REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ DES MATHEMATIQUES ET DE l'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunication

Par :
Zenina Chaimaa
Labbadi Ourida

Sur le thème

La proposition d'une approche pour l'optimisation de l'efficacité des systèmes de détection d'intrusion en basant sur les graphes de connaissance

Soutenu publiquement le .. / .. / 2023 à Tiaret devant le jury composé de :

Mr MEGHAZI Hadj Madani	MCB Université de Tiaret	Président
Mr DAOUD Mohamed Amine	MAA Université de Tiaret	Encadrant
Mr BEKKAR Mohamed	MAA Université de Tiaret	Examinateur

2023-2024

Sample output to test PDF Combine only

Dédicace

Je dédie ce travail à :

Mes chers parents pour leur amour, leurs encouragements et leurs sacrifices.

Ma deuxième maman Fatma que j'aime beaucoup.

À mon frère Mohamed et à mes sœurs Aicha, Tourkia, Nourhan et Imen.

À mes neveux et nièces, chacun à son prénom.

Je ne peux exprimer à travers ces lignes tous mes sentiments d'amour et de tendresse. Je vous souhaite la réussite dans votre vie privée et professionnelle.

À Ourida, chère amie avant d'être binôme, et à tous mes amis.

À tous les membres de ma famille.

Chaimaa

Je dédie ce travail à :

Ma très chère mère

Honneur à mon défunt père

Pour leurs amour, encouragements, soutiens, prières et sacrifices

Je vous exprime ma gratitude éternelle et mon amour infini à travers cet humble effort.

À mon frère Radeouane, ma belle-sœur Fatima Zohra, et mon cher neveu Mohamed Akram

Que Dieu tout-puissant veille sur vous, vous accorde santé, bonheur et longue vie

À mes chères amies chacune son prénom

À tous les membres de la famille Labbadi et Kharroubi

À mon cher binôme Chaimaa

Pour sa patience, et soutien moral que dieu nous garde toujours unis

Ourida

Remerciement

Nous tenons tout d'abord à remercier Dieu, le Tout-Puissant, qui nous a aidés à réaliser ce travail.

La présentation de ce modeste travail nous offre l'occasion d'exprimer notre profonde gratitude à Monsieur ***Daoud Mohamed Amine****, qui a bien voulu diriger ce travail pendant toute la durée de l'expérimentation et la mise en forme du document final. Ses nombreux conseils ne nous ont jamais fait défaut. Nous sommes heureux de lui exprimer ici notre respectueuse reconnaissance.*

Nous remercions également les membres du jury de notre soutenance de mémoire, Monsieur ***Bekkar Mohamed*** *et Monsieur* ***Meghazi Hadj Medani****, pour avoir accepté de faire partie du jury.*

Nous désirons aussi exprimer notre gratitude envers les professeurs du département Informatique de l'université Ibn Khaldoun, qui nous ont fourni les outils nécessaires à la réussite de nos études universitaires.

Nous ne saurions oublier notre famille, Zenina et Labbadi, pour leur soutien indéfectible et pour avoir toujours cru en nous. Leurs encouragements ont été notre refuge et notre motivation durant tout le parcours académique.

Un merci spécial à nos camarades de classe de 2 RT, pour leurs encouragements, leurs échanges intellectuels stimulants et pour tous les moments partagés.

Enfin, nous tenons à remercier tous ceux qui ont participé de près ou de loin à la réalisation de ce modeste travail.

Résumé

Dans un environnement cybernétique en perpétuelle évolution, un défi majeur persiste : comment renforcer efficacement les systèmes de détection d'intrusion (IDS) pour contrer les attaques sophistiquées et émergentes ? Les méthodes classiques basées sur l'apprentissage automatique rencontrent des limites dans la détection des schémas d'attaques complexes et la prévention des menaces évolutives. Face à cette problématique, il est impératif de renforcer les capacités des systèmes de détection d'intrusion (IDS) en exploitant les potentialités des graphes de connaissance. Cette approche est évaluée par le biais d'une analyse comparative de ses performances par rapport aux méthodes traditionnelles d'apprentissage automatique.

Mots clés : système de détection d'intrusion, graphe de connaissance, apprentissage automatique.

Abstract

In an ever-changing cyber environment, a major challenge persists: how to effectively strengthen intrusion detection systems (IDS) to counter sophisticated and emerging attacks? Traditional machine learning methods encounter limitations in detecting complex attack patterns and preventing evolving threats. Faced with this problem, it is imperative to strengthen the capabilities of intrusion detection systems (IDS) by exploiting the potential of knowledge graphs. This approach is evaluated by benchmarking its performance against traditional machine learning methods.

Keywords: Intrusion detection system, knowledge graph, machine learning.

ملخص:

في بيئة إلكترونية دائمة التغير، لا يزال هناك تحدٍ كبير: كيفية تعزيز أنظمة الكشف عن التسلل (IDS) بشكل فعال لمواجهة الهجمات المتطورة والناشئة ؟ تواجه طرق التعلم الآلي التقليدية قيودًا في اكتشاف أنماط الهجوم المعقدة ومنع التهديدات المتطورة. في مواجهة هذه المشكلة، من الضروري تعزيز قدرات أنظمة الكشف عن التسلل (IDS) من خلال استغلال إمكانات الرسوم البيانية المعرفية. يتم تقييم هذا النهج من خلال قياس أدائه مقابل طرق التعلم الآلي التقليدية.

الكلمات المفتاحية : نظام الكشف عن التسلل، الرسم البياني المعرفي، التعلم الآلي.

Table des matiéres

Dédicace

Remerciement

Résumé

Liste des figures

Liste des tableaux

Liste des abbreviations

Introduction générale.....1

1 Chapitre 01 : systèmes de détection d'intrusion.....4

1.1 Introduction 4

1.2 Concepts 4

1.2.1 Intrusion 4

1.2.2 Détection d'intrusion 4

1.2.3 Attaques 4

1.2.3.1 Types d'attaque.....4

Attaque directe.....5

Attaque indirecte par rebond.....5

Attaque indirect par réponse.....6

1.3 La sécurité 6

1.3.1 Sécurité de réseau 6

1.3.2 Propriété de sécurité 7

1.3.3 Types de sécurité de réseau 8

1.4 Système de détection des intrusions (IDS) 10

1.4.1 Définition 10

1.4.2 L'architecture des IDS.....10

1.4.3 Les différents types des IDS 11

1.4.3.1 Les systèmes de détection d'intrusion hôte (HIDS).....11

1.4.3.2 Les systèmes de détection d'intrusion réseau (NIDS).....12

1.4.3.3 Les IDS hybride.....13

1.4.4 Les méthodes de détection d'intrusion 14

1.4.4.1 L'approche par signature.....14

1.4.4.2 L'approche comportementale.....14

1.4.5 Réaction et comportement après une attaque 15
1.4.6 Mesures performance des IDS 16
1.5 Conclusion 18
2 Chapitre 02 : Apprentissage automatique 20
2.1 Introduction 20
2.2 Définitions 20
2.2.1 Intelligence artificielle 20
2.2.2 L´apprentissage automatique 20
2.2.3 L´apprentissage profond 20
2.3 Les types d'apprentissage automatique 21
2.3.1 Apprentissage supervisé (Supervised Learning) 21
2.3.2 Apprentissage non supervisé (Unsupervised Learning) 22
2.3.3 Apprentissage par renforcement (Reinforcement Learning) 23
2.4 Les algorithmes de machine Learning 23
2.4.1 Algorithmes supervisés 24
2.4.1.1 Algorithmes de classification 24
2.4.1.2 Algorithmes de régression 27
2.4.2 Les algorithmes non-supervisés 28
2.4.3 Algorithme par renforcement 30
2.5 L´apprentissage en profondeur 30
2.5.1 Définition 30
2.5.2 Les algorithmes d´apprentissage en profondeur 31
2.5.2.1 Algorithmes Supervisé 31
2.5.2.2 Algorithmes non supervisés 32
2.6 Conclusion 34
3 Chapitre 03 : Graphes de connaissances 37
3.1 Introduction 37
3.2 Les graphes 37
3.2.1 Définition 37
3.2.2 Les concepts fondamentaux d'un graphe 37
3.2.3 Définir l'analytique de graphe et la science des données de graphe 38
3.2.4 L'émergence de la science des données de graphe 38
3.3 Les graphes de connaissances 39

3.3.1 Définition 39
3.3.2 Historique 39
3.3.3 L'architecture de graphe de connaissance 40
3.3.4 Points forts des graphes de connaissance 41
3.3.5 La construction des graphes de connaissance 41
3.3.5.1 Collection et l'extraction de l'information 41
3.3.5.2 Vérification et inférence 42
3.3.6 Modèles de graphes de connaissances 42
3.3.6.1 Resource Description Framework (RDF) 42
3.3.6.2 Modèle de données de graphe de propriétés (PGM) 43
3.3.7 Les algorithmes de graphe de connaissance 43
3.3.7.1 Algorithmes de centralité 43
3.3.7.2 Algorithmes de détection de communauté 44
3.3.7.3 Algorithmes de connectivité 44
3.3.7.4 Algorithme de similarité 44
3.4 Conclusion 44
4 Chapitre 04 : L'intégration des graphes de connaissances avec l'intelligence artificielle 46
4.1 Introduction 46
4.2 La valeur de la combinaison des graphes de connaissances avec l'IA 46
4.3 La convergence de l'IA et des graphes de connaissances 46
4.4 Infusion d'intelligence dans les données via l'utilisation de graphes de connaissances 46
4.5 Le Rôle central des données dans l´apprentissage automatique 47
4.6 Défis dans l'intégration du contexte 47
4.7 L'apport des graphes de connaissances dans l'amélioration de l'apprentissage automatique 48
4.8 Un processus amélioré de l'apprentissage automatique grâce aux graphes de connaissances 48
4.9 L'importance du contexte pour l'IA 48
4.9.1 Graphes de Connaissances : Contexte pour les Décisions 50
4.9.2 Apprentissage automatique accéléré par les graphes : Contexte pour l'efficacité 50
4.9.3 Caractéristiques Connectées : Contexte pour la Précision 51
4.9.4 Explicabilité de l'IA : Contexte de Crédibilité 52
4.10 L´intelligence artificielle améliorée par les Graphes 52
4.11 Conclusion 53
5 Chapitre 05 : Approche proposée 55

5.1 Introduction 55
5.2 Description de l'ensemble de données CICIDS2017 55
5.3 Matériel et logiciels 57
5.3.1 Environnement d´exécution 57
5.3.2 Outils utilisés 57
5.3.3 Langages utilisés 58
5.3.4 Les bibliothèques utilisées 58
5.1 Etude d'ablation 59
5.2 Les expériences 59
5.2.1 Approche 01 : Application des méthodes d'apprentissage automatiques 59
5.2.2 Approche 02 : Application des algorithmes des graphes de connaissance 65
5.2.3 Approche 03 : L'application des techniques de ML sur les KG 77
5.2.3.1 Modèle de construction 78
5.3 Comparaison des résultats 84
5.4 Conclusion 84
Conclusion générale 86
Bibliographique 87

Liste des figures

Figure 1 : Une attaque directe. [3]....5
Figure 2: Attaque indirecte par rebond. [3]....5
Figure 3: Attaque indirect par réponse. [3]....6
Figure 4: Propriété de la sécurité informatique. [12]....8
Figure 5: Architecture de pare-feu….....8
Figure 6: L'architecture d'IDS proposé par L'IDWG….....10
Figure 7: Système de détection d'intrusion hôte(HIDS)….....12
Figure 8: Système de détection d'intrusion réseau(NIDS)….....12
Figure 9: Matrice de confusion [14]....16
Figure 10: La relation entre l´intelligence artificielle, l´apprentissage automatique et l´apprentissage profond….....21
Figure 11: Les types d´apprentissage automatique….....21
Figure 12: Schéma d´apprentissage supervisé….....22
Figure 13: Schéma d´apprentissage non supervisé….....23
Figure 14: Schéma d´apprentissage par renforcement….....23
Figure 15: L´algorithme SVM….....24
Figure 16: Le diagramme de structure de l'algorithme Random Forest….....25
Figure 17: L'algorithme d'apprentissage automatique KNN….....27
Figure 18: Régression linéaire versus régression logistique. [26]....28
Figure 19: L´algorithme de K-means….....30
Figure 20: L'architecture de base d'un réseau de neurones convolutionnel (CNN)….....32
Figure 21: Un simple réseau neuronal récurrent. [31]....32
Figure 22: L´algorithme RNAe.[32]....33
Figure 23: Flux de travail d´un réseaux génératifs antagonistes (GAN)….....34
Figure 24: Exemple d'un graphe simple....37
Figure 25: Exemple d'un graphe de connaissance. [36]....39
Figure 26: l'architecture des graphes de connaissance [38]....40
Figure 27: Graphique représentant RDF [43]....43
Figure 28: Les apports de la science des données orientée graphe. [48]....47
Figure 29: quatre façons dont les graphes fournissent du contexte [47]….....49
Figure 30: Filtrage des données stockées dans des tables avec celle des graphes….....51
Figure 31: Méthode traditionnelle par rapport à la méthode basée sur les graphes pour la prise de décision. [48]....52
Figure 32: Architecture de composants de l'approche 01….....60
Figure 33: Visualisation des performances améliorées du modèle….....62
Figure 34: La courbe ROC du modèle CNN….....63
Figure 35: Matrice de confusion pour le modèle amélioré (KNN)….....64
Figure 36: La courbe ROC du modèle KNN….....64
Figure 37: La courbe ROC du modèle RF….....65
Figure 38: Architecture de composants de l'approche 02….....65
Figure 39: Visualisation du graphe local….....67
Figure 40: Visualisation du graphe global….....68
Figure 41: Projection de Graph….....69
Figure 42: Mode d'éxecution estimate….....70
Figure 43: Éxecution en mode Stream……....71
Figure 44: Mode d'exécution Stats….....71

Figure 45: L'éxecution en mode Mutate….72
Figure 46: L'éxecution en mode Write….73
Figure 47: Projection de graphe "myGraph"….74
Figure 48: Résultat d'application de l'algorithme KNN….75
Figure 49: Mode write estimate….76
Figure 50: Louvain avec le mode flux….76
Figure 51: Louvain résultat en mode d'écriture….77
Figure 52: Architecture de composants de l'approche 03….78
Figure 53: Nombre de nœuds par type….79
Figure 54: Nombre total de relation pour chaque type de relation….79
Figure 55: Les premières lignes du DataFrame résultant….81
Figure 56: Visualisation des performances améliorées du modèle sur le nouveau jeu de données….82
Figure 57: La courbe ROC de modele CNN de nouveau jeu de données ….82
Figure 58: La courbe ROC de modele KNN de nouveau jeu de données….83
Figure 59: La courbe ROC de modele RF de nouveau jeu de données….84

Liste des tableaux

Tableau 1 : Comparaison entre NIDS et HIDS….13
Tableau 2: Les avantages et les inconvénients des méthodes de détection….15
Tableau 3: Caractéristiques générales de l'ensemble de jeu de données CICIDS2017….55
Tableau 4: Occurrence des instances par classe dans l'ensemble de données CICIDS2017….56
Tableau 5: Caractéristiques présentes dans l'ensemble de données CICIDS2017….57
Tableau 6: Les bibliothèques utilisées dans notre approche….59
Tableau 7: Rendement des mesures d'évaluation pour le modèle amélioré(CNN)….63
Tableau 8: Rendement des mesures d'évaluation pour le modèle amélioré(KNN)….63
Tableau 9: Mesures d'évaluation pour chaque étiquette….63
Tableau 10: Rendement des mesures d'évaluation pour le modèle amélioré(RF)….65
Tableau 11: Caractéristiques présentes dans nouveau jeu de données graphique….81
Tableau 12: Rendement des mesures d'évaluation pour le modèle(CNN) sur le nouveau jeu de données ….82
Tableau 13: Rendement des mesures d'évaluation pour le modèle(knn) sur nouveau jeu de données….83
Tableau 14: Rendement des mesures d'évaluation pour le modèle (RF) sur nouveau jeu de données….83

Liste des abbreviations

IDS: Intrusion Detection System

HIDS: Host Intrusion Detection System

NIDS: Network Intrusion Detection System

AI: Artificial Intelligence

ML: Machine Learning

KG: Knowledge Graph

CICIDS 2017: Cyber Intrusion Detection Data Sets 2017

IDWG: Intrusion Detection Working Group

CNN**:** Convolutional Neural Network

KNN: K-Nearest Neighbors

RF: Random Forest

SVM: Support Vector Machine

RNN: Recurrent Neural Network

GMM: Gaussian Mixture Model

GAN: Generative Adversarial Network

ROC (Receiver Operating Characteristic)

AUC (Area Under the Curve)

Introduction générale

Introduction générale

La sécurité informatique est devenue une préoccupation majeure à l'ère numérique, alors que les menaces en ligne continuent de se multiplier et de devenir de plus en plus sophistiquées. Parmi les outils essentiels dans la protection des réseaux informatiques, les systèmes de détection d'intrusion (Intrusion Detection System) occupent une place centrale. Ces systèmes jouent un rôle crucial dans la défense des réseaux en informant les responsables de la sécurité pour les alerter sur des comportements malveillants tels que les attaques, les logiciels malveillants et les intrusions.

L'importance des IDS dans la protection des réseaux vitaux est indéniable. Ces systèmes fournissent une deuxième ligne de défense essentielle, permettant aux équipes de sécurité de réagir rapidement face aux menaces et de prévenir les dommages potentiels. Cependant, malgré leur importance, les IDS sont confrontés à des défis et des limitations tels que la gestion des faux positifs et négatifs, la détection des attaques sophistiquées et chiffrées, ainsi que des contraintes de déploiement, de performances et d'adaptation aux évolutions constantes des menaces.
Afin d'assurer cet objectif, une question de recherche sera poser afin de traiter la problématique du notre contexte de travail :

Comment peut-on améliorer l'efficacité de la détection des intrusions par l'intégration des techniques de l'apprentissage automatique et les graphes de connaissances ?

Dans ce contexte, une étude se penche sur la proposition d'une approche visant à optimiser l'efficacité des systèmes de détection d'intrusion en se basant sur les graphes de connaissance. Cette approche vise à combler le fossé entre la recherche et la pratique en fournissant une méthodologie pour renforcer la capacité des IDS à détecter et à répondre aux menaces émergentes.

Le but de travail

L'objectif de ce mémoire est d'analyser des approches classiques de détection d'intrusion qui reposent sur l'utilisation du Machine Learning (ML), en particulier leur incapacité à saisir les relations complexes entre les données. Nous cherchons à explorer ces limites et à proposer une approche basée sur les graphes de connaissance (Knowledge Graphs). Nous avons choisi d'utiliser le jeu de données CICIDS 2017, qui contient un trafic réseau réel avec différentes attaques courantes.
Les objectifs spécifiques sont les suivants :

- Développer une méthodologie pour la création et l'utilisation des graphes de connaissance dans le contexte de la détection d'intrusion, avec pour objectif d'améliorer la capacité des systèmes à identifier les schémas d'attaques et à anticiper les comportements malveillants.

- Évaluer l'efficacité de notre approche en comparant ses performances avec celles des méthodes ML traditionnelles, en utilisant des métriques telles que l'exactitude, la précision et le rappel, sur des ensembles de données représentatifs de scénarios réels d'intrusion.

Notre mémoire est organisé comme suit :

- Chapitre 1 : décrit une introduction à un système de détection d'intrusion, à leurs principes de fonctionnement, ainsi qu'à divers concepts liés aux détections d'intrusion.
- Chapitre 2 : introduit l'apprentissage automatique, l'apprentissage profond et l'intelligence artificielle, en exposant en détail leurs différentes catégories et algorithmes qui leur sont associés.
- Chapitre 3 : met en évidence les principaux concepts des graphes de connaissances, en analysant les techniques employées pour leur élaboration, leur structure et les divers modèles de KG.
- Chapitre 4 : analyse les diverses facettes de l'incorporation des graphes dans des domaines essentiels de l'apprentissage automatique.
- Chapitre 5 : nous exposons la création et l'évaluation de notre modèle, en mettant l'accent sur l'explication de la méthodologie employée et des divers paramètres pris en compte, ainsi que sur les résultats.

- En dernier lieu, une conclusion générale et des perspectives de ce travail seront présentées.

Chapitre 01

Systèmes de détection d'intrusion

1 *Chapitre 01 : systèmes de détection d'intrusion*

1.1 Introduction

De nos jours, avec le développement croissant de l'informatique, il ne fait aucun doute que la sécurisation des entreprises contre les attaques malveillantes est devenue une nécessité pressante. D'autant plus que les menaces se produisent fréquemment chaque année et ne peuvent pas être arrêtées. Dans ce contexte, la sécurité informatique revêt une importance cruciale pour assurer la protection des données sensibles et prévenir toute utilisation malveillante des systèmes. Cela passe notamment par l'utilisation d'outils tels que les systèmes de détection d'intrusion (IDS).

Dans ce premier chapitre, nous introduisons les concepts élémentaires de la sécurité informatique. Ensuite, nous présentons le système de détection d'intrusion (IDS).

1.2 Concepts

1.2.1 Intrusion

Dans le contexte de la sécurité informatique, une intrusion désigne toute activité illégale menée sur un système ou un réseau pour obtenir des privilèges non autorisés et le vol de ressources réseau précieuses ou détruire les données du système. [1]

1.2.2 Détection d'intrusion

La détection d'intrusion est une technologie relativement récente, la majeure partie de la recherche et du développement en matière de détection d'intrusion a lieu depuis 1980.

La détection d'intrusion est le processus de surveillance des événements survenant dans un système informatique ou un réseau, en l'analysant pour détecter les problèmes de sécurité. Il y d'autres systèmes de surveillance analogues dans d'autres domaines, y compris les alarmes antivols et les systèmes de surveillance vidéo trouvés dans les magasins et les banques. [1]

1.2.3 Attaques

Une attaque est définie comme une faute d'interaction malveillante visant à violer une/ou plusieurs propriétés de sécurité veulent dire que les attaques représentent les moyens d'exploiter une vulnérabilité. Ils s'appuient sur divers types de faiblesses telles que les faiblesses des protocoles, faiblesses d'authentification, faiblesses d'implémentation ou bugs et les mauvaises configurations. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées (par des virus, chevaux de Troie, vers, etc.) [2]

1.2.3.1 Types d'attaque

Les attaques peuvent être classées en trois familles différentes :

Attaque directe

Dans ce cas, l'hacker, à l'aide d'un logiciel ou d'un script, attaque directement l'ordinateur de sa victime. De nombreux logiciels circulent sur internet et permettant ce type de méfait sans connaissance particulière, c'est une méthode très prisée des hackers. [3]

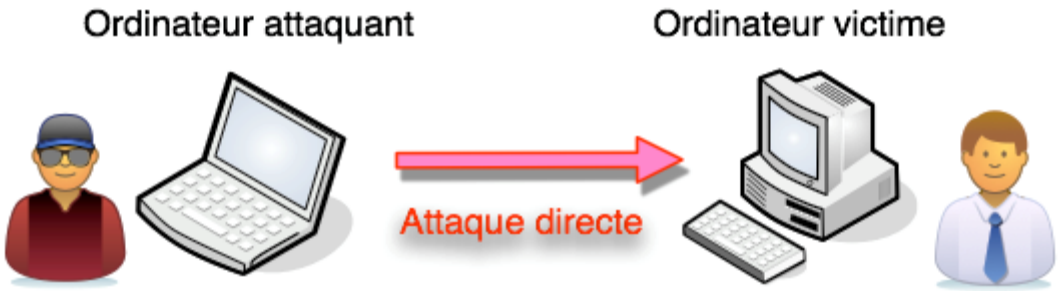


Figure 1 : Une attaque directe. [3]

Attaque indirecte par rebond

Ce type d'attaque est très prisé des hackers. En effet, le rebond a deux avantages :

- Masquer l'identité du pirate.
- Utiliser les ressources de l'ordinateur intermédiaire car il est plus puissant pour attaquer.

Le principe-est simple car les paquets d'attaque sont envoyés à l'ordinateur intermédiaire, qui répercute l'attaque vers la victime. D'où le terme par rebond. [2]

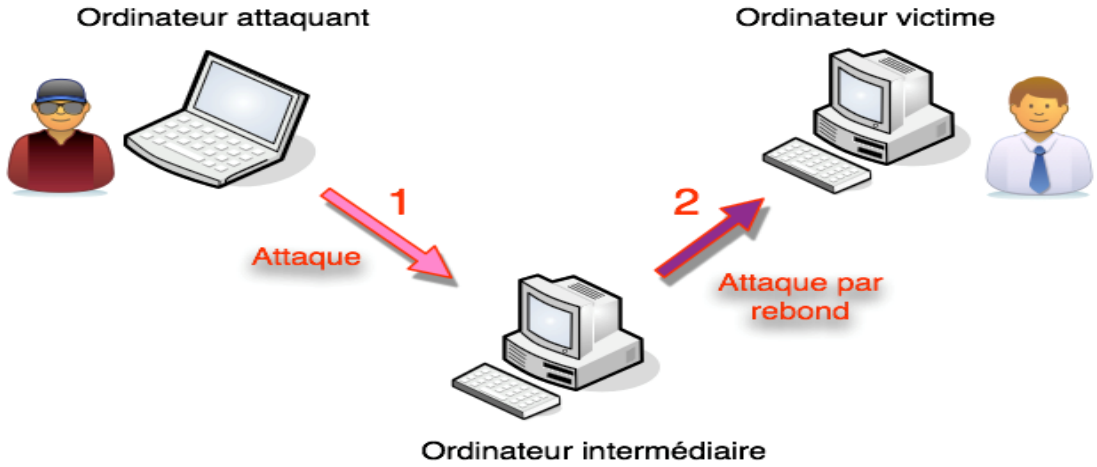


Figure 2: Attaque indirecte par rebond. [3]

Attaque indirect par réponse

Ce type d'attaque est un dérivé de l'attaque par rebond. Elle offre les mêmes avantages, du point de vue d'hacker. Mais au lieu d'envoyer une attaque à l'ordinateur intermédiaire pour qu'il la répercute, l'attaquant va lui envoyer une requête.
L'ordinateur victime va exécuter la requête vis-à-vis d'un autre ordinateur ou d'un site auquel il a l'accès normal. C'est cette réponse à la requête qui va être envoyée à l'ordinateur de l'hacker. [3]

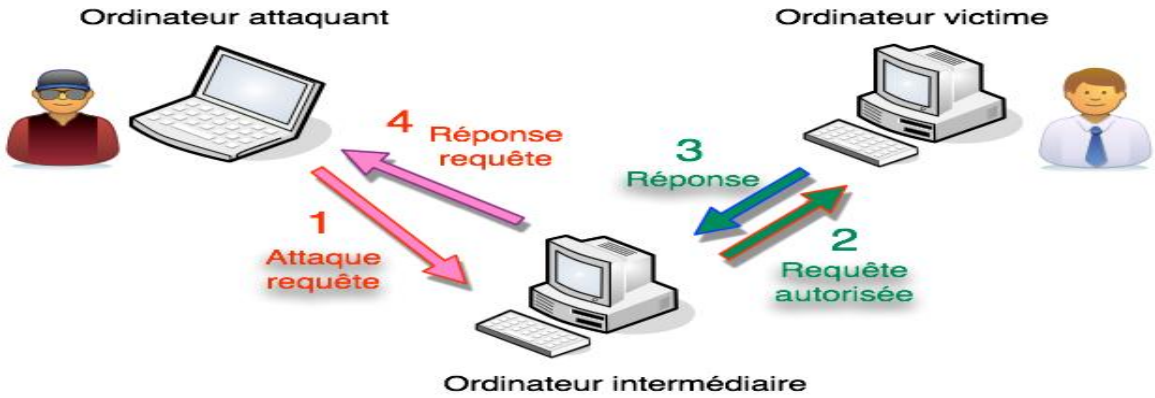


Figure 3: Attaque indirect par réponse. [3]

1.3 La sécurité

La sécurité est une notion qui concerne la protection des personnes, des biens, des informations et des infrastructures contre les menaces, les risques ou les dommages potentiels qui peuvent survenir. Il peut s'appliquer à divers domaines comme la sécurité personnelle, la sécurité informatique, la sécurité nationale, la sécurité des réseaux, etc.

