RÉPUBLIQUE DU SÉNÉGAL
Un Peuple – Un But – Une Foi



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR, DE LA RECHERCHE ET DE L'INNOVATION



UNIVERSITE CHEIKH ANTA DIOP DE DAKAR

****

FACULTE DES SCIENCES ET TECHNIQUES

****

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE SECTION INFORMATIQUE

MÉMOIRE DE MASTER INFORMATIQUE

Spécialité : Réseaux et Télécommunications

Option : Réseaux, Systèmes et Services

Sujet : Évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements

Présenté et soutenu publiquement par :

Mme. Adama BA

Encadré par Professeur Chérif DIALLO

Le : 15 / 02 / 2025

Jury

Pr. Karim KONATE	Professeur Titulaire, UCAD	Président
Dr. Mohamed Ould DEYE	Maître Assistant , UCAD	Membre
M. Mohamed Lamine DIOP	Professionnel, UCAD	Membre
Pr. Cherif DIALLO	Professeur Titulaire, UGB	Encadreur

Année universitaire 2023 – 2024

A la mémoire de :

Tous ceux qui ont contribué à mon éducation, à ma formation et à ma réussite et

qui ne sont plus là malheureusement :

- *A mon cher père **Amadou Ndao BA** pour l'encouragement, le soutien et les prières qu'il m'a accordés.*
- *A ma chère grande mère **Hawa Seydou TOURE** une amie, une confidente, une conseillère, une idole pour son amour sa tendresse et ses prières qu'elle ne cessait de formuler pour ma réussite.*
- *A mon cher oncle **Ousmane Samba BA** qui restera toujours dans nos cœurs.*

Paix à leur âme(ALLAH y rahmou warhamhou) et que ALLAH les accueille au paradis. Amine

Dédicaces

Gloire à Allah Seigneur des mondes, le Tout Puissant pour Sa grâce, Son amour infini, Son assistance, Sa miséricorde grâce à qui j'ai pu arriver à réaliser ce modeste travail que je dédie pour témoin ma profonde gratitude à :

Ma mère ***Lola Mariam OLADAPO BA*** *et mon père* ***Amadou NDAO BA*** *en guise de reconnaissance et de gratitude pour leur amour, leur patience, leur soutien, leur encouragement et aux sacrifices qu'ils ont consentis à mon égard, pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance.*

*Mes oncles (****Aba BA, Cheikh Tidiane BA, Mouhamadal Habib BA****) et à mes tantes****(Farmata TOURE, Ndeye Ami NDIAYE, Coumba THIOUB, Khady SEYE, Molel BA)***

Mon cher époux, mon ami, mon frère, ***Moustapha KANE****, dans les moments où j'ai faibli, tu as été ma force ,dans mes moments de doute tu étais ma quiétude, je veux prendre un moment pour te dire à quel point je suis reconnaissante pour tout ce que tu fais pour moi, merci pour ta patience et ton amour inébranlables, merci d'être mon compagnon constant et fidèle, ta capacité à rester calme et aimant fait toute la différence merci aussi pour le soutien et l'encouragement qu'****ALLAH*** *nous donne une progéniture pieuse.*

Mon cher frère ***Mouhamadou Oluwafémi BA*** *mon booster la lumière qui éclaire mon chemin je t'adore ainsi que mes chères sœurs(****Hawa BA****,* ***Aishat BA****,* ***Dieynaba BA ,Amina ,BB Maye, Hawa Cheikh, Hawa Malal, Aicha*** *)*

Mes ***cousines Hawa Molel DIOP,Hawa SY, Diary SY, Dieynaba DIALLO,Coumbel SY, Coumba Aw****, mes cousins* ***Cheikh bou ndao ,Papa, ABA, Bibou***

Mes chères amies (***Ndeye Aminata SARR DIALLO****,* ***Rama BA DIAKHATE*** *,* ***Abibatou MBAYE****,* ***Ndèye Fatou DIASSE KEBE****,* ***Mame Khar NDIAYE****,* ***Ami DIONE BABOU****)*

Diarra DIOP KANE *qui est une personne spéciale qui m'a toujours soutenu soit heureuse auprès de* ***Omar Kane*** *alias (cours) qu'Allah ta'âla vous accorde une vie remplie d'amour et de bienfaits.*

Ma belle-famille merci de m'avoir accueilli dans votre famille avec autant d'amour mention spécial à ma belle-mère ***Coumba BEYE*** *dite* ***Fatou SEYE*** *pour son amour, son respect elle m'a toujours considéré comme sa fille je prie Allah pour qu'il lui accorde longue vie avec une santé de fer et mon beau-père* ***Dialor KANE*** *pour son amour.*

A la petite sœur de mon bien-aimé ***Coumba KANE*** *une personne adorable la douceur en question ai une vie prospère et plein de réussite.*

A mes beaux frère qui sont des personnes vraiment gentille m'ont facilité mon intégration au sein de la famille ***Amary KANE, Adama KANE, Mame Omar KANE, Mouhamed KANE « thiato thiat », Fallou KANE, Ngounda KANE.***

Tous les membres de la ***UFEMS-Juvénile(Union des Femmes Musulmanes du Sénégal)*** *spécialement* ***Fatima BITEYE, Maty SARR***

A ***Mouhamed DIALLO « AhmadAmina »*** *je tiens à exprimer toute ma gratitude pour tout ce que tu as fait pour nous.*

Tous les membres de la ***UFEMS-Juvénile(Union des Femmes Musulmanes du Sénégal)*** *spécialement* ***Fatima BITEYE, Maty SARR***

A toute personne qui occupe une place dans mon cœur et a toutes les personnes qui m'ont aidé à achever ce niveau...

Remerciement

Je remercie tout d'abord ALLAH le tout puissant de m'avoir donné la force, le courage, la volonté et la santé pour pouvoir accomplir ce travail.

