deployment and integration in heterogeneous environments, maintaining model updates, and complying with regulatory requirements represent further obstacles to the practical adoption of IDS solutions.

Although machine learning offers considerable potential to enhance intrusion detection, it is essential to address limitations related to data quality, robustness, computational complexity, and deployment to maximize IDS effectiveness.

VI. Conclusion

This article provides an overview of intrusion detection systems (IDS) based on machine learning. It highlights the effectiveness of methods used in intrusion detection, such as recurrent neural networks, convolutional neural networks, and classifier ensembles, to protect increasingly vulnerable confidential data from cyber threats. This study also helps identify the most effective algorithms for designing an intrusion detection model based on each type of data.

Despite the improvement of IDS through machine learning, challenges remain, such as data quality, false alert management, and the detection of new attacks. Therefore, it is essential to continue improving these techniques to ensure robust protection of computer networks and confidential information.

VII. References:

1) Patel, A., Alhussian, H., Pedersen, J.M., Bounabat, B., Júnior, J.C., and Katsikas, S. (2017). An intelligent collaborative architecture for intrusion detection and prevention in smart network ecosystems. *In Proceedings of the 2017 9th International Conference on Computer Science and Information Technology (CSIT)* (pp. 92-109). Elsevier. https://doi.org/10.1016/j.cose.2016.07.002.
2) Bridges, R.A., Glass-Vanderlan, T.R., Iannacone, M.D., Vincent, M.S., and Chen, Q. (2020). A study of host data-driven intrusion detection systems. *In Proceedings of the 2020 52nd ACM Computing Surveys* (pp. 1-35). ACM. https://doi.org/10.1145/3344382.
3) Aldweesh, A., Derhab, A., and Emam, A.Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *In Proceedings of the 2020 189th Knowledge-Based Systems* (pp. 105124). Elsevier. https://doi.org/10.1016/j.knosys.2019.105124.
4) Otoum, Y., Liu, D., and Nayak, A. (2022). DL-IDS: A deep learning-based intrusion detection framework for securing IoT. *In Proceedings of the 2022 Transactions on Emerging Telecommunications Technologies* (pp. e3803). Wiley.
5) Azeroual, H., Daha Belghiti, I., and Berbiche, N. (2022). A framework for implementing an ML or DL model to improve intrusion detection systems (IDS) in the NTMA context, with an example on the dataset (CSE-CIC-IDS2018). *In Proceedings of the 2022 ITM Web of Conferences* (Vol. 46, pp. EDP Sciences).
6) Rashid, A., Siddique, M.J., and Ahmed, S.M. (2020). Machine and deep learning-based comparative analysis using hybrid approaches for intrusion detection systems. *In 2020 3rd International Conference on Advancements in Computational Sciences (ICACS)* (pp. 1-9). IEEE. https://doi.org/10.1109/ICACS47775.2020.9055946.
7) Abdulganiyu, O.H., Ait Tchakoucht, T., and Saheed, Y.K. (2023). A systematic literature review for network intrusion detection systems (IDS). *In Proceedings of the 2023 International Journal of Information Security* (pp. 1125-1162). Springer. https://doi.org/10.1007/s10207-023-00682-2.
8) Ansari, M. S., Bartoš, V., and Lee, B. (2022, March). GRU-based deep learning approach

for network intrusion alert prediction. *Future Generation Computer Systems, 128*, 235–247. https://doi.org/10.1016/j.future.2021.09.040.

9) Matey, A. H., Danquah, P., and Koi-Akrofi, G. Y. (2022). Predicting Cyber-Attack using Cyber Situational Awareness: The Case of Independent Power Producers (IPPs). *International Journal of Advanced Computer Science and Applications, 13*(1). https://doi.org/10.14569/IJACSA.2022.0130181.
10) Yeboah-Ofori, A., Ismail, U. M., Swidurski, T., and Opoku-Boateng, F. (2021, July). Cyber Threat Ontology and Adversarial Machine Learning Attacks: Analysis and Prediction Perturbance. *In 2021 International Conference on Computing, Computational Modelling and Applications (ICCMA)* (pp. 71-77). IEEE. https://doi.org/10.1109/ICCMA53594.2021.00020.
11) Husak, M., and Kaspar, J. (2018, June). Towards Predicting Cyber Attacks Using Information Exchange and Data Mining. *In 2018 14th International Wireless Communications and Mobile Computing Conference (IWCMC)* (pp. 536-541). IEEE. https://doi.org/10.1109/IWCMC.2018.8450512.
12) Zhao, J., et al. (2021, March). Automatically predicting cyber attack preference with attributed heterogeneous attention networks and transductive learning. *Computers and Security, 102*, 102152. https://doi.org/10.1016/j.cose.2020.102152.
13) Bebeshko, B., Khorolska, K., Kotenko, N., Kharchenko, O., and Zhyrova, T. (n.d.). Use of Neural Networks for Predicting Cyberattacks.
14) Ben Fredj, O., Mihoub, A., Krichen, M., Cheikhrouhou, O., and Derhab, A. (2020, November). CyberSecurity Attack Prediction: A Deep Learning Approach. *In 13th International Conference on Security of Information and Networks* (pp. 1-6). ACM. https://doi.org/10.1145/3433174.3433614.
15) Kasongo, S. M. (2023, February). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computers and Communications, 199*, 113–125. https://doi.org/10.1016/j.comcom.2022.12.010.
16) Xu, C., Shen, J., Du, X., and Zhang, F. (2018). An Intrusion Detection System Using a Deep Neural Network With Gated Recurrent Units. *IEEE Access, 6*, 48697–48707. https://doi.org/10.1109/ACCESS.2018.2867564.
17) Jia, Y., Wang, M., and Wang, Y. (2019, January). Network intrusion detection algorithm based on deep neural network. *IET Information Security, 13*(1), 48–53. https://doi.org/10.1049/iet-ifs.2018.5258.
18) Kravchik, M., and Shabtai, A. (2018, December 10). Detecting Cyberattacks in Industrial Control Systems Using Convolutional Neural Networks. *arXiv*. Consulted on: December 21, 2023. Available at: http://arxiv.org/abs/1806.08110.
19) Prabha, P. S., and Kumar, S. M. (2022). A Novel Cyber-attack Leads Prediction System using Cascaded R2CNN Model. *International Journal of Advanced Computer Science and Applications, 13*(2). https://doi.org/10.14569/IJACSA.2022.0130260.
20) Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., and Sabrina, F. (2021). Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset. *IEEE Access, 9*, 140136–140146. https://doi.org/10.1109/ACCESS.2021.3116612.
21) Mohammed, B., and Gbashi, E. (2021, July). Intrusion Detection System for NSL-KDD Dataset Based on Deep Learning and Recursive Feature Elimination. *Engineering Technology Journal, 39*(7), 1069–1079. https://doi.org/10.30684/etj.v39i7.1695.
22) Rawat, S., Srinivasan, A., Ravi, V., and Ghosh, U. (2022, January). Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network. *Internet Technology Letters, 5*(1), e232. https://doi.org/10.1002/itl2.232.
23) Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majitar, Sikkim, India, Gurung, S., Ghose, M. K., and Subedi, A. (2019, March). Deep Learning Approach on Network Intrusion Detection System using NSL-KDD Dataset. *International Journal of Computer Networks and Information Security, 11*(3), 8–14. https://doi.org/10.5815/ijcnis.2019.03.02.
24) Khan, S. S., and Mailewa, A. B. (2023, March). Detecting Network Transmission Anomalies using Autoencoders-SVM Neural Network on Multi-class NSL-KDD Dataset. *In 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0835–0843). IEEE. https://doi.org/10.1109/CCWC57344.2023.10099056.
25) Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., and Yang, B. (2016, November). Predicting network attack patterns in SDN using machine learning approach. *In 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)* (pp. 167–172). IEEE. https://doi.org/10.1109/NFV-SDN.2016.7919493.
26) Pranto, Md. B., Ratul, Md. H. A., Rahman, Md. M., Diya, I. J., and Zahir, Z.-B. (2022). Performance of Machine Learning Techniques in Anomaly Detection with Basic Feature Selection Strategy - A Network Intrusion Detection System. *Journal of Advanced Information Technology, 13*(1), 36–44. https://doi.org/10.12720/jait.13.1.36-44.
27) Rajwar, S. K., Mukherjee, Dr. I., and Manjhi, Dr. P. K. (2020). Machine Learning Methods for Network Intrusion Detection. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3610618.
28) Zhang, C., and al. (2021, January). A Novel Framework Design of Network Intrusion Detection Based on Machine Learning Techniques. *Security and Communication Networks, 2021*, 1–15.

https://doi.org/10.1155/2021/6610675.