1.3.1 Sécurité de réseau

La sécurité du réseau peut être définie comme le processus de conception d'une stratégie défensive et mettre en œuvre les mesures et les garanties nécessaires pour protéger les infrastructures de réseau d'accès non autorisé [4], afin d'assurer la confidentialité, l'intégrité, la disponibilité et la traçabilité de l'information traitée. En général, la sécurité d'un réseau englobe celle du système informatique sur lequel il s'appuie.

Il peut s'agir :

- D'empêcher des personnes non autorisées d'agir sur le système de façon malveillante.
- D'empêcher les utilisateurs d'effectuer des opérations involontaires capables de nuire au système.
- De sécuriser les données pour éviter la perturbation ou des pannes.

- De garantir la non-interruption d'un service. [2]

1.3.2 Propriété de sécurité

La sécurité des systèmes d'information est fondée sur trois propriétés fondamentales désignées par le terme CIA (confidentialité, intégrité et disponibilité).
Ces trois propriétés varient suivant le contexte dans lequel elles sont utilisées.

Intégrité : D'une manière générale, l'intégrité désigne le fait que les données, lors de leur traitement, de leur conservation ou de leur transmission, ne doivent subir aucune altération ou destruction volontaire ou accidentelle [5], en les ayant dans leur état prévu et à l'abri de toute modification inappropriée, Cela peut être accompli grâce à des mesures de chiffrement ou de hachage.

Confidentialité : Il assure que les informations confidentielles des utilisateurs restent secrètes. Autrement dit, cela consiste à rendre l'information inintelligible à d'autres personnes que son propriétaire. [6]
En effet, toutes les entreprises, quel que soit le secteur dans lequel elles opèrent, disposent d'informations importantes qu'elles souhaitent garder secrètes. Ils impliquent généralement l'entreprise elle-même, les clients ou les employés. Par exemple, il peut s'agir des types de données suivants :

- Identifiant bancaire.
- Information produit.
- Informations sur le contrat... etc.

Disponibilité : Le but de la disponibilité est de garantir l'accès à un ou des services ressources afin que les systèmes d'information puissent être maintenus opérationnels tout en accédant aux informations en cas de besoin.

Un autre ensemble de garanties doit également être mis en œuvre pour garantir la sécurité des systèmes d'information se sont :

L'authentification : assure que lorsqu'un utilisateur agit sur les services mis en place, le système est capable de garantir que l'utilisateur est bien celui qu'il prétend être. [6]

La non-répudiation : assure que lorsqu'un utilisateur agit sur les services, il ne lui est pas possible de nier d'avoir fait cette action. [6]

Traçabilité : Il s'agit de s'assurer que les opérations effectuées sur les données sont traçables. Cela peut être fait via des journaux d'audit ou des outils de surveillance du réseau.

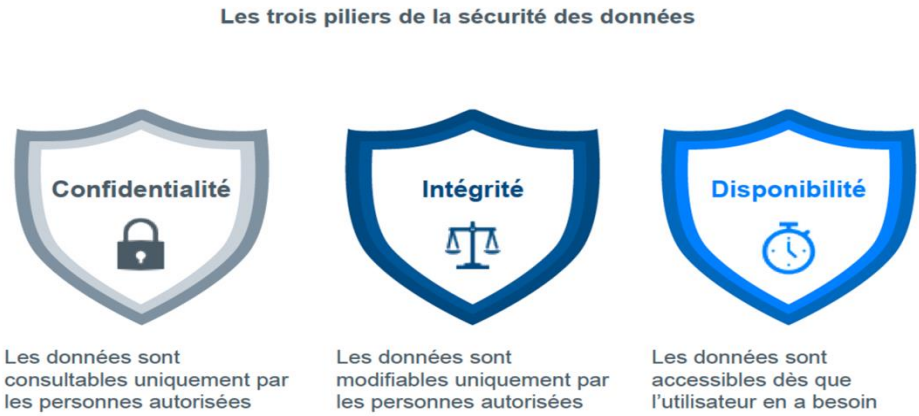


Figure 4: Propriété de la sécurité informatique. [12]

1.3.3 Types de sécurité de réseau

Il existe de nombreux types de sécurité de réseau, dont les plus courants sont ce qui suit **:**

Un pare-feu

Un pare-feu (ou firewall en anglais) est une collection de composants placée entre deux réseaux. Un firewall doit filtrer l'ensemble du trafic qui provient du réseau externe dirigé vers le réseau interne (et vice-versa). De même, un firewall filtrera le trafic non autorisé. [6]

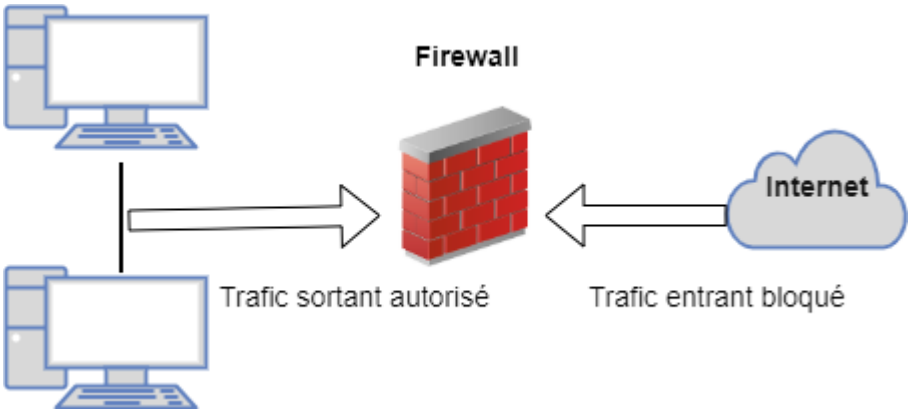


Figure 5: Architecture de pare-feu.

Contrôle d'accès au réseau (Network Access Control)

En première ligne de la défense, le contrôle d'accès au réseau, comme son nom l'indique, qui limite les ressources du réseau et l'accès à l'infrastructure seuls les terminaux conformes,

authentifiés et de confiance, ceci est réalisé en déployant un mot de passe, utilisateur unique, processus d'identification et d'authentification pour accéder au réseau. [4]

Réseaux privés virtuels (VPN)

Un réseau privé virtuel (VPN) est le logiciel le plus simple et le plus efficace pour protéger l'identité de l'utilisateur, il permet un échange sécurisé et anonyme des informations de l'utilisateur en cryptant les données et en masquant son adresse IP et son emplacement, il est utilisé par de nombreuses entreprises pour protéger l'information.
Un VPN protège les utilisateurs contre les pirates qui peuvent voler n'importe quoi, des e-mails et des photos aux numéros de carte de crédit et aux identités des utilisateurs.

Logiciels Antivirus et antimalware

Les logiciels antivirus et antimalware sont deux types de programmes de sécurité conçus pour protéger les ordinateurs et autres appareils numériques contre les logiciels malveillants, mais ils ont des priorités et des capacités légèrement différentes.

- **Antivirus**: est l'un des principaux dispositifs de sécurité pour garantir la protection des données de l'utilisateur et une navigation optimale sur le web. Ce logiciel élimine ou réduit le risque de cyberattaques sur l'ordinateur, le téléphone ou la tablette qui disposent d'un accès à Internet. [7]
- **Antimalware**: Le terme "anti malware" est plus inclusif, indiquant que le logiciel est conçu pour faire face à un spectre plus large de menaces numériques au-delà des seuls virus traditionnels. En conséquence, le logiciel antimalware peut offrir une protection plus complète et être mieux équipé pour détecter et bloquer les formes de logiciels malveillants plus récents et plus complexes.

Détection et prévention des intrusions

Technologie de sécurité utilisée pour surveiller et protéger les réseaux et les systèmes informatiques contre les menaces, les attaques ou l'accès non autorisé. IDPS analyse le trafic réseau, les enregistrements système et d'autres données pour détecter les activités suspectes ou nuisibles.
Le système de gestion du programme comporte deux composantes principales:

- Système de détection des intrusions **(IDS)** : L'élément IDS est chargé de surveiller les activités du réseau ou du système pour identifier les comportements suspects, l'IDS émet des alertes ou des avis pour une enquête plus approfondie.

- Système de prévention des intrusions **(IPS)** : Le composant IPS dépend des capacités IDS en détectant non seulement les activités suspectes, mais aussi en prenant des

mesures immédiates pour prévenir les menaces potentielles. Il peut bloquer ou arrêter le trafic nuisible.

1.4 Système de détection des intrusions (IDS)

1.4.1 Définition

Un système de détection d'intrusion (IDS) est un système de surveillance qui détecte les activités suspectes et génère des alertes lorsqu'elles sont détectées. En fonction de ces alertes, un analyste du centre des opérations de sécurité ou un intervenant en cas d'incident peut enquêter sur le problème et prendre les mesures appropriées pour remédier à la menace.
Les fonctions de l'IDS sont les suivantes: offrir des informations sur les menaces, prendre des mesures correctives lorsqu'il détecte des menaces et enregistrer tous les événements importants au sein d'un réseau. [8]

Des termes utilisés lorsqu'on parle de système de détection d'intrusion :

- Faux positif : Détection d'une activité normale comme une activité suspecte.
- Faux négative : Toute activité suspecte acceptée comme une activité normale.

1.4.2 L'architecture des IDS

Il existe plusieurs outils pour détecter des intrusions, L'IDWG (Intrusion Détection Working Group) de L'IETF (L'Internet Engineering Task Force) a défini un modèle fonctionnel de la détection d'intrusion qui compose de 4 éléments de base.

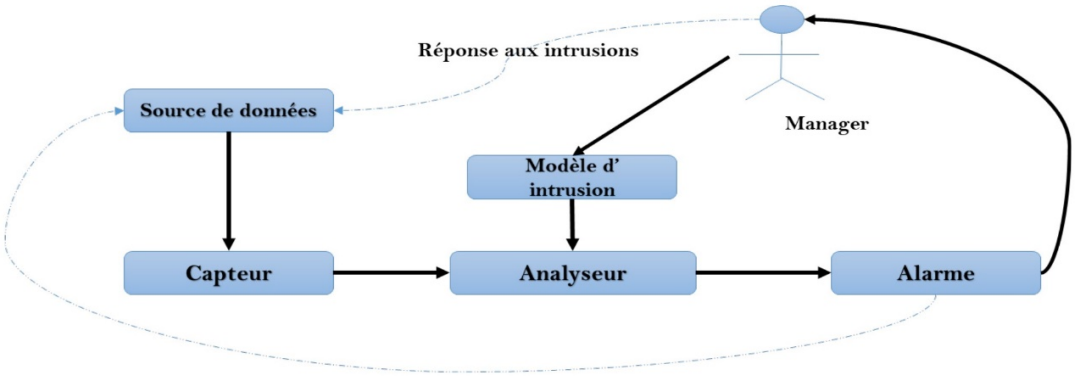


Figure 6: L'architecture d'IDS proposé par L'IDWG.

Source des données : Informations utilisées par le système de détection d'intrusion pour détecter les activités indésirables, ces données provenant plusieurs sources.

Capteur : Un capteur est chargé de collecter des informations sur l'évolution de l'état du système et de fournir une séquence d'événements qui traduit l'évolution de l'état du système.

L'avantage principal des capteurs réseau réside dans leur capacité à surveiller un grand ensemble de machines. Cette caractéristique simplifie le déploiement et la maintenance d'une solution de détection visant à garantir une couverture optimale du réseau surveillé. [9]

Analyseur : l'objectif de l'analyseur est d'analyser les données collectées par le capteur et déterminer si le flux d'événement contient des activités non autorisées, ont été proposé deux approches, l'approche comportementale et l'approche par signature.

Manager : c'est aussi un composant clé permet de collecter les alertes produites par le capteur, les met en forme et les présente à l'opérateur. Éventuellement, le manager est chargé de la réaction à adopter, par exemple :

- Confinement de l'attaque, qui a pour but de limiter les effets de l'attaque.
- Elimination de l'attaque, qui tente d'arrêter l'attaque.
- Diagnostic, qui est la phase d'identification du problème, de ses causes et qui peut éventuellement être suivi d'actions contre l'attaquant (fonction de réaction). [9]

1.4.3 Les différents types des IDS

1.4.3.1 Les systèmes de détection d'intrusion hôte (HIDS)

Ces IDS fonctionnent au niveau des hôtes individuels (ordinateurs ou serveurs) et de ce fait il analyse l'activité se passant sur cette machine pour surveiller les processus malveillants, en plus la visibilité profonde dans les internes de l'hôte. Si le système est compromis par des pirates, HIDS cessera de fonctionner pour prévenir de telles attaques.

Exemples de HIDS

- CrowdSec
- DarkSpy
- IceSword
- Chkrootkit

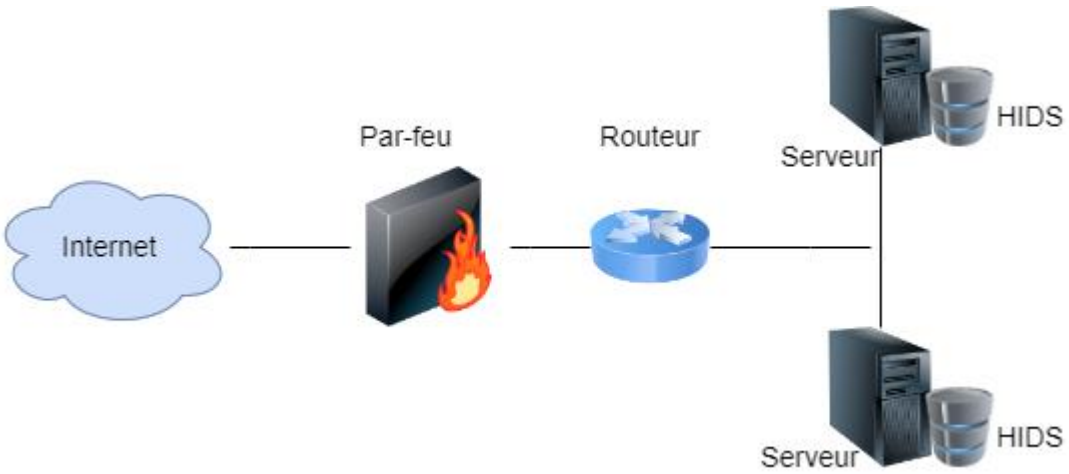


Figure 7: Système de détection d'intrusion hôte(HIDS).

1.4.3.2 Les systèmes de détection d'intrusion réseau (NIDS)

Le système réseau de détection des intrusions (NIDS) surveillent l'état de la sécurité du réseau, est situé sur un réseau isolé qui vérifie continuellement les activités malveillantes en inspectant et en analysant le flux de paquets circulant sur le réseau, bloquant les attaques si nécessaires, et création de rapports. [4]

Exemples de NIDS

- EtRanger
- Dragon
- NFR
- Snort

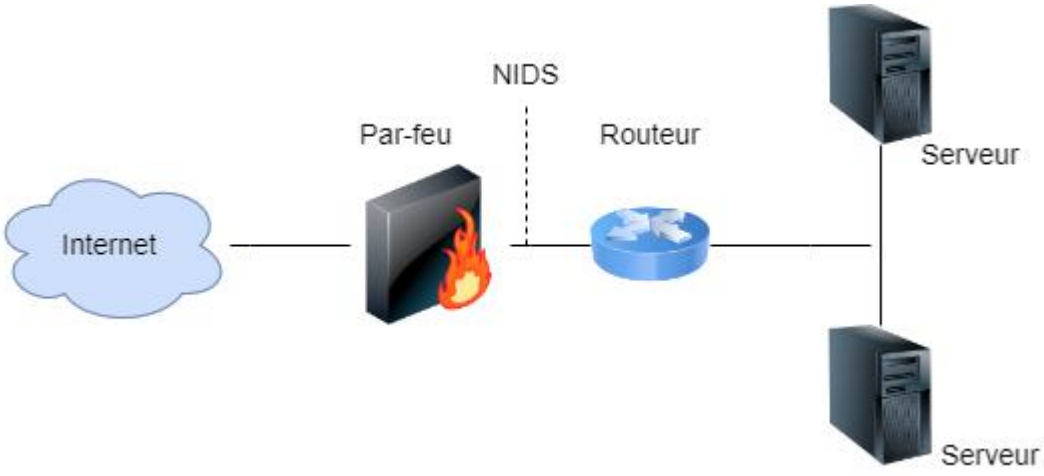


Figure 8: Système de détection d'intrusion réseau(NIDS).

1.4.3.3 Les IDS hybride

Les IDS hybride est un système de détection d'intrusion qui combine plusieurs méthodes de détection pour améliorer la précision de la détection des menaces. Ces méthodes peuvent inclure une détection basée sur les signatures (recherche de modèles connus de code malveillant) et une détection comportementale (surveillance du trafic réseau ou du comportement du système pour détecter des anomalies).

Les IDS hybrides peuvent être utilisés pour collecter les informations en provenance d'un système HIDS et NIDS, d'où l'appellation hybride.

Exemples d'IDS hybride

- OSSIM
- Prelude

Les avantages des IDS hybrides

- Moins de faux positifs.
- Meilleure corrélation.
- Possibilité de réaction sur les analyseurs.

	NIDS	HIDS
Avantage	- Peut surveiller de grands réseaux. - Très sûr des contre-attaques et invisible pour de nombreux attaquants. - Le NIDS présente l'avantage d'être un système temps réel et ont la capacité à détecter les attaques qui ciblent plusieurs machines simultanément.	- Les HIDS peuvent souvent fonctionner dans des environnements avec un trafic réseau crypté. - Peuvent inspecter les données cryptées lorsqu'elles passent sur le réseau. - Fournir des alertes plus pertinentes.
Inconvénient	- Ils ne peuvent donner d'alarme que si le trafic correspond aux règles ou aux signatures préconfigurées. [2] - Faux positive et faux négatif. - Il ne peut pas inspecter le trafic crypté.	- Ils sont assez gourmands en CPU et peuvent parfois altérer les performances de la machine hôte. [2] - Difficile à gérer. - Dépendance aux signatures ou règles préconfigurées.

Tableau 1 : Comparaison entre NIDS et HIDS.

1.4.4 Les méthodes de détection d'intrusion

Afin de détecter un intrus, nous devons utiliser un modèle de détection d'intrusion, qui est un processus essentiel pour protéger les systèmes contre les attaques et les menaces potentielles. La détection d'intrusion peut être classée en deux catégories principales, chacune avec ses propres défauts et avantages.

La première catégorie c'est l'approche par signature est basé sur la recherche des traces d'attaque ou d'intrusion, Quant à la deuxième catégorie, on trouve les méthodes d'analyse qui s'intéressent aux actions autorisées et on parlera donc d'une approche comportementale ou approche par détection d'anomalies.

1.4.4.1 L'approche par signature

La détection d'intrusion basée sur les signatures est une méthode utilisée pour détecter les activités malveillantes ou non autorisées dans un réseau ou sur un système informatique en se basant sur des signatures spécifiques associées à des attaques connues. Cette approche repose sur faire correspondre une vaste collection de modèles connus de données malveillantes aux données stockées sur un système ou en transit sur un réseau.

Voici comment fonctionne la détection d'intrusion basée sur les signatures :

1- Collecte de données : collecte des données réseau ou système, qui peuvent inclure des journaux d'événements, des paquets réseau, des fichiers de configuration, etc.
2- Création de signatures : La création des signatures qui correspondent à des schémas ou à des caractéristiques spécifiques associées à des attaques connues
3- Comparaison avec les signatures : les données collectées sont ensuite comparées à la signature stockée dans la base de données. Si une correspondance est trouvée, cela indique une éventuelle intrusion ou activité malveillante.
4- Alertes : lorsqu'une correspondance est détectée, le système génère une alerte, qui est généralement envoyée à un administrateur de sécurité ou à un système de gestion des événements de sécurité. Les administrateurs peuvent alors prendre des mesures pour enquêter sur l'incident et prendre des mesures correctives.

1.4.4.2 L'approche comportementale

Contrairement à l'approche par scénario, initialement proposée par JP. ANDERSON puis repris et étendus par D.E. DENNING, la méthode comportementale consiste à détecter une intrusion en fonction du comportement passé de l'utilisateur. L'idée sous-jacente est de parvenir à dresser un profil utilisateur établi selon ses habitudes de travail et à déclencher des alertes lorsque des évènements hors gabarit se produisent. [10]

Cette technique peut être appliquée non seulement à des utilisateurs, mais aussi à des applications et services. Plusieurs métriques sont possibles : la charge CPU, le volume de données échangées, le temps de connexion sur des ressources, la répartition statistique des protocoles et applications utilisés, les heures de connexion... [11]
L'avantage majeur de cette technique est la détection d'intrusion inconnue.

	Signature	Comportemetale
Avantages	- Faible coût en temps et en ressources. - Efficace pour détecter des attaques connues. - Faible taux de faux positifs	- La détection d'intrusion inconnue. - Aucune connaissance préalable requise.
Inconvénients	- Ne détecte pas les attaques nouvelles ou modifiées. -Effort considérable pour identifier et examiner les nouveaux logiciels malveillants afin de créer une signature. -les signatures doivent être définies pour toutes les attaques.	- Peuvent produire un grand nombre de faux positifs, Cela peut entraîner une perte de temps et de ressources pour gérer des alertes non pertinentes. - À l'inverse, il existe également des faux négatifs, Cela peut entraîner des vulnérabilités en matière de sécurité du système.

Tableau 2: Les avantages et les inconvénients des méthodes de détection.

1.4.5 Réaction et comportement après une attaque

Il existe deux approches principales pour la réaction d'un IDS (système de détection d'intrusion) : l'approche active et l'approche passive, et le choix dépend des besoins de sécurité spécifiques d'une organisation.

Approche passive : dans cette approche l'IDS se contente de détecter les menaces et informer l'administrateur par une alarme ou un email et il devra alors prendre les mesures nécessaires.

Approche active : l'IDS envoie des alertes et prend des mesures actives pour répondre aux menaces détectées, par exemple réinitialiser la connexion, bloquer du trafic, etc.

1.4.6 Mesures performance des IDS

La mesure de la performance des IDS (Intrusion Detection Systems) est essentielle pour évaluer leur efficacité dans la détection des intrusions dans un réseau.

Matrice de confusion

La matrice de confusion est une représentation tabulaire qui évalue le rendement du modèle de classification pour un problème donné. Il compare les étiquettes de classe prévues du modèle avec les étiquettes de classe réelles de l'ensemble de données de test, résumant les résultats dans un format matriciel.

Pour les problèmes de classification binaire, la matrice de confusion se compose de deux lignes et de deux colonnes qui représentent les classes prévues et réelles (voir la figure 9).
Les lignes indiquent les véritables étiquettes de classe, tandis que les colonnes indiquent la classe prédite.
Chaque cellule de la matrice représente le nombre d'instances qui appartiennent à un combinaison particulière d'étiquettes de classe prévues et réelles. [13]

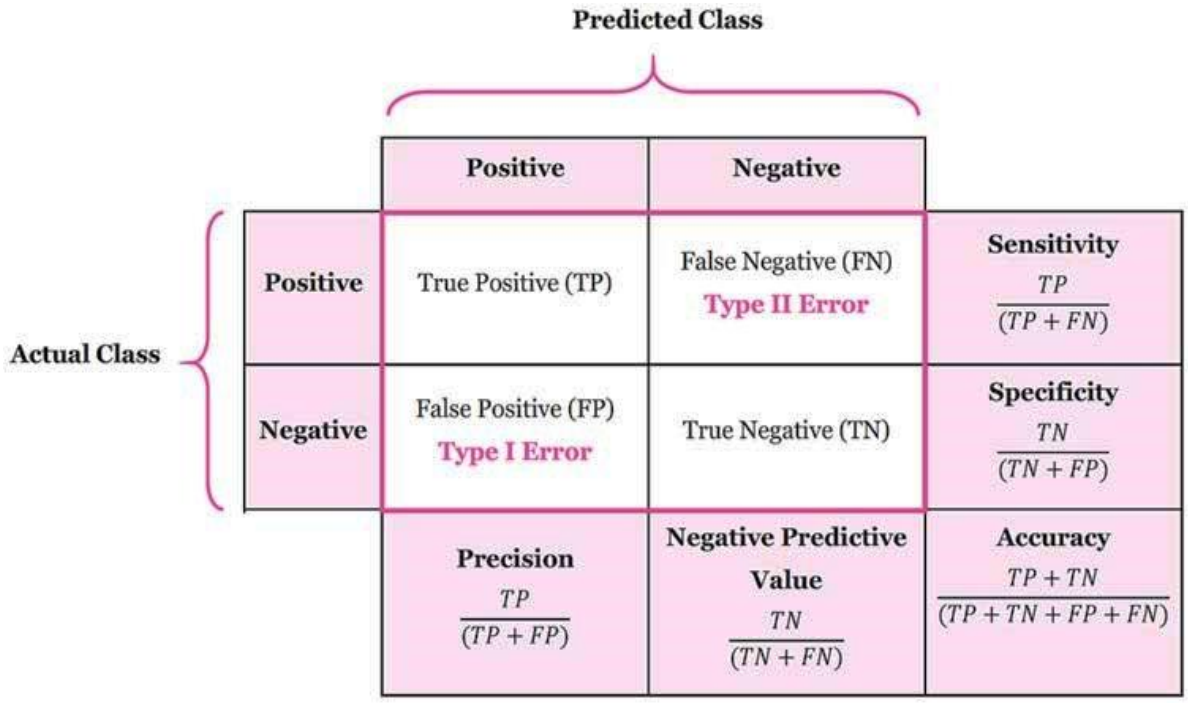


Figure 9: Matrice de confusion [14]

- Vrai positive (TP) : les instances qui sont correctement classés comme positifs par le modèle.
- Vrai négative (TN) : les instances qui sont correctement classés comme négatifs par le modèle.
- Faux positive (FP) : les instances qui sont incorrectement classés comme positifs par le modèle.

- Faux négatif (FN) : les instances qui sont incorrectement classées comme positives par le modèle.

À partir de cette matrice, plusieurs métriques de performance peuvent être calculées pour évaluer la qualité du modèle de classification, telles que :

Exactitude

L'exactitude est l'une des mesures les plus couramment utilisées pour les problèmes de classification. Il mesure le pourcentage de prédictions correctes faites par le modèle. Il est calculé en divisant la somme des prédictions positives et négatives réelles par le nombre total de prédictions effectuées [13]

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

Précision

La précision évalue la proportion de vrais positifs sur toutes les positions prédites elle se calcule en divisant le nombre de vrais positifs par la somme des vrais positifs et des faux positifs. La précision est une métrique pertinente lorsqu'on cherche à réduire les faux positifs.

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

Rappel

Le rappel, également connu sous le nom de sensibilité ou taux positif réel, est une mesure de performance utilisée pour évaluer la capacité d'un modèle à identifier correctement les instances positives. Il mesure la proportion d'instances positives réelles que le modèle classe avec succès comme positives, et il est calculé comme le nombre de vraies prédictions positives divisées par la somme des vraies prédictions positives et fausses négatives. [13]

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

F1 Score

Le score F1 est la moyenne harmonique de précision et de rappel, calculée comme suit :

$$\boldsymbol{F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}}$$

Le score F1 est une mesure appropriée lorsque la précision et le rappel sont tout aussi importants. Il équilibre les deux mesures et fournit une seule mesure du rendement.
Le score F1 est couramment utilisé dans les tâches de récupération d'informations. [13]

ROC et AUC

Le ROC (Receiver Operating Characteristic) est une courbe qui représente la performance d'un modèle de classification binaire à différents seuils de classification.

L'AUC (Area Under the Curve) est une mesure de la performance d'un modèle de classification binaire qui représente la probabilité que le modèle classe correctement un exemple positif aléatoire plus haut qu'un exemple négatif aléatoire. [14]

1.5 Conclusion

Au cours de ce chapitre nous avons fait un survol sur la sécurité et leurs propriétés qui doit assurer à chaque système, ensuite nous avons détaillé le système de détection d'intrusion qui est un sujet principal dans cette mémoire.