*Je remercie mon encadreur « **M. chérif Diallo** » , pour la confiance qu'il m'a accordé en proposant ce travail, son encadrement et le temps qu'il m'a consacré durant la réalisation de ce projet.*

Je souhaite formuler mes remerciements les plus affectueux à ma famille surtout mes très chers parents et à ma belle famille

A mes chers frères et sœurs qui ont toujours été à nos côtés m'ont soutenu sur tous les plans sans oublier mes amies qui m'ont toujours encouragée au cours de la réalisation de mon travail.

*Mes remerciements s'adressent aux membres de la **CCI**(Comité Culturel Islamique)*

Mes remerciements s'adressent également aux membres de la section informatique pour les moyens qu'ils ont mis à notre disposition et sympathie à notre formation.

*Je tiens à remercier mes promotionnaires de la **M2RETEL 2023** pour leurs soutiens et encouragement .*

Je tiens à exprimer ma profonde gratitude à tous les professeurs qui ont contribué à ma formation durant tout mon cursus.

Enfin, j'adresse mes sincères remerciements et ma gratitude la plus profonde à tous ceux qui m'ont apporté leurs aides et qui ont contribué à l'élaboration de ce mémoire. Merci à tous et à toutes.

Résumé

La corrélation des événements informatiques automatise le processus d'analyse des événements de l'infrastructure informatique et d'identification des relations qu'ils entretiennent pour détecter les problèmes et découvrir leur cause profonde. Un outil de corrélation d'événements peut aider les organisations à superviser plus efficacement leurs systèmes et leurs applications tout en améliorant leur disponibilité et leurs performances. Les infrastructures IT d'entreprise génèrent d'énormes volumes de données dans divers formats. Ces données sont produites par des serveurs, des bases de données, des machines virtuelles, des appareils mobiles, des systèmes d'exploitation, des applications, des capteurs et bien d'autres composants réseau. Un événement est un élément de données qui fournit des informations sur un changement d'état survenu au sein de cette infrastructure . Un bon nombre de ces événements sont normaux et bénins, mais certains traduisent un problème au sein de l'infrastructure.

Ce mémoire porte sur l'Evaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements afin de répondre à ce problème. Un logiciel de corrélation des événements informatiques importe les données d'infrastructure et utilise le machine learning pour identifier des modèles et des relations, ces techniques permettent de localiser et de résoudre plus facilement les incidents et les interruptions, de superviser les performances et d'améliorer la disponibilité et la stabilité de l'infrastructure.

Le but de ce mémoire est de faire une généralité sur les logs, sur l'IA, sur les SIEM ainsi que la mise en place d'une solution adapté aux besoins de l'entreprise.

Mots clés : IT, SIEM, Logs, IA

Abstract

The correlation of IT events automates the process of analyzing IT infrastructure events and identifying their relationships to detect problems and discover root causes. An event correlation tool can help organizations to more effectively monitor their systems and applications while improving their availability and performance. Enterprise IT infrastructures generate huge volumes of data in various formats. This data is generated by servers, databases, virtual machines, mobile devices, operating systems, applications, sensors and many other network components. An event is a data item that provides information about a change in status within this infrastructure. Many of these events are normal and benign, but some reflect a problem within the infrastructure.

This thesis focuses on the evaluation of an artificial intelligence model for event correlation to address this problem. Computer event correlation software imports infrastructure data and uses machine learning to identify patterns and relationships, these techniques make it easier to locate and resolve incidents and interruptions, to monitor performance and improve infrastructure availability and stability.

The purpose of this thesis is to make a generality on logs, AI, SIEM and the implementation of a solution adapted to the needs of the company.

Keywords: IT, SIEM, Logs, IA

Tables des matières

A la mémoire de : i
Dédicaces ii
Remerciement iv
Résumé v
Abstract vi
Liste des figures x
Liste des tableaux xi
Glossaire xii
Introduction générale 1
Partie 1 : Cadre théoriques et méthodologiques 2
Chapitre 1 :Cadre théorique 3
1.1 Contexte 3
1.2 Problématique 4
1.3 Objectif 5
Chapitre 2 : Cadre méthodologique 6
2.1 Pertinence du sujet 6
2.2 Délimitation du sujet 6
2.3 Méthodologie de recherche 7
2.4 Difficultés rencontrées 7
Partie 2 : Etat de l'art et études comparatives 8
Chapitre 3 : Généralité sur les Logs et SIEM 9
3.1 Généralités sur le Logs 9
3.1.1 Définition d'un Log 9
3.1.2 Les types de logs 9
3.1.3 Catégorisations des logs 10
3.1.4 Contenue de base d'un message journal 11
3.1.5 Centralisation des logs 11
3.2 LE SIEM 13
3.2.1 Définition du SIEM 13
3.2.2 Rôle du SIEM 13
3.2.3 Fonctionnalités d'un SIEM 13
3.2.4 Avantages du SIEM 15

Chapitre 4 :Généralité sur l'IA 16
4.1 Définition de l'IA 16
4.2 Fonctionnalités de l'IA 16
4.3 Les différentes types d'IA 17
4.3.1 Intelligence artificielle étroite (ANI) / IA faible / IA étroite 18
4.3.2 Intelligence générale artificielle (AGI) / IA forte / IA profonde 19
4.3.3 Superintelligence artificielle (ASI) 20
Chapitre 5: Techniques de classification 21
5.1 L'apprentissage supervisé 21
5.1.1 Les réseaux neurones 22
5.1.2 Les arbres de décision 22
5.1.3 Les SVM*(*support-vector machine) 23
5.1.4 Le naive bayes 24
5.2 L'apprentissage non supervisé 25
5.2.1 K-Means*(* k-moyennes) 26
5.2.2 ACP(Analyse en Composantes Principales) 27
5.2.3 GMM (Gaussian Mixture Models) 27
5.2.4 Algorithme Apriori 28
5.2.5 Algorithme des K-Médoïdes 29
Chapitre 6 : Etude comparative des algorithmes de corrélation 32
6.1 Présentation de quelques techniques de corrélation 32
6.1.1 Pearson 32
6.1.2 Spearman 32
6.1.3 Kendall (Tau) 33
6.1.4 Corrélation Partielle 33
6.1.5 Information mutuelle 33
Partie 3 : Conception et mise en œuvre 35
Chapitre 7 : Mise en place de la solution 36
7.1 Architecture de déploiement 36
7.2 Maquette de test 38
7.3 Conception de la solution 39
7.3.1 Choix de la solution 39
7.3.2 Installation et configuration 39
7.3.1.1 Installation et configuration de TensorFlow 39