29) Azam, Z., Islam, Md. M., and Huda, M. N. (2023). Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree. *IEEE Access, 11*, 80348–80391. https://doi.org/10.1109/ACCESS.2023.3296444.
30) Yihunie, F., Abdelfattah, E., and Regmi, A. (2019, May). Applying Machine Learning to Anomaly-Based Intrusion Detection Systems. *In 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* (pp. 1–5). IEEE. https://doi.org/10.1109/LISAT.2019.8817340.
31) Meena, G., and Choudhary, R. R. (2017, July). A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA. *In 2017 International Conference on Computer, Communications and Electronics (Comptelix)* (pp. 553–558). IEEE. https://doi.org/10.1109/COMPTELIX.2017.8004032.
32) Chandra, P., Lilhore, U. K., and Agrawal, N. (n.d.). Network Intrusion Detection System Based on Modified Random Forest Classifiers for KDD Cup-99 and NSL-KDD Dataset, 04(08).
33) Ghafir, I., et al. (2019). Hidden Markov Models and Alert Correlations for the Prediction of Advanced Persistent Threats. *IEEE Access, 7*, 99508–99520. https://doi.org/10.1109/ACCESS.2019.2930200.
34) Jemili, F., Zaghdoud, M., and Ahmed, M. B. (n.d.). Attack Prediction based on "Hybrid" Propagation in Bayesian Networks.
35) Okutan, A., Yang, S. J., and McConky, K. (2018, March 26). Forecasting Cyber Attacks with Imbalanced Data Sets and Different Time Granularities. *arXiv*. Consulted on: December 21, 2023. Available at: http://arxiv.org/abs/1803.09560.
36) Almukaynizi, M., and al. (2018, October 29). DARKMENTION: A Deployed System to Predict Enterprise-Targeted External Cyberattacks. *arXiv*. Consulted on: December 21, 2023. Available at: http://arxiv.org/abs/1810.12492.
37) Goyal, P., and al. (2018, June 8). Discovering Signals from Web Sources to Predict Cyber Attacks. *arXiv*. Consulted on: December 21, 2023. Available at: http://arxiv.org/abs/1806.03342.
38) Fang, X., Xu, M., Xu, S., and Zhao, P. (2019, December). A deep learning framework for predicting cyber attack rates. *EURASIP Journal on Information Security, 2019*(1), 5. https://doi.org/10.1186/s13635-019-0090-6.
39) Su, T., Sun, H., Zhu, J., Wang, S., and Li, Y. (2020). BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. *IEEE Access, 8*, 29575–29585. https://doi.org/10.1109/ACCESS.2020.2972627.
40) Kwon, D., Natarajan, K., Suh, S. C., Kim, H., and Kim, J. (2018, July). An empirical study on network anomaly detection using convolutional neural networks. *In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)* (pp. 1595–1598). IEEE. https://doi.org/10.1109/ICDCS.2018.00178.
41) Magdy, M. Eshak, Matter, A. M., Hussin, S., Hassan, D., and Elsaid, S. (2022, September). A comparative study of intrusion detection systems applied to NSL-KDD dataset. *Egyptian International Journal of Engineering Sciences and Technology*, 0(0), 0–0. https://doi.org/10.21608/eijest.2022.137441.1156.

CHAPITRE 2

ARTICLE 2 : AMELIORATION DES PERFORMANCES DES SYSTEMES DE DETECTION D'INTRUSION (IDS) GRACE A UNE ANALYSE COMPARATIVE DES FORETS ALEATOIRES, XGBOOST ET DES RESEAUX NEURONAUX PROFONDS

2.1 RESUME EN FRANÇAIS DU DEUXIEME ARTICLE

Les IDS font face à des défis majeurs en matière de sécurité des réseaux, notamment la nécessité de combiner un taux de détection élevé avec une performance fiable. Cette fiabilité est souvent affectée par des déséquilibres de classes et une optimisation insuffisante des hyperparamètres.

Cet article aborde la question de l'amélioration du taux de détection des IDS en évaluant et comparant trois algorithmes d'apprentissage automatique : RF, XGBoost et DNN, en utilisant le jeu de données NSL-KDD.

Dans notre méthodologie, nous intégrons SMOTE (Synthetic Minority Oversampling Technique) pour traiter la nature déséquilibrée des données, garantissant ainsi une représentation plus équilibrée des différentes classes. Cette approche aide à optimiser les performances du modèle, à réduire les biais et à renforcer la robustesse. De plus, l'optimisation des hyperparamètres est réalisée à l'aide d'Optuna, garantissant que chaque algorithme fonctionne à son niveau optimal.

Les résultats montrent que notre modèle, utilisant l'algorithme Random Forest, atteint une précision de 99,80 %, surpassant les performances de XGBoost et DNN. Cela fait de notre approche un véritable atout pour les méthodes de détection d'intrusion dans les réseaux informatiques.

Ainsi, notre contribution principale réside dans la conception d'un cadre d'évaluation combinant SMOTE et Optuna, permettant d'améliorer la précision, et la généralisation des modèles de détection d'intrusion.

Mots-clés : Cybersécurité, Système de détection d'intrusion (IDS), Apprentissage automatique, Apprentissage profond, NSL-KDD, SMOTE.

2.2 ENHANCING IDS PERFORMANCE THROUGH A COMPARATIVE ANALYSIS OF RANDOM FOREST, XGBOOST, AND DEEP NEURAL NETWORKS

Machine Learning with Applications 22 (2025) 100738

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks

Sow Thierno Hamidou *, Adda Mehdi

Department of Mathematics, Computer Science, and Engineering, University of Quebec at Rimouski, 300, allée des Ursulines, Rimouski, QC G5L 3A1, Québec, Canada

ARTICLE INFO

Keywords:
Cybersecurity
Intrusion Detection System (IDS)
Machine learning
Deep learning
NSL-KDD
Smote

ABSTRACT

Intrusion Detection Systems (IDS) face major challenges in network security, notably the need to combine a high detection rate with reliable performance. This reliability is often affected by class imbalances and inadequate hyperparameter optimization. This article addresses the issue of improving the detection rate of IDS by evaluating and comparing three machine learning algorithms: Random Forest (RF), XGBoost, and Deep Neural Networks (DNN), using the NSL-KDD dataset. In our methodology, we integrate SMOTE (Synthetic Minority Oversampling Technique) to tackle the unbalanced nature of the data, ensuring a more balanced representation of the different classes. This approach helps optimize model performance, reduce bias, and enhance robustness. Additionally, hyperparameter optimization is performed using Optuna, ensuring that each algorithm operates at its optimal level. The results show that our model, using the Random Forest algorithm, achieves an accuracy of 99.80%, surpassing the performance of XGBoost and Deep Neural Networks (DNN). This makes our approach a true asset for intrusion detection methods in computer networks.

1. Introduction

Intrusion represents a major threat in the field of cybersecurity, constituting a crucial challenge for the protection of computer systems. Indeed, a single malicious breach can compromise sensitive data, leading to its theft or destruction, and threatening critical infrastructures in a few seconds (Ahmad et al., 2018). In this context, the effectiveness of Intrusion Detection Systems (IDS) is essential to ensure the security of computer networks, as they play a crucial role in identifying cyber-attacks and protecting sensitive data. According to Catania and Garino (2012), IDS must be able to detect intrusions effectively and quickly to mitigate the risks associated with cyber threats.

However, these systems face many challenges, including the need to maintain a high detection rate while ensuring reliable performance. These aspects, particularly detection accuracy and performance reliability, represent fundamental issues. Indeed, a problem related to effectiveness in data classification lies in class imbalance, especially when one class is markedly less represented than the others, such as in the case of fraud detection (Chan et al., 1999) or medical diagnosis (Srinivas & Katarya, 2022). Attacks often represent a small portion of the recorded events in security applications, which explains the imbalance in the datasets. Thus, Raju et al. (2024), in their study, highlighted the issue of class imbalance in datasets, in the context of facial and vocal emotion recognition. This imbalance can make it difficult to learn the minority classes, leading to poor model performance that favors the majority classes. In a similar intrusion detection context, this imbalance complicates the learning of models, which tend to be biased toward the majority classes, potentially reducing their effectiveness in detecting anomalies. This limitation becomes problematic in security contexts where it is essential to detect the rare intrusion incidents.

Moreover, hyperparameter optimization is a key element in improving the performance of classification models, as it helps balance the complexity of the model and its ability to generalize. This process reduces overfitting and ensures more reliable predictions on new data. As Tran et al. (2020) highlight, the importance of an adequate optimization approach to maximize IDS performance while minimizing costs and the risks of overfitting, emphasizing the strategic impact that precise and appropriate hyperparameter tuning can have on the effectiveness of intrusion detection models. Srinivas and Katarya (2022) also demonstrate the importance of hyperparameter optimization to improve the effectiveness of the detection model, here applied to predicting heart disease, but with similar implications for IDS.

To address these challenges, this study evaluates three machine learning algorithms: Random Forest (RF), XGBoost, and Deep Neural Networks (DNN) on the NSL-KDD dataset. Indeed, these algorithms have achieved good results in previous studies, such as the one by Azam et al. (2023), which explored various algorithms and the challenges

* Corresponding author.
E-mail addresses: thiernohamidou.sow@uqar.ca (S.T. Hamidou), mehdi_adda@uqar.ca (A. Mehdi).

https://doi.org/10.1016/j.mlwa.2025.100738
Received 20 May 2025; Received in revised form 6 September 2025; Accepted 16 September 2025
Available online 27 September 2025

related to big data in order to overcome the limitations of traditional IDS in the face of current cyber threats. The authors reported a detection accuracy of 99.60% achieved by XGBoost, using a feature selection strategy based on a discernibility function. Similarly, the study by Pranto et al. (2022) achieved an accuracy of 99.50% with Random Forest on the KDD'99 dataset by applying a dimensionality reduction and hyperparameter optimization approach. Their approach included Min-Max data normalization as well as feature selection based on variance and correlation. The hyperparameters were optimized using cross-validation and grid search. Moreover, the DNN method was used in the study by Mohammed and Gbashi (2021) on the NSL-KDD dataset, achieving an accuracy of 94%. Their approach included Min-Max normalization, categorical attribute encoding, and feature selection through Recursive Feature Elimination (RFE) with Random Forest, along with dropout regularization applied to reduce overfitting. Another study by Kikissagbe et al. (2024) achieved an accuracy of 99.62% in detecting Denial of Service (DoS) attacks on the Edge IIoT dataset. This performance was achieved through six combinations of techniques, including data balancing with Synthetic Minority Oversampling Technique (SMOTE) and Random Under-Sampling (RUS), as well as feature selection using RF, DNN, PCA (Principal Component Analysis), and classifiers such as SVM (Support Vector Machine), RF, XGBoost, and DNN.