Chapitre 02

Apprentissage automatique

2 Chapitre 02 : Apprentissage automatique

2.1 Introduction

L'Intelligence Artificielle est devenue une réalité omniprésente dans notre quotidien. Des algorithmes intelligents façonnent nos expériences en ligne, contribuent à la prise de décision dans divers secteurs, et inspirent une nouvelle ère d'innovation. À la base de cette intelligence informatique se trouvent les concepts plus spécifiques de Machine Learning et de Deep Learning. Dans ce chapitre, nous explorerons les principes de base, les applications concrètes, et les perspectives futures de ces domaines.

2.2 Définitions

2.2.1 Intelligence artificielle

L'intelligence artificielle (IA) se réfère à la conception de systèmes informatiques capables d'accomplir des tâches qui exigent normalement l'intelligence humaine. Ces tâches incluent la résolution de problèmes, l'apprentissage, la compréhension du langage naturel, la reconnaissance de formes et la prise de décisions. Une définition courante de l'IA a été formulée par John McCarthy, l'un des pionniers de ce domaine, dans un atelier de recherche en 1956. Il a défini l'IA comme "le faire par des machines de tout travail qui aurait normalement besoin d'intelligence humaine." [15]

2.2.2 L´apprentissage automatique

L'apprentissage automatique (Machine Learning) est une branche de l'intelligence artificielle qui se concentre sur le développement de modèles et d'algorithmes capables d'apprendre à partir de données et d'améliorer leurs performances au fil du temps sans être explicitement programmés. Une définition générale peut être attribuée à Tom M. Mitchell, qui a déclaré que "Les ordinateurs sont dits apprendre à partir de l'expérience E par rapport à une tâche T et une mesure de performance P, si leur performance à la tâche T, telle que mesurée par P, s'améliore avec l'expérience E." [16]

2.2.3 L´apprentissage profond

L'apprentissage profond, ou Deep Learning, est une branche de l'intelligence artificielle qui repose sur l'utilisation de réseaux de neurones artificiels pour traiter des tâches complexes. Ces réseaux, inspirés du fonctionnement du cerveau humain, sont caractérisés par des architectures en couches, permettant une représentation hiérarchique des données. Un exemple notable d'application réussie de l'apprentissage profond est la victoire du réseau de neurones convolutionnel AlphaGo, développé par DeepMind (une filiale de Google), contre des champions mondiaux au jeu de Go, démontrant ainsi la puissance de cette approche dans la résolution de problèmes complexes. [17]

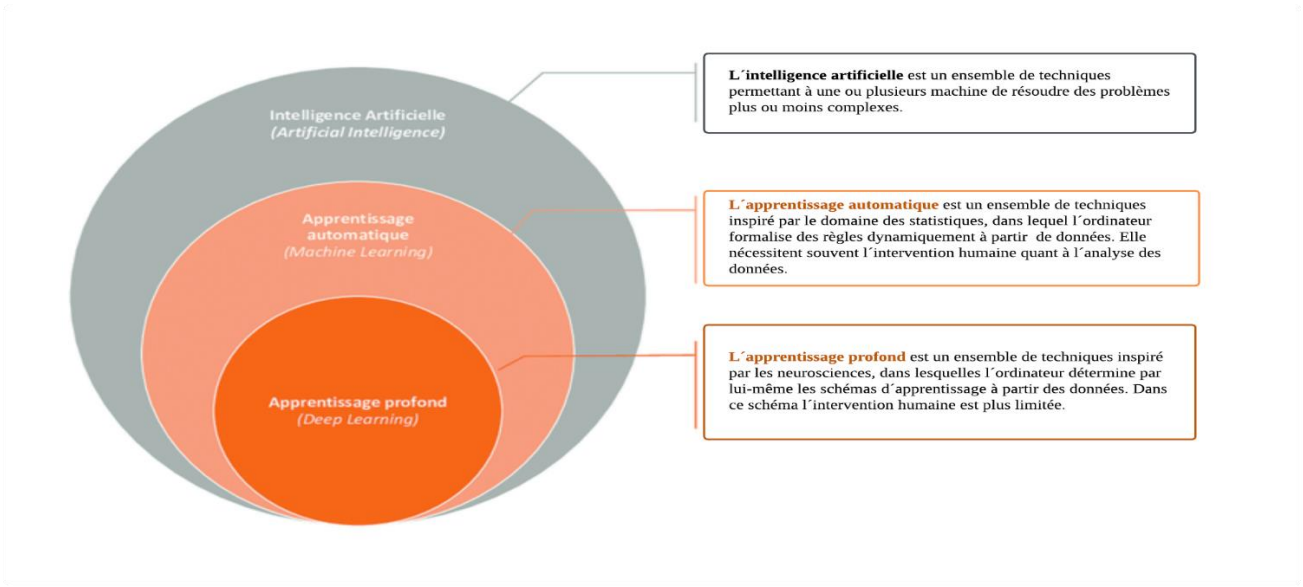


Figure 10: La relation entre l´intelligence artificielle, l´apprentissage automatique et l´apprentissage profond.

2.3 Les types d'apprentissage automatique

Selon Mitchell [18], il existe trois types d'apprentissage automatique :

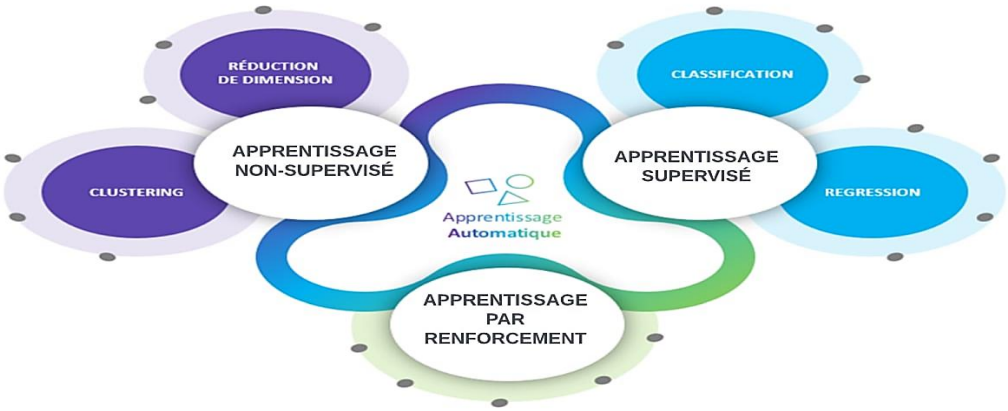


Figure 11: Les types d´apprentissage automatique.

2.3.1 Apprentissage supervisé (Supervised Learning)

L'apprentissage supervisé est un type d'apprentissage automatique où l'algorithme est formé sur un ensemble de données étiqueté. Chaque exemple d'entrée du jeu de données est associé à une sortie désirée. L'objectif est de permettre à l'algorithme de généraliser à de nouvelles données non étiquetées en apprenant des relations entre les entrées et les sorties.

Cette catégorie se divise en deux sous catégories principales soit la classification et la régression :

- **Classification :** La classification est une tâche d'apprentissage supervisé où un modèle est entraîné sur un ensemble de données étiqueté pour prédire la classe ou la catégorie d'une nouvelle observation. [18]
- **Régression :** La régression est une tâche d'apprentissage supervisé où un modèle est formé pour prédire une valeur numérique continue en fonction des caractéristiques d'entrée. [19]

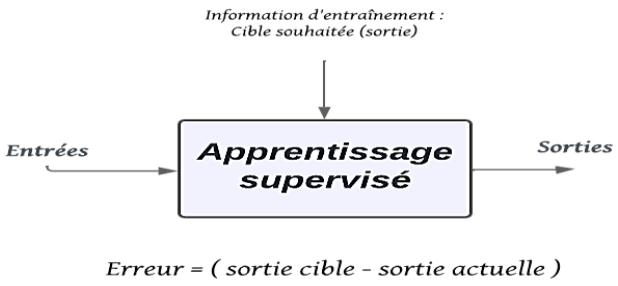


Figure 12: Schéma d'apprentissage supervisé.

2.3.2 Apprentissage non supervisé (Unsupervised Learning)

Définition : L'apprentissage non supervisé implique des données non étiquetées. L'algorithme explore la structure inhérente aux données sans connaître les résultats attendus. Les techniques courantes incluent le regroupement (clustering) et la réduction de dimensionnalité pour découvrir des modèles et des relations intrinsèques. Ce dernier se divise lui aussi en deux principales catégories le regroupement et la réduction de dimension (dimensionality reduction).

- **Clustering :** Le clustering est une tâche d'apprentissage non supervisé qui vise à regrouper des données similaires en clusters distincts en fonction de mesures de similarité. [20]
- **Réduction de dimension :** La réduction de dimension est une technique d'apprentissage non supervisé qui vise à réduire le nombre de variables tout en conservant l'information essentielle. [21]

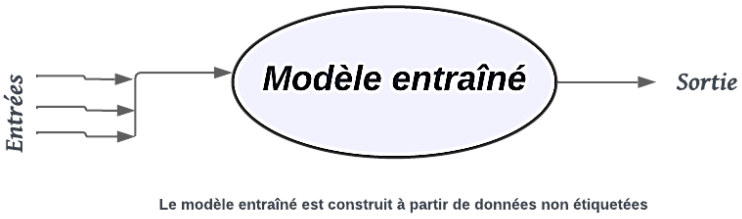


Figure 13: Schéma d'apprentissage non supervisé.

2.3.3 Apprentissage par renforcement (Reinforcement Learning)

Définition : L'apprentissage par renforcement consiste à apprendre par l'interaction avec un environnement dynamique. Un agent prend des décisions successives pour maximiser une récompense cumulative. L'agent apprend par essais et erreurs, ajustant ses actions en fonction des récompenses obtenues.

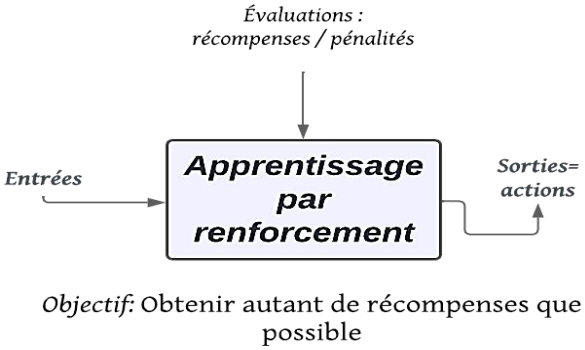


Figure 14: Schéma d'apprentissage par renforcement.

Ces catégories définissent les principaux paradigmes d'apprentissage automatique, chacun avec ses propres caractéristiques et applications spécifiques.

2.4 Les algorithmes de machine Learning

L'univers des algorithmes de Machine Learning constitue le cœur dynamique de l'intelligence artificielle moderne, offrant aux machines la capacité d'apprendre, d'analyser des données et de prendre des décisions autonomes. Ces algorithmes sont les outils fondamentaux qui permettent aux systèmes informatiques de détecter des schémas complexes, de généraliser des informations à partir de données brutes, et d'adapter leurs comportements en fonction de l'expérience.

2.4.1 Algorithmes supervisés

2.4.1.1 Algorithmes de classification

1- Machines à Vecteurs de Support (Support Vector Machines, SVM) :

Description : Les SVM sont des modèles qui cherchent à trouver un hyperplan optimal pour séparer les données en classes. Ils sont particulièrement efficaces dans les espaces de grande dimension.

Les SVM sont largement utilisées pour la classification et la régression, mais elles présentent des avantages et des inconvénients qu'il convient de considérer. [22]

Avantages

- Efficacité dans les espaces de grande dimension.
- Capacité à gérer des données non linéaires.
- Robustesse aux problèmes de surajustement (Overfitting).
- Efficacité en présence de données non équilibrées.

Inconvénients

- Sensibilité à l'échelle des caractéristiques.
- Complexité de l'entraînement sur de grands ensembles de données.
- Choix du noyau.
- Interprétabilité réduite.

Exemple : Classification des tumeurs cérébrales à partir d'IRM.

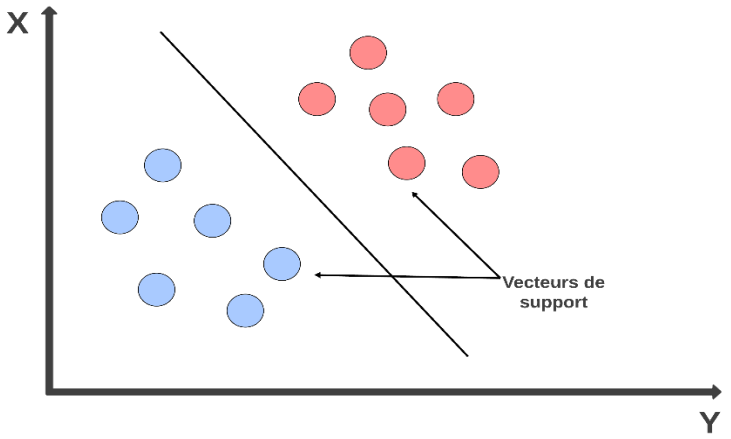


Figure 15: L´algorithme SVM.

2- Forêts d'Arbres de Décision (Random Forests) :

Description : Les Random Forests sont des modèles puissants utilisés pour la classification et la régression, mais elles présentent des caractéristiques distinctes en termes d'avantages et d'inconvénients. [23]

Avantages

- Haute précision.

- Robustesse aux valeurs aberrantes.
- Capacité à gérer des données non linéaires et multidimensionnelles.
- Estimation de l'importance des caractéristiques.

Inconvénients

- Manque d'interprétabilité.
- Taille du modèle.
- Moins efficace sur des ensembles de données très dispersés.
- Peut-être plus lent pour la prédiction individuelle.

Exemple : Classification de la couverture du sol avec Random Forest.

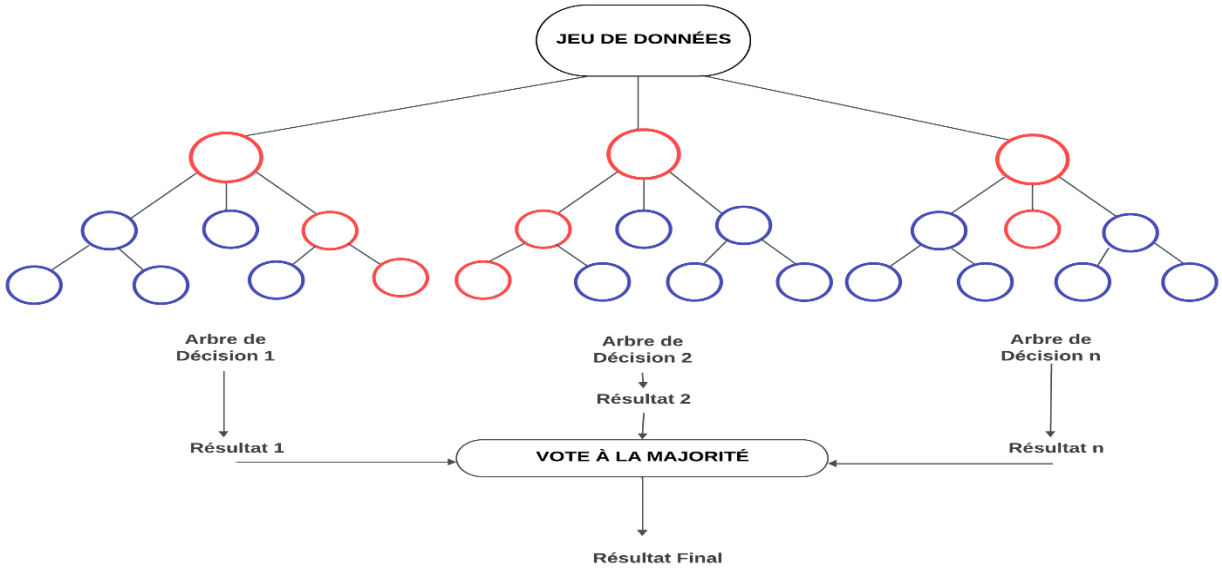


Figure 16: Le diagramme de structure de l'algorithme Random Forest.

3- K-NN (k plus proches voisins) :

Description : L'algorithme k-NN est basé sur le principe que des objets similaires ont tendance à se regrouper dans des espaces similaires. Pour une tâche de classification, il attribue une classe à une observation en se basant sur la classe majoritaire parmi ses k voisins les plus proches dans l'espace des caractéristiques. Pour une tâche de régression, il prédit la valeur en faisant la moyenne des valeurs de ses k voisins les plus proches.

Paramètre k : Il s'agit du nombre de voisins à considérer lors de la prise de décision. La valeur de k est un paramètre ajustable qui influence la sensibilité de l'algorithme aux variations locales et peut affecter ses performances.

Métrique de distance : La mesure de la distance entre les points est cruciale dans k-NN. Les distances euclidiennes ou d'autres métriques peuvent être utilisées selon le contexte de la tâche.

Algorithme KNN

```
// Entrées :
// - X_new : Nouvel exemple à classer
// - X_train : Ensemble de données d'entraînement avec des exemples étiquetés
// - y_train : Étiquettes correspondantes pour chaque exemple dans X_train
// - k : Nombre de voisins à considérer
FONCTION CalculerDistance(X1, X2) :
   // Calcule la distance euclidienne entre deux exemples X1 et X2
   distance = 0
   POUR chaque dimension i de X1 et X2 FAIRE :
      distance += (X1[i] - X2[i])^2
   FIN POUR
   RETOURNER racine carrée de distance
FONCTION KNN(X_new, X_train, y_train, k) :
   // Pour un nouvel exemple, prédit la classe en utilisant l'algorithme KNN
   distances = ListeVide()
   // Calcul des distances entre X_new et chaque exemple de X_train
   POUR chaque exemple (X_train[i], y_train[i]) FAIRE :
      distance = CalculerDistance(X_new, X_train[i])
      AJOUTER (distance, y_train[i]) à distances
   FIN POUR
// Tri des distances par ordre croissant
   distances = TRIER(distances, selon la première composante)
 // Sélection des k plus proches voisins
   k_plus_proches = PREMIERS_K_ELEMENTS(distances, k, selon la première composante
   // Vote majoritaire pour la classification (ou moyenne pondérée pour la régression)
   classe_majoritaire = VOTE_MAJORITAIRE(k_plus_proches)
   RETOURNER classe_majoritaire
// Exemple d'utilisation :
X_new = [valeur1, valeur2, ...]  // Nouvel exemple à classer
resultat = KNN(X_new, X_train, y_train, k)
```

Avantages

- Simplicité.
- Adaptabilité.
- Apprentissage non paramétrique.

Inconvénients

- Sensibilité aux valeurs aberrantes.
- Choix de k crucial.
- Calcul intensif.

Exemple : Recommandation de films avec l'algorithme KNN.

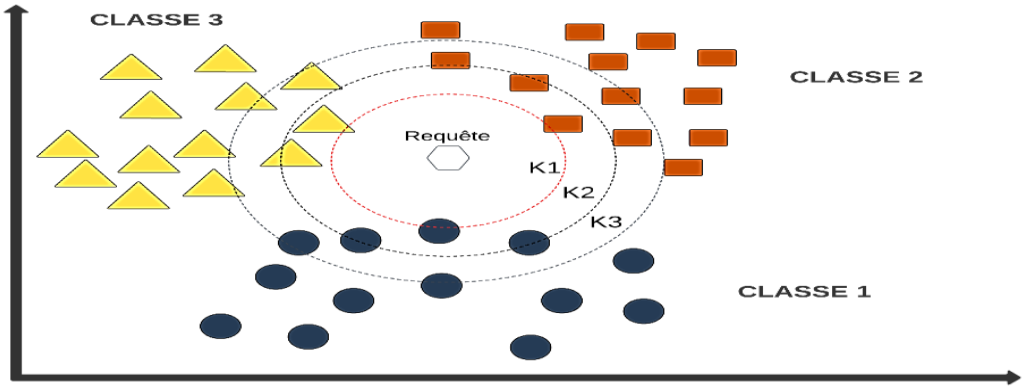


Figure 17: L'algorithme d'apprentissage automatique KNN.

2.4.1.2 Algorithmes de régression

1- Régression linéaire :

Description : Un modèle de régression linéaire cherche à établir une relation linéaire entre les variables d'entrée et la variable de sortie.

Avantages

- Simplicité et interprétabilité : la régression linéaire est simple à comprendre et à interpréter, ce qui facilite la communication des résultats. [24]
- Efficacité en présence de relations linéaires.
- Identification des relations de cause à effet.
- Utilisation prédictive.
- Prise en compte des incertitudes.

Inconvénients

- Sensibilité aux valeurs aberrantes.
- Assumptions strictes.
- Limitation à des relations linéaires.
- Overfitting dans des modèles complexes.
- Dépendance à la qualité des données.

2- Régression logistique

Description: L'algorithme de régression logistique utilise une fonction logistique pour transformer la sortie d'un modèle linéaire en une probabilité comprise entre 0 et 1. Cette fonction logistique prend la forme d'une courbe en forme de S et permet de modéliser la probabilité que la variable dépendante soit égale à 1 (ou à la classe positive dans le cas d'une classification binaire) en fonction des variables indépendantes. L'algorithme est généralement optimisé en maximisant la vraisemblance ou en minimisant une fonction de perte comme l'entropie croisée.

Avantages

- Adaptabilité : La régression logistique est adaptée aux problèmes de classification binaire et peut être étendue à des cas de classification multiclasse.
- Interprétabilité : Les coefficients de régression peuvent être interprétés pour évaluer l'importance des variables indépendantes dans la prédiction de la classe cible.
- Efficacité : L'algorithme est rapide à entraîner et à prédire, ce qui le rend adapté à des ensembles de données de taille moyenne à grande.

Inconvénients

- Linéarité : Comme la régression logistique est basée sur un modèle linéaire, elle peut ne pas être adaptée à des relations complexes entre les variables.
- Sensibilité aux valeurs aberrantes : Les valeurs aberrantes peuvent avoir un impact significatif sur les résultats de la régression logistique.
- Assomption de linéarité : L'algorithme suppose que la relation entre les variables indépendantes et la probabilité de la classe cible est linéaire, ce qui peut ne pas toujours être le cas dans la pratique. [25]

Exemple : Modélisation des valeurs de taille et de poids fournies (régression linéaire versus régression logistique). [26]

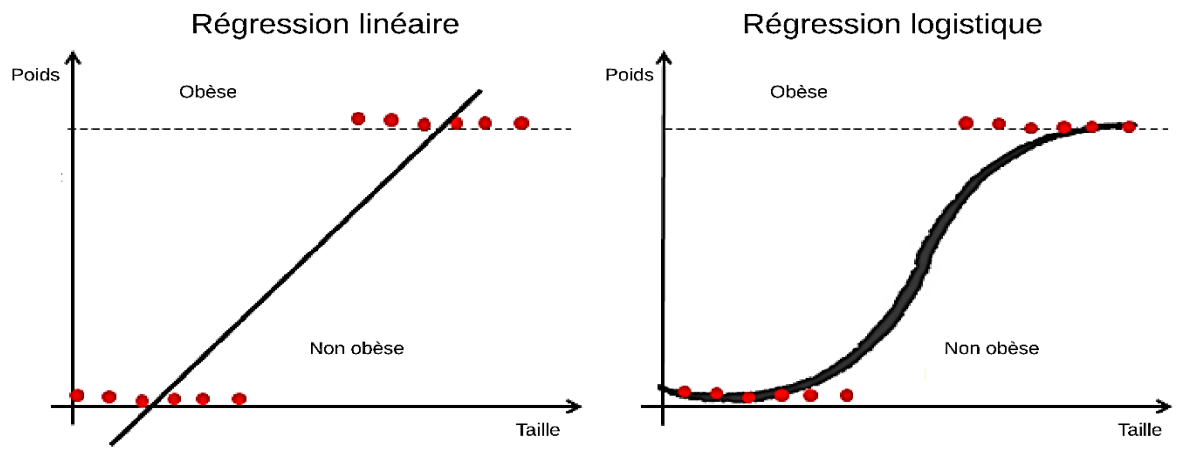


Figure 18: Régression linéaire versus régression logistique. [26]

2.4.2 Les algorithmes non-supervisés

1- K-Means :

Description : Le K-Means est un algorithme de clustering largement utilisé, mais il présente des avantages et des inconvénients qu'il est important de considérer. [27]

Algorithme K-Means

// Entrées : // - X : Ensemble de données à clusteriser // - k : Nombre de clusters	AJOUTER nouveau_centroide à nouveaux_centroides FIN POUR

```
FONCTION InitialiserCentroides(X, k) :
// Choix initial des centroides en sélectionnant k exemples aléatoires de X
centroides = CHOISIR_K_EXEMPLES(X, k)
RETOURNER centroides

FONCTION AffecterCluster(X, centroides) :
// Affecte chaque exemple de X au cluster du centroïde le plus proche
clusters = ListeVide()
POUR chaque exemple x de X FAIRE :
cluster_assigne = ARG_MIN(DISTANCE(x, c) pour c dans centroides)
AJOUTER (x, cluster_assigne) à clusters
FIN POUR
RETOURNER clusters

FONCTION MettreAJourCentroides(clusters) :
// Recalcule les centroides en prenant la moyenne des exemples de chaque cluster
nouveaux_centroides = ListeVide()
POUR chaque cluster dans clusters FAIRE :
nouveau_centroide = MOYENNE(cluster)
RETOURNER nouveaux_centroides
FONCTION KMeans(X, k, max_iterations) :
// Applique l'algorithme k-means pour clusteriser les données X en k clusters
centroides = InitialiserCentroides(X, k)
POUR chaque itération de 1 à max_iterations FAIRE :
clusters = AffecterCluster(X, centroides)
nouveaux_centroides = MettreAJourCentroides(clusters)
SI centroides == nouveaux_centroides ALORS
// Convergence atteinte, terminer l'algorithme
RETOURNER clusters
SINON
centroides = nouveaux_centroides
FIN SI
FIN POUR
RETOURNER clusters

// Exemple d'utilisation :
X = [[valeur1, valeur2, ...], ...] // Ensemble de données à clusteriser
k = nombre_de_clusters
max_iterations = nombre_maximum_iterations
resultat = KMeans(X, k, max_iterations)
```

Pseudo algorithmique. [27]

Avantages

- Simplicité et efficacité.
- Évolutivité.
- Rapidité de convergence.
- Adaptabilité aux formes géométriques.

Inconvénients

- Sensibilité au choix initial des centres.
- Incapacité à gérer des formes de cluster complexes.
- Dépendance à la normalisation.
- Nombre de clusters prédéfini.

Exemple : Segmentation de la clientèle avec K-Means.

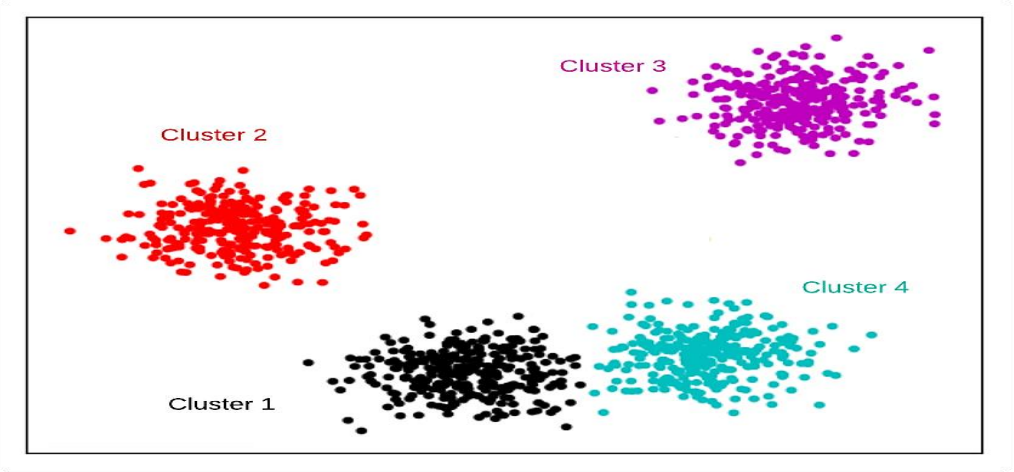


Figure 19: L´algorithme de K-means.