7.3.3 Interprétation du code 42
7.3.3 Interprétation des résultats 59
Conclusion et perspectives 63
Webographie 64

Liste des figures

Figure 1:apprentissage supervisé.....21
Figure 2:réseaux neurones.....22
Figure 3: arbres de décision.....23
Figure 4: Support-vector machine(SVM).....24
Figure 5: Le naive bayes.....25
Figure 6:apprentissage non supervisé.....26
Figure 7:K-Means.....27
Figure 8 : ACP(Analyse en Composantes Principales).....27
Figure 9: GMM (Gaussian Mixture Models).....28
Figure 10 : Algorithme Apriori.....29
Figure 11: Algorithme des K-Médoïdes.....29
Figure 12:Les composantes de l'architecture.....36
Figure 13:Maquette de test.....38
Figure 14: Format Dataset.....39
Figure 15 :Visualisation des erreurs de reconstruction.....50
Figure 16: erreurs de reconstruction pour la caractéristique taille des paquets.....52
Figure 17: distribution d'une variable.....53
Figure 18: epoch_learning_rate.....55
Figure 19: epoch_loss.....57
Figure 20: epoch_mae.....59
Figure 21: distribution des anomalies.....61

Liste des tableaux

Tableau 1: Comparatif des Techniques de Classification Algorithmes supervisés 30
Tableau 2 : Comparatif des Techniques de Classification :Algorithmes non Supervisés 30
Tableau 3 : Comparatif des Techniques de Classification :Algorithmes Hybrides 31
Tableau 4: comparatif des Techniques de corrélation 34

Glossaire

ACP: Analyse en Composantes Principales

AE: Auto-Encodeurs

AGI : Artificial General Intelligence

ANI : Artificial Narrow Intelligence

ASI : Artificial Super Intelligence

DDN: Deep Neural Network

GMM: Gaussian Mixture Models

IA : Intelligence Artificielle

IoT : Internet of Things

ML: Machine Learning

MAE : Mean Absolute Error

NLP : Natural Language Processing

P**YNQ** : Python Productivity for Zynq Zynq (famille de System-on-Chip de Xilinxintégrant FPGA et processeur)

SEM: Security Event Management

SI: System Information

SIEM: Security Information & Event Management

SIM : Security Information Management

SMS : Short Message Service

SVM: support-vector machine

UEBA: User and Entity Behavior Analytics

Introduction générale

Dans le monde moderne, la quantité de données générées par les systèmes numériques, les capteurs, les transactions financières, et d'autres sources est en constante augmentation. Analyser et interpréter ces vastes ensembles de données pour en extraire des informations significatives est devenu un défi majeur. L'évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements consistent à détecter, analyser et interpréter des relations entre divers événements enregistrés dans un système. Un événement peut être défini comme une occurrence particulière enregistrée à un moment donné, telle qu'une connexion réseau, une alerte de sécurité, ou toute autre action pertinente dans le contexte d'un système. L'évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements surpassent largement les méthodes traditionnelles en termes de capacité à gérer des volumes massifs de données, d'identifier des relations complexes, de fournir des analyses prédictives, et d'améliorer l'efficacité opérationnelle. Leur application dans divers domaines permet de découvrir des insights précieux, de prévenir des incidents, et d'optimiser les processus, ce qui est crucial dans un monde de plus en plus axé sur les données. L'adoption de ces techniques est donc essentielle pour les organisations qui cherchent à rester compétitives et à tirer pleinement parti des opportunités offertes par les big data et l'IA

Le but principal est de découvrir des relations significatives et souvent non linéaires entre les événements pour améliorer la détection des anomalies, la prédiction des comportements futurs, et la prise de décisions basée sur des données.

Dans ce contexte ce mémoire de fin d'étude se concentrera sur l'évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements. Ce présent document comprend trois parties : Partie1 : Cette partie est composée d'une part du contexte, problématique, objectifs de notre sujet et d'autre part du cadre méthodologique.

Partie2 : Etat de l'art et étude comparative : cette partie contient une généralité sur les logs et SIEM, une généralité sur l'IA, les techniques de classification et enfin d'une étude comparative des algorithmes de corrélation.

Partie 3 : Conception et mise en œuvre : Elle est la dernière partie de notre mémoire et est la phase de mise en œuvre, configuration et test de notre solution.

Partie 1 : Cadre théoriques et méthodologiques

Chapitre 1 :Cadre théorique

1.1 Contexte

Avec l'augmentation exponentielle des volumes de données générés par les systèmes informatiques, les entreprises et organisations cherchent des moyens de comprendre les relations entre divers événements. Les techniques de corrélation traditionnelles, basées sur des approches statistiques simples, ne sont souvent pas suffisantes pour capturer les relations complexes et dynamiques dans ces données. C'est ici que les techniques basées sur l'intelligence artificielle (IA) prennent toute leur importance. Les approches classiques de corrélation, comme l'analyse de Pearson ou les coefficients de Spearman, sont souvent inadaptées aux environnements modernes caractérisés par des volumes massifs de données et des relations non linéaires. L'émergence de l'intelligence artificielle offre des outils puissants, capables de détecter des motifs complexes, de s'adapter à des données non structurées et de fonctionner en temps réel. Ces techniques reposent sur des algorithmes d'apprentissage supervisé, comme les forêts aléatoires et les réseaux neuronaux profonds, ainsi que sur des approches non supervisées, comme les autoencodeurs et les algorithmes de clustering. Elles permettent de capturer des relations complexes entre des événements, qu'ils soient temporels, contextuels ou spatiaux. Bien que l'IA puisse améliorer la corrélation d'événements, plusieurs défis et limites doivent être pris en compte : faux positifs et faux négatifs, explicabilité des résultats, besoins en données d'entraînement, consommation de ressources, dépendance aux mises à jour.