Therefore, we hypothesize that by using these algorithms, we will be able to improve the performance of IDS, making them more precise and robust. We have integrated the SMOTE technique, which generates synthetic samples for the minority class, thereby balancing the data and improving the model's ability to detect rare attacks. This also reduces training time and enhances the overall model performance by avoiding biases related to class imbalance (Jiang et al., 2020).

Indeed, for a reliable performance evaluation, we used stratified K-Fold cross-validation, which divides the data into several folds while preserving the class distribution. This approach allows the model to be trained and tested on different subsets, ensuring accurate evaluations by minimizing biases caused by unbalanced data splits. Mahesh et al. (2023). We also used Optuna for hyperparameter optimization, a method that dynamically defines the search space, allowing flexibility in adjusting parameters during trials. Unlike grid or random search, Optuna is faster, less computationally expensive, and suitable for complex models such as neural networks, thus improving the accuracy and efficiency of prediction models (Hanifi et al., 2024).

Our results demonstrate that the Random Forest algorithm achieves an accuracy of 99.80%, which indicates that our approach provides an effective alternative to the different approaches we have explored. This improves the robustness of IDS to address the growing challenges of cybersecurity.

This work makes the following contributions:

- We demonstrate that optimal hyperparameter tuning via Optuna, applied to IDS models on the NSL-KDD dataset, achieves detection accuracy surpassing the best reported values in the literature, positioning our approach as a reference for high precision in the IDS context.
- We propose a reproducible experimental protocol that uses identical hyperparameters to fairly compare configurations with and without SMOTE, for both grid search and Optuna optimization. This ensures that observed benefits stem directly from the applied techniques, providing a reliable evaluation rarely addressed with such precision in existing works.
- Our analysis shows that even on a relatively balanced dataset like NSL-KDD, SMOTE provides a modest but essential improvement for detecting minority classes. This improvement is crucial because training a model without addressing class imbalance can bias its performance, especially in the IDS context. As noted by Chawla et al. (2002), imbalanced datasets pose a significant challenge: most examples belong to the majority class, while minority class examples are rare, and misclassifying minority examples often incurs a higher cost. SMOTE mitigates this issue by generating synthetic minority samples, thereby improving model sensitivity to rare cases and enhancing overall reliability.
- Our detailed comparison of three models, considering both classification performance and computational costs, offers a clear and practical framework to select IDS models adapted to operational constraints. This guide aids practitioners in making informed decisions balancing efficiency and performance.

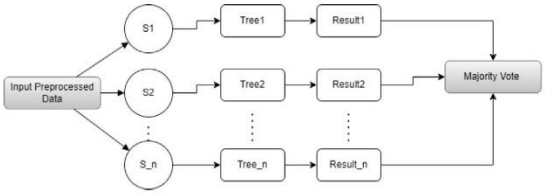

Fig. 1. Architecture of the RF.

Nevertheless, this study has limitations. The NSL-KDD dataset is widely used as a benchmark in IDS research, it presents certain limitations. Its relatively old design does not reflect the evolution of current cyber threats or the increasing complexity of real-world network traffic. This obsolescence may limit the generalizability of the results obtained to more recent production environments.

The structure of the paper is as follows: Section 2 presents the context of the topic, Section 3 addresses the review of the literature, Section 4 describes the proposed approach, and Section 5 details the experiments conducted. The results are analyzed in Section 6, ollowed by a discussion and future work in 7, while the conclusion is addressed in Section 8.

2. Background

The importance of IDS in the field of cybersecurity is paramount, especially as cyberattacks grow increasingly sophisticated and targeted. IDS play a crucial role in detecting various types of network threats, such as DoS attacks, port scans, malware, and advanced persistent threats (APT).

These systems are designed to continuously monitor data flows and identify malicious behaviors or anomalies that may indicate potential intrusions. Traditionally, IDS relied on predefined rules and signatures to detect known attacks (Khraisat et al., 2019). However, these approaches show their limitations in the face of emerging threats. This has led to the integration of machine learning techniques, which allow IDS to learn and adapt to new forms of attacks.

In this study, the NSL-KDD dataset used simulates real attacks, such as DoS, Remote to Local (R2L), Probe, and User to Root (U2R). It provides a comprehensive framework to evaluate the models' ability to accurately distinguish between normal and malicious behaviors. Each of the algorithms we applied has distinct characteristics that make them particularly effective for classification.

2.1. Random Forest

Random Forest is an ensemble machine learning algorithm used for classification and regression tasks. Its operation is based on the generation of multiple decision trees during the learning phase, with each tree contributing to the final prediction (Ahmad et al., 2018). In the context of classification, the final prediction is obtained through a majority vote of the predictions made by each decision tree in the forest.

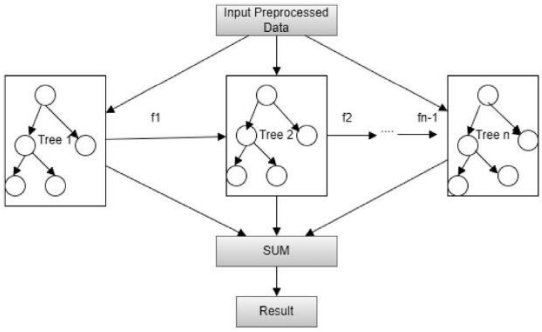

Fig. 2. Architecture of the XGBoost.

Fig. 1 presents the architecture of the Random Forest model used in an IDS. It describes the process in several steps:

The input data (samples) are preprocessed before being used in the model. Next, this data is divided into several subsets $S_1, S_2, \ldots, S_n$, each intended for a distinct decision tree. From these subsets, unique decision trees $Tree_1, Tree_2, \ldots, Tree_n$ are created, with each performing classification independently. The classification results obtained by each tree $Result_1, Result_2, \ldots, Result_n$ are then generated. Finally, to obtain a final decision, the results from the various trees are combined through a majority vote, allowing the dominant class to be selected.

2.2. XGBoost

XGBoost is a learning algorithm used for classification and regression tasks. It relies on the sequential construction of decision trees, where each tree improves the predictions of the previous one using gradient boosting. With XGBoost, the final prediction is obtained by summing the predictions of each individual tree in the model. Mathematically, this can be expressed as follows:

$$\hat{y} = \sum_{k=1}^{K} f_k(x)$$

where:

- $\hat{y}$ is the final prediction,
- K represents the number of trees,
- f_k is the prediction of tree k for the input x,
- x denotes the input features of the instance.

Each tree f_k contributes an additional prediction, and the overall result is the sum of all the predictions from the trees. This model was chosen for this study due to its ability to achieve high accuracy, which has already been demonstrated in previous works on intrusion detection (Dhaliwal et al., 2018).

Fig. 2 illustrates the architecture of the XGBoost model used in an IDS. It details the process in several steps:

The input data is first preprocessed before being used in the model. Unlike Random Forest, the decision trees in XGBoost are created sequentially. Each tree $Tree_1, Tree_2, \ldots, Tree_n$ is constructed by taking into account the errors of the previous trees. Each new tree adjusts its predictions to correct the errors of the earlier trees. At each step, a function $f_1, f_2, \ldots, f_{n-1}$ is applied to improve the accuracy of the model. The intermediate results provided by each tree are then summed to obtain a final score.

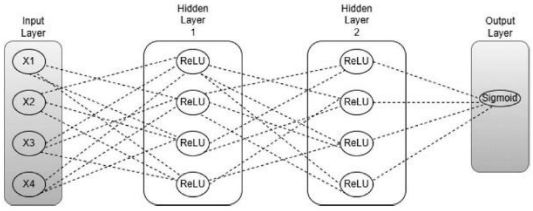

Fig. 3. Architecture of a DNN.

2.3. Deep Neural Network (DNN)

A Deep Neural Network (DNN) is a type of artificial neural network, structured in a sequence of layers called Multilayer Perceptrons (MLP). Each layer refines the representations acquired from the previous one, allowing the model to capture complex relationships in the data (Al-Maksousy et al., 2018). The output of a deep neural network (DNN) is evaluated based on the weights and inputs of a neuron using the following function:

$$y = \delta\left(\sum_{n=1}^{N} W_n x_n + b\right) = \delta(W^T X + b)$$

where:

- W is the weight vector.
- X is the input vector.
- b denotes the bias.
- δ represents the activation function (often a sigmoid or ReLU function).
- $W^T X$ is the dot product between the weight vector W and the input vector X. This product yields a value representing the weighted sum of the inputs.

Thus, the output of a DNN at each neuron is a weighted linear combination of the inputs, modified by a non-linear activation function to enhance the network's ability to capture relationships. This model was chosen for its ability to achieve high accuracy, as demonstrated by numerous studies.

Fig. 3 illustrates the architecture of a deep neural network (DNN) designed for an IDS. The process unfolds in several steps:

The input features $X_1, X_2, \ldots, X_n$ are fed into the first hidden layer. In this layer, each neuron, which is connected to all the input neurons, applies the ReLU activation function to learn complex relationships. The outputs of this first layer are then passed to a second hidden layer, also composed of neurons using the ReLU function, which refines the acquired representations.

Finally, the data flows through an output layer, consisting of a single neuron utilizing a sigmoid activation function, intended for binary classification.

The performance of these models has been evaluated using several key indicators: accuracy, precision, recall, F1 score, and the ROC curve.