2- Modèle de mélange de Gaussienne (GMM):

Description : Un modèle de mélange de gaussienne GMM sert à estimer para métriquement la distribution des variables aléatoires en les modélisant comme une somme de plusieurs gaussiennes (appelées noyaux). Il s'agit alors de déterminer la variance, la moyenne et l'amplitude de chaque gaussienne. Ces paramètres sont optimisés selon un critère de maximum de Vraisemblance pour approcher le plus possible la distribution recherchée. Cette procédure se fait le plus souvent itérativement via l'algorithme EM (Expectation-Maximization). [28]

Exemple : la segmentation d'image.

2.4.3 Algorithme par renforcement

Q-Learning : est une méthode d'apprentissage par renforcement qui vise à apprendre une politique d'action optimale pour un agent interagissant avec un environnement. L'algorithme attribue des valeurs (appelées Q-values) à chaque paire état-action et ajuste ces valeurs à mesure que l'agent explore l'environnement pour maximiser les récompenses à long terme.

Description : Le Q-learning est un algorithme d'apprentissage monoagent, qui peut être utilisé en environnement multiagent, mais sans prendre en compte explicitement la présence des autres agents.

Exemple : Q-learning peut être appliqué pour optimiser les décisions de routage dans les réseaux de communication, améliorant ainsi l'efficacité et la qualité du service.

2.5 L´apprentissage en profondeur

2.5.1 Définition

Le Deep learning fait l'objet d'importants investissements privés, notamment de la part des grands acteurs du net, mais aussi publics. « De plus en plus d'entreprises ont des masses de données gigantesques à exploiter, trier, indexer, et cela demande des ressources considérables.

L'intelligence artificielle et le Deep learning peuvent aider à le faire de façon automatisée et plus efficace », confirme Yann LeCun qui reste prudent quant aux fantasmes que suscitent ces développements. « De grands progrès ont été fait notamment en matière de reconnaissance visuelle et vocale - dans la reconnaissance automatique d'images, des réseaux neuronaux artificiels ont produit des algorithmes meilleurs que ceux conçus par des ingénieurs humains. [29]
Cette approche a connu un grand succès dans divers domaines tels que la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale, et d'autres applications où la complexité des données requiert une capacité de modélisation plus sophistiquée.

2.5.2 Les algorithmes d´apprentissage en profondeur

En Deep Learning, comme dans l'apprentissage machine en général, il existe des approches supervisées et non supervisées. Voici quelques exemples d'algorithmes pour chacune de ces catégories :

2.5.2.1 Algorithmes Supervisé

1- Réseaux de neurones convolutifs (CNN) :
Description: Les réseaux de neurones convolutifs (CNN) sont des architectures puissantes largement utilisées pour la vision par ordinateur, mais comme toute méthode, ils présentent des avantages et des inconvénients. [30]

Avantages

- Extraction de caractéristiques automatique.
- Translation invariante.
- Capacité à gérer des données complexes.
- Performances élevées en classification d'images.

Inconvénients

- Requiert des quantités importantes de données.
- Calcul intensif.
- Manque d'interprétabilité.
- Risque de surajustement (Overfitting).

Exemple : Classification objets capturés par des caméras de sécurité.

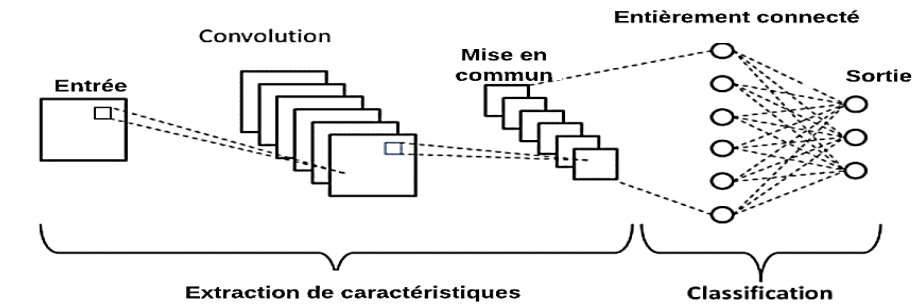


Figure 20: L'architecture de base d'un réseau de neurones convolutionnel (CNN).

2- Réseaux de neurones récurents (RNN) :

Description *:* Les réseaux de neurones récurrents (RNN) sont des architectures de réseaux de neurones conçues pour traiter des séquences de données.

Avantages

- Traitement de séquences.
- Partage de paramètres.
- Capacité à gérer des entrées de taille variable.
- Mémoire à court terme.

Inconvénients

- Problème de disparition/explosion du gradient.
- Calculs séquentiels.
- Manque de capturation de contexte global.
- Complexité d'entraînement.

Exemple : Modèle de langage avec réseaux de neurones récurrents (RNN).

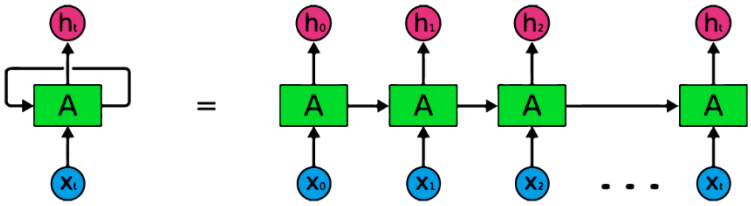


Figure 21: Un simple réseau neuronal récurrent. [31]

2.5.2.2 Algorithmes non supervisés

1- Réseaux de neurones autoencodeurs :

Description : Les autoencodeurs sont des modèles de réseaux de neurones utilisés pour la réduction de dimension et l'apprentissage de représentations latentes.

Avantages

- Réduction de dimension.
- Apprentissage non supervisé.
- Génération de données.
- Détecteur d'anomalies.

Inconvénients

- Apprentissage difficile.
- Interprétabilité limitée.
- Sensibilité aux données bruitées.
- Choix de l'architecture.

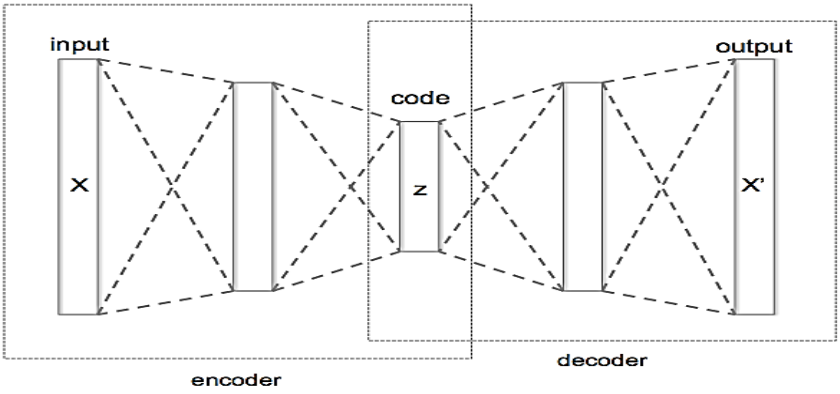


Figure 22: L´algorithme RNAe.[32]

2- Réseaux de neurones génératifs (GAN) :

Description : Les réseaux générateurs antagonistes (GAN) sont des modèles de réseaux de neurones utilisés pour générer de nouvelles données, mais ils ont également leurs avantages et inconvénients.

Avantages

- Génération de données réalistes.
- Apprentissage non supervisé.
- Adaptabilité à diverses tâches.
- Apprentissage de représentations utiles.

Inconvénients

- Instabilité de l'entraînement.
- Mode collapse.
- Évaluation difficile.

- Sensibilité aux données d'entraînement.

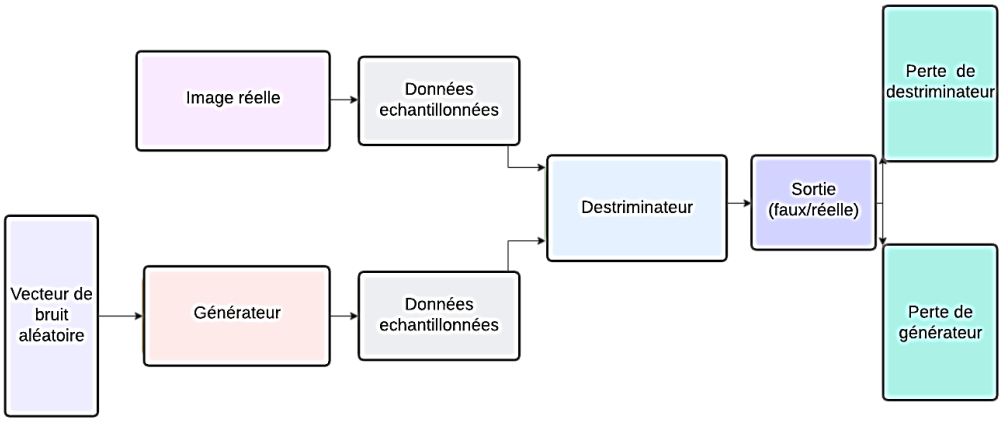


Figure 23: Flux de travail d'un réseaux génératifs antagonistes (GAN).

3- Méthodes de clustering basées sur le Deep Learning :

Description *:* Les méthodes de clustering basées sur le deep learning utilisent des architectures de réseaux de neurones pour regrouper automatiquement des données. Bien que ces approches soient encore en évolution, elles présentent certaines caractéristiques. [33]

Avantages

- Représentations latentes complexes.
- Adaptabilité à des données non linéaires.

Inconvénients

- Besoin de grandes quantités de données.
- Complexité de l'entraînement.
- Interprétabilité limitée.
- Sensibilité aux hyperparamètres.

Bien que les méthodes de clustering basées sur le deep learning offrent des avantages en termes de capture de complexités, il est important de considérer les défis liés à la nécessité de données massives et à la complexité de l'entraînement.
L'utilisation d'approches supervisées ou non supervisées dépend largement de la nature des données et des objectifs de la tâche.

2.6 Conclusion

En conclusion, Le machine learning et le deep learning sont étroitement liés, offrant un potentiel innovant. Le premier, par son adaptabilité, et le second, par sa capacité à comprendre les données complexes, ouvrent la voie à des applications révolutionnaires.

Le prochain chapitre nous plongera dans l'univers des graphes et des graphes de connaissance. Ces structures complexes nous permettront une compréhension plus nuancée et contextuelle de la connaissance.

Chapitre 03

Graphes de connaissance

3 Chapitre 03 : Graphes de connaissances

3.1 Introduction

Au cours de la dernière décennie, l'utilisation massive des données graphiques a compliqué leur analyse. Les graphes de connaissances émergent comme une solution majeure pour démystifier cette complexité. Ils simplifient l'extraction de connaissances à partir de vastes ensembles de données en proposant une méthode structurée pour les organiser, les interpréter et en tirer des conclusions significatives. Ce chapitre se penche sur les bases des graphes de connaissances et explore leur lien avec l'analytique et la science des données.

3.2 Les graphes

3.2.1 Définition

Un graphe est une structure mathématique composée d'un ensemble de sommets ou de nœuds, et d'un ensemble d'arêtes ou d'arc reliant ces sommets. Les arêtes peuvent être orientées ou non orientées, et elles représentent généralement des relations ou des connexions entre les sommets. Les graphes sont largement utilisés pour modéliser des relations entre des entités dans divers domaines tels que les réseaux sociaux, la logistique, la biologie, l'informatique, etc. [34]

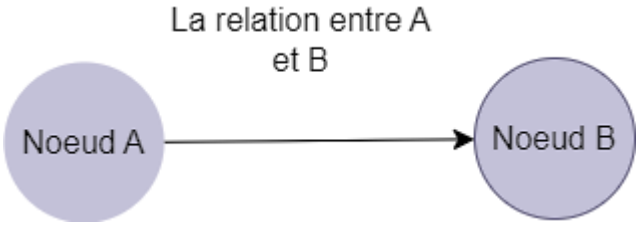


Figure 24: Exemple d'un graphe simple

3.2.2 Les concepts fondamentaux d'un graphe

1. **Sommets (ou nœuds)** : Les points fondamentaux d'un graphe, représentant des entités distinctes telles que des objets, des lieux ou des événements.
2. **Arêtes (ou arcs)** : Les connexions entre les sommets, indiquant des relations ou des liens entre les entités représentées par ces sommets.
3. **Orientation :** La propriété d'un graphe où les arêtes peuvent avoir une direction spécifique, indiquant une relation asymétrique, ou être non orientées, indiquant une relation symétrique.
4. **Graphes pondérés** : Des graphes dans lesquels des valeurs numériques, appelées poids, sont associées aux arêtes pour représenter des coûts, des distances, ou d'autres mesures liées à la relation entre les sommets.
5. **Graphes connexes** : Des graphes dans lesquels, il existe un chemin entre chaque paire de sommets, assurant une connexion globale dans la structure du graphe.

6. **L'ontologie :** Les ontologies fournissent une explication détaillée du contenu, des propriétés essentielles et des relations entre les termes au sein d'une base de connaissance.
7. **Intégration de graphe** : Cette technique de transformation de graphes offre une solution efficace pour analyser les graphes en les représentant dans un espace de dimensions réduites tout en préservant leurs informations cruciales. Elle permet de générer des vecteurs de faible dimension qui capturent les caractéristiques des graphes, facilitant ainsi des tâches telles que la classification des nœuds, la prédiction des liens et la détection de communautés. [60]

3.2.3 Définir l'analytique de graphe et la science des données de graphe

La modélisation des graphes ne représente qu'une partie de l'histoire. On les analyse pour révéler des insights qui ne sont pas immédiatement évidents.

La science des données de graphe est une approche basée sur la science pour acquérir des connaissances à partir des relations et des structures dans les données, généralement dans le but d'alimenter des prédictions. Elle utilise des workflows multidisciplinaires qui peuvent inclure des requêtes, des statistiques, des algorithmes et de l'apprentissage automatique. La science des données de graphe peut généralement être décomposée en trois domaines [35] :

- Les statistiques de graphe fournissent des mesures de base sur un graphe, telles que le nombre de nœuds et la distribution des relations. Ces insights peuvent influencer la configuration et l'exécution d'analyses plus complexes, ainsi que l'interprétation des résultats.
- Les analyses de graphe s'appuient sur les statistiques de graphe en répondant à des questions spécifiques et en tirant des enseignements des connexions dans les données existantes ou historiques. Les requêtes et les algorithmes de graphe sont généralement appliqués ensemble dans des "recettes" pendant les analyses de graphe, et les résultats sont directement utilisés pour l'analyse.
- L'apprentissage automatique (ML) et l'intelligence artificielle (IA) améliorés par le graphe consistent en l'application des données de graphe et des résultats d'analyse de graphe pour entraîner des modèles ML ou soutenir des décisions probabilistes au sein d'un système d'IA.

Les statistiques et les analyses de graphe sont souvent utilisées de concert pour répondre à certains types de questions sur des systèmes complexes, et les enseignements qui en résultent sont ensuite appliqués pour améliorer l'apprentissage automatique.

3.2.4 L'émergence de la science des données de graphe

La capacité accrue à effectuer des calculs sur d'énormes ensembles de données de graphes, et d'une prise de conscience du pouvoir des graphes pour déduire du sens et améliorer les prévisions. Les chercheurs jouent un rôle essentiel dans le développement de cette prise de conscience et préconisent les meilleures techniques. À mesure que les scientifiques des données reconnaissent la puissance de l'information structurelle, ils intègrent de plus en plus

les graphes dans leurs pratiques de statistiques, d'analyses et d'apprentissage automatique. En effet, l'utilisation de la technologie des graphes dans la recherche en intelligence artificielle s'accélère. [35]

3.3 Les graphes de connaissances

3.3.1 Définition

Sont la pierre angulaire de la science des données de graphe et offrent une manière de rationaliser les flux de travail, d'automatiser les réponses et de mettre à l'échelle les décisions intelligentes. À un niveau élevé, les graphes de connaissances sont des ensembles interconnectés de points de données décrivant des entités du monde réel, des faits ou des éléments et leurs relations entre eux sous une forme compréhensible par les humains. Contrairement à une base de connaissances simple avec des structures plates et un contenu statique, un graphe de connaissances acquiert et intègre des informations adjacentes en utilisant les relations de données pour déduire de nouvelles connaissances.
En tant que première phase de la science des données de graphe, les graphes de connaissances sont souvent mis en œuvre pour rassembler des informations diverses afin d'aider les experts du domaine à trouver du contenu connexe et à explorer les connexions dans leurs données. Les graphes de connaissances peuvent également ajouter du contexte à des applications, telles que celles dans les systèmes d'intelligence artificielle (IA), afin qu'elles puissent prendre des décisions approximatives meilleures et plus rapides. Cette approche est utilisée dans des systèmes d'IA tels que les chatbots qui utilisent un graphe de connaissances.

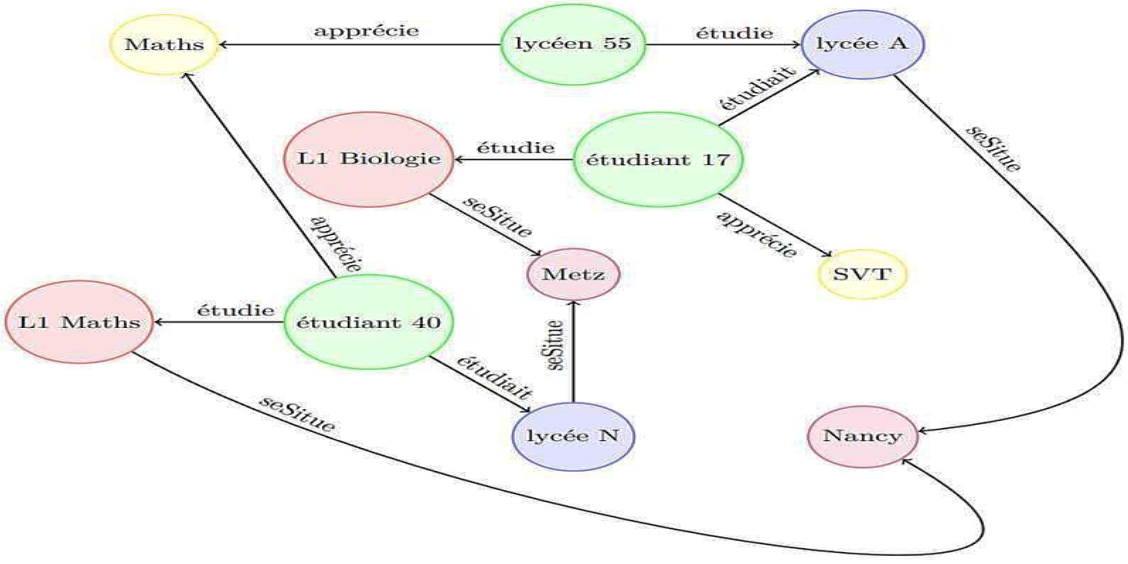


Figure 25: Exemple d'un graphe de connaissance. [36]

3.3.2 Historique

Le terme de « graphe de connaissance » existe depuis des décennies mais son utilisation par Google en 2012 pour un nouveau service, puis par un nombre grandissant d'autres entreprises,

l'ont rendu extrêmement populaire dernièrement. De plus, son couplage avec différentes techniques d'intelligence artificielle contribue à en faire un sujet d'intérêt d'actualité. Si, à l'instar de cette expression « intelligence artificielle », le terme « graphe de connaissance » ou Knowledge Graph est utilisé avec différentes acceptions et identifie actuellement une ressource numérique très différente d'un cas d'usage à un autre, le domaine de la représentation des connaissances à base de graphes existe depuis longtemps et étudie l'expressivité de ces modèles et la complexité de leurs traitements avec des interactions multidisciplinaires et des applications dans de nombreux domaines. [37]

3.3.3 L'architecture de graphe de connaissance

Généralement, la structure des graphes de connaissances peut être conceptualisée conformément à ce qui est illustré dans la figure 26.
La section encadrée en pointillés représente le processus de création du graphe de connaissances, englobant simultanément la phase de mise à jour de KG, Cet architecture se concentrera sur les quatre processus de construction du graphe de connaissances.

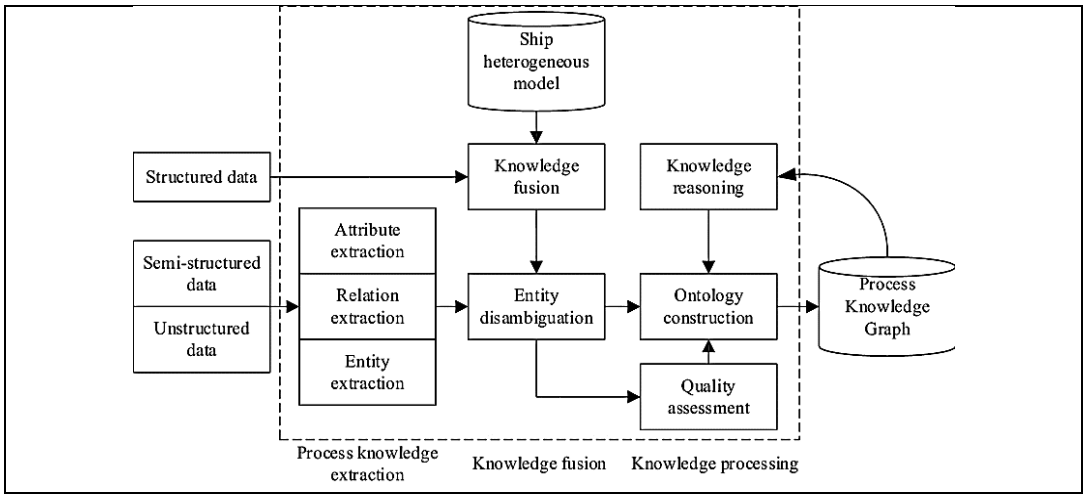


Figure 26: l'architecture des graphes de connaissance [38]

- **Technologie de représentation des connaissances**

La représentation des connaissances constitue le fondement de la construction et de l'application de la cartographie des connaissances. Elle a été largement utilisée dans le traitement du langage naturel et la reconnaissance d'images. Cependant, la représentation des connaissances basée sur les triplets ne peut pas entièrement et complètement rendre compte des relations sémantiques entre les entités. [38]

- **Technologie d'extraction d'informations**

La technologie d'extraction d'informations est cruciale pour la construction des graphes de connaissances, qui puisent leurs sources dans des données telles que le texte, les images, les

capteurs et les vidéos. L'essentiel réside dans la manière d'extraire les entités, attributs et relations nécessaires à partir de différentes sources de données. Plus les connaissances extraites sont complètes, plus les graphes de connaissances construits seront exhaustifs. L'objectif est d'extraire des entités, des attributs et des relations entre différentes entités à partir de diverses sources de données.

- **Technologie de fusion des connaissances**

La fusion des connaissances tout comme l'intégration ontologique, lors de l'extraction de connaissances à partir d'un graphique des connaissances les sources de données peuvent être variées entraînant parfois une disparité de qualité dans les connaissances recueillies. Afin de remédier à cette situation, la fusion des connaissances intervient pour éliminer ces ambiguïtés et de réaliser la fusion des données, des informations, des méthodes, de l'expérience et des pensées humaines et de former une base de connaissances de haute qualité.

- **Technologie de raisonnement des connaissances**

Il constitue un moyen essentiel et un lien clé dans la construction du graphe de connaissances et permet de découvrir de nouvelles connaissances à partir du des connaissances existant, contribuant ainsi à enrichir et améliorer le graphe de connaissances.

3.3.4 Points forts des graphes de connaissance

- Le graphe de connaissances utilise des entités, des relations et des attributs pour organiser les informations de manière structurée ce qui permet une compréhension plus approfondie du contenu.
- Le graphe de connaissances sert de base de données pour les systèmes d'intelligence artificielle, fournissant une source de données structurées pour la formation et l'amélioration des modèles.
- Des coûts de mise en œuvre et de maintenance significativement plus bas que sur des bases de données relationnelles. [39]
- Les graphiques de connaissances sont conçus pour être extensibles, ce qui signifie qu'ils peuvent être mis à jour et étendus au fil du temps pour intégrer de nouvelles connaissances.
- La capacité de fusionner et relier des données variées issues de différentes sources.

3.3.5 La construction des graphes de connaissance

La construction d'un graphe de connaissance se fait en deux étapes :

3.3.5.1 Collection et l'extraction de l'information

Cette partie extrait des instances de connaissances à partir de ressources de connaissances. Après la fusion des connaissances des instances peuplées, les ontologies de niveau supérieur sont construites au moyen d'instances de connaissances pour créer l'ensemble des KG.

Une définition en est donnée par Wilks (1997) comme suit : « étant donné un ensemble de traits structuraux représenté par des formes graphiques, comment l'organiser de façon à ce qu'il s'ajuste le mieux au texte » [40] L'objectif est d'extraire les informations nécessaires à la construction d'un graphe à partir de textes non structurés. Le traitement du langage naturel (NLP : Natural Language Processing) est utilisé pour analyser les sources textuelles et générer des graphes de connaissances. Trois tâches NLP sont particulièrement pertinentes pour la construction d'un graphe de connaissances : l'extraction d'entités, l'extraction de relations et la résolution d'entités.

- **L'extraction d'entités :** cette tâche est essentielle pour identifier les éléments clés du texte pouvant être utilisés comme nœuds dans les graphiques de connaissances. (Par exemples noms de personnes, lieux, organisations, dates, etc.)
- **L'extraction de relations :** une fois les entités extraites, les relations d'extraction aident à déterminer comment ces entités sont liée Les relations et propriétés extraites deviendront généralement des arêtes du graphe de connaissance.
- **La résolution d'entités :** consiste à déterminer si plusieurs mentions dans le texte font référence à la même entité.

3.3.5.2 Vérification et inférence

La dernière étape de la construction d'un graphe de connaissances est d'utiliser l'apprentissage automatique, consistant à déduire de nouvelles relations entre les nœuds en fonction des relations qui existent déjà dans le graphe.
La création de graphes de connaissance est complexe et présente quelques challenges :

- La qualité des données, pour garantir la qualité d'un graphe de connaissance il faut:
  - Assurer la mise à jour permanente des données.
  - Assurer que les données sont correctes.
  - Assurer que les données sont complètes et couvrent suffisamment le concept que l'on veut ajouter au graphe (problèmes de relations manquantes et nœuds manquants).
- La vérification et l'enrichissement
  - La vérification est difficile à réaliser manuellement à grande échelle.
  - Il faut pouvoir détecter les doublons.
  - Il faut pouvoir gérer les conflits.
- L'établissement de contraintes sur les relations. [41]

3.3.6 Modèles de graphes de connaissances

3.3.6.1 Resource Description Framework (RDF)

Resource Description Framework (RDF) est un cadre ou un modèle de données pour présenter les données sous forme de graphiques, et a été initialement développé pour décrire les métadonnées des ressources Web (à savoir le Web sémantique). Aujourd'hui, le W3C propose de nombreuses technologies autour de RDF qui aident à construire et à utiliser des graphiques de connaissances soit dans le cadre du Linked Data Cloud, soit dans un

environnement encapsulé. Les KGs sont représentés par un ensemble de triplets <sujet, prédicat, objet> qui représentent uniformément des relations nommées (prédicats) d'entités (sujets) pour attribuer des valeurs (objet). [42]

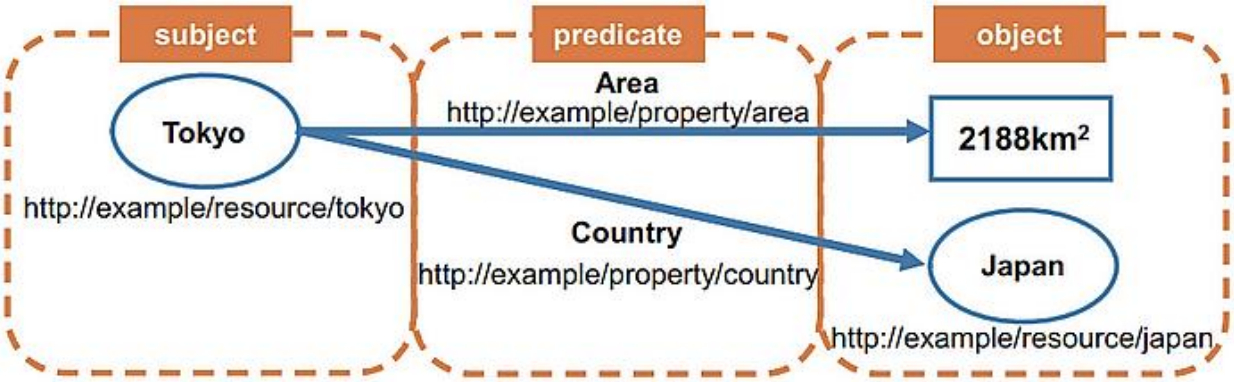


Figure 27: Graphique représentant RDF [43]

Dans cet exemple, nous pouvons voir que Tokyo est située dans le pays Japon et a une superficie de 2188 km². Les entités (Tokyo et Japon) et les prédicats ont chacun des URI pour les identifier dans un format lisible par machine. L'aire dans le rectangle n'a cependant pas d'importance puisqu'il s'agit de ce qu'on appelle un littéral (une valeur constante). [43]

3.3.6.2 Modèle de données de graphe de propriétés (PGM)

Un modèle de données de graphe de propriétés se compose de nœuds, de relations et de propriétés. Chaque nœud a une étiquette et un ensemble de propriétés sous la forme de paires clé-valeur arbitraires. Les clés sont des chaînes et les valeurs sont des types de données arbitraires. Une relation est une arête dirigée entre deux nœuds, a une étiquette et peut avoir un ensemble de propriétés. [44]

3.3.7 Les algorithmes de graphe de connaissance

On examine brièvement quelques algorithmes graphiques populaires qui sont pertinents et applicables aux graphes de connaissances :

3.3.7.1 Algorithmes de centralité

L'analyse de centralité vise à identifier les nœuds ou bords les plus importants « centraux » d'un graphique. Les mesures spécifiques de centralité des nœuds comprennent le degré, l'intermédiation, la proximité, le vecteur propre, le PageRank et autre.
La centralité peut également être appliquée aux bords. Une mesure de centralité des nœuds permettrait, par exemple, de prédire les centres de transport les plus fréquentés, alors que la centralité nous permettrait de trouver les bords sur lesquels de nombreux itinéraires les plus courts dépendent pour prédire le trafic. [45]

3.3.7.2 Algorithmes de détection de communauté

La détection des communautés vise à identifier des sous-graphes qui sont plus étroitement connectés à l'interne qu'au reste du graphe. Les algorithmes de détection de la communauté incluent les algorithmes de coupe minimale, la propagation des étiquettes, la modularité Louvain, etc. On essaye de prendre cet algorithme dans notre cas pour détection la communauté des attaques (Louvain).