Le SIEM, ou Système de Gestion des Informations et des Événements de Sécurité (Security Information and Event Management en anglais), est une solution technologique utilisée pour collecter, agréger, analyser et corréler les données de sécurité provenant de diverses sources au sein d'un environnement informatique. Son objectif principal est de fournir une vue d'ensemble de la posture de sécurité d'un système, d'une infrastructure réseau ou d'une application, ainsi que de détecter et de répondre aux incidents de sécurité. Il centralise les évènements et permet de détecter les menaces connues et inconnues en temps réel via du Machine Learning, de l'intelligence de sécurité et de l'analyse comportementale des utilisateurs et des entités (UEBA User and Entity Behavior Analytics en anglais). Il surveille les accès non autorisés en corrélant de multiples événements pour déclencher une alerte. Pour permettre aux équipes sécurité d'agir plus rapidement et efficacement, les alertes sont ensuite

hiérarchisées en fonction de leur degré de criticité. L'intégration de l'IA dans les SIEM permet d'étendre leurs capacités grâce à des technologies comme l'apprentissage automatique (machine learning), l'analyse comportementale et le traitement du langage naturel (NLP).

Dans ce mémoire, nous proposons d'explorer l'évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements pour améliorer la détection des anomalies dans les réseaux informatiques. Notre objectif est de démontrer que ces approches surpassent les méthodes classiques en termes de précision, de scalabilité et de robustesse. C'est dans la perspective de pouvoir gagner du temps en les analysant qu'on utilise l'Evaluation d'un modèle d'Intelligence Artificielle pour la Corrélation d'Evénements, cette technique permet de tirer parti de la puissance de l'IA pour automatiser la détection de corrélations complexes dans de grandes quantités de données, ce qui peut aider les organisations à prendre des décisions plus éclairées et à réagir plus rapidement aux événements critiques.

1.2 Problématique

Depuis quelques années le monde subit un changement suite à une évolution à grande vitesse dans l'utilisation et la perfection des nouvelles technologies de l'information et de la communication qui sont actuellement visible presque dans tous les domaines de la vie et font parties d'un système d'information et de production.

L'intelligence artificielle est un phénomène qui touche certes le monde des entreprise elle est d'une importance considérable car elle améliore les performances et la productivité de l'entreprise en automatisant des processus ou des tâches qui nécessitaient auparavant des ressources humaines.

L'évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements est un processus qui utilise différentes méthodes pour mesurer la performance du modèle.

Ceci conduit donc à poser les questions suivantes :

Dans quelle mesure l'évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements peuvent-elles être utilisées pour améliorer la détection précoce d'incidents à partir de multiples sources de données ?

Dans quelle mesure l'intégration de techniques d'apprentissage peut-elle améliorer la précision , la scalabilité et la performance d'un système de corrélation d'événements basé sur l'IA face à l'augmentation continue du volume de données et de la complexité des patterns ?

1.3 Objectif

L'objectif de notre mémoire est de pouvoir réaliser les points suivants :

- Mettre en place des algorithmes de détection d'anomalies
- Améliorer la corrélation par les techniques d'apprentissage
- Faire une étude comparative des algorithmes de corrélation et sur les techniques de corrélation
- Détecter les comportements anormaux
- Déployer, tester et valider la solution proposée.

Chapitre 2 : Cadre méthodologique

2.1 Pertinence du sujet

La pertinence de l'évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements est justifiée par les nombreux défis auxquels les entreprises font face en matière de :

- Volume de données : les systèmes informatiques génèrent des quantités massives de logs et de métriques de monitoring. L'IA permet de traiter et d'analyser ces big data (volume de données)efficacement.
- Complexité : les SI sont de plus en plus distribués et interconnectés. Les chaînes de causalité entre événements sont complexes. L'IA peut détecter des corrélations non évidentes.
- Rapidité : l'IA permet d'automatiser et d'accélérer grandement l'analyse de corrélation, là où un humain serait limité.
- Précision : grâce à l'apprentissage, les algorithmes d'IA peuvent détecter des anomalies et des corrélations avec une granularité très fine.
- Généricité : une fois entraînés, les modèles d'IA sont réutilisables pour n'importe quel flux d'événements informatiques.
- Proactivité : en anticipant certains incidents, l'IA permet de passer d'une posture réactive à prédictive.

L'IA apporte donc de la précision, de la rapidité et de l'extensibilité à l'analyse de corrélation, difficilement atteignables avec des approches conventionnelles.

2.2 Délimitation du sujet

Notre étude porte sur l' Évaluation d'un Modèle d'Intelligence Artificielle pour la Corrélation d'Evénements. Elle permettra de classifier les données ou évènements en fonction de leurs types ou en fonction de la phase de menace, ce qui conduit au gain du temps, à l’efficacité et à la rapidité.

2.3 Méthodologie de recherche

La recherche documentaire à travers les sites internet, les exemples de mémoires et les ouvrages consultés et l'observation sont les techniques d'investigations qui nous ont permis de réaliser ce mémoire intitulé évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements.

2.4 Difficultés rencontrées

Durant ce mémoire, nous avons rencontré des difficultés dans un premier temps dans de la partie recherche pour trouver les informations fiables, les recherches pour la comparaison des algorithmes de corrélation, le choix du type données à exploiter avec open data ça pas été faciles.

Dans un deuxième temps, il y'a eu des difficultés dans la compréhension et l'exploitation du données choisi. Certes simple d'utilisation, il nous a pris un peu de temps pour comprendre certaines fonctionnalités car étant un nouveau pour nous. Une autre difficulté a été la partie collecte de données.