Accuracy represents the total percentage of correctly classified instances and is calculated as follows:

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

Precision, on the other hand, focuses on positive results and is calculated by the following ratio:

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

Recall is determined by dividing the number of true positives by the sum of true positives and false negatives:

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

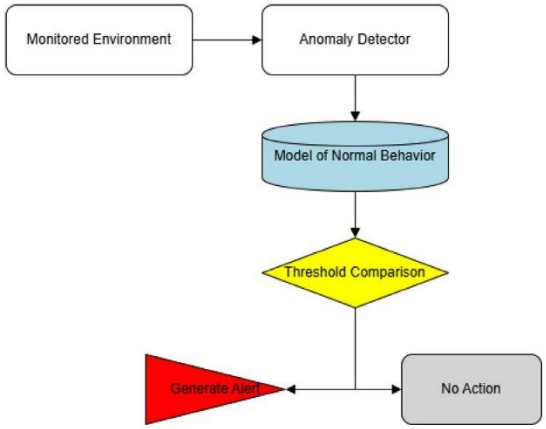

Fig. 4. IDS anomaly detection process.

The F1 score is defined by the formula:

$$\text{F1 Score} = \frac{2 \times TP}{2 \times TP + FN + FP}$$

A high F1 score indicates a good balance between recall and precision. Finally, the ROC curve represents the true positive rate against the false positive rate (FPR).

The results of this research aim to improve IDS by ensuring continuous monitoring and rapid response to threats. Fig. 4 illustrates the intrusion detection process through behavioral analysis, where suspicious actions are compared to a model of normal behavior to detect anomalies (Aldallal & Alisa, 2021).

3. Literature review

The necessity for efficient IDS has led to the integration of machine learning techniques to optimize detection accuracy in complex network environments. Numerous studies have proposed frameworks using various algorithms to enhance IDS performance. In this section, we will examine the different approaches developed for intrusion detection.

Ahmad et al. (2018) compared SVM, Random Forest, and Extreme Learning Machine (ELM) on the NSL-KDD dataset, revealing that ELM offered the best accuracy 99.67%, particularly for large datasets. SVM with an Radial Basis Function (RBF) kernel showed better performance on reduced subsets. However, the study is limited to metrics such as accuracy and F1-score without including the AUC-ROC, which is essential for evaluating the ability to differentiate classes in imbalanced data contexts, which would have allowed a more comprehensive assessment of the models for intrusion detection.

Similarly, Azam et al. (2023) evaluated the effectiveness of various machine learning algorithms applied to the KDD Cup 99 and NSL-KDD datasets. Their study focuses on methods such as C4.5, Naive Bayes, Multi-Layer Perceptron (MLP), NB-Tree, and Decision Tree for detecting attacks, including DoS and Distributed Denial of Service (DDoS). The Decision Tree, combined with feature selection via RFE, achieved an accuracy of 99%, while the XGBoost algorithm, optimized in a parallel computing environment, recorded a detection rate of 99.60%. Although the results are promising, the authors highlight that traditional machine learning methods have limitations compared to deep learning techniques, which are more suited for modeling complex data.

Moreover, Pranto et al. (2022) proposed a method using classifiers such as k-NN, Decision Tree, Naive Bayes, Logistic Regression, and Random Forest. Their approach, applied to the NSL-KDD dataset, showed that Random Forest with 1400 trees achieved an accuracy of 99.5% and an alarm rate of 0.6%. However, although performance metrics like the ROC curve and AUC were included, the study did not address potential risks of overfitting, especially with such a high number of trees. Stricter regularization or reducing the number of trees could have improved the model's generalization.

Similarly, Mohammed and Gbashi (2021) proposed a method combining (DNN) and Recurrent Neural Networks (RNN) with RFE. Their DNN model achieved 94% accuracy in binary classification, while the RNN model, used to classify five categories (Normal, DoS, Probe, R2L, U2R), demonstrated strong results, with 96% accuracy for the DoS class and 94% for U2R. However, although preprocessing and normalization techniques were applied, their method does not address the imbalance among attack classes, which may limit the model's generalization for less frequent attack types.

Kikissagbe et al. (2024) proposed an approach aimed at improving the DoS attacks in IoT systems based on the use of the Edge IIoT dataset. Their method integrates class balancing techniques, including SMOTE and RUS, to address class imbalance issues. At the same time, they applied feature selection techniques such as DNN, RF, and PCA and tested four classifiers (SVM, DNN, XGBoost, RF) across six combinations of techniques, where the SMOTE + DNN combination achieved the best results with an accuracy of 0.9962. However, the study is limited to DoS attacks. Other types of threats in IoT systems, such as adversarial attacks and Man-in-the-Middle attacks, could be further explored.

Similarly, Meena and Choudhary (2017) proposed an intrusion detection framework using the J48 Graft decision tree, validated on the NSL-KDD datasets. Their study highlighted J48's effectiveness in detecting attacks like DoS and SYN Flood, achieving 99.435% accuracy, outperforming Naive Bayes. However, the study did not address class imbalance in the dataset, where normal connections are predominant and some attacks are underrepresented, potentially skewing results in favor of majority classes.

In parallel, Chandra et al. (2017) used the MRFA (Modified Random Forest Algorithm), an enhanced version of Random Forest, for intrusion detection on the KDD-Cup 99 and NSL-KDD datasets. They achieved an accuracy of 97.9% on KDD-Cup 99 and 99.25% on NSL-KDD. Although the MRFA algorithm yielded promising results, the study did not address certain limitations, such as the computation time compared to simpler models, which could pose a problem for real-time applications. Additionally, the impact of class imbalance was not considered, a crucial factor that can affect the model's robustness in real-world scenarios where intrusions are generally rare.

Zhang et al. (2021) developed a deep learning model using a stacked autoencoder (SAE) for dimensionality reduction and a Random Forest classifier for intrusion detection, achieving 99.92% accuracy in binary classification and 99.90% in multiclass classification on the CICIDS2017 dataset. This approach effectively handles high-dimensional data. However, the study does not analyze the computational cost, which could limit its applicability for real-time intrusion detection. Additionally, while min–max normalization improves scaling, it does not necessarily ensure the model's effectiveness in the presence of outliers, which can degrade the model's performance.

Similarly, Xu et al. (2021) proposed a model using a five-layer autoencoder (AE) to detect network anomalies, aiming to enhance cybersecurity against cyberattacks. Their method includes preprocessing to remove outliers, thus reducing data bias. This model achieved 90.61% accuracy and an F1 score of 92.26% on the NSL-KDD dataset. However, in intrusion detection, outliers may correspond to abnormal or unusual behavior and should therefore be handled carefully, as they may represent either legitimate anomalies or errors that require specific handling.

Rawat et al. (2022) evaluated an IDS based on classical techniques and a DNN on the NSL-KDD dataset. Their method, combining PCA and a five-layer DNN, achieved an accuracy of 79.3% with all features, and 75.9% when limited to six features specific to SDN networks. However, although the model displayed good performance on the training set,

its accuracy decreased on the test data. This phenomenon limits the model's ability to generalize to other intrusion datasets. Moreover, using all the features did not considerably improve the results, suggesting that a more relevant feature selection could strengthen the model.

Similarly, Kasongo (2023) also used RNN for IDS, noting that RNN outperformed Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) methods. By applying these approaches to the UNSW-NB15 and NSL-KDD datasets, promising results were obtained, particularly with attacks such as Exploits, Fuzzers, DoS, Reconnaissance, Backdoor, and Shellcode. The LSTM-XGBoost and GRU-XGBoost combinations achieved accuracies of 85.93% and 85.65%, respectively. However, while RNN can capture temporal dependencies, they require intensive computations. Additionally, the study does not consider class imbalance, which could affect detection performance for some less frequent attack classes.

Moreover, Jia et al. (2019) proposed an intrusion detection method based on a DNN applied to the KDD99 and NSL-KDD datasets, targeting DoS, R2L, U2R, and Probe attacks. Their approach achieved an accuracy of over 98%, with an F1 score reaching 98.84%, highlighting the effectiveness of deep learning techniques for intrusion detection in complex network environments. However, this method may face challenges when dealing with imbalanced data or rare attacks, which could affect its overall performance.

Gurung et al. (2019) proposed a IDS based on a deep learning model, combining a sparse autoencoder and logistic regression, validated on the NSL-KDD dataset and achieving an overall accuracy of 87.2%. They argue that this approach enhances the ability to distinguish normal traffic from malicious traffic. However, the model uses the 115 features derived from one-hot encoding all categorical variables in the dataset, without any prior feature selection process. While this allows for a comprehensive representation of the data, the lack of relevant feature selection may introduce unnecessary noise, which could harm the model's accuracy due to the inclusion of superfluous variables.

Additionally, Khan and Mailewa (2023) proposed a hybrid method combining PCA with SVM and Deep Autoencoders (DAE), achieving a 95% accuracy for DoS attacks, thereby improving the effectiveness of combined approaches. However, this method presents an important limitation, namely the model achieves a low detection rate for U2R and L2R attacks, as these attacks are less represented in the training data, which is due to a lack of class balancing in the data.

4. Methodology

The approach that we have proposed for intrusion detection is structured in six key steps, each playing an essential role in the classification process and the evaluation of the performance of the models. The proposed model is illustrated in Fig. 5. The methodology steps are as follows: Section 4.1 presents the data collection process, while Section 4.2 addresses the data preprocessing steps, including handling categorical variables, feature selection, and data normalization, followed by Section 4.3, which deals with the stratified cross-validation used for splitting the data. The class balancing using SMOTE is addressed in Section 4.4, followed by the classification process in Section 4.5. Finally, the model's performance evaluation is detailed in Section 4.6.