3.3.7.3 Algorithmes de connectivité

Permet d'estimer dans quelle mesure le graphique est connecté et résilient. Les techniques spécifiques comprennent la mesure de la densité du graphique, la détection de composants fortement connectés et de composants faiblement connectés.

3.3.7.4 Algorithme de similarité

Dans le contexte des graphes de connaissances (KGs), les algorithmes de similarité sont des méthodes utilisées pour mesurer la similitude entre deux entités (nœuds) du graphe. Ces algorithmes sont importants pour de nombreuses tâches, telles que la recommandation d'entités similaires, la détection de similarité entre des concepts ou des entités, ou encore la recherche d'entités liées.

3.4 Conclusion

Depuis que Google a introduit le concept de graphique des connaissances, sa popularité a augmenté. Les graphes de connaissances sont donc des ressources numériques, destinés à accumuler et à transmettre des connaissances, dont les sommets représentent des entités d'intérêt et dont les arêtes représentent leurs relations, dans ce chapitre nous avons introduit la notion de graphe et ses concepts fondamentales. Après, nous avons détaillé les graphes de connaissance (KG).

Le champ de l'intelligence artificielle (IA) connaît une progression rapide, avec les organismes spécialisés en science des données reconnaissant les graphes de connaissances comme une compétence essentielle pour la réussite des projets liés à l'IA, Et voici ce dont nous parlerons au chapitre prochain.

Chapitre 04

L'intégration des graphes de connaissances avec l'intelligence artificielle

4 Chapitre 04 : L'intégration des graphes de connaissances avec l'intelligence artificielle

4.1 Introduction

L'apprentissage automatique sur les graphes est une approche qui a gagné en importance ces dernières années. Il se distingue par sa capacité à extraire des informations utiles à partir de données complexes organisées sous forme de graphes. Cette méthode se révèle particulièrement efficace dans divers domaines, y compris la cybersécurité. En combinant les potentialités des graphes de connaissance avec les techniques d'apprentissage automatique, les systèmes de détection d'intrusion peuvent ainsi améliorer leur capacité à sécuriser les réseaux vitaux face à une multitude de menaces cybernétiques.

4.2 La valeur de la combinaison des graphes de connaissances avec l'IA

La combinaison des graphes de connaissances et la technologie d'apprentissage automatique peut améliorer la précision des résultats et augmenter le potentiel des approches d'apprentissage automatique. Selon Stephanie Simone, les graphes de connaissances, les modèles linguistiques d'IA sont capables de représenter les relations et la signification précise des données plutôt que de simplement générer des mots en fonction de modèles. Cela permet à l'IA d'être un partenaire plus fiable lors de nos recherches sur le web. [46]

4.3 La convergence de l'IA et des graphes de connaissances

Les entreprises utilisent de plus en plus des applications d'IA pour la prise de décision. Cependant, en raison du manque d'informations contextuelles, les systèmes d'IA n'ont pas encore pu atteindre leur plein potentiel en tant que solutions fiables pour des problèmes complexes.

Les graphes de connaissances insufflent de l'intelligence dans les données elles-mêmes et fournissent à l'IA le contexte nécessaire pour être plus explicative, précise et reproductible.

Ni l'IA ni les graphes de connaissances ne sont de nouvelles technologies, mais récemment, elles ont atteint leur maturité et ont uni leurs forces. Bien que les données et la puissance de calcul aient contribué à leur essor au cours de la dernière décennie, c'est la puissante combinaison des deux qui suscite une explosion d'intérêt pour l'IA contextuelle. [46]

4.4 Infusion d'intelligence dans les données via l'utilisation de graphes de connaissances

Les graphes de connaissances mettent explicitement en évidence les relations riches entre les données que les experts du domaine expérimentés considèrent naturellement. En réalité, il n'y a pas de données isolées, mais seulement des domaines riches et connectés tout autour de

nous. Un graphe de connaissances replace les données dans un contexte en établissant des liens entre les données.
Ensuite, un graphe de connaissances enrichit la signification et l'utilité des données en ajoutant une couche de sémantique, permettant ainsi aux agents logiciels de raisonner à leur sujet. En ajoutant des relations aux données et en les enrichissant de sémantique, les graphes de connaissances insufflent de l'intelligence dans les données, les rendant plus intelligentes.
Les graphes de connaissances améliorent l'apprentissage automatique de la collecte à l'entraînement jusqu'aux prédictions. [46]

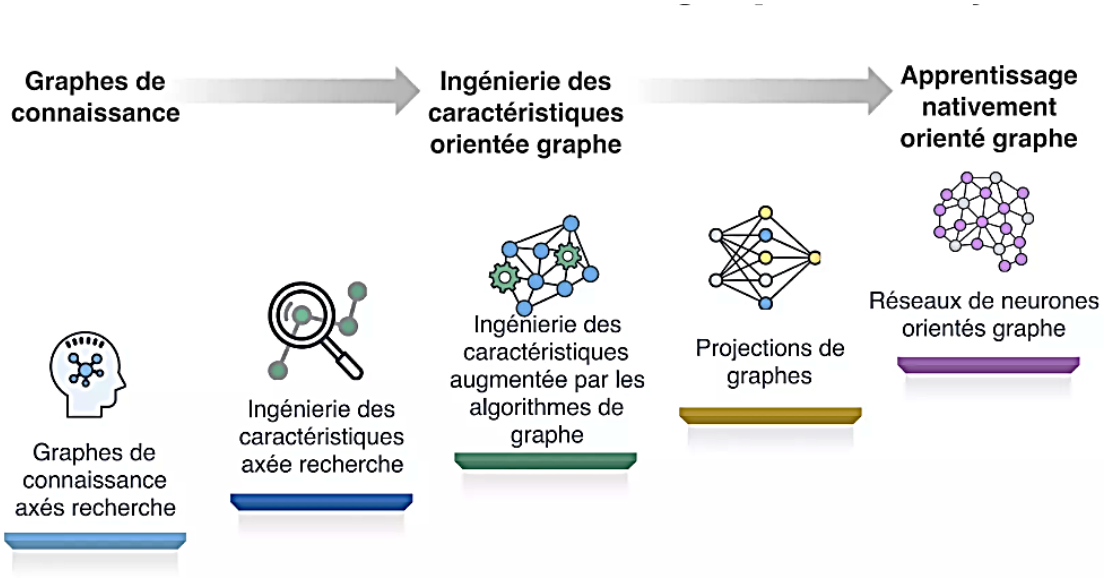


Figure 28: Les apports de la science des données orientée graphe. [48]

4.5 Le Rôle central des données dans l´apprentissage automatique

Les données jouent un rôle fondamental dans le processus d'apprentissage automatique. En effet, la qualité et la quantité des données disponibles déterminent largement les performances et la fiabilité des modèles générés. Plus il y a de données disponibles, plus le modèle peut apprendre de manière précise et généraliser ses connaissances à de nouvelles situations. De plus, la diversité des données est également cruciale : des données variées permettent au modèle d'acquérir une compréhension plus complète et robuste du problème, lui permettant ainsi de produire des résultats plus fiables et pertinents dans des contextes diversifiés. En somme, les données constituent le carburant essentiel qui alimente le processus d'apprentissage automatique, et leur qualité et leur diversité sont des facteurs déterminants pour obtenir des performances optimales.

4.6 Défis dans l'intégration du contexte

Un obstacle majeur dans de nombreuses approches de la science des données réside dans la difficulté à intégrer efficacement les informations contextuelles. Souvent, les données sont

analysées sans prendre en compte leur contexte, ce qui peut entraîner une interprétation erronée ou incomplète des résultats. Par exemple, les relations entre les différentes variables ou les tendances temporelles peuvent être négligées si le contexte n'est pas pris en considération. De plus, les structures complexes des données, telles que les réseaux sociaux ou les systèmes interconnectés, peuvent être difficiles à modéliser et à analyser sans une compréhension approfondie de leur contexte. Cette lacune dans l'intégration du contexte peut conduire à une perte d'informations cruciales et limiter la capacité des modèles à fournir des insights précis et utiles. Ainsi, pour relever ce défi, il est essentiel de développer des méthodes et des techniques qui permettent une intégration efficace du contexte dans l'analyse des données, afin d'obtenir des résultats plus pertinents et fiables.

4.7 L'apport des graphes de connaissances dans l'amélioration de l'apprentissage automatique

Les graphes de connaissances offrent une approche prometteuse pour surmonter les défis de l'apprentissage automatique. Leur capacité à représenter les relations entre les entités et à capturer le contexte permet d'enrichir les modèles d'apprentissage automatique en fournissant des informations supplémentaires. Cette richesse contextuelle améliore la précision des prédictions et la flexibilité des systèmes décisionnels. De plus, l'utilisation de graphes de connaissances offre un cadre permettant de suivre la traçabilité des données, ce qui est essentiel pour garantir la transparence et la fiabilité des résultats.

4.8 Un processus amélioré de l'apprentissage automatique grâce aux graphes de connaissances

Les graphes de connaissances enrichissent chaque phase du processus, de la collecte des données à l'entraînement des modèles, en passant par l'analyse des prédictions et l'application des résultats. Cette contextualisation renforce la fiabilité, la robustesse, l'explicabilité et la confiance des systèmes d'IA. [46]

4.9 L'importance du contexte pour l'IA

En intelligence artificielle, le contexte joue un rôle crucial pour permettre aux systèmes de prendre des décisions informées en tenant compte des informations environnantes. Pour qu´elle prenne des décisions de manière similaire à celle des humains, elle doit intégrer beaucoup de contexte. Sans informations périphériques et connexes, l'IA nécessite un apprentissage plus approfondi, des règles plus prescriptives et des applications plus spécifiques. [47]

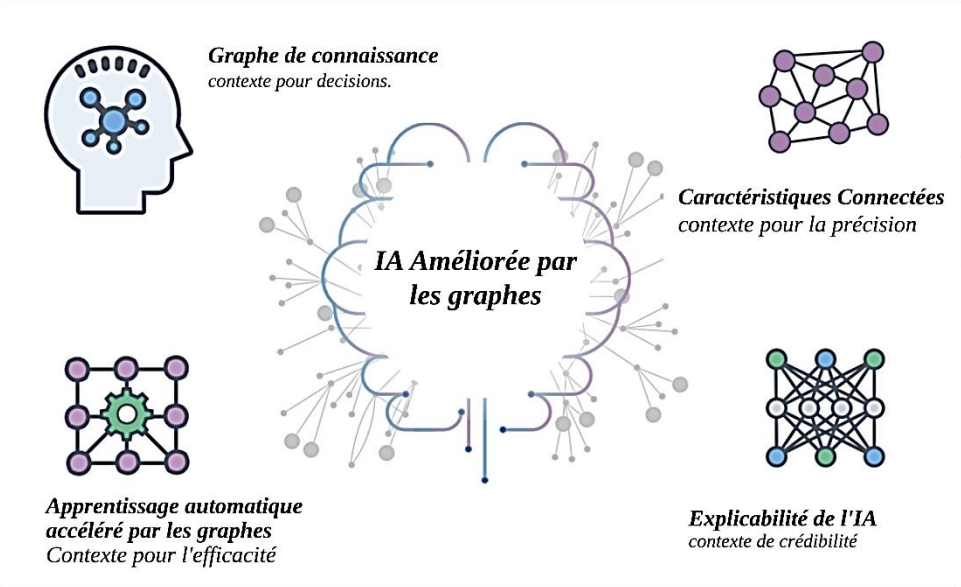


Figure 29: quatre façons dont les graphes fournissent du contexte [47].

Les graphes fournissent du contexte de plusieurs manières, améliorant ainsi la compréhension des données et des relations. Voici quatre façons dont les graphes apportent du contexte :

- Graphes de connaissance : Les graphes de connaissances organisent les données en entités reliées par des relations. En reliant les informations de manière sémantique, ils offrent un contexte riche sur les entités et leurs interactions, facilitant la compréhension des données.
- Efficacité de traitement : Les graphes accélèrent le traitement en permettant de traverser et d'analyser rapidement les relations entre les entités. Cette efficacité facilite la recherche d'informations contextuelles pertinentes, rendant le processus plus rapide et plus précis.
- Extraction de caractéristiques connectées : Les graphes analysent les données pour identifier les éléments les plus prédictifs. En examinant les relations connectées, les graphes extraient des caractéristiques significatives, améliorant ainsi la qualité des données utilisées pour l'apprentissage automatique.
- Explicabilité de l'IA : Lors de la prise de décision par des modèles d'IA, les graphes permettent de suivre et d'expliquer le raisonnement. En mettant en évidence les connexions et le contexte, ils rendent les résultats des modèles plus compréhensibles et transparents pour les utilisateurs. [47]

4.9.1 Graphes de Connaissances : Contexte pour les Décisions

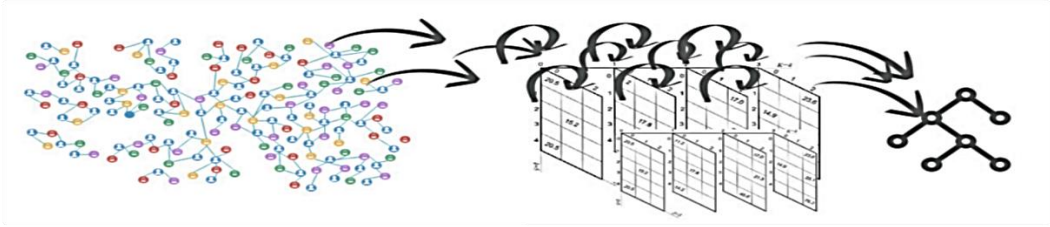
Les graphes de connaissances fournissent un cadre essentiel pour prendre des décisions éclairées dans divers domaines. Ils organisent les informations sous forme de nœuds (entités) et d'arêtes (relations), offrant ainsi une représentation structurée des connaissances. Chaque nœud représente un concept spécifique, tandis que les arêtes décrivent les liens entre ces concepts.

Le contexte pour les décisions émerge de la capacité des graphes de connaissances à capturer les relations et les interconnexions entre différentes entités. Par exemple, dans un contexte médical, un graphe de connaissances pourrait relier des entités telles que les patients, les maladies, les traitements, les symptômes, etc. Ces relations permettent de comprendre le contexte global des informations médicales, facilitant ainsi les décisions médicales éclairées. [47]

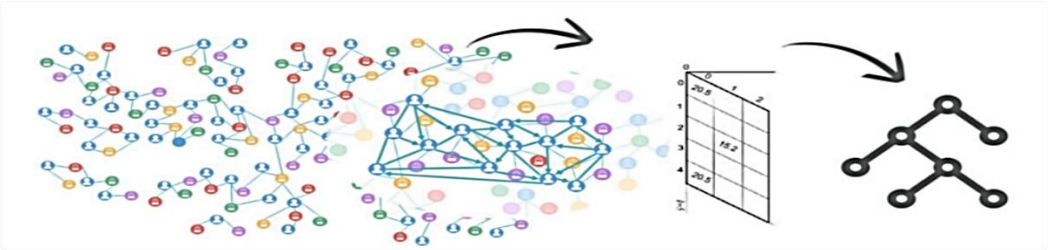
4.9.2 Apprentissage automatique accéléré par les graphes : Contexte pour l'efficacité

Lorsqu'on évoque l'apprentissage automatique accéléré par les graphes, il s'agit de l'intégration de structures de graphes dans les processus d'apprentissage automatique. Cette approche apporte un contexte particulier qui contribue à accroître l'efficacité globale du processus. Les graphes sont des représentations puissantes des relations entre les données, et les exploiter dans le cadre de l'apprentissage automatique offre plusieurs avantages en termes d'efficacité.

En particulier, le contexte pour l'efficacité émerge de la capacité des graphes à représenter les interconnexions complexes entre différentes entités. Plutôt que de traiter les données de manière isolée, les relations entre les entités sont prises en compte, permettant ainsi d'extraire des informations significatives. Par exemple, dans un graphe représentant les relations sociales, l'apprentissage automatique peut être accéléré en analysant les connexions entre les individus pour identifier des schémas de comportement ou des tendances.



Les données connectées du monde réel sont stockées sous forme de tables, puis connectées de manière itérative pour produire des arbres de décision.



L'entraînement sur des données connectées stockées dans une base de données de graphes est bien plus efficace. Le filtrage par graphe est très efficace.

Figure 30: Filtrage des données stockées dans des tables avec celle des graphes.

4.9.3 Caractéristiques Connectées : Contexte pour la Précision

Lorsque l'on aborde les caractéristiques connectées dans le contexte de l'apprentissage automatique, on se réfère à l'analyse des relations entre différentes entités pour améliorer la précision des modèles. Les caractéristiques connectées fournissent un contexte crucial pour évaluer le comportement et les interactions entre les entités, ce qui contribue à une compréhension plus approfondie des données.

Le contexte pour la précision émerge de la capacité à exploiter les liens entre les entités pour identifier des motifs significatifs. Par exemple, dans un réseau social, les caractéristiques connectées pourraient inclure le nombre d'amis communs, la fréquence des interactions, ou d'autres relations sociales pertinentes. En intégrant ces caractéristiques dans les modèles d'apprentissage automatique, on peut améliorer la capacité du modèle à faire des prédictions plus précises.

Les caractéristiques connectées offrent un contexte essentiel pour augmenter la précision des modèles d'apprentissage automatique en prenant en compte les relations entre les entités. Cette approche permet une meilleure capture des nuances des données, contribuant ainsi à des prédictions plus précises et fiables. [47]

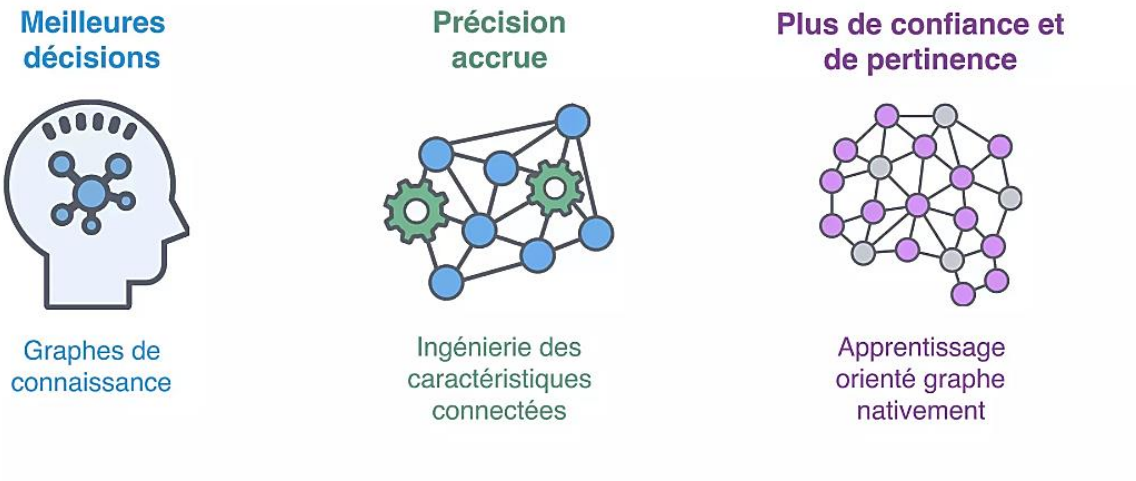


Figure 31: Méthode traditionnelle par rapport à la méthode basée sur les graphes pour la prise de décision. [48]

4.9.4 Explicabilité de l'IA : Contexte de Crédibilité

Lorsque l'on évoque l'explicabilité de l'intelligence artificielle, il s'agit de la capacité à comprendre et à expliquer les décisions prises par les modèles d'IA. Cette dimension joue un rôle crucial dans la crédibilité des systèmes d'IA, car elle permet de rendre les processus de prise de décision plus transparents et compréhensibles.

Le contexte de la crédibilité émerge de la nécessité de comprendre comment les modèles d'IA arrivent à leurs conclusions. L'explicabilité permet aux utilisateurs et aux parties prenantes de retracer le raisonnement du modèle, d'identifier les facteurs pris en compte, et de déterminer la fiabilité des résultats. Cela devient particulièrement important dans des domaines sensibles tels que la santé, la finance ou la justice, où la transparence des décisions est essentielle.

L'explicabilité de l'IA crée un contexte essentiel pour renforcer la crédibilité des modèles d'IA en permettant une compréhension claire des processus décisionnels. Cela contribue à établir la confiance des utilisateurs et à favoriser une adoption responsable de l'intelligence artificielle. [47]

4.10 L´intelligence artificielle améliorée par les Graphes

Lorsqu'on parle d'IA améliorée par les graphes, on fait référence à l'intégration de structures de graphes dans les systèmes d'intelligence artificielle pour améliorer leur performance et leur compréhension contextuelle. Cette approche vise à exploiter les relations et les connexions entre les données, représentées sous forme de graphes, pour renforcer les capacités des modèles d'IA.

L'amélioration par les graphes offre un contexte plus riche pour l'IA en permettant aux modèles de prendre en compte les interconnexions entre différentes entités. Par exemple, dans un contexte de recommandation, l'IA améliorée par les graphes pourrait considérer les relations entre les utilisateurs, les produits, les évaluations, etc., pour fournir des recommandations plus pertinentes et personnalisées.

L'IA améliorée par les graphes exploite la structure relationnelle des graphes pour enrichir le contexte des modèles d'IA, améliorant ainsi leur capacité à comprendre et à traiter les données de manière plus sophistiquée. [47]

4.11 Conclusion

Dans de ce chapitre, nous avons exploré les différents aspects de l'intégration des graphes dans les domaines clés de l'apprentissage automatique, mettant en lumière l'importance du contexte pour l'amélioration de la précision, de l'efficacité et de la crédibilité des modèles d'intelligence artificielle. Dans le prochain chapitre, nous passerons à l'implémentation pratique de notre approche.

Chapitre 05

Approche proposée

5 Chapitre 05 : Approche proposée

5.1 Introduction

Dans ce dernier chapitre, on présente une approche pour l'analyse des réseaux informatiques en utilisant le jeu de données CICIDS 2017. L'objectif de cette approche est d'analyser et de détecter les anomalies dans les réseaux informatiques en utilisant des techniques avancées de traitement des données et d'analyse de graphes de connaissances. Ce chapitre présentera les différentes étapes de notre approche, depuis la préparation des données jusqu'à l'analyse des résultats, en mettant en évidence les avantages et les performances de notre approche par rapport aux méthodes traditionnelles.

5.2 Description de l'ensemble de données CICIDS2017

L'ensemble de données CICIDS2017 "Cyber Intrusion Détection Data Sets 2017" est produit par l'Institut canadien de cyber sécurité. Chaque ensemble de données contient des attaques bénignes et les plus récentes, telles que DoS, DDoS, brute force SSH, brute force FTP, Heartbleed, infiltration et botnet, qui en font le plus récent, par rapport à d'autres ensembles de données et conçu pour la détection d'intrusion et la classification d'un large éventail d'attaques [49]

La période de saisie des données a débuté à 9 h le lundi 3 juillet 2017 et s'est terminée à 17 h le vendredi 7 juillet 2017, pour un total de cinq jours. Le lundi est le jour normal et ne comprend que la circulation bénigne. Les attaques ont été exécutés le matin et l'après-midi, mardi, mercredi, jeudi et vendredi. [50]

L'ensemble de données comprend 2 099 971 instances avec 83 fonctionnalités, et comprend 16 étiquettes de classe différentes, 1 étiquette normale et 15 étiquettes d'attaque, les tableaux 3, 4 et 5 offrent une analyse détaillée des différentes caractéristiques présentes dans l'ensemble de données.

Nom d'ensemble de données	CIC-IDS2017
Type d'ensemble de données	Multi classe
Année de lancement	2017
Nombre total d'instances	2099971
Nombre de function	83
Nombre de classes distinctes	16

Tableau 3: Caractéristiques générales de l'ensemble de jeu de données CICIDS2017.

Normal/ Attaque Labels	***Nombre d'instances***
BENIGN	1432918
DoS Hulk	158468

DDoS	95144
DoS GoldenEye	7567
Infiltration – Portscan	6127
FTP-Patator	3972
DoS Slowloris	3859
SSH-Patator	2961
DoS Slowhttptest	1740
Portscan	1683
Botnet	736
Web Attack - Brute Force	73
Infiltration	36
Web Attack – XSS	18
Web Attack - SQL Injection	13
Heartbleed	11

Tableau 4: Occurrence des instances par classe dans l'ensemble de données CICIDS2017.