Partie 2 : Etat de l'art et études comparatives

Chapitre 3 : Généralité sur les Logs et SIEM

3.1 Généralités sur le Logs

3.1.1 Définition d'un Log

Un log est un fichier généré par ordinateur qui capture l'activité au sein du système d'exploitation ou des applications logicielles. Le fichier journal documente automatiquement toutes les informations conçues par les administrateurs système, notamment : les messages, les rapports d'erreurs, les demandes de fichiers et les transferts de fichiers. L'activité est également horodatée, ce qui aide les professionnels de l'informatique et les développeurs à comprendre ce qui s'est passé ainsi que le moment où cela s'est produit. [29]

Les logs ont un intérêt et une importance cruciale en informatique, car il s'agit là de savoir ce qu'il s'est passé sur un ensemble d'applications ou systèmes pour pouvoir, par exemple :

- Expliquer une erreur, un comportement anormal, un crash sur un service comme un service web ;
- Retracer la vie d'un utilisateur, d'une application, d'un paquet sur un réseau sur les logs d'un proxy et des éléments réseau par exemple ;
- Comprendre le fonctionnement d'une application, d'un protocole, d'un système comme les étapes de démarrage d'un service SSH sous Linux;
- Être notifié d'un comportement, d'une action, d'une modification tel qu'une extinction ou un démarrage système ;
- etc.

3.1.2 Les types de logs

Chaque technologie génère un type de données différent et chaque technologie collecte ces données dans son propre fichier logs (journal). Pour cette raison, il existe de nombreux types de logs (journaux), notamment :

- Logs d'événements : est un log de haut niveau qui enregistre des informations sur le trafic et l'utilisation du réseau, telles que les tentatives de connexion, les tentatives de login échouées et les événements d'application.
- Logs serveur : est un document texte contenant un enregistrement des activités liées à un serveur spécifique pendant une période spécifique.

- Logs système : ou Syslogs, est un enregistrement des événements du système d'exploitation. Il comprend les messages de démarrage, les modifications du système, les arrêts inattendus, les erreurs et les avertissements, ainsi que d'autres processus importants. Windows, Linux et MacOs génèrent tous des Syslogs.
- Logs d'autorisation et d'accès : comprennent une liste de personnes ou de robots accédant à certaines applications ou à certains fichiers.
- Logs de modification : comprennent une liste chronologique des modifications apportées à une application ou à un fichier.
- Logs de disponibilité : permettent de suivre les performances, le temps de fonctionnement et la disponibilité du système.
- Logs de ressources : Les logs de ressources fournissent des informations sur les problèmes de connectivité et les limites de capacité.
- Logs de menaces : contiennent des informations sur le trafic des systèmes, des fichiers ou des applications qui correspondent à un profil de sécurité prédéfini dans un pare-feu.

3.1.3 Catégorisations des logs

Il existe actuellement plusieurs manières de catégoriser les logs ; selon leur type, selon leur degré de criticité ou selon leur niveau de journalisation.

Les logs Windows par exemple sont divisés sur quatre catégories : les logs système, les logs d'installation, les logs de sécurité et les logs d'application. Les évènements systèmes contiennent tous les évènements relatifs aux pilotes de périphériques et au système d'exploitation. Les évènements d'installation contiennent à leurs tours les logs relatifs aux paramètres de configuration du système d'exploitation. Les évènements de sécurités contiennent les logs d'authentifications, les logs des règles d'audit ainsi ceux de l'accès aux ressources systèmes. Les évènements d'application contiennent les logs relatifs aux applications installées sur le système.

Les logs sont catégorisés ainsi selon leur degré de criticité en commençant par DEBUG, INFO,

WARN, ERROR jusqu'on arrive au niveau FATAL.

- Le niveau de journalisation DEBUG désigne les évènements d'information les plus utiles pour déboguer une application.

- Le niveau de journalisation INFO désigne les logs qui concernent la progression de fonctionnement de l'application. Le but est de fournir une vision permettant de comprendre quel utilisateur, a fait quoi, à quelle heure, sur quelles données, avec quelles saisies, a obtenu quel résultat, etc.
- Le niveau de journalisation WARN désigne les situations potentiellement dangereuses.
- Le niveau ERROR contient les évènements des erreurs d'application mais qui se génèrent permettent à l'application de continuer à s'exécuter.
- Le niveau FATAL contient les évènements des erreurs fatals et qu'empêchent l'application à continuer de s'exécuter.

3.1.4 Contenue de base d'un message journal

Un message de log contient généralement les informations suivantes :

- Date et heure : La date et l'heure à laquelle le message de log a été généré.
- Niveau de log : Le niveau de log associé au message, comme DEBUG, INFO, WARN, ERROR, FATAL.
- Identifiant de processus ou de thread : L'identifiant du processus ou du thread qui a généré le message de log.
- Source du message : Le nom de la classe ou de la méthode qui a généré le message de log.
- Message : Le contenu du message de log. Il peut s'agir d'une description de l'événement ou d'une erreur spécifique.
- Données supplémentaires : Certaines implémentations de journalisation peuvent inclure des informations supplémentaires telles que les variables d'environnement, les paramètres d'appel de méthode, les valeurs de retour, les informations de trace de pile, etc.

Il est important de noter que cette liste n'est pas exhaustive, et que les implémentations de journalisation peuvent varier en fonction des besoins de l'application et de la plateforme utilisée.

3.1.5 Centralisation des logs

Aujourd'hui, la grande majorité des systèmes, applications et éléments réseau produisent des logs. Dans la quasi-totalité des cas, les logs sont stockés de manière locale sur la machine :

- Les journaux d'événements sous Windows stockent par défaut les logs des OS Windows (Clients et serveur) via l'observateur d'événement.
- Le fameux dossier "/var/log " sous linux qui stocke l'ensemble des logs systèmes et applicatifs sous Linux via syslog (mais pas que, cela dépend des distributions)
- Sous Cisco, quelques messages de logs sont inscrits directement dans le terminal, mais le reste des logs n'est pas sauvegardé, faute de stockage.