4.1. Data collection

In the current literature, the NSL-KDD dataset (Gurung et al., 2019; Mohammed & Gbashi, 2021; Xu et al., 2021) is widely used as a benchmark standard for evaluating IDS, particularly in experimental studies involving various machine learning techniques. For this reason, we selected this dataset to conduct our own evaluations.

NSL-KDD is an improved version of the KDD Cup 99 dataset (Tavallaee et al., 2009), from which redundant records have been removed. It consists of simulated data representing computer network connections.

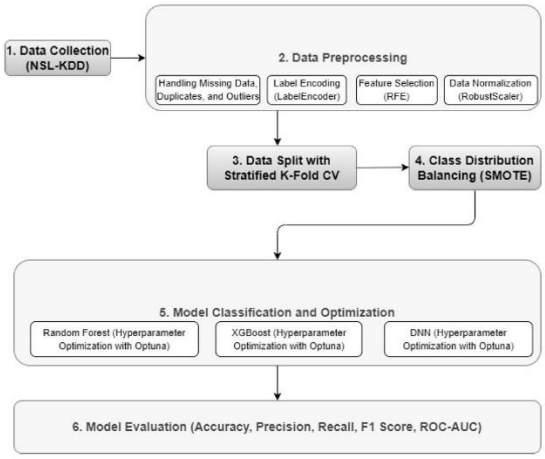

Fig. 5. Proposed approach for an intrusion detection system.

The dataset contains a total of 25,192 instances, described by 42 features. These features include protocol type, connection duration, network service type, and the number of failed connections, among which three are categorical in nature.

In addition, a target variable is provided to indicate whether a connection is considered normal or anomalous. Attacks are categorized into four types: DoS, Probe, R2L, and U2R, all grouped under a single anomalous class. This is illustrated in the first step: Data Collection (NSL-KDD) in Fig. 5.

4.2. Data preprocessing

In this study, we applied various preprocessing techniques to prepare the data for the training of models. This included addressing missing values, removing duplicates, handling outliers, and managing categorical variables. We also performed feature selection to ensure that only the most relevant features were used during model training. These steps are essential for enhancing the performance of machine learning models by reducing noise and optimizing the data for improved generalization. This process is represented in step 2: Data Preprocessing in Fig. 5.

4.2.1. Management of categorical variables

The classifier cannot efficiently process raw data due to certain qualitative variables. Thus, we applied the LabelEncoder to transform these categorical values into numerical representations. This approach allows us to retain categorical information while ensuring that machine learning algorithms can effectively process the data. This is illustrated in step 2: Preprocessing Label Encoding (LabelEncoder) in Fig. 5.

4.2.2. Feature selection

Feature selection is a crucial step in preprocessing as it helps identify and retain the most relevant features. This reduces model complexity while improving its accuracy and generalization ability. The feature selection method used in this research is RFE. RFE is a selection technique that iteratively eliminates the least relevant features while retaining those that contribute the most to the model's performance (Urmi et al., 2024). This approach is particularly effective for high-dimensional datasets as it reduces overfitting while enhancing the model's interpretability. This is represented in step 2: Preprocessing Feature Selection (RFE) in Fig. 5.

4.2.3. Data normalization

After selecting the relevant features, it is necessary to scale the data, an essential preprocessing step that normalizes the data to a specific range. This optimizes the calculation speed of the algorithms. We chose the RobustScaler because it is less sensitive to outliers compared to other scaling methods, making it suitable when the dataset contains outliers. The RobustScaler scales the data by removing the median and adjusting the values based on the interquartile range (IQR). This method is robust to outliers because it relies on statistics that are less sensitive to them (Rashmi & Shantala, 2024). This is represented in step 2: Data Normalization (RobustScaler) in Fig. 5.

4.3. Data splitting with stratified cross-validation

In our approach, we implemented K-fold cross-validation to split the dataset into k subsets of nearly equal sizes. In each iteration, one of the k subsets serves as the test set, while the others form the training set. The stratified version ensures a class distribution in each subset that reflects that of the complete dataset, which is particularly beneficial in cases of class imbalance (Mahesh et al., 2023). This is represented in step 3: Data Split with Stratified K-Fold CV in Fig. 5.

4.4. Class balancing with SMOTE

To correct class imbalances, we applied the SMOTE technique, which generates synthetic examples for the minority class. This improves model performance by preventing it from being biased toward the majority class, making the model more robust, especially when certain classes are under-represented (Jiang et al., 2020). This is represented in step 4: *Class Distribution Balancing (SMOTE)* in Fig. 5.

4.5. Classification

This step focuses on training classification models with hyperparameter optimization. We compared the performance of classification algorithms such as Random Forest, XGBoost, and DNN, using Optuna to find the best hyperparameters. Optuna relies on an advanced technique, the Tree-structured Parzen Estimator (TPE), which focuses on the most promising hyperparameter combinations while discarding less relevant trials (Hanifi et al., 2024). This approach allows for a more efficient exploration of hyperparameter space regions where the likelihood of improving model performance is higher, thus reducing the time and resources required for optimization. The choice of hyperparameter values depends on the complexity of the data, the need to prevent overfitting, and the available computing resources. These models are represented in step 5: *Model Classification and Optimization* in Fig. 5.

4.6. Model evaluation

During model building, each model was trained k times, as we divided our data into k folds using cross-validation. Thus, performance is evaluated at each iteration, and an average is calculated to provide an overall performance assessment of the model. The models were evaluated using several performance metrics such as accuracy, precision, recall, F1-Score, and ROC-AUC curve. These metrics allow for comparison of model performance and identification of which is the most effective. This process is represented in the final step: *Model Evaluation* in Fig. 5.

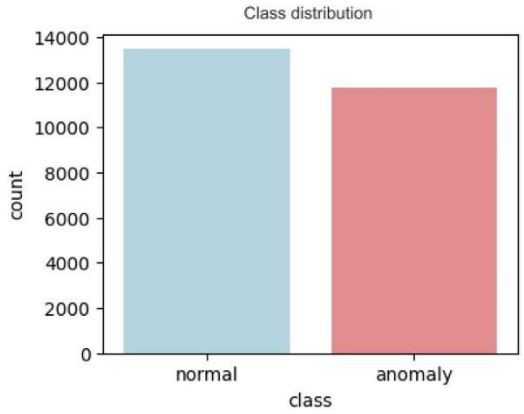

Fig. 6. Class distribution.

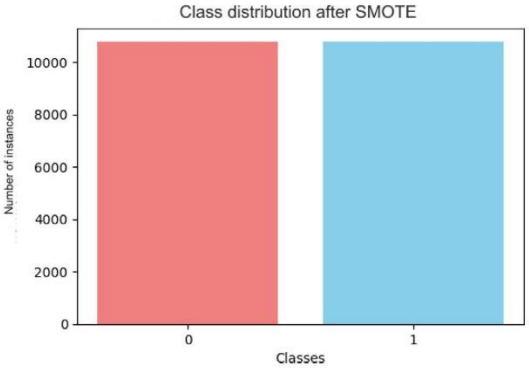

Fig. 7. Class distribution after SMOTE.

5. Experiments

To implement and evaluate the models, we used Python with the Scikit-learn and Keras libraries. Before training the models, the dataset was preprocessed, which reduced the data to 20 columns with 20,153 instances for training and 5039 for testing. The dataset contains no missing values or duplicates; however, some columns contain outlier values. The class distribution before and after the application of SMOTE is presented in Figs. 6 and 7.

For each model, the hyperparameters were optimized using Optuna, which employs the TPE method to find optimal combinations. After optimization, the best parameters obtained are as follows:

- **Random Forest**: *n_estimators = 85, max_depth = 23, min_samples_split = 5, min_samples_leaf = 1, max_features = 0.5866, bootstrap = False*
- **XGBoost**: *n_estimators = 140, max_depth = 10, learning_rate = 0.1184, gamma = 0.0041, colsample_bytree = 0.3235, subsample = 0.8982, reg_alpha = 0.00042, reg_lambda = 1.5426e−08*
- **DNN**: *n_layers = 2, n_units = 128, dropout_rate = 0.3303, activation = tanh, learning_rate = 0.00071*

To provide further insight into the model's decision-making process, we performed a detailed analysis of the features selected using RFE. This technique enabled us to identify the most influential features based on the best-performing model, Random Forest. Table 1 presents the

Table 1
Selected features by RFE with their importance scores (Random Forest).

Feature	Importance score
src_bytes	0.440841
dst_bytes	0.175580
flag	0.077083
same_srv_rate	0.055170
protocol_type	0.033939
dst_host_srv_count	0.031876
diff_srv_rate	0.031439
count	0.022363
dst_host_same_src_port_rate	0.020331
srv_count	0.018410
service	0.018236
hot	0.016697
logged_in	0.012228
dst_host_same_srv_rate	0.012056
dst_host_srv_diff_host_rate	0.009594
dst_host_diff_srv_rate	0.008571
dst_host_rerror_rate	0.005163
dst_host_count	0.004471
dst_host_serror_rate	0.003475
dst_host_srv_serror_rate	0.002477

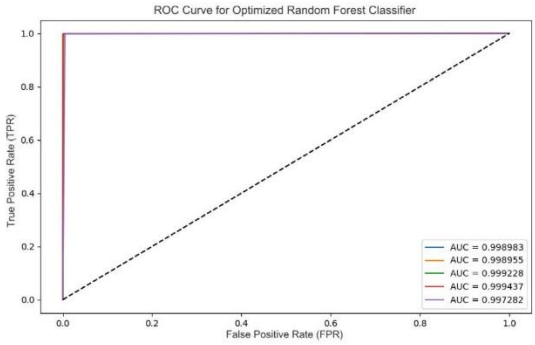

Fig. 8. ROC curve for Random Forest classifier.