FONCTION	TYPE	FONCTION	TYPE
Protocol	int64	Packet Length Mean	float64
Flow Duration	int64	Packet Length Std	float64
Total Fwd Packet	int64	Packet Length Varience	float64
Total Bwd packets	int64	FIN Flag Count	int64
Total Length of Fwd Packet	int64	SYN Flag Count	int64
Total Length of Bwd Packet	int64	RST Flag Count	int64
Fwd Packet Length Max	int64	PSH Flag Count	int64
Fwd Packet Length Min	int64	ACK Flag Count	int64
Fwd Packet Length Mean	float64	RST Flag Count	int64
Fwd Packet Length Std	float64	URG Flag Count	int64
Bwd Packet Length Max	int64	CWR Flag Count	int64
Bwd Packet Length Min	int64	ECE Flag Count	int64
Bwd Packet Length Mean	float64	Down/Up Ratio	float64
Bwd Packet Length Std	float64	Average Packet Size	float64
Flow Bytes/s	float64	Fwd Segment Size Avg	float64
Flow Packets/s	float64	Bwd Segment Size Avg	float64
Flow IAT Mean	float64	Fwd Bytes/Bulk Avg	int64
Flow IAT Std	float64	Fwd Packet/Bulk Avg	int64
Flow IAT Max	int64	Fwd Bulk Rate Avg	int64
Flow IAT Min	int64	Bwd Bytes/Bulk Avg	int64
Flow IAT Total	int64	Bwd Packet/BulkAvg	int64

Fwd IAT Mean	float64	Bwd Bulk Rate Avg	int64
Fwd IAT Std	float64	Subflow Fwd Packets	int64
Fwd IAT Max	int64	Subflow Fwd Bytes	int64
Fwd IAT Min	int64	Subflow Bwd Packets	int64
Bwd IAT Total	int64	Subflow Bwd Bytes	int64
Bwd IAT Mean	float64	FWD Init Win Bytes	int64
Bwd IAT Std	float64	Bwd Init Win Bytes	int64
Bwd IAT Max	int64	Fwd Act Data Pkts	int64
Bwd IAT Min	int64	Fwd Seg Size Min	int64
Fwd PSH Flags	int64	Active Mean	float64
Bwd PSH Flags	int64	Active Std	float64
Fwd URG Flags	int64	Active Max	int64
Bwd URG Flags	int64	Active Min	int64
Fwd RST Flags	int64	Idle Mean	float64
Bwd RST Flags	int64	Idle Std	float64
Fwd Header Length	int64	Idle Max	int64
Bwd Header Length	int64	Idle Min	int64
Fwd Packets/s	float64	ICMP Code	int64
Bwd Packets/s	float64	ICMP Type	int64
Packet Lengtth Min	int64	Total TCP Flow Time	int64
Packet Length Max	int64	Label	Object

Tableau 5: Caractéristiques présentes dans l'ensemble de données CICIDS2017.

5.3 Matériel et logiciels

5.3.1 Environnement d´exécution

Notre expérience dépend fortement du matériel pour accomplir efficacement leurs tâches. Avant d'entreprendre l'exploration approfondie de notre approche, il est crucial de vérifier que certaines exigences matérielles essentielles sont satisfaites.
Ci-dessous sont les propriétés du PC utilisé pour accomplir ce travail :
Nom de l'appareil : DESKTOP-35BGH9D.
Processeur : Intel(R) Core(TM) i5-7200U CPU @ 2,50 GHz 2,71 GHz.
RAM installée : 8,00 Go.

5.3.2 Outils utilisés

- **Anaconda**

Anaconda est une distribution de Python qui comprend un grand nombre de bibliothèques et d'outils populaires pour le calcul scientifique, l'analyse de données et l'apprentissage automatique. Il est livré avec son propre gestionnaire de packages appeler Conda, qui facilite l'installation et la gestion des packages Python et des environnements virtuels. [51]

- **Jupyter**

Jupyter Notebook est une application web open-source qui vous permet de créer et de partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif. C'est un outil populaire parmi les scientifiques des données, les chercheurs pour l'informatique interactive et l'analyse des données. Le nom « Jupyter » est dérivé des trois principaux langages de programmation qu'il supportait à l'origine : Julia, Python et R. [52]

- **Neo4j**

Neo4j est une base de données graphique open-source implémentée en Java. Les fondateurs de Neo4j le décrivent comme une base de données entièrement transactionnelle, un moteur Java persistant où il est possible de stocker des données sous la forme de graphiques au lieu de tableaux. Le Neo4j est considéré comme la base de données de graphes la plus populaire et la plus utilisée au monde. [53]

5.3.3 Langages utilisés

- **Python**

Python est un langage de programmation de haut niveau, polyvalent et très populaire. Le langage de programmation Python (dernier Python 3) est utilisé dans le développement Web, les applications de Machine Learning, ainsi que toute la technologie de pointe dans l'industrie du logiciel. Le langage Python est utilisé par presque toutes les grandes entreprises technologiques comme Google, Amazon, Facebook, Instagram, Dropbox, Uber, etc.
La plus grande force de Python est l'énorme collection de bibliothèque standard utilisées telles que NumPy, Pandas, Scikit-learn, Tensor-Flow et Keras. [52]

- **Cypher**

Cypher est un langage de requête de graphe déclaratif utilisé par les développeurs du monde entier. Créé par Neo4j, Cypher fournit des requêtes expressives et efficaces pour les graphiques de propriétés. [54]

5.3.4 Les bibliothèques utilisées

Bibliothèque	Description
TensorFlow	Bibliothèque open source pour les machines Learning. [55]
Keras	Interface haut niveau pour TensorFlow. [56]

Scikit-learn	Bibliothèque d'apprentissage automatique avec des implémentations efficaces d'algorithmes de ML. [57]
Pandas	Bibliothèque pour la manipulation et l'analyse des données. [58]
NumPy	Bibliothèque pour le calcul numérique en Python. [59]

Tableau 6: Les bibliothèques utilisées dans notre approche.

5.1 Etude d'ablation

Dans cette étude d'ablation, nous avons conçu trois expériences distinctes en fonction de notre objectif principal. Dans l'approche 1, nous avons appliqué des techniques de ML et de DL sur le jeu de données CIDIDS2017, en explorant les méthodes traditionnelles et modernes pour la détection des intrusions. Pour l'approche 2, elle s'est concentrée sur la classification des attaques en fonction des flux du réseau. Cela a impliqué le stockage de notre ensemble de données dans un format adapté pour l'analyse des flux, permettant une identification plus précise des comportements malveillants. Enfin, dans l'approche 3, nous avons affiné notre méthode en extrayant uniquement les aspects les plus pertinents du jeu de données, comme décrit dans le chapitre précédent. Les résultats obtenus dans cette troisième approche ont montré une amélioration significative par rapport aux précédentes, soulignant l'efficacité de notre approche de traitement et d'analyse des données.

5.2 Les expériences

5.2.1 Approche 01 : Application des méthodes d'apprentissage automatiques

Nous préparons les données brutes pour les algorithmes de machine Learning. Ensuite, en divisant les données en ensembles d'entraînement et de test, on évalue les performances de divers algorithmes tels que les réseaux de neurones convolutionnels (CNN), les k plus proches voisins (KNN) et les forêts aléatoires (RF) pour améliorer les mesures de performance telles que l'exactitude, la précision, le rappel et le score F1. Notre approche est présentée par la figure 32.

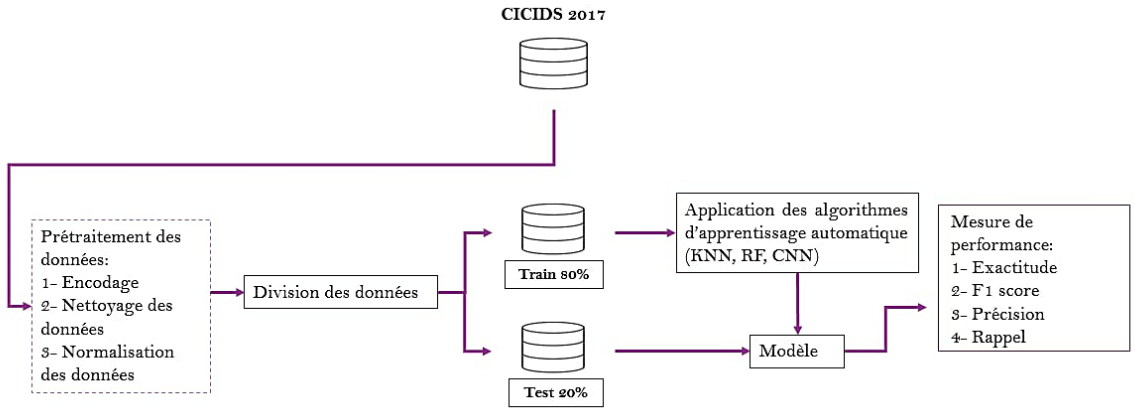


Figure 32: Architecture de composants de l'approche 01.

Étape 01 : Prétraitement des données

Le prétraitement des données est une étape cruciale dans le processus de développement de modèles de machine learning. Cette étape vise à nettoyer, transformer et organiser des données brutes pour les rendre prêtes à être utilisées dans les algorithmes de Deep Learning et obtenir des prédictions plus précises et plus fiables, cela inclut des techniques tels que l'encodage, le nettoyage de données et la normalisation.

- **Encodage**

Dans cette étape les variables catégorielles doivent être converties en une forme numérique pour être utilisées efficacement par les algorithmes d'apprentissage automatique. Cela peut être fait à l'aide de techniques telles que l'encodage one-hot ou l'encodage ordinal. Dans notre cas on a utilisé l'encodage ordinal.

Syntaxe :

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df.Label = le.fit_transform(df.Label)
```

- **Nettoyage des données**

Après avoir effectué le prétraitement pour assurer sa qualité, on passe a l'étapes de nettoyage. Voici les étapes que nous avons suivies, accompagnées de leur code correspondant :

1- **Suppression des doublons :** Nous avons éliminé les valeurs en double de notre ensemble de données.

Syntaxe 1 :

```
df = df.drop_duplicates()
```

2- **Vérification des valeurs manquantes :** Toutes les valeurs manquantes (NaN) ont été remplacées par des zéros.

Syntaxe 2 :

```
df = df.fillna(0)
```

3- **Détection et traitement des valeurs aberrantes :** Nous avons cherché à identifier les valeurs extrêmes qui pourraient fausser notre analyse statistique, et nous avons pris des mesures pour les traiter et en les supprimant.

- **Normalisation des données**

La normalisation des données est une étape cruciale du prétraitement des données qui vise à rendre les variables comparables en les mettant sur une même échelle, ce qui facilite l'analyse et la modélisation des données. Nous nous sommes appuyés sur la normalisation StandardScaler comme étape clé dans le prétraitement des données.
Le score standard d'un échantillon x est calculé comme suit : z = (x - u) / s.

Syntaxe :

```
from sklearn.preprocessing import StandardScaler
x = df.iloc[:, df.columns != 'Label']
y = df[['Label']].to_numpy()
Scaler = StandardScaler()
x = scaler.fit_transform(x)
return x, y
```

Étape 02 : Division des données

La division des données est couramment utilisée dans l'apprentissage automatique pour évaluer et valider les modèles que construire. Dans cette étape, nous séparons l'ensemble de données en deux parties : 80 % des données sont réservées pour l'entraînement du modèle, tandis que les 20 % restants sont réservés pour tester la performance du modèle.
Nous avons utilisé la fonction train_test_split de la bibliothèque scikit-learn comme suit:

Syntaxe :

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, stratify=y,
test_size=0.20,
random_state=np.random.randint(10))
```

Une fois que les données sont prétraitées nous avons appliqué les algorithmes d'apprentissage automatique tels que les réseaux de neurones convolutionnels (CNN), les k plus proches voisins (KNN) et les forêts aléatoires (RF) sur notre ensemble de données

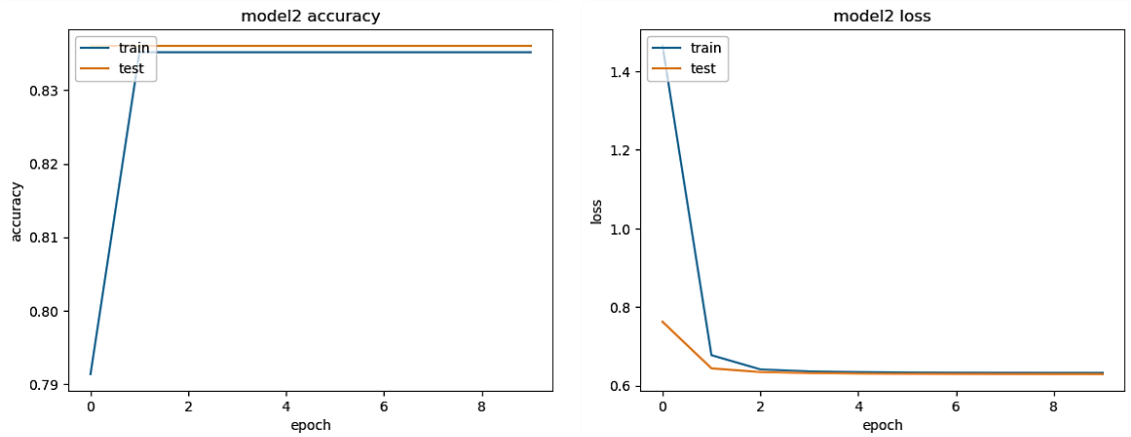
prétraité pour améliorer les mesures de performance telles que l'exactitude, la précision, le rappel et le score F1.

Expérience 01 : Dans cette expérience nous avons appliqué l'algorithme d'apprentissage profond les réseaux de neurones convolutionnels (CNN) et voici les résultats :

Algorithme	Exactitude	Précision	Rappel	F1-Score
CNN	83,54%	69,78%	83,54%	76,04%

Tableau 7: Rendement des mesures d'évaluation pour le modèle amélioré (CNN).

La figure 33 montre les performances améliorées du modèle en termes d'exactitude et de la perte du modèle.



a) L' exactitude de train et test **b) La perte de train et test**

Figure 33: Visualisation des performances améliorées du modèle.

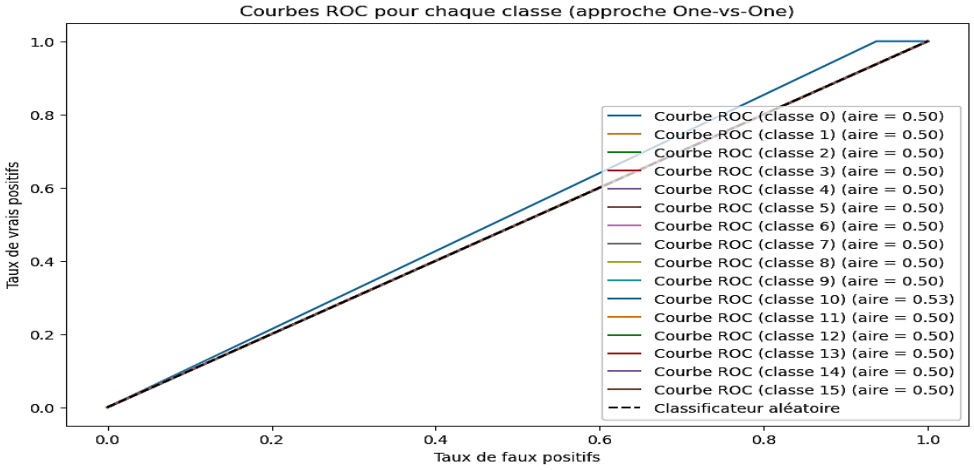


Figure 34: La courbe ROC du modèle CNN.

Expérience 2 : Dans la deuxième expérience, nous avons appliqué les k plus proches voisins (KNN) sur notre ensemble de données et nous avons obtenu les résultats suivants :

Algorithme	Exactitude	Precesion	Rappel	F1-Score
KNN	99,98%	99,56%	90,88%	92,88%

Tableau 8: Rendement des mesures d'évaluation pour le modèle amélioré(KNN).

Type d'attaque	Precesion	Recall	F1-Score
BENIGN	100%	100%	100%
Botnet	96%	99%	97%
DDoS	100%	100%	100%
DoS GoldenEye	99%	99%	99%
DoS Hulk	100%	100%	100%
DoS Slowhttptest	100%	99%	99%
DoS Slowloris	100%	99%	100%
FTP-Patator	100%	100%	100%
Heartbleed	100%	100%	100%
Infiltration	100%	29%	44%
Infiltration_Portscan	99%	99%	99%
Portscan	99%	97%	98%
SSH-Patator	100%	100%	100%
Web Attack - Brute Force	100%	100%	100%
Web Attack - SQL Injection	100%	33%	50%
Web Attack – XSS	100%	100%	100%

Tableau 9: Mesures d'évaluation pour chaque étiquette.

La figure 35 illustrent la matrice de confusion du modèle KNN qui permet d'identifier les classes pour lesquelles le modèle performe bien et celles pour lesquelles il y a des lacunes, ce qui peut orienter les actions de correction ou d'amélioration du modèle.

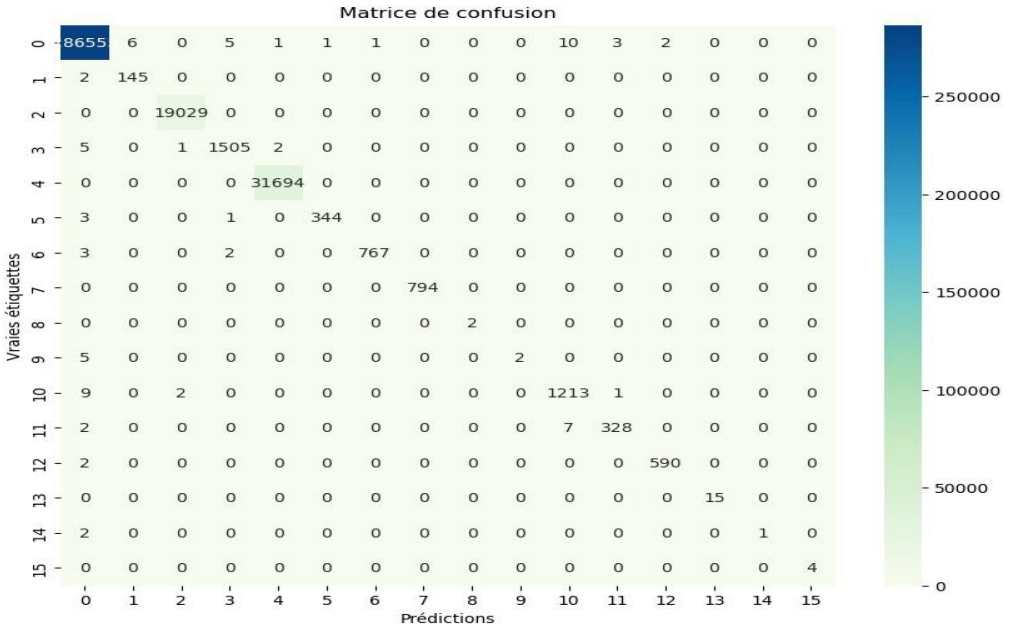


Figure 35: Matrice de confusion pour le modèle amélioré (KNN).

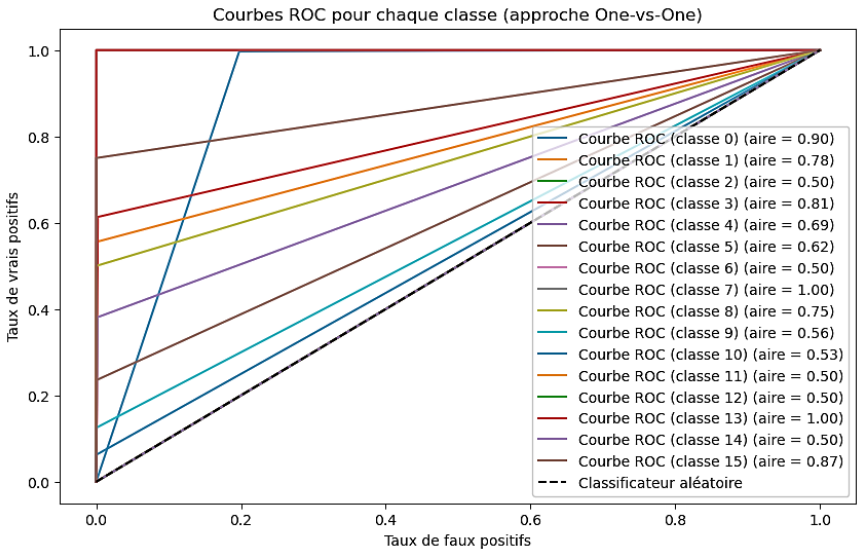


Figure 36: La courbe ROC du modèle KNN.

Expérience 3 : Dans la troisième expérience, nous avons appliqué l'algorithme de forêts aléatoire (RF) à notre ensemble de données et obtenu un exactitude élevée de 99,99%.

Algorithme	Exactitude	Précision	Rappel	F1-Score
RF	99,99%	99,99%	99,99%	99,99%

Tableau 10: Rendement des mesures d'évaluation pour le modèle amélioré(RF).

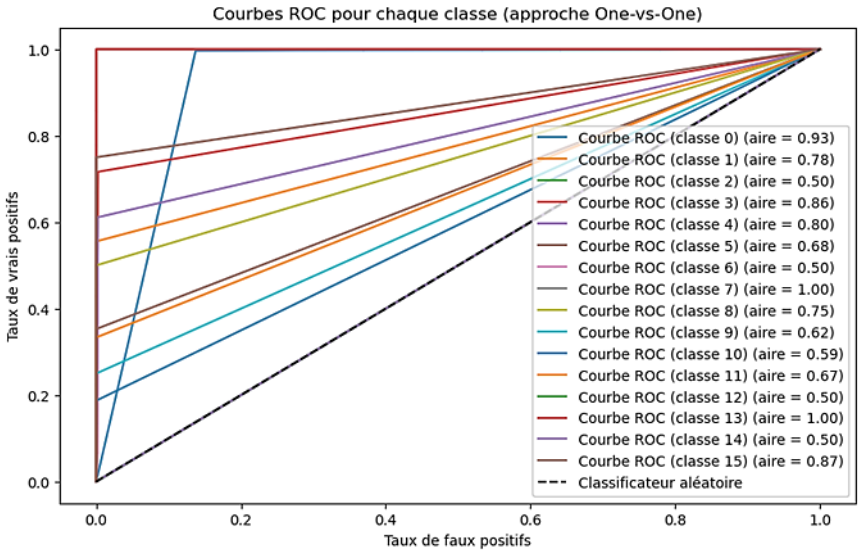


Figure 37: La courbe ROC du modèle RF.

5.2.2 Approche 02 : Application des algorithmes des graphes de connaissance

Nous avons transféré nos données depuis l'ensemble de données vers les graphes Neo4j. Cela implique de convertir les données tabulaires en entités et relations dans Neo4j. Ensuite, nous pouvons explorer et analyser ces données en utilisant des requêtes Cypher et en visualisant les relations entre les entités. Voici la figure représentant notre seconde approche :

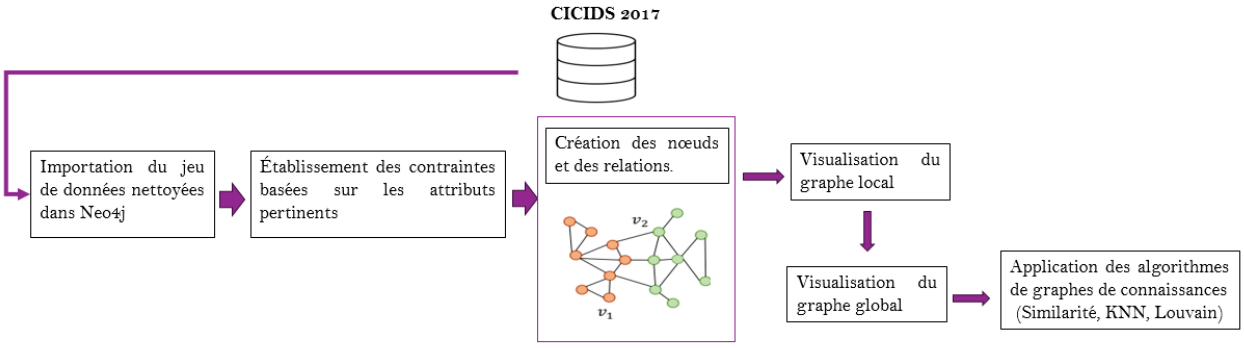


Figure 38: Architecture de composants de l'approche 02.

Dans ce processus :

1. **Importation du jeu de données nettoyées dans neo4j** : Les données nettoyées sont intégrées dans Neo4j pour former la base de votre graphe de connaissances de la facon suivante :
   - Création d'un nouveau projet.
   - Création d'une base de données local.
   - L'importation du fichier CSV.

Après l'ouverture du projet, on charge le jeu de données CICIDS2017 nettoyée.

Syntaxe :

```
LOAD CSV WITH HEADERS FROM  'file:///Cleaned_CICIDS2017.csv'
AS row RETURN
count(row);
```

2. **Établissement des contraintes basées sur les attributs pertinents** : Comme le jeu de données CICIDS 2017 à 83 attributs, on s´est basé sur les attributs les plus pertinents (11 attributs sélectionnés).

Syntaxe :

```
CREATE CONSTRAINT Label IF NOT EXISTS FOR (lab:Label) REQUIRE
lab.name IS UNIQUE
```

3. **Création des nœuds dans le graphe :**

Des nœuds sont créés dans Neo4j pour représenter nos entités principales (Average_Packet_Size,Bwd_Header_Length,Bwd_Segment_Size_Avg,Fwd_Header_Length,Fwd_Segment_Size_Avg,ICMP_Code,ICMP_Type,Label,Protocol,Total_Length_of_Bwd_Packet,Total_Length_of_Fwd_Packet). Leur création se fait comme se suit :

Syntaxe:

```
LOAD CSV WITH HEADERS FROM 'file:///Cleaned_CICIDS2017.csv' AS
row
MERGE (lab:Label {name: row.Label})
SET lab.Label=toInteger(row.Label);
```

4. **Établissement des relations entre les nœuds :** Des relations sont établies entre les nœuds pour capturer les interactions et les liens entre les différentes entités. La construction des relations se fait comme se suit :

Syntaxe :

```
load csv with headers from 'file:///Cleaned_CICIDS2017.csv' AS row
MATCH (lab:Label {name: row.Label})
MATCH (pro:Protocol {name: row.Protocol})
MERGE (pro)-[:define]->(lab)
```

5. **Visualisation du graphe local** : Le graphe est visualisé localement à l'aide d'outils de visualisation graphique pour examiner sa structure et ses caractéristiques.

Syntaxe :

```
CALL db.schema.visualization
```

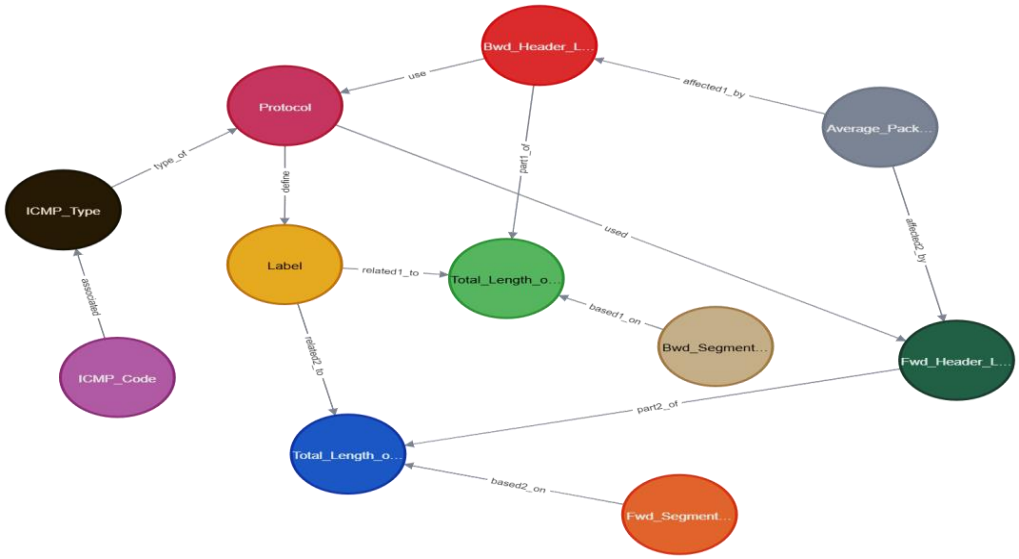


Figure 39: Visualisation du graphe local.

6. **Visualisation du graphe global** : Le graphe dans son ensemble est visualisé en utilisant l'interface Neo4j Browser avec la commande qui se suit :

Syntaxe :

```
MATCH (n)
WITH labels(n) AS types, n
LIMIT 498853
UNWIND types AS type
RETURN type, collect(n)[0..50] AS nodes_de_ce_type
```

Figure 40: Visualisation du graphe global.

7. **Application des algorithmes de graphes de connaissances :** Des algorithmes tels que la similarité, le KNN (k plus proches voisins), et Louvain sont utilisés pour analyser et extraire des informations pertinentes à partir du graphe construit.