Ces informations venues de plusieurs dispositifs sont souvent difficiles à gérer individuellement, une idée de les centraliser permettra alors de faciliter cette analyse. Dans ce cadre, la centralisation des logs consiste simplement à mettre sur un même système, une même plateforme, l'ensemble des logs des systèmes, applications et services des machines environnantes.

La centralisation des logs peut donc nous aider à :

- **Détection rapide des problèmes** : En centralisant les logs de différents systèmes et applications, les administrateurs système peuvent facilement repérer des problèmes tels
  que des erreurs, des pannes ou des comportements anormaux, ce qui peut leur permettre
  de réagir rapidement avant que le problème ne se propage et n'affecte d'autres parties du système.
- **Résolution plus rapide des problèmes** : En centralisant les logs, les administrateurs système peuvent également accéder à une vue plus complète du système, ce qui peut leur permettre de détecter des tendances ou des modèles dans les événements qui peuvent aider à identifier la source d'un problème. Cela peut réduire le temps nécessaire pour résoudre un problème, car les administrateurs peuvent se concentrer rapidement sur les parties du système les plus susceptibles d'être affectées.
- **Amélioration de la sécurité** : La centralisation des logs peut également être utilisée pour surveiller les activités du système et détecter les incidents de sécurité. Les administrateurs peuvent rechercher des modèles d'activité malveillante, des tentatives de piratage ou des violations de sécurité et prendre des mesures pour y remédier avant qu'ils ne causent des dommages.
- **Amélioration des performances** : La centralisation des logs peut également aider les administrateurs à surveiller les performances du système, en identifiant les problèmes

de performance tels que les goulots d'étranglement ou les ressources surutilisées. Cela peut permettre aux administrateurs de prendre des mesures pour optimiser les performances du système, améliorant ainsi l'expérience des utilisateurs et la productivité globale de l'entreprise.[30]

3.2 LE SIEM

3.2.1 Définition du SIEM

SIEM est l'abréviation de 'Security Information & Event Management' est une solution qui combine des outils existants ; à savoir le SIM (Security Information Management) et le SEM (Security Event Management). Un SIEM doit donc avoir la capacité de gérer les évènements de sécurité, de les stocker ainsi de les analyser afin d'extraire les informations de sécurité depuis eux pour les utiliser dans la détection des menaces et des incidents de sécurité.[31]

3.2.2 Rôle du SIEM

La solution S.I.E.M. permet de surveiller des applications, des comportements utilisateurs et des accès aux données. A travers les fonctionnalités fournis par cette solution, il est donc possible de collecter, de mettre dans un format de normalisation, d'agréger, de corréler et d'analyser les données des événements issus des machines, systèmes et applications (pare-feu, systèmes de prévention d'intrusion, machines réseau, machines de sécurité, applications, bases de données, serveurs). Le principe d'un S.I.E.M. consiste à examiner depuis un guichet unique les données relatives à la sécurité de l'entreprise qui sont générées en de nombreux points. Cette approche facilite l'identification d'éventuels comportements inhabituels au sein du système informatique de l'entreprise. Concrètement, le S.I.E.M. permettra à une entreprise de centraliser toutes les informations de sécurité en un seul outil.[32]

3.2.3 Fonctionnalités d'un SIEM

Pour permettre au système d'identifier des événements anormaux, il est important que l'administrateur S.I.E.M. élabore en premier lieu un profil du système dans des conditions normales de fonctionnement. Au niveau le plus simple, un système S.I.E.M peut reposer sur des règles ou utiliser un moteur de corrélation statistique pour établir des relations entre les entrées du journal des événements [26]. Sur certains systèmes, un prétraitement peut intervenir au niveau des collecteurs. Seuls certains événements sont alors transmis au nœud d'administration centralisé. Ce prétraitement permet de réduire le volume d'informations

communiquées et stockées. Toutefois, cette approche comporte le risque de filtrer des événements pertinents de manière précoce. Les SIEM permettent :

- **La collecte** : Le principe de la collecte est de fournir au S.I.E.M. des données à traiter. Ces données peuvent être de nature diverse en fonction de l'équipement ou du logiciel, mais aussi être envoyées de manières tout à fait différentes.

On distingue deux modes de fonctionnement :

- **Mode actif** : la plupart des S.I.E.M. fonctionnent en déployant une multitude d'agents de collecte de manière hiérarchique sur les équipements à superviser. Ces agents ont pour fonction de récupérer les informations des appareils des utilisateurs, des serveurs, des équipements réseau, voire des équipements spécialisés de sécurité, tels que les pare-feu, ou encore les systèmes antivirus et anti-intrusions, et aussi des logiciels de sécurité et de les envoyer au S.I.E.M. Un élément de sécurité qui a été conçu nativement pour être un agent du S.I.E.M. est appelé une « sonde » ;
- **Mode passif** : le S.I.E.M. est en écoute directe sur les équipements à superviser. Pour cette méthode, c'est l'équipement ou le logiciel qui envoie des informations sans intermédiaire au S.I.E.M.
- **La normalisation** : Les informations collectées viennent d'équipements et logiciels hétérogènes ayant pour la plupart leurs propres moyens de formater les données. Cette étape permet d'uniformiser les informations selon un format unique pour faciliter le traitement par le S.I.E.M. La normalisation permet ainsi d'effectuer des recherches multi-critères, sur un champ ou sur une date. Ce sont ces événements qui seront enrichis avec d'autres données puis envoyés vers le moteur de corrélation.
- **L'agrégation** : L'agrégation est le premier traitement des évènements de sécurité. Il consiste en un regroupement d'évènements de sécurité selon certains critères. Ces critères sont généralement définis à travers des règles appelées « règles d'agrégation » et s'appliquent à des évènements ayant des similarités. Plusieurs règles de filtrage peuvent être appliquées. Ils sont ensuite agrégés selon les solutions, puis envoyés vers le moteur de corrélation.
- **La corrélation** : La corrélation correspond à l'analyse d'évènements selon certains critères. Ces critères sont généralement définis à travers des règles appelées « règles de corrélation ». Le but de cette étape est d'établir des relations entre évènements, pour ensuite pouvoir créer des alertes de corrélations, des incidents de sécurités, des

rapports d'activité. Les règles de corrélation permettent d'identifier un événement qui a causé la génération de plusieurs autres (un hacker qui s'est introduit sur le réseau, puis a manipulé tel équipement, etc). Elles permettent aussi de remonter une alerte via un courriel, SMS ou ouvrir un ticket si la solution S.I.E.M. est interfacée avec un outil de gestion de tickets