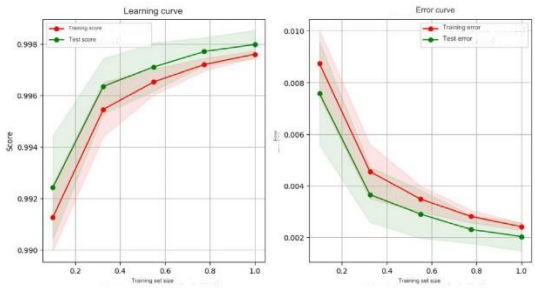

Fig. 9. Learning curve and error curve of the Random Forest classifier.

selected features along with their corresponding importance scores. This information can guide future feature engineering efforts in IDS.

6. Results and interpretation

The performances of the optimized models were evaluated on the test set (5039 instances). The metrics used include accuracy, precision, recall, F1-score, and the ROC-AUC curve. The experimental results are detailed in this section where we compare the models based on these metrics.

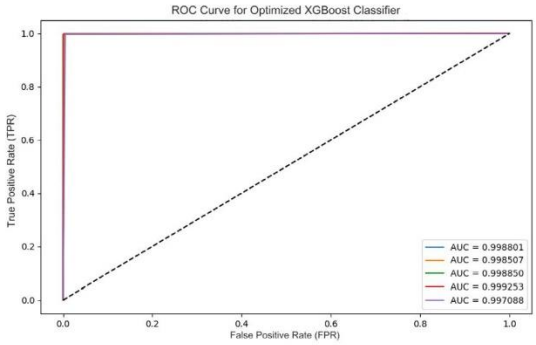

Fig. 10. ROC curve for XGBoost.

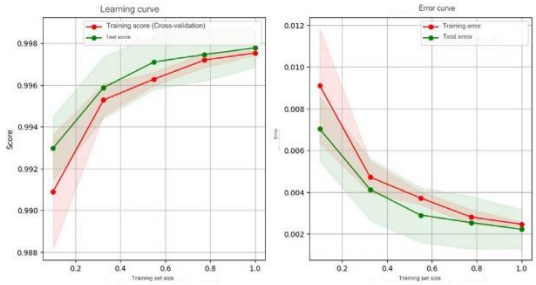

Fig. 11. Learning curve and error curve of the XGBoost classifier.

The Random Forest classifier achieved outstanding results, with an accuracy of 99.80%. Fig. 8 presents the ROC curve of the RF model, highlighting its ability to effectively distinguish between classes, with an average AUC of 0.9988 ± 0.0008, indicating high predictability. Regarding the learning curve, shown in Fig. 9, it demonstrates that the training and testing results converge with a consistent decrease in errors. This suggests that the model improves with more training data and reduces the gap between training and testing errors, indicating good generalization without overfitting.

The performance of the XGBoost classifier was also remarkable, with an accuracy of 99.79% and an average AUC of 0.9985 ± 0.0007, as shown in Fig. 10, indicating a high predictive capability. In Fig. 11, continuous improvement in performance can be observed as the data evolves, with decreasing errors, demonstrating good generalization of the model.

The accuracy of the DNN classifier reached 98.66%, reflecting a high level of precision. According to Fig. 12, the ROC curve of the DNN model shows an average AUC of 0.9872 ± 0.0007, which is slightly lower than those of Random Forest and XGBoost, but still performs well. The learning curve, represented in Fig. 13, illustrates effective generalization of the model, while the error curve reflects a continuous reduction in errors, indicating effective learning.

The comparison of ROC curves displayed in Fig. 14 shows that Random Forest and XGBoost offer very similar performances, with Random Forest slightly ahead, while the DNN model falls slightly below.

Table 2 shows that Random Forest and XGBoost display almost equivalent performances, with a slight advantage for Random Forest. Although the DNN shows good performance, it is slightly behind. The goal is to achieve the best possible performance, making Random Forest the preferred choice in this case, closely followed by XGBoost. The DNN may require additional adjustments to reach a comparable level of performance.

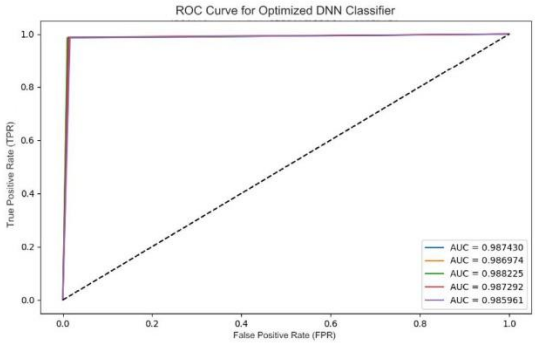

Fig. 12. ROC curve for DNN.

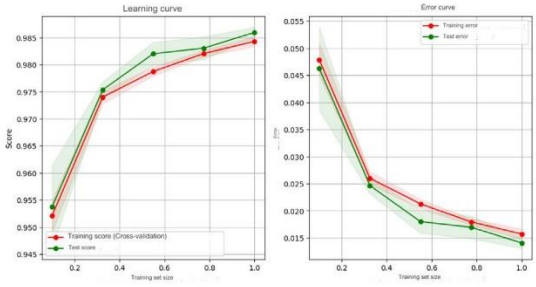

Fig. 13. Learning curve and error curve of the DNN classifier.

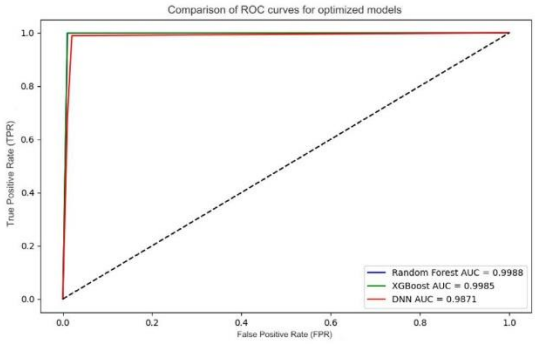

Fig. 14. Comparison of ROC curves.

Table 2
Comparison of model performance metrics.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.9980	0.9973	0.9989	0.9981
XGBoost	0.9979	0.9971	0.9989	0.9980
DNN	0.9865	0.9878	0.9868	0.9873

Table 3
Comparison of model performance using the NSL-KDD dataset.

Ref.	Techniques	Accuracy
Ahmad et al. (2018)	SVM, RF, ELM	99.67%
Azam et al. (2023)	DT, XGB	99.60%
Pranto et al. (2022)	RF, k-NN, NB, DT	99.50%
Mohammed and Gbashi (2021)	DNN, RNN	94.00%
Meena and Choudhary (2017)	J48, NB	99.43%
Chandra et al. (2017)	MRFA	99.25%
Xu et al. (2021)	AE, RF	90.61%
Rawat et al. (2022)	PCA, DNN	79.30%
Kasongo (2023)	LSTM, GRU	85.93%
Jia et al. (2019)	NDNN	98%
Gurung et al. (2019)	Sparse AE	87.20%
Khan and Mailewa (2023)	PCA, DAE, SVM	95.00%
Our approach	RF, XGB, DNN	**99.80%**

Table 4
Comparison of RF performance based on optimization method and use of SMOTE.

Method	Accuracy	Precision	Recall	F1-score
GridSearch No SMOTE	0.9945	0.9927	0.9970	0.9948
GridSearch + SMOTE	0.9946	0.9928	0.9972	0.9950
Optuna No SMOTE	0.9977	0.9967	0.9987	0.9977
Optuna + SMOTE	0.9980	0.9973	0.9989	0.9981

Table 3 presents a comparison of intrusion detection techniques applied to the NSL-KDD dataset and shows the accuracies achieved by each approach. Our method, based on Random Forest (RF), stands out with an accuracy of 99.80%, surpassing all other techniques.

6.1. Ablative experiments: Respective impact of SMOTE and Optuna

To assess the individual effects of SMOTE and Optuna on our final model, we performed ablative experiments. The results comparing their impacts are shown in Table 4.

We conducted experiments using the same hyperparameters for configurations with and without SMOTE, both for GridSearch and Optuna optimization, to ensure a reliable comparison.

Impact of SMOTE:

The addition of SMOTE results in a slight improvement in recall, increasing from 0.9970 to 0.9972 with GridSearch, and from 0.9987 to 0.9989 with Optuna. Similarly, the F1-score shows a minor increase, from 0.9948 to 0.9950 with GridSearch, and from 0.9977 to 0.9981 with Optuna. This modest improvement is mainly due to the NSL-KDD dataset being relatively balanced, which limits the overall effect of SMOTE on performance.

Although these improvements in recall and F1-score appear subtle, they have a crucial impact in the context of IDS, as even a marginal increase in recall can lead to correct detection and more attacks, thereby reducing false negatives that could critically compromise network security. In effect, SMOTE enhances the model's ability to detect underrepresented minority attack classes, directly improving detection reliability and operational performance.

Thus, these findings highlight the important role of SMOTE in enhancing IDS effectiveness, representing incremental yet essential gains in attack detection.