- **Algorithme de similarité** : La syntaxe générale de l'algorithme nécessite la référence à un graphe nommé précédemment chargé. De plus, divers modes d'exécution sont disponibles.

a. Projection des données de graphe dans le domaine de la science des données :

Syntaxe :

```
CALL gds.graph.project(
'Graph', {
Bwd_Header_Length:{label: 'Bwd_Header_Length'},
 Bwd_Segment_Size_Avg: {label: 'Bwd_Segment_Size_Avg'},
 Protocol: {label: 'Protocol'},
 Fwd_Header_Length:{label: 'Fwd_Header_Length'},
Fwd_Segment_Size_Avg:{label: 'Fwd_Segment_Size_Avg'},
Label: {label: 'Label'},
Total_Length_of_Bwd_Packet:{label: 'Total_Length_of_Bwd_Packet'},
 Total_Length_of_Fwd_Packet:{label: 'Total_Length_of_Fwd_Packet'}},
{ATTACK: {
  type: '*',
  orientation: 'UNDIRECTED'
 }});
```

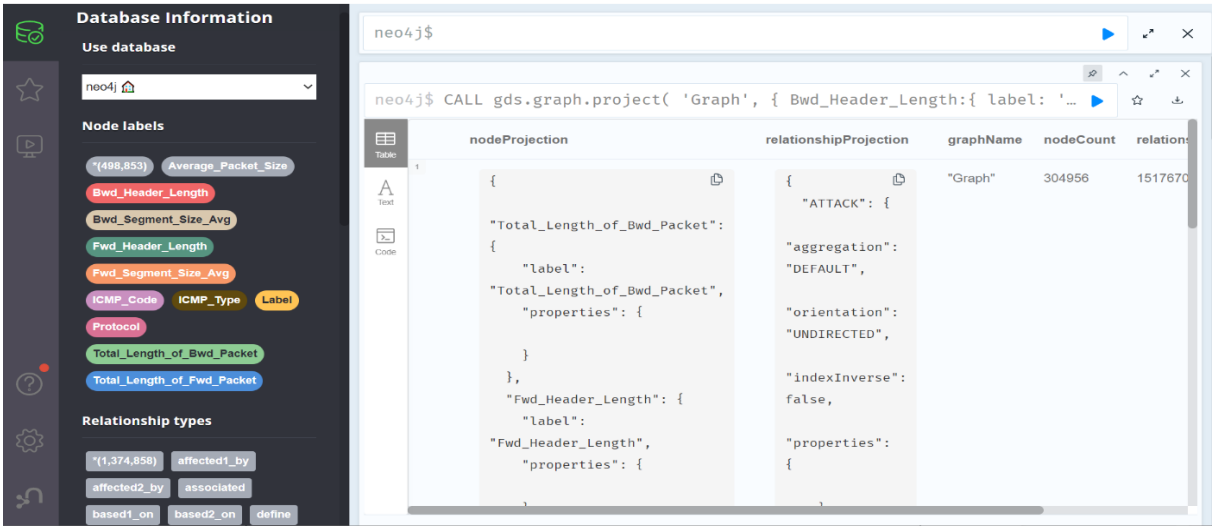


Figure 41: Projection de Graph.

b. L'exécution de la fonction "estimate" permet d'estimer le coût d'exécution de l'algorithme (similitude des nœuds). Nous utiliserons le mode d'écriture.

Syntaxe :

```
CALL gds.nodeSimilarity.write.estimate('Graphe', {
writeRelationshipType: 'SIMILAR',
writeProperty: 'score'
})
YIELD nodeCount, relationshipCount, bytesMin, bytesMax,
requiredMemory
```

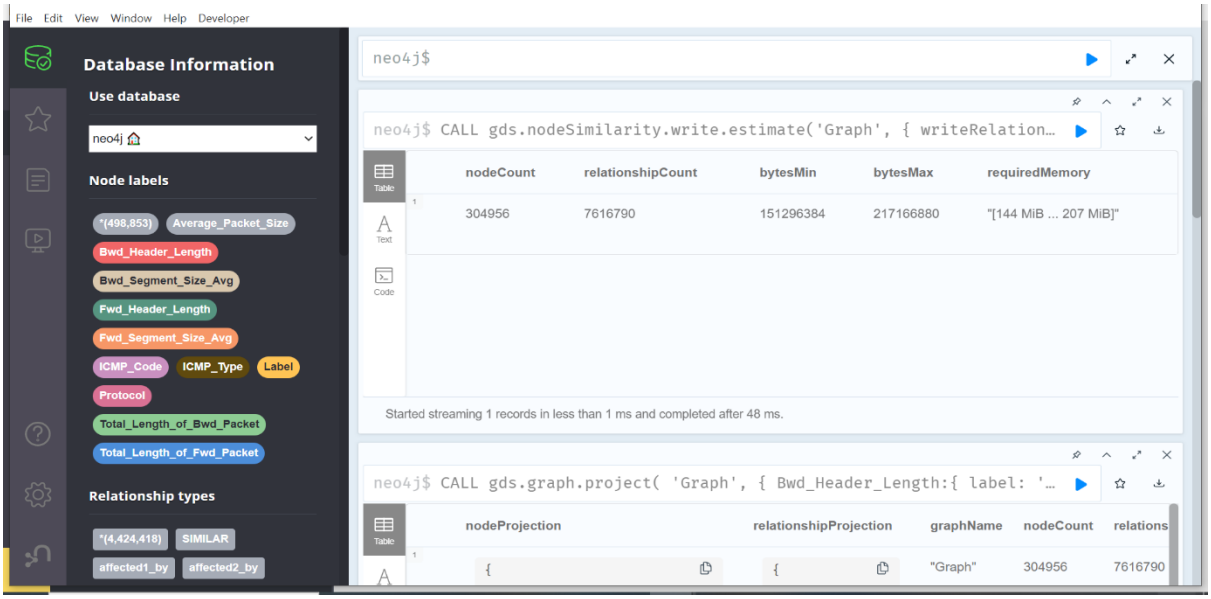


Figure 42: Mode d'éxecution estimate.

c. En mode exécution en streaming, l'algorithme renvoie le score de similarité pour chaque relation.

Syntaxe :

```
CALL gds.nodeSimilarity.stream('Graph')
YIELD node1, node2, similarity
RETURN gds.util.asNode(node1).name AS colonne1,
gds.util.asNode(node2).name AS colonne2, similarity
ORDER BY similarity DESCENDING, colonne1, colonne2
LIMIT 1000
```

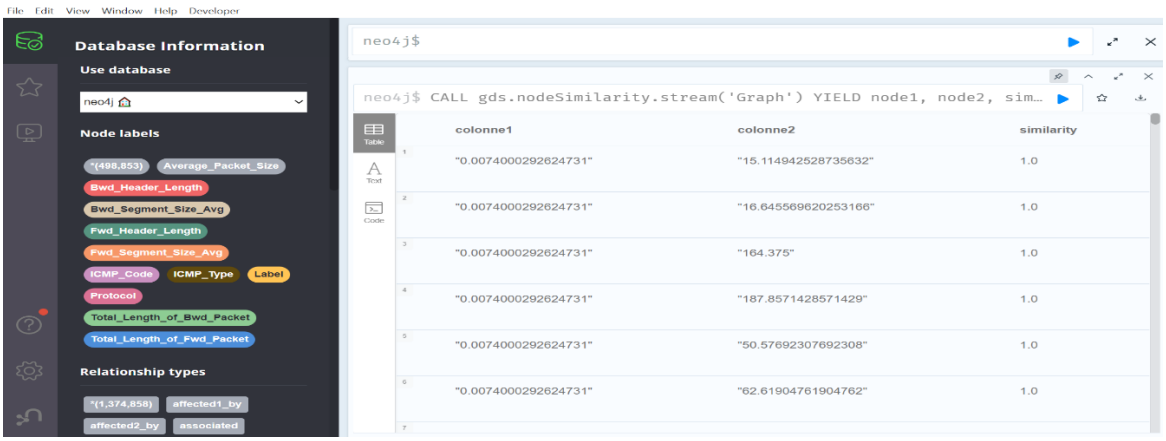


Figure 43: Éxecution en mode Stream.

d. En mode Stats, l'algorithme génère une seule ligne qui résume le résultat de l'algorithme.
Syntaxe :

```
CALL gds.nodeSimilarity.stats('Graph')
YIELD nodesCompared, similarityPairs
```

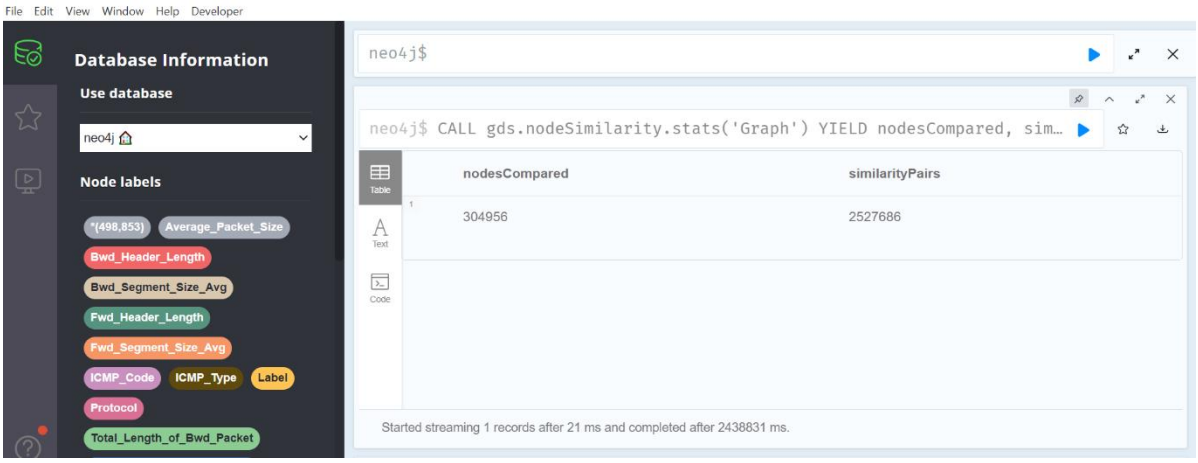


Figure 44: Mode d'exécution Stats.

e. En mode mutate, le graphe nommé sera modifié en ajoutant une nouvelle propriété de relation qui contient le score de similarité pour cette relation. Le nom de la nouvelle propriété est spécifié à l'aide du paramètre de configuration obligatoire "mutateProperty".

Syntaxe :

```
CALL gds.nodeSimilarity.mutate(
IDSgraph: String,
configuration: Map
)
```

```
YIELD
 preProcessingMillis: Integer,
computeMillis: Integer,
 mutateMillis: Integer,
 postProcessingMillis: Integer,
 relationshipsWritten: Integer,
 nodesCompared: Integer,
 similarityDistribution: Map,
 configuration: Map
```

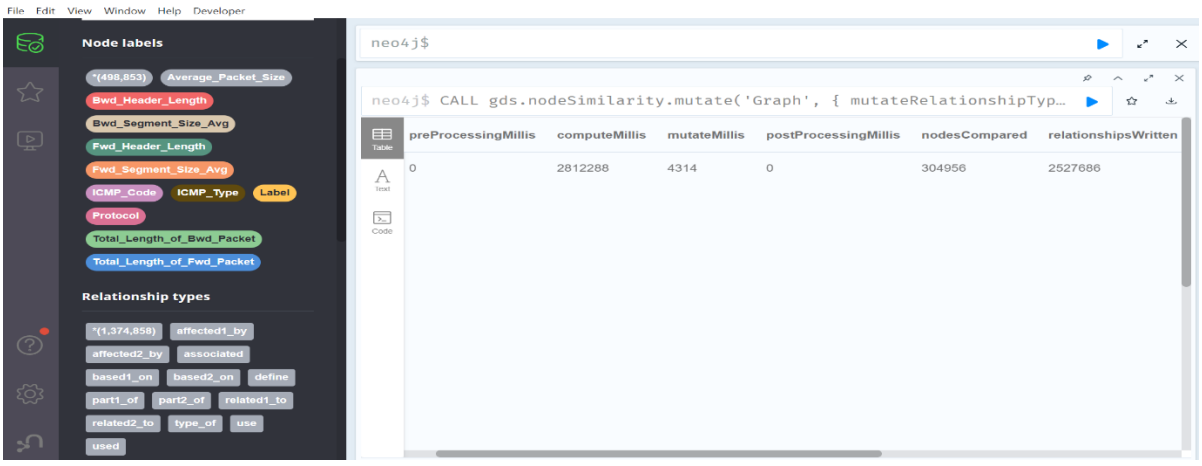


Figure 45: L'éxecution en mode Mutate.

f. Le mode d'écriture exécute chaque paire de nœuds et crée une relation avec leur score de similarité comme propriété dans la base de données Neo4j. Le type de la nouvelle relation est défini par le paramètre de configuration obligatoire "writeRelationshipType".

Syntaxe :

```
CALL gds.nodeSimilarity.write('Graphe', {
 writeRelationshipType: 'SIMILAR',
writeProperty: 'score'
})
YIELD nodesCompared, relationshipsWritten
```

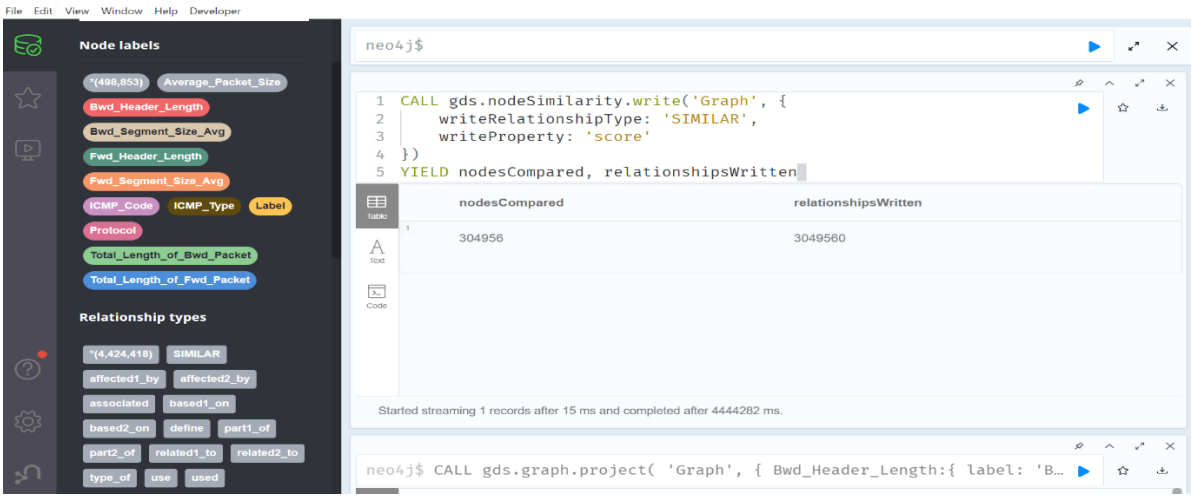


Figure 46: L'éxecution en mode Write.

- **Algorithme KNN :** Les résultats de l'algorithme se présentent sous la forme de nouvelles relations entre les nœuds et leurs k-voisins les plus proches. Les scores de similarité sont représentés par les propriétés des relations.

a. Projection du graphe nommé (myGraph) comme se suit :

```
CALL gds.graph.project(
'myGraph',
{Total_Length_of_Bwd_Packet:{
properties:['Total_Length_of_Bwd_Packet']
}}, '*'
);
```

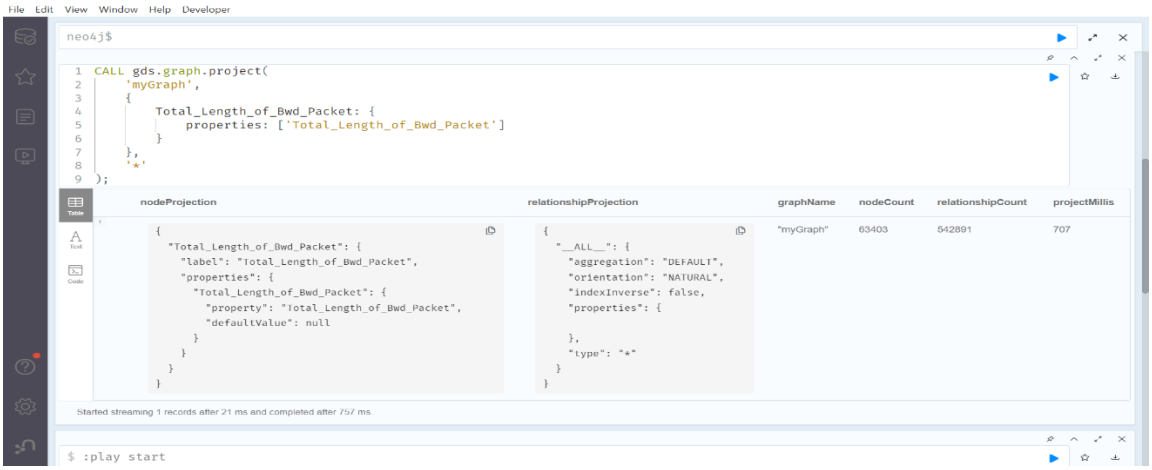


Figure 47: Projection de graphe "myGraph".

b. En mode d'écriture : ce dernier revêt une importance capitale dans notre processus. Pour chaque paire de nœuds, une nouvelle relation est instaurée, portant en elle le précieux score de similarité en tant que propriété au sein de notre base de données Neo4j. Il est à noter que le type de cette relation émane du paramètre de configuration incontournable, writeRelationshipType. Chaque liaison nouvellement formée entre deux nœuds ne se contente pas de connecter ces derniers ; elle se charge également de préserver leur affinité via le stockage du score de similarité. De plus, la clé de propriété de cette relation est minutieusement déterminée par le paramètre writeProperty, lequel ne laisse aucune place à l'improvisation.

Syntaxe :

```
CALL gds.knn.write('myGraph', {
 writeRelationshipType: 'SIMILAR_KNN',
 writeProperty: 'score_KNN',
 topK: 10,
 randomSeed: 42,
 concurrency: 1,
 nodeProperties: 'Total_Length_of_Bwd_Packet'
})
YIELD nodesCompared, relationshipsWritten
// Explore the similarity results
match (tlbp1:Total_Length_of_Bwd_Packet)-[:SIMILAR_KNN]->(tlbp2)
RETURN tlbp1, tlbp2
LIMIT 500
```

La figure ci-dessous récapitule le déroulement de l'algorithme knn :

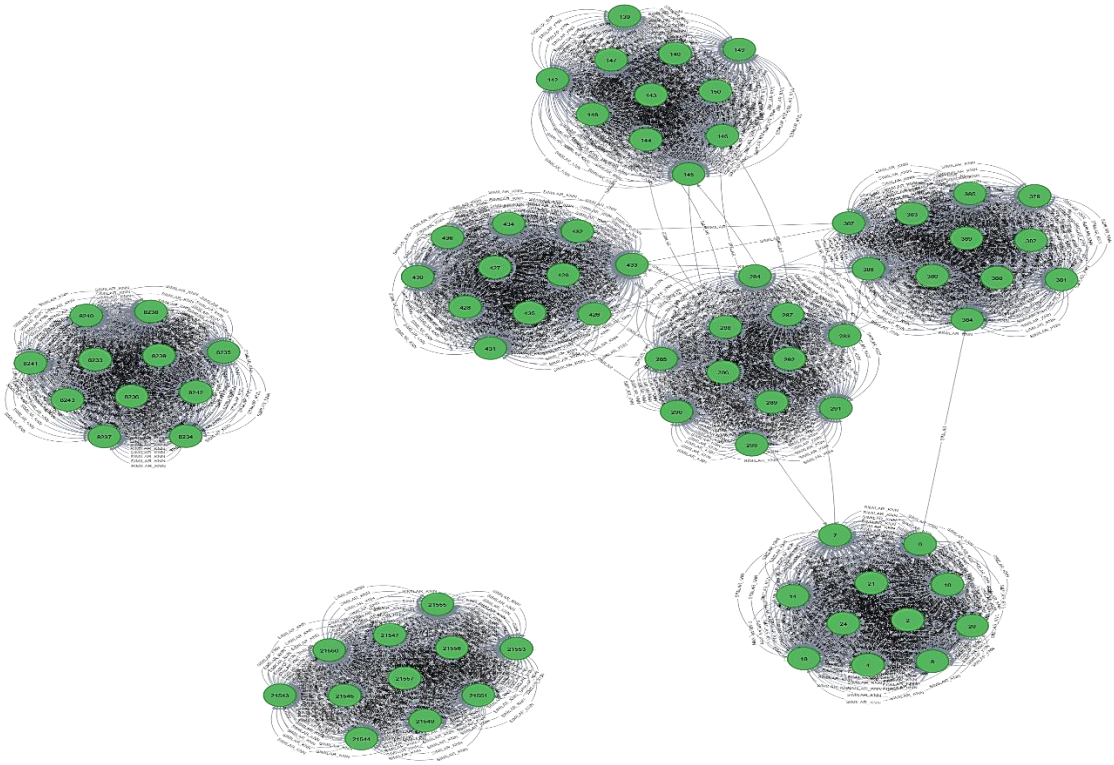


Figure 48: Résultat d'application de l'algorithme KNN.

- **Algorithme de Louvain :** on a utilisé cet algorithme pour détecter les communautés. Il partitionne le graphe de manière à maximiser la cohésion intra-communautaire tout en minimisant les connexions inter-communautaires, ce qui permet d'identifier les groupes de nœuds densément connectés.

a. Mode write.estimate : permettra d'estimer les besoins en mémoire pour l'exécution de l'algorithme.

Syntaxe :

```
CALL gds.louvain.write.estimate('myGraph', { writeProperty: 'community'
})
YIELD nodeCount, relationshipCount, bytesMin, bytesMax,
requiredMemory
```

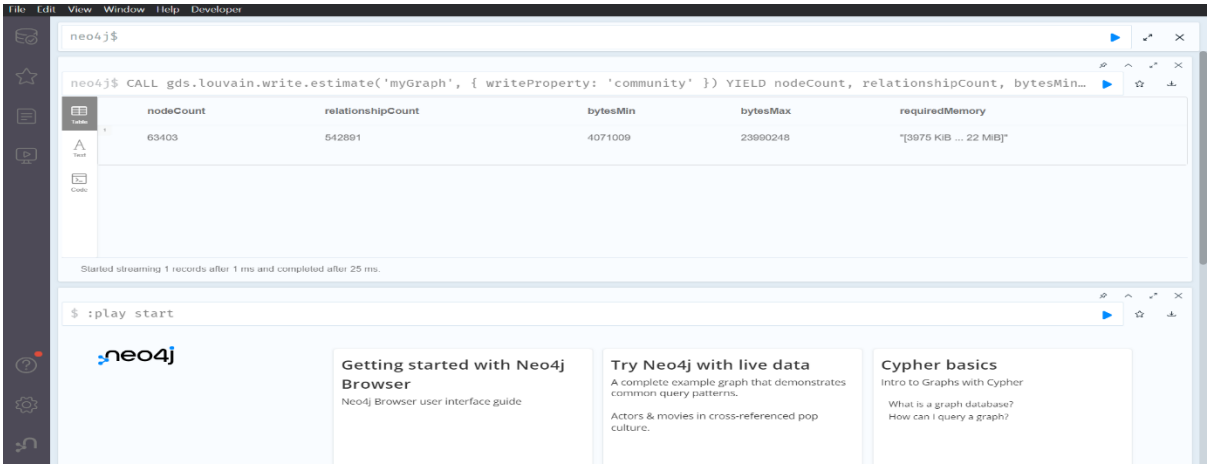


Figure 49: Mode write estimate.

b. Mode flux : Dans le mode d'exécution en flux, l'algorithme renvoie l'identifiant de la communauté pour chaque nœud. Cela nous permet d'inspecter directement les résultats ou de les traiter ultérieurement dans Cypher sans aucun effet secondaire.

Syntaxe :

```
CALL gds.louvain.stream('myGraph')
YIELD nodeId, communityId, intermediateCommunityIds
RETURN gds.util.asNode(nodeId).name AS name, communityId
ORDER BY name ASC
```

Figure 50: Louvain avec le mode flux.

c. Mode d'écriture : Ce qui permet de persister directement les résultats dans la base de données.

Syntaxe :

```
CALL gds.louvain.write('myGraph', { writeProperty: 'community' })
```

```
YIELD communityCount, modularity, modularities
```

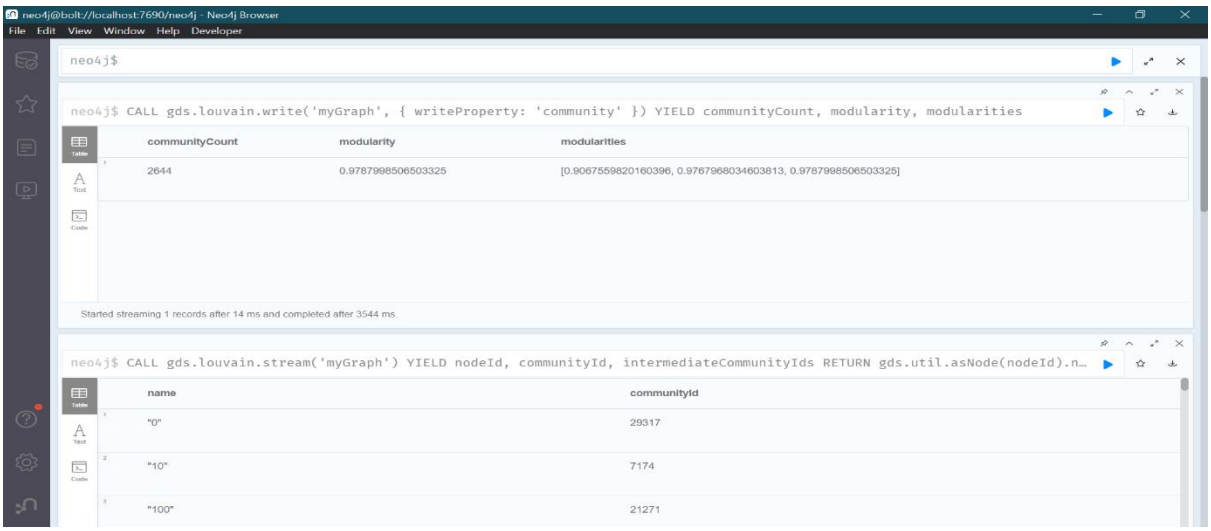


Figure 51: Louvain résultat en mode d'écriture.

5.2.3 Approche 03 : L'application des techniques de ML sur les KG

Cette approche est adaptée aux réseaux de systèmes de détection d'intrusion. Dans un premier temps, il explique comment extraire des fonctionnalités en utilisant des graphes de connaissances et de l'analyse d'apprentissage automatique. Par la suite, il expose les principales étapes du processus de détection, en commençant par le prétraitement des données afin d'améliorer leur qualité, puis en extraisant des fonctionnalités en utilisant les relations sémantiques entre diverses formes d'attaques. On insiste sur le fait que la transition vers un graphe de connaissances global constitue une avancée majeure dans la structuration et la compréhension des informations. Par la suite, on traite de l'alignement des caractéristiques afin d'améliorer l'interprétabilité. Dans la prochaine étape, on élabore un modèle qui intègre des classificateurs de machine et des techniques d'apprentissage profond afin de saisir la sémantique contextuelle. Finalement, on utilise des méthodes de classification multiclasse pour identifier les différents types d'attaque.

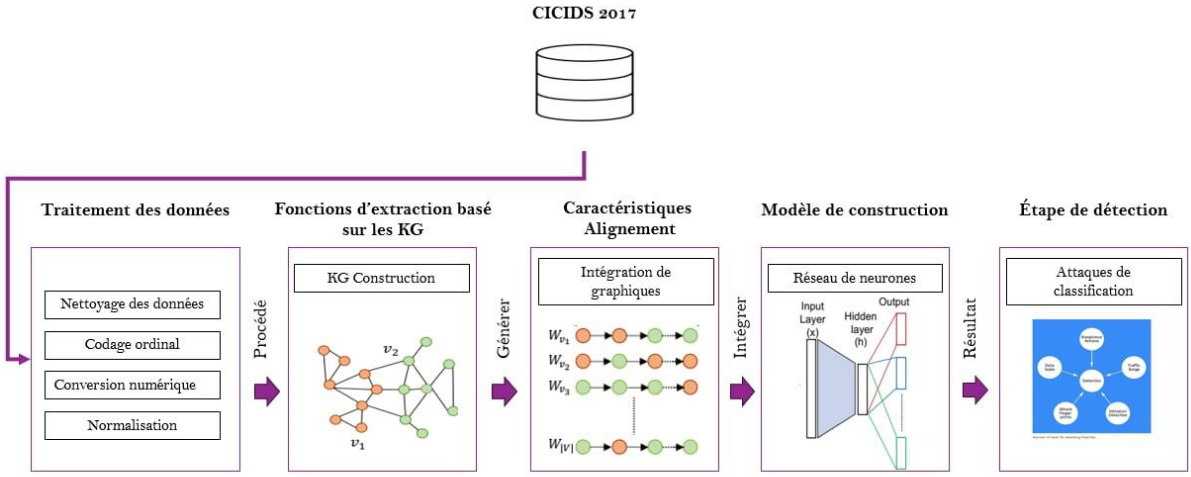


Figure 52: Architecture de composants de l'approche 03.