- **Le reporting** : Les S.I.E.M. permettent également de créer et générer des tableaux de bord et des rapports. Ainsi, les différents acteurs du système d'information, responsables sécurité du système d'information, administrateurs, utilisateurs peuvent avoir une visibilité sur le système d'information (nombre d'attaques, nombre d'alertes par jour...).
- **L'archivage** : Les solutions S.I.E.M. sont utilisées également pour des raisons juridiques et réglementaires. Un archivage à valeur probante permet de garantir l'intégrité des traces. Les solutions peuvent utiliser du chiffrement par exemple pour garantir l'intégrité des traces.
- **Le rejeu des évènements** : La majorité des solutions permettent également de rejouer les anciens événements pour mener des enquêtes post-incidentes. Il est également possible de modifier une règle et de rejouer les événements pour voir son comportement. [33]

3.2.4 Avantages du SIEM

Les avantages d'utiliser un système d'information d'événements de sécurité (SIEM):

- **Centralisation des journaux** : Les journaux de sécurité de différents systèmes et réseaux sont collectés et stockés dans un emplacement central pour une analyse et une enquête plus faciles. Analyse en temps réel : Les alertes de sécurité peuvent être générées et analysées en temps réel pour une détection rapide des incidents de sécurité.
- **Corrélation automatisée** : Les SIEM peuvent automatiquement corréler les événements de sécurité pour identifier les tendances et les menaces potentielles.
- **Archivage et conformité** : Les journaux de sécurité peuvent être archivés pour une durée définie pour respecter les exigences de conformité et pour une analyse ultérieure.
- **Automatisation de la réponse aux incidents** : Les SIEM peuvent automatiser les processus de réponse aux incidents pour une réponse rapide et efficace. [34]

Chapitre 4 : Généralité sur l'IA

4.1 Définition de l'IA

L'intelligence artificielle représente tout outil utilisé par une machine afin de « reproduire des comportements liés aux humains, tels que le raisonnement, la planification et la créativité ».

L'intelligence artificielle (ou IA) est de plus en plus présente dans notre quotidien, notamment au travers de nouveaux produits ou services. Elle repose cependant sur des algorithmes gourmands en données, souvent personnelles, et son usage nécessite le respect de certaines précautions.

Ce n'est pas une technologie à proprement parler mais plutôt un domaine scientifique dans lequel des outils peuvent être classés lorsqu'ils respectent certains critères [13].

4.2 Fonctionnalités de l'IA

L'intelligence artificielle (IA) est en train de révolutionner de nombreux domaines de notre vie, y compris le monde des affaires. L'une des applications les plus prometteuses de l'IA est dans la gestion des connaissances au sein des entreprises.

Il englobe une variété de fonctionnalités et de capacités qui lui permettent d'accomplir une grande diversité de tâches. On peut en citer quelques :

- **Reconnaissance de motifs :** L'IA peut être utilisée pour identifier des modèles et des structures dans des ensembles de données complexes, permettant ainsi de détecter des tendances, des corrélations et des anomalies.
- **Apprentissage automatique (Machine Learning) :** L'IA est capable d'apprendre à partir de données et d'améliorer ses performances au fil du temps sans être explicitement programmée pour chaque tâche. Les algorithmes d'apprentissage automatique comprennent l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.
- **Traitement du langage naturel (NLP) :** Cette fonctionnalité permet aux ordinateurs de comprendre, d'interpréter et de générer un langage humain de manière efficace. Le NLP est utilisé dans les applications telles que la traduction automatique, la reconnaissance de la parole, la compréhension de documents et la génération de texte.

- **Vision par ordinateur :** L'IA peut interpréter et comprendre des informations visuelles à partir d'images ou de vidéos. Cela inclut la détection d'objets, la reconnaissance faciale, la segmentation d'images et la classification d'images.
- **Planification et prise de décision :** Les systèmes d'IA peuvent être programmés pour prendre des décisions en fonction de critères spécifiques et de données disponibles. Cela peut inclure la planification d'itinéraires, la gestion des ressources, la recommandation de produits, etc.
- **Raisonnement logique :** L'IA peut utiliser des techniques de raisonnement logique pour déduire des conclusions à partir de données et de règles prédéfinies. Cela peut être utilisé dans le diagnostic médical, la résolution de problèmes complexes, etc.
- **Robotique :** L'intégration de l'IA dans les systèmes robotiques permet aux robots d'effectuer des tâches complexes de manière autonome. Cela inclut la navigation autonome, la manipulation d'objets, l'interaction avec l'environnement, etc.
- **Systèmes experts :** Les systèmes experts utilisent des bases de connaissances et des règles pour imiter le raisonnement d'un expert humain dans un domaine spécifique. Ils sont souvent utilisés dans le diagnostic médical, l'assistance à la clientèle, etc.
- **Analyse prédictive :** L'IA peut être utilisée pour prévoir des événements futurs en analysant des tendances passées et des données historiques. Cela est couramment utilisé dans le domaine de la finance, du marketing, de la logistique, etc.
- **Génération de contenu :** L'IA peut être utilisée pour générer du contenu textuel, visuel ou audio de manière automatique, comme la rédaction d'articles, la création d'œuvres artistiques, etc.