Impact of Optuna compared to GridSearch:

When comparing models without SMOTE, Optuna optimization notably improves overall performance, with accuracy rising from 99.45% to 99.77%, and precision increasing from 0.9927 to 0.9967. These improvements are also observed in models using SMOTE, where Optuna

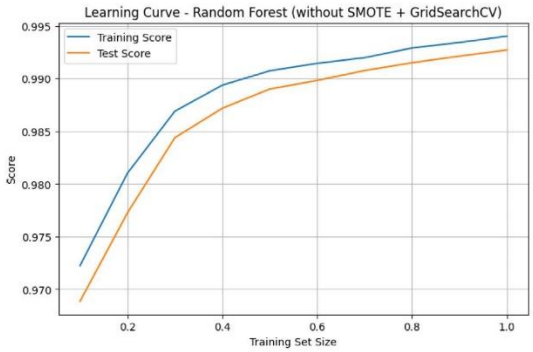

Fig. 15. Learning curve of the Random Forest model without SMOTE + Grid search.

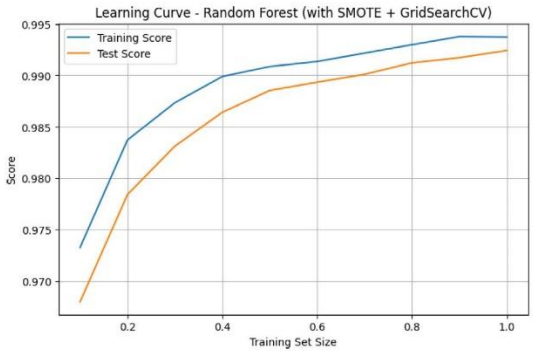

Fig. 16. Learning curve of the Random Forest model with SMOTE and Grid search.

achieves the best performance in terms of accuracy (from 99.46% to 99.80%) and precision (from 0.9928 to 0.9973).

While SMOTE specifically strengthens the detection of minority classes, as evidenced by the improved recall and F1-score, these results clearly demonstrate that Optuna enables more effective hyperparameter tuning than GridSearch, substantially enhancing the model's predictive capability. The gains in precision translate into more reliable network traffic classification. In the context of an IDS, greater precision reduces false alarms, improves the detection of actual attacks, and directly reinforces the overall robustness and effectiveness of the security system (see Figs. 15–17).

6.2. Computational cost analysis of the evaluated models

This subsection presents a comparative analysis of the training, inference, and optimization times required by each evaluated model. These three stages respectively reflect how long it takes to fit the model, make predictions, and perform hyperparameter tuning. Table 5 provides an overview that helps assess the computational cost of each model and highlights the trade-offs between predictive accuracy and processing efficiency.

As shown in Table 5, the trade-offs between performance and computational efficiency among the three evaluated models are highlighted. XGBoost stands out for its very fast training and inference

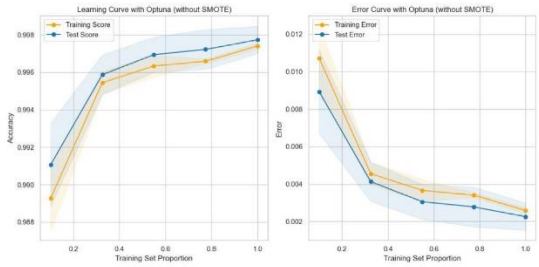

Fig. 17. Learning curve of the RF model with Optuna (without SMOTE).

Table 5
Comparison of training, inference, and hyperparameter optimization times (in seconds) for each model.

Model	Train (s)	Inference (s)	Optimization (s)
RF	21.37	0.0297	683.32
XGBoost	3.04	0.0163	541.94
DNN	27.75	0.2274	512.46

times, making it particularly suitable for scenarios requiring rapid deployment or real-time detection. The DNN, although slower especially during inference benefits from faster hyperparameter optimization compared to the RF.

RF offers the highest accuracy, but at the cost of longer optimization time due to the complexity of its ensemble architecture and hyperparameter tuning.

Thus, RF ensures optimal detection performance, while XGBoost provides a compelling alternative for applications where computational resources or latency are major constraints. The DNN may require further tuning to match the effectiveness of tree-based methods.

7. Discussion and future work

The experiments conducted in this research project focused specifically on the NSL-KDD dataset, which limits the applicability of the results to other datasets and real-world scenarios. Testing these models on diverse and real-world data would provide a more accurate assessment of their performance in practical situations. Future work could also consider using more recent and comprehensive datasets, such as CICIDS2017 or CSE-CIC-IDS2018, as well as evaluating the models on real network traffic when such data become available.

Furthermore, despite demonstrating good performance, machine learning models for intrusion detection face a significant limitation due to their vulnerability to adversarial attacks. These attacks involve malicious inputs crafted to deceive machine learning models while maintaining a legitimate appearance (Kurakin et al., 2016). Two scenarios can be distinguished: white-box attacks, where the attacker knows the model's architecture and parameters, and black-box attacks, where only the outputs are observed to infer the model's behavior. In both cases, an attacker can modify characteristics of malicious traffic (e.g., by masking certain patterns or adding controlled noise) to cause it to be misclassified as benign.

To make IDS more robust against these attacks, several research directions should be explored. Training models on augmented datasets containing adversarial examples appears particularly promising. Another approach involves modeling attacks and designing appropriate countermeasures to better protect IDS against intelligent adversaries

within the framework of adversarial machine learning (Huang et al., 2011).

Finally, adapting to unknown attacks remains a significant challenge. A hybrid approach that combines supervised and unsupervised learning particularly through clustering techniques prior to classification could enhance the detection of emerging threats.

8. Conclusion

The implementation of an effective IDS model remains a significant challenge in the field of cybersecurity. The objective of this study was to develop machine learning models for intrusion detection namely Random Forest, XGBoost, and DNN and to evaluate their performance using the NSL-KDD dataset.

The results obtained in this study demonstrate the effectiveness of the adopted methodology and the selected techniques for intrusion detection, particularly considering the issue of data imbalance. Random Forest achieved an accuracy of 99.80% and an AUC of 0.9988, outperforming XGBoost (99.79%, AUC of 0.9985) and DNN (98.66%, AUC of 0.9872). This highlights the superior capability of Random Forest to accurately distinguish between normal connections and intrusions compared to the other models.

In comparison with related work, approaches based on XGBoost (Azam et al., 2023) and Random Forest (Pranto et al., 2022) have reported accuracies of around 99.60% and 99.50%, respectively, while DNN (Mohammed & Gbashi, 2021) achieved an accuracy of 94%. It is also worth noting that other models, such as SVM and ELM (Ahmad et al., 2018), achieved an accuracy of 99.67%, whereas AE (Xu et al., 2021) reported a lower accuracy of 90.61%. Furthermore, some hybrid models like LSTM or GRU (Kasongo, 2023), although effective, did not exceed 90% accuracy.

This work proposes a comparative framework to evaluate three machine learning algorithms Random Forest, XGBoost, and DNN under consistent conditions, including ablation studies with and without SMOTE, as well as hyperparameter optimization using GridSearch and Optuna. This approach enables a reproducible and balanced assessment of the effects of data balancing and parameter tuning techniques. The detailed comparison of classification performance alongside computational costs offers practical insights to guide the selection of IDS models tailored to operational constraints.

By applying this methodology to the NSL-KDD dataset, the study identifies the most effective algorithm for intrusion detection, contributing to the broader literature on machine learning in IDS. The results also show an improvement in detection accuracy compared to related works, highlighting the relevance of the chosen approach.

CRediT authorship contribution statement

Sow Thierno Hamidou: Conceptualization, Methodology, Writing – review & editing. **Adda Mehdi:** Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access*, *6*, 33789–33795. http://dx.doi.org/10.1109/ACCESS.2018.2841987.

Al-Maksousy, H. H., Weigle, M. C., & Wang, C. (2018). NIDS: Neural network based intrusion detection system. In *2018 IEEE international symposium on technologies for homeland security* (pp. 1–6). IEEE.

Aldallal, A., & Alisa, F. (2021). Effective intrusion detection system to secure data in cloud using machine learning. *Symmetry*, *13*(12), 2306.

Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree. *IEEE Access*, *11*, 80348–80391.

Catania, C. A., & Garino, C. G. (2012). Automatic network intrusion detection: Current techniques and open issues. *Computers & Electrical Engineering*, *38*(5), 1062–1072.

Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and their Applications*, *14*(6), 67–74.

Chandra, P., Lilhore, U. K., & Agrawal, N. (2017). Network intrusion detection system based on modified random forest classifiers for KDD Cup-99 and NSL-KDD dataset. *International Research Journal of Engineering and Technology*, *4*, 786–791.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Dhaliwal, S. S., Nahid, A. A., & Abbas, R. (2018). Effective intrusion detection system using XGBoost. *Information*, *9*(7), 149.

Gurung, S., Ghose, M. K., & Subedi, A. (2019). Deep learning approach on network intrusion detection system using NSL-KDD dataset. *International Journal of Computer Network and Information Security*, *11*(3), 8–14.

Hanifi, S., Cammarono, A., & Zare-Behtash, H. (2024). Advanced hyperparameter optimization of deep learning models for wind power prediction. *Renewable Energy*, *221*, Article 119700.

Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., & Tygar, J. D. (2011). Adversarial machine learning. In *Proceedings of the 4th ACM workshop on security and artificial intelligence* (pp. 43–58).

Jia, Y., Wang, M., & Wang, Y. (2019). Network intrusion detection algorithm based on deep neural network. *IET Information Security*, *13*(1), 48–53. http://dx.doi.org/10.1049/iet-ifs.2018.5176.

Jiang, K., Wang, W., Wang, A., & Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access*, *8*, 32464–32476.

Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a recurrent neural networks based framework. *Computer Communications*, *199*, 113–125. http://dx.doi.org/10.1016/j.comcom.2023.03.017.

Khan, S. S., & Mailewa, A. B. (2023). Detecting network transmission anomalies using autoencoders-SVM neural network on multi-class NSL-KDD dataset. In *2023 IEEE 13th annual computing and communication workshop and conference* (pp. 0835–0843). IEEE.