5.2.3.1 Modèle de construction

Étape 01 : Connexion avec py2neo

Nous utiliserons py2neo pour connecter le notebook Jupyter au serveur Neo4j. C'est une bibliothèque très simple utiliser pour connecter votre application Python à la base de données Neo4j, on l'obtienra en éxécutons "pip install py2neo". Ensuite, connecter notre programme au graphe.

Syntaxe :

```
graph = Graph("bolt://localhost:7690", auth=("neo4j", "1234567890"))
```

Étape 02: Manipulation des nœuds et relations

Accéder aux nœuds et aux relations et effectuer des opérations sur eux, comme se suit :
Syntaxe :

```
# Définition d'une fonction pour exécuter une requête et récupérer les
résultats
def run_query(query):
data = graph.run(query).data()
 return data
# Exemple de requête pour récupérer des données
query = """ MATCH (n) RETURN labels(n) AS type, count(n) AS
nombre_de_noeuds"""
```

```
query = """MATCH ()-[r]->()RETURN type(r) AS Type_de_Relation,
COUNT(r) AS Total """
```

Nous avons simplement soumis des requêtes Cypher à notre objet Graph, nous obtenons le décompte de chaque nœud et de chaque type de relation.

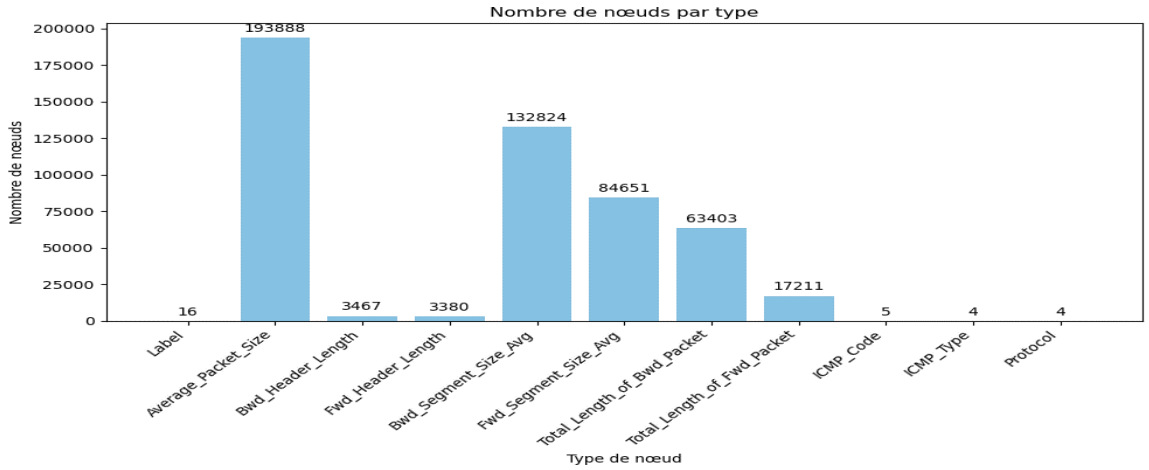


Figure 53: Nombre de nœuds par type.

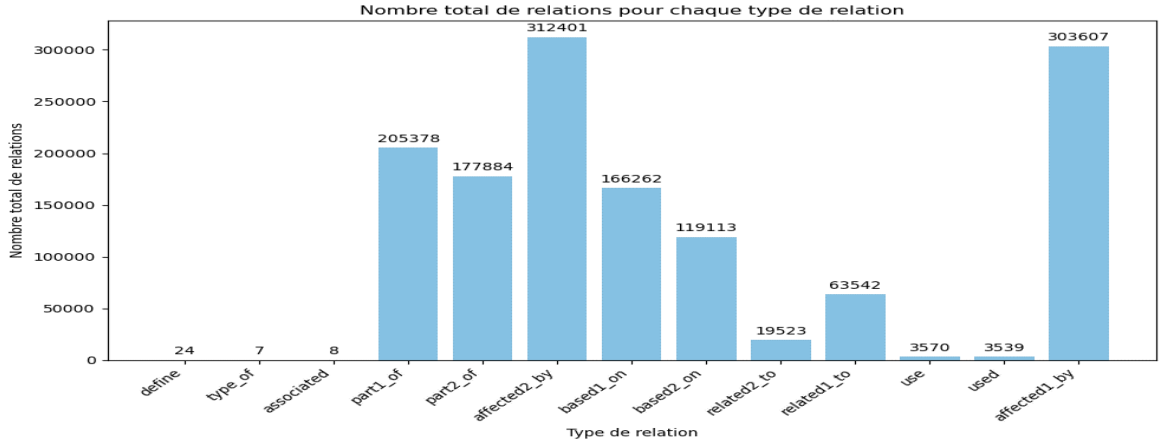


Figure 54: Nombre total de relation pour chaque type de relation.

Étape 03 : Extraction et stockage de données

Nous avons extrait les nœuds1 reliés par des relations aux nœuds2, puis nous avons stocké ces données dans un DataFrame pour une manipulation ultérieure, exemple :

Syntaxe :

```
query = """
```

```
MATCH (ict:ICMP_Type)-[rel:type_of]->(pro:Protocol) RETURN ict.name
as ICMP_Type, pro.name as Protocol ORDER BY ICMP_Type ASC
"""
data = run_query(query)
df1 = pd.DataFrame(data)
display(df1)
```

De manière détaillée, nous avons créé 13 DataFrames distincts, chacun contenant des données spécifiques obtenues à partir des nœuds et des relations dans notre base de données. Chaque DataFrame est conçu pour traiter et analyser un aspect particulier de nos données.

Étape 04 : Concaténation et exportation des données

Nous avons combiné les 13 DataFrames en un seul DataFrame en les concaténant, ce qui nous permet d'avoir une vue consolidée de toutes les données extraites et traitées à partir de notre base de données. Puis, nous avons procédé à la sauvegarde du DataFrame en l'exportant vers un fichier au format CSV.

Syntaxe :

```
# Initialisez le DataFrame concaténé avec le premier DataFrame df1
df_concatene = df1.copy()
# Utilisez une boucle pour concaténer les DataFrames suivants (de df2 à df13)
for i in range(2, 14):
 nom_df = 'df' + str(i)
 df_suivant = globals()[nom_df]
 df_concatene = pd.concat([df_concatene, df_suivant], ignore_index=True)
# Sauvegarder le DataFrame dans un fichier CSV
df_concatene.to_csv('Neo4jCICIDS2017.csv', index=False)
```

Étape 05 : Chargement et affichage des données

Nous chargeons les données à partir d'un fichier CSV nommé 'Neo4jCICIDS2017.csv' en utilisant la fonction read_csv() de la bibliothèque pandas. Une fois chargées, nous affichons les premières lignes du DataFrame résultant. Ceci nous permet d'avoir un aperçu des premières lignes du jeu de données pour une inspection initiale de ses caractéristiques et de sa structure.

	ICMP_Type	Protocol	Average_Packet_Size	Bwd_Header_Length	ICMP_Code	Total_Length_of_Bwd_Packet	Fwd_Header_Length	Total_Length_of_Fwd_Packet
0	0.0	0.0	1153.413223	2408.0	0.0	0.0	0.0	0.0
1	0.0	0.0	741.933333	332.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.000000	0.0	0.0	0.0	456.0	1342.0
3	0.0	0.0	219.857143	320.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.000000	0.0	0.0	5470.0	0.0	0.0

Figure 55: Les premières lignes du DataFrame résultant.

Notre nouveau jeu de données est de dimension (1099886, 11).

Attributs	Type
ICMP_Type	float64
Protocol	float64
Average_Packet_Size	float64
Bwd_Header_Length	float64
ICMP_Code	float64
Total_Length_of_Bwd_Packet	float64
Fwd_Header_Length	float64
Total_Length_of_Fwd_Packet	float64
Bwd_Segment_Size_Avg	float64
Fwd_Segment_Size_Avg	float64
Label	int64

Tableau 11: Caractéristiques présentes dans nouveau jeu de données graphique.

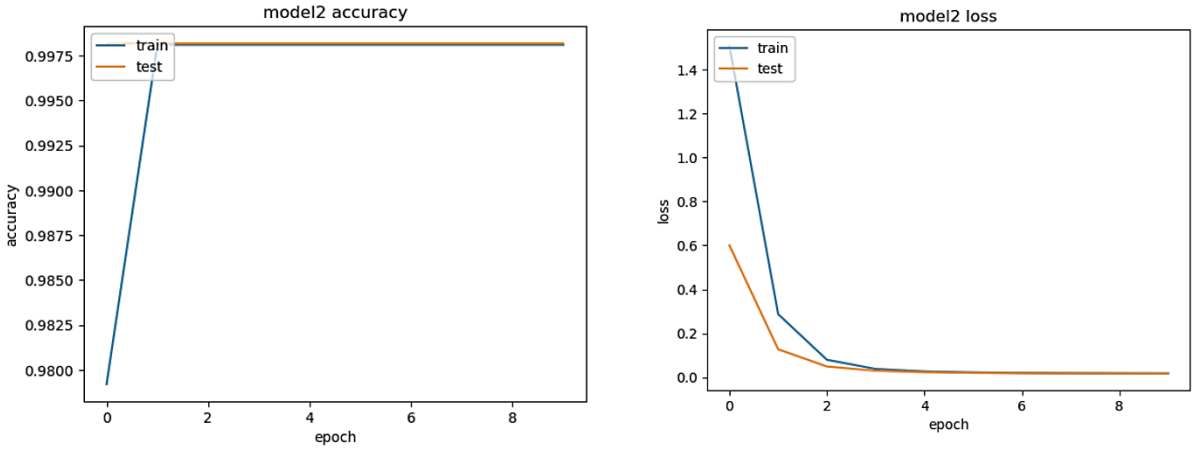
Etape 06 : Prétraitement et évaluation des performances
Nous avons appliqué le même processus de prétraitement, comprenant le nettoyage des données et la séparation en ensembles d'entraînement et de test, que celui utilisé dans l'approche 1. Le fractionnement a été effectué avec une répartition de 80% pour l'ensemble d'entraînement et de 20% pour l'ensemble de test.

Après cela, nous avons appliqué les algorithmes d'apprentissage automatique de la même manière que dans la première approche pour améliorer les mesures de performances (Exactitude, Précision, Rappel, F1-score) et nous avons obtenu les résultats suivants :

Expérience 01 : Nous avons procédé à l'application de l'algorithme CNN sur notre jeu de données graphique et voici les résultats :

Algorithme	Exactitude	Précision	Rappel	F1-Score
CNN	99,81%	99,62%	99,81%	99,72%

Tableau 12: Rendement des mesures d'évaluation pour le modèle(CNN) sur le nouveau jeu de données .



a) L'exactitude de train et test b) La perte de train et test

Figure 56: Visualisation des performances améliorées du modèle sur le nouveau jeu de données.

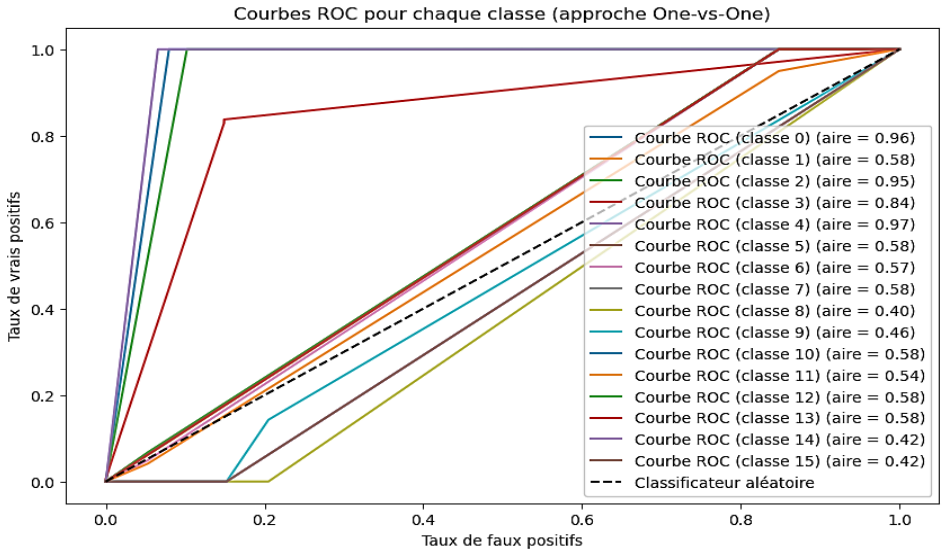


Figure 57: La courbe ROC de modele CNN de nouveau jeu de données .

Expérience 02 : l'application du KNN sur notre jeu de données et voici les résultats :

Algorithme	Exactitude	Précision	Rappel	F1-Score
KNN	99,77%	99,66%	99,77%	99,71%

Tableau 13: Rendement des mesures d'évaluation pour le modèle(knn) sur nouveau jeu de données

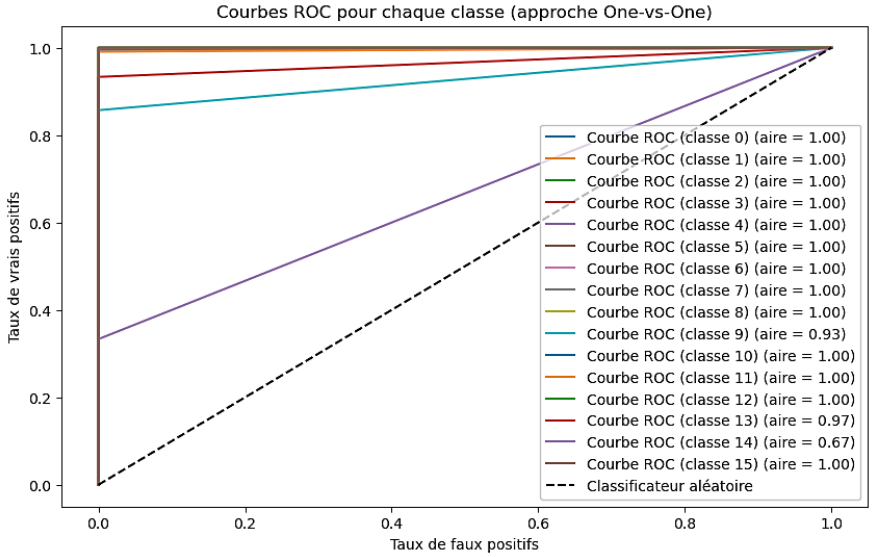


Figure 58: La courbe ROC de modele KNN de nouveau jeu de données.

Expérience 03 : en ce qui concerne cette expérience, et après l'application de l'algorithme de forêts aléatoire (RF) nous avons obtenu ces résultats :

Algorithme	Exactitude	Precesion	Rappel	F1-Score
RF	99,68%	99,68%	99,68%	99,68%

Tableau 14: Rendement des mesures d'évaluation pour le modèle (RF) sur nouveau jeu de données.

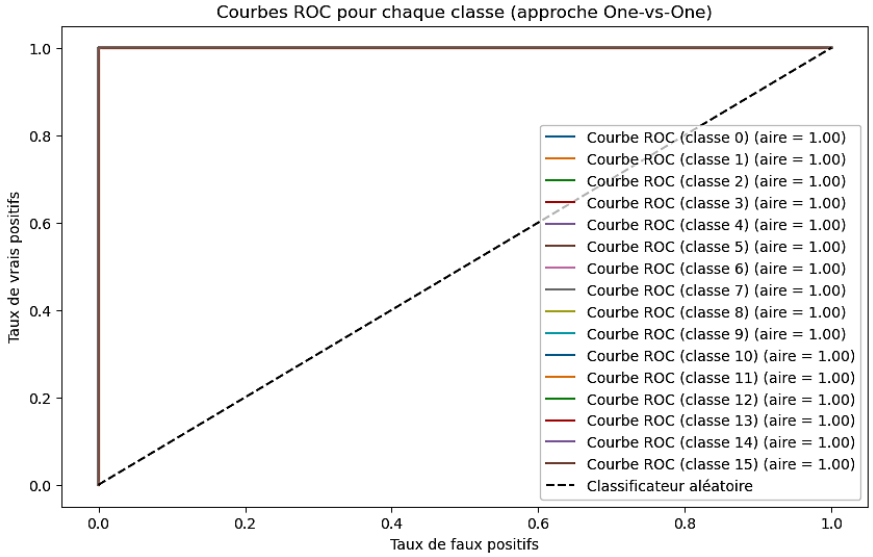


Figure 59: La courbe ROC de modele RF de nouveau jeu de données.

5.3 Comparaison des résultats

Expérience 1 : On peut voir que le modèle (CNN) sur le nouveau jeu de données graphiques atteint des performances exceptionnelles avec des scores très élevés dans toutes les mesures d'évaluation. Cela suggère que les améliorations apportées à l'ensemble de données ont été efficaces.

Expérience 2 : L'algorithme knn appliqué sur le nouveau jeu de données maintien des performances élevées dans toutes les mesures d'évaluation, bien qu'il montre une légère baisse du rappel par rapport au modèle de base. Cependant, cette baisse est compensée par une amélioration de la précision et du F1-score. Dans l'ensemble, l'algorithme knn semble être une amélioration significative et plus adaptée pour le nouveau jeu de données graphiques.

Expérience 3 : les mesures indiquent une performance très élevée et équilibrée du modèle Random Forest sur le nouveau jeu de données graphiques. L'exactitude, la précision, le rappel et le F1-Score sont tous très proches les uns des autres, ce qui suggère que le modèle RF est robuste et cohérent dans ses prédictions sur ce jeu de données. Comparé aux performances du modèle RF sur un ensemble de données antérieur (avec une exactitude de 99,99%), il semble y avoir une légère diminution de la performance, mais cela reste négligeable étant donné les scores déjà très élevés.

5.4 Conclusion

Le dernier chapitre visait à proposer une solution capable de surmonter les limitations des systèmes de détection d'intrusion. Cette solution vise à améliorer les performances générales des IDS en accroissant la précision de détection et de catégorisation d'un large spectre d'attaques, tout en réduisant au minimum les fausses alarmes. Les expériences menées ont confirmé l'efficacité de cette approche, fournissant des résultats extrêmement satisfaisants.

Conclusion générale

Conclusion générale

Dans cette étude, nous avons plongé dans l'univers de la détection des intrusions dans les réseaux informatiques, une préoccupation de premier plan en matière de sécurité informatique. Notre exploration a débuté par une analyse minutieuse des systèmes de détection d'intrusions traditionnels et des algorithmes qui les soutiennent. Par la suite, nous avons orchestré trois expérimentations distinctes, chacune exploitant des méthodes de machine learning (ML) et de deep learning (DL) pour scruter les comportements suspects dans les données réseau.

Ces expériences ont sollicité des algorithmes de renom tels que Random Forest (RF), k-plus proches voisins (KNN) et les réseaux de neurones convolutionnels (CNN), reconnus pour leur efficacité dans des contextes similaires. Malgré les avancées prometteuses constatées avec chaque approche, nous avons été confrontés à des limites en termes de capacité de traitement de nos machines, ce qui a entravé notre aptitude à gérer de vastes volumes de données. Toutefois, malgré ces défis, nous avons su tirer des conclusions éclairées grâce à des stratégies de gestion de données judicieuses et à une exploitation optimale des ressources à notre disposition.

Ces résultats mettent en lumière l'importance cruciale de la sélection et du traitement adéquats des données dans l'élaboration de systèmes de détection des intrusions fiables et performants. En conclusion, cette étude offre des perspectives essentielles pour l'évolution continue des systèmes de détection des intrusions, tout en soulignant les défis inhérents à la manipulation de vastes ensembles de données dans ce domaine vital de la sécurité informatique.

Bibliographique

[1] Bace, R. G. (2000). Intrusion Detection. Sams Publishing.

[2] Bedreddine Imad Eddine, & Wissam Brahmi. (2020). Implémentation de politiques de sécurité réseaux nCISCO. Récupéré de http://dspace1.univ-tlemcen.dz/

[3] René. (n.d.). Sécurité informatique - Les techniques d'attaque. Le blog de René. Récupéré dehttps://www.rene-reyt.fr/documents/informatique/petit-resume-de-securite-informatique/securite-informatique-les-techniques-dattaque/

[4] KORTI, S. M. M., & MEDIANI, M. N. E. (2022). Intrusion Detection System Using Machine Learning Techniques. Récupéré de http://dspace.univ-tlemcen.dz/

[5] Rouzaud-Cornabas, J. (2010). Formalisation de propriétés de sécurité pour la protection des systèmes d'exploitation (Thèse de doctorat). Université d'Orléans.

[6] Riquet, D. (2015). Une architecture de détection d'intrusions réseau distribuée basée sur un langage dédié.

[7] Futura Sciences. (n.d.). Antivirus. Récupéré de https://www.futura-sciences.com/tech/definitions/informatique-antivirus-10999/

[8] Hodo, E., Bellekens, X., Hamilton, A., et al. (2017). Shallow and deep networks intrusion detection system: A taxonomy and survey. ArXiv preprint arXiv:1701.02145.

[9] Hiet, G. (2008). Détection d'intrusions paramétrée par la politique de sécurité grâce au contrôle collaboratif des flux d'informations au sein du système d'exploitation et des applications: mise en œuvre sous Linux pour les programmes Java (Thèse de doctorat). Université Rennes 1.

[10] Debock-Marcant. (n.d.). [wapiti.enic]. Récupéré de http://wapiti.enic.fr/Commun/ens/peda/options/ST/RIO/pub/exposes/exposesrio2005ttnfa2006/debock-marcant/lien.html

[11] Brahima Embarka, B., & Selyna, A. (2017). Mise en place d'une solution de détection d'intrusion (Thèse de doctorat). Université Mouloud Mammeri.

[12] Outscale. (2022). Les trois piliers de la donnée. Récupéré de https://blog.outscale.com/les-trois-piliers-de-la-donnee/janvier2022

[13] Krim, S. M., & Yahlali, A. (2023). Anomaly- Intrusion Detection Systems based on CSE-CIC-IDS2018 Dataset using Deep Learning Model. Récupéré de http://dspace.univ-tlemcen.dz/

[14] Shajihan, N. (2020). Classification of stages of Diabetic Retinopathy using Deep learning.

[15] McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1956). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. Récupéré de https://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html

[16] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[17] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489.

[18] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.

[19] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

[20] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. ACM Computing Surveys (CSUR), 31(3), 264-323.

[21] Jolliffe, I. T. (2002). Principal component analysis. Springer.

[22] Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press.

[23] Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. R News, 2(3), 18–22.

[24] Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). Applied Linear Statistical Models.

[25] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. John Wiley & Sons.

[26] Régression linéaire vs logistique | Régression linéaire et logistique | Haut-parleur de données (datapeaker.com)

[27] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Applied Statistics, 28(1), 100-108. doi:10.2307/2346830

[28] Dalila, G. H. A. L. E. M., & Yousra, Z. E. G. A. D. I. (2021). Identification des appareils électriques basée sur les modèles GMM (Doctoral dissertation, Faculté des Sciences et Technologies).

[29] LeCun, Y. (2015). L'apprentissage profond, une révolution en intelligence artificielle. La lettre du Collège de France, 41. Récupéré de http://journals.openedition.org/lettre-cdf/3227

[30] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning. MIT press Cambridge.

[31] Tutorial for Recurrent Neural Network. (n.d.). Récupéré de https://www.datacamp.com/tutorial/tutorial-for-recurrent-neural-network

[32] Chervinskii, P. (Travail personnel). (2016). [Nom de l'image]. Récupéré de https://commons.wikimedia.org/w/index.php?curid=45555552

[33] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[34] Bondy, J. A., & Murty, U. S. R. (2008). Graph theory. Springer Science & Business Media.

[35] Frame, A., & Blumenfeld, Z. (2022). Graph Data Science For Dummies Neo4j 2nd Edition.

[36] Hubert, N., et al. (2022). Vers un système de recommandation explicable pour l'orientation scolaire.

[37] Gandon, F. (2021). Dessine-moi un graphe de connaissances ! Binaire.

[38] Zhou, H., Shen, T., Liu, X., Zhang, Y., Guo, P., & Zhang, J. (2020). Survey of Knowledge Graph Approaches and Applications. Journal on Artificial Intelligence, 2(2), 89–101. https://doi.org/10.32604/jai.2020.09968.

[39] Graphe de connaissance. (n.d.). Récupéré de https://www.jean-delahousse.net/graphe-de-connaissance-ontologie-vocabulaires-controles/

[40] Crié, D. (2003). De l'extraction des connaissances au Knowledge Management. Revue française de gestion.

[41] Les graphes de connaissance, incontournables pour l'intelligence artificielle. (n.d.). Récupéré de https://www.smals.be/nl/content/les-graphes-de-connaissance-incontournable-pour-lintelligence-artificielle

[42] Hofer, M., Obraczka, D., Saeedi, A., Köpcke, H., & Rahm, E. (2023, October 11). Construction of Knowledge Graphs: State and Challenges. arXiv. http://arxiv.org/abs/2302.11509

[43] Understanding Linked Data Formats: RDF/XML vs Turtle vs N-Triples. (n.d.). Récupéré de https://medium.com/wallscope/understanding-linked-data-formats-rdf-xml-vs-turtle-vs-n-triples-eb931dbe9827

[44] Daoudi, A. (2020). Vers la construction des graphes de connaissances à partir des textes. Université Dr. Tahar Moulay Saida.

[45] Hogan, A., et al. (2021). Knowledge graphs. ACM Computing Surveys (Csur).

[46] The Future of AI and Machine Learning with Knowledge Graphs. (n.d.). Récupéré de https://neo4j.com/blog/future-ai-machine-learning-knowledge-graphs/

[47] Hodler, A. E., Needham, M., & Graham, J. (2021). Artificial Intelligence & Graph Technology: Enhancing AI with Context & Connections.

[48] Needham, M., & Hodler, A. E. (2019). Graph algorithms: practical examples in Apache Spark and Neo4j. O'Reilly Media.

[49] Abdulrahman, A. A., & Ibrahem, M. K. (2020). Toward constructing a balanced intrusion detection dataset based on CICIDS2017. Samarra Journal of Pure and Applied Science, 2(3), 132-142.

[50] Kaggle, CICIDS2017 (kaggle.com), Consulté le 02/01/2024 .

[51] Anaconda, Distribution | Anaconda, Consulté le 02/01/2024.

[52] Geeksforgeeks, Python Tutorial | Learn Python Programming (geeksforgeeks.org), Consulté le 05/01/2024.

[53] Guia, J., Soares, V. G., & Bernardino, J. (2017, April). Graph Databases: Neo4j Analysis. In *ICEIS (1)* (pp. 351-356).

[54] Neo4j, Learn Why Cypher is the Leading Language for Graph Databases (neo4j.com), Consulté le 01/02/2024.

[55] TensorFlow, Consulté le 05/01/2024.

[56] Keras, Keras: Deep Learning for humans,Consulté le 24/01/2024.

[57] scikit-learn: machine learning in Python — scikit-learn 1.4.1 documentation. Consulté le 03/02/2024.

[58] pandas documentation — pandas 2.2.1 documentation (pydata.org). Consulté le 03/02/2024

[59] NumPy Documentation. Consulté le 06/02/2024.

[60] Xu, M. (2021). Understanding graph embedding methods and their applications. SIAM Review, 63(4), 825-853.

Bibliographique