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*, 20.

Kikissagbe, B. R., Adda, M., Célicourt, P., Haman, I. T., & Najjar, A. (2024). Machine learning for DoS attack detection in IoT systems. *Procedia Computer Science*, *241*, 195–202.

Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236.

Mahesh, T. R., Geman, O., Margala, M., & Guduri, M. (2023). The stratified K-folds cross-validation and class-balancing methods with high-performance ensemble classifiers for breast cancer classification. *Healthcare Analytics*, *4*, Article 100247.

Meena, G., & Choudhary, R. R. (2017). A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA. In *2017 international conference on computer, communications and electronics* (pp. 553–558). IEEE.

Mohammed, B., & Gbashi, E. K. (2021). Intrusion detection system for NSL-KDD dataset based on deep learning and recursive feature elimination. *Engineering and Technology Journal*, *39*(7), 1069–1079.

Pranto, M. B., Ratul, M. H. A., Rahman, M. M., Diya, I. J., & Zahir, Z. B. (2022). Performance of machine learning techniques in anomaly detection with basic feature selection strategy–a network intrusion detection system. *Journal of Advances in Information Technology*, *13*(1).

Raju, V. N., Saravanakumar, R., Yusuf, N., Pradhan, R., Hamdi, H., Saravanan, K. A., & Askar, M. A. (2024). Enhancing emotion prediction using deep learning and distributed federated systems with SMOTE oversampling technique. *Alexandria Engineering Journal*, *108*, 498–508.

Rashmi, C. R., & Shantala, C. P. (2024). Evaluating deep learning with different feature scaling techniques for EEG-based music entrainment brain computer interface. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, *7*, Article 100448.

Rawat, S., Srinivasan, A., Ravi, V., & Ghosh, U. (2022). Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network. *Internet Technology Letters*, *5*(1), Article e232. http://dx.doi.org/10.1002/itl2.232.

Srinivas, P., & Katarya, R. (2022). hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost. *Biomedical Signal Processing and Control*, *73*, Article 103456.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1–6). IEEE.

Tran, N., Schneider, J. G., Weber, I., & Qin, A. K. (2020). Hyper-parameter optimization in classification: To-do or not-to-do. *Pattern Recognition*, *103*, Article 107245.

Urmi, W. F., Uddin, M. N., Uddin, M. A., Talukder, M. A., Hasan, M. R., Paul, S., & Imran, F. (2024). A stacked ensemble approach to detect cyber attacks based on feature selection techniques. *International Journal of Cognitive Computing in Engineering*, *5*, 316–331.

Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., & Sabrina, F. (2021). Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. *IEEE Access*, *9*, 140136–140146. http://dx.doi.org/10.1109/ACCESS.2021.3118587.

Zhang, C., Chen, Y., Meng, Y., Ruan, F., Chen, R., Li, Y., & Yang, Y. (2021). A novel framework design of network intrusion detection based on machine learning techniques. *Security and Communication Networks*, *2021*(1), Article 6610675. http://dx.doi.org/10.1155/2021/6610675.

CONCLUSION GÉNÉRALE

7. SYNTHESE DES RESULTATS

La mise en œuvre d'un modèle IDS efficace demeure un défi dans le domaine de la cybersécurité. L'objectif de cette étude était de développer des modèles d'apprentissage automatique pour la détection d'intrusion, à savoir RF, XGBoost et les réseaux de neurones profonds (DNN) et d'évaluer la performance de ces modèles dans le contexte du jeu de données NSL-KDD.

Les résultats obtenus dans cette étude montrent l'efficacité de la méthodologie adoptée et les techniques choisies pour la détection d'intrusions en tenant compte du déséquilibre des données. RF a présenté une précision de 99,80 % et une AUC de 0,9988, surpassant XGBoost (99,79 %, AUC de 0,9985) et DNN (98,66 %, AUC de 0,9872). Cela met en évidence la capacité de RF à distinguer efficacement, comparativement aux autres, les connexions normales des intrusions.

En comparaison avec les travaux connexes, les approches telles que celles basées sur XGBoost [10] et RF [11], ont obtenu des précisions respectivement autour de 99,60 % et 99,50 %, tandis que DNN [12] atteint une précision de 94 %. Il est également à noter que d'autres modèles, comme le SVM et les machines à apprentissage extrême (ELM) [20], ont obtenu une précision de 99,67 %, tandis que les autoencodeurs [14] ont enregistré une précision de 90,61 %. Par ailleurs, certains modèles tels que le réseau de neurones à mémoire à long terme combiné à XGBoost (LSTM-XGBoost) et le réseau de neurones récurrent à portes combiné à XGBoost (GRU-XGBoost) [21], bien qu'efficaces, n'ont pas dépassé les 90 % de précision.

Ainsi, ce mémoire met en évidence la contribution essentielle de l'apprentissage automatique à l'amélioration des IDS, offrant un compromis efficace entre précision, robustesse et temps d'exécution, et ouvre la voie à des approches plus adaptatives et robustes, capables de répondre efficacement aux défis de la cybersécurité moderne.

8. LIMITES ET PERSPECTIVES

Les expérimentations réalisées dans le cadre de ce projet de recherche ont porté spécifiquement sur le jeu de données NSL-KDD, ce qui restreint son applicabilité à d'autres données et dans des contextes réels. Tester ces modèles sur des données réelles et variées permettrait d'offrir une vision plus précise de leur performance dans des situations réelles. Par ailleurs, bien que les modèles d'apprentissage automatique aient montré de bonnes performances, ils restent vulnérables aux attaques adversariales qui peuvent compromettre la fiabilité du système. L'intégration d'exemples adversariaux dans l'apprentissage

constituerait une piste pour renforcer leur robustesse face à ces manipulations malveillantes.

L'étude n'a pas non plus abordé la complexité algorithmique et le temps d'exécution des modèles. Pour garantir une détection en temps réel efficace, il est crucial que les modèles soient non seulement précis, mais aussi rapides et économes en ressources. Enfin, l'adaptation aux attaques inconnues demeure un défi. Une approche hybride combinant apprentissage supervisé et non supervisé, notamment via des techniques de clustering avant la classification, pourrait améliorer la détection des menaces émergentes.

RÉFÉRENCES

[1] Devan, P., & Khare, N. (2020). An efficient XGBoost–DNN-based classification model for network intrusion detection system. *Neural Computing and Applications, 32*(16), 12499-12514.

[2] Louk, M. H. L., & Tama, B. A. (2023). Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Systems with Applications, 213*, 119030.

[3] Zhang, Y., & Liu, Q. (2022). On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples. *Future Generation Computer Systems, 133*, 213-227.

[4] Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications, 14*(6), 67-74.

[5] Srinivas, P., & Katarya, R. (2022). hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost. *Biomedical Signal Processing and Control, 73*, 103456.

[6] Raju, V. N., et al. (2024). Enhancing emotion prediction using deep learning and distributed federated systems with SMOTE oversampling technique. *Alexandria Engineering Journal, 108*, 498-508.

[7] Kikissagbe, B. R., Adda, M., Célicourt, P., Haman, I. T., & Najjar, A. (2024). Machine learning for DoS attack detection in IoT systems. *Procedia Computer Science, 241*, 195-202.

[8] Jiang, K., Wang, W., Wang, A., & Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access, 8*, 32464-32476.

[9] Hanifi, S., Cammarono, A., & Zare-Behtash, H. (2024). Advanced hyperparameter optimization of deep learning models for wind power prediction. *Renewable Energy, 221*, 119700.

[10] Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree. *IEEE Access, 11*, 80348-80391.

[11] Pranto, M. B., Ratul, M. H. A., Rahman, M. M., Diya, I. J., & Zahir, Z.-B. (2022). Performance of machine learning techniques in anomaly detection with basic feature selection strategy—a network intrusion detection system. *Journal of Advanced Information Technology, 13*(1).

[12] Mohammed, B., & Gbashi, E. K. (2021). Intrusion detection system for NSL-KDD dataset based on deep learning and recursive feature elimination. *Engineering and Technology Journal, 39*(7), 1069-1079.

[13] Magdy, M. E., Matter, A. M., Hussin, S., Hassan, D., & Elsaid, S. (2023). A comparative study of intrusion detection systems applied to NSL-KDD dataset. *Egyptian International Journal of Engineering Science and Technology, 43*(2), 88-98.

[14] Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., & Sabrina, F. (2021). Improving performance of autoencoder-based network anomaly detection on NSL-KDD dataset. *IEEE Access, 9*, 140136-140146.

[15] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1-6). IEEE.

[16] Urmi, W. F., et al. (2024). A stacked ensemble approach to detect cyber attacks based on feature selection techniques. *International Journal of Cognitive Computing in Engineering, 5*, 316-331.

[17] Rashmi, C., & Shantala, C. (2024). Evaluating deep learning with different feature scaling techniques for EEG-based music entrainment brain computer interface. *E-Prime Advances in Electrical Engineering, Electronics and Energy, 7*, 100448.

[18] Mahesh, T., Geman, O., Margala, M., Guduri, M., et al. (2023). The stratified K-folds cross-validation and class-balancing methods with high-performance ensemble classifiers for breast cancer classification. *Healthcare Analytics, 4*, 100247.

[19] Zhang, H., Huang, L., Wu, C. Q., & Li, Z. (2020). An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Computer Networks, 177*, 107315.

[20] Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access, 6*, 33789-33795.

[21] Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a recurrent neural networks based framework. *Computer Communications, 199*, 113-125.

[22] Hamidou, S. T., & Mehdi, A. (2025). Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks. *Machine Learning with Applications*, 100738.