4.3 Les différentes types d'IA

Il existe trois types d'intelligence artificielle (IA) : l'IA étroite ou faible, l'IA générale ou forte et la superintelligence artificielle.

- L'intelligence artificielle étroite (ANI), qui possède une gamme étroite de capacités
- L'intelligence artificielle générale (AGI), qui est à la hauteur des capacités humaines.
- La superintelligence artificielle (ASI), dont les capacités sont supérieures à celles de l'homme.

4.3.1 Intelligence artificielle étroite (ANI) / IA faible / IA étroite

L'intelligence artificielle étroite (ANI), également appelée IA faible ou IA étroite, est le seul type d'intelligence artificielle que nous ayons réussi à réaliser à ce jour. L'IA étroite est orientée vers un but précis, elle est conçue pour accomplir des tâches uniques – par exemple, la reconnaissance faciale, la reconnaissance vocale/les assistants vocaux, la conduite d'une voiture ou la recherche sur Internet – et elle est très intelligente pour accomplir la tâche spécifique pour laquelle elle est programmée.

Bien que ces machines puissent sembler intelligentes, elles fonctionnent avec un ensemble restreint de contraintes et de limitations, ce qui explique pourquoi ce type d'IA est communément appelé IA faible. L'IA étroite n'imite ni ne reproduit l'intelligence humaine, elle se contente de simuler le comportement humain sur la base d'une gamme étroite de paramètres et de contextes.

L'intelligence artificielle de l'IA étroite provient de l'utilisation du traitement du langage naturel (NLP) pour effectuer des tâches. Le NLP est évident dans les chatbots et les technologies d'IA similaires. En comprenant la parole et le texte en langage naturel, l'IA est programmée pour interagir avec les humains de manière naturelle et personnalisée.

L'IA étroite peut être réactive où avoir une mémoire limitée. L'IA réactive est incroyablement basique ; elle n'a aucune capacité de mémoire ou de stockage de données, imitant la capacité de l'esprit humain à répondre à différents types de stimuli sans expérience préalable. L'IA à mémoire limitée est plus avancée, dotée de capacités de stockage de données et d'apprentissage qui permettent aux machines d'utiliser des données historiques pour prendre des décisions.

La plupart des IA sont des IA à mémoire limitée, où les machines utilisent de grands volumes de données pour l'apprentissage profond. L'apprentissage profond permet des expériences d'IA personnalisées, par exemple des assistants virtuels ou des moteurs de recherche qui stockent vos données et personnalisent vos expériences futures.

Exemples d'IA étroite :

- Rankbrain de Google / Google Search
- Siri d'Apple, Alexa d'Amazon, Cortana de Microsoft et autres assistants virtuels.
- Watson d'IBM

- Logiciels de reconnaissance d'images et de visages
- Outils de cartographie et de prédiction des maladies
- Robots de fabrication et drones
- Filtres anti-spam pour les courriels / outils de surveillance des médias sociaux pour les contenus dangereux
- Recommandations de contenu de divertissement ou de marketing basées sur le comportement de visionnage, d'écoute ou d'achat.
- Voitures à conduite autonome

4.3.2 Intelligence générale artificielle (AGI) / IA forte / IA profonde

L'intelligence générale artificielle (AGI), également appelée IA forte ou IA profonde, est le concept d'une machine dotée d'une intelligence générale qui imite l'intelligence et/ou les comportements humains, avec la capacité d'apprendre et d'appliquer son intelligence pour résoudre tout problème. L'IA peut penser, comprendre et agir d'une manière indiscernable de celle d'un être humain dans une situation donnée.

Les chercheurs et les scientifiques spécialisés dans l'IA n'ont pas encore atteint une IA forte. Pour y parvenir, ils devraient trouver un moyen de rendre les machines conscientes, en programmant un ensemble complet de capacités cognitives. Les machines devraient faire passer l'apprentissage par l'expérience à un niveau supérieur, en améliorant non seulement l'efficacité de certaines tâches, mais en acquérant la capacité d'appliquer les connaissances acquises par l'expérience à un éventail plus large de problèmes différents.

L'IA forte utilise un cadre d'IA basé sur la théorie de l'esprit, qui fait référence à la capacité de discerner les besoins, les émotions, les croyances et les processus de pensée d'autres personnes intelligentes. L'IA de niveau théorie de l'esprit ne consiste pas à reproduire ou à simuler, mais à entraîner les machines à comprendre réellement les humains.

L'immense défi que représente la réalisation d'une IA forte n'est pas surprenant si l'on considère que le cerveau humain est le modèle de création de l'intelligence générale. En raison du manque de connaissances approfondies sur la fonctionnalité du cerveau humain, les chercheurs s'efforcent de reproduire les fonctions de base de la vue et du mouvement.

4.3.3 Superintelligence artificielle (ASI)

La superintelligence artificielle (ASI) est l'IA hypothétique qui ne se contente pas d'imiter ou de comprendre l'intelligence et le comportement humains ; l'ASI est l'endroit où les machines deviennent conscientes d'elles-mêmes et dépassent la capacité de l'intelligence et des aptitudes humaines.

La superintelligence est depuis longtemps la muse de la science-fiction dystopique dans laquelle les robots envahissent, renversent et/ou asservissent l'humanité. Le concept de superintelligence artificielle voit l'IA évoluer pour devenir si proche des émotions et des expériences humaines qu'elle ne se contente pas de les comprendre, mais suscite des émotions, des besoins, des croyances et des désirs qui lui sont propres.

En plus de reproduire l'intelligence à multiples facettes des êtres humains, l'ASI serait théoriquement bien meilleure dans tout ce que nous faisons : mathématiques, sciences, sports, art, médecine, loisirs, relations émotionnelles, tout. L'ASI aurait une meilleure mémoire et une capacité plus rapide à traiter et à analyser les données et les stimuli. Par conséquent, les capacités de prise de décision et de résolution de problèmes des êtres super intelligents seraient bien supérieures à celles des êtres humains. La possibilité de disposer de machines aussi puissantes peut sembler séduisante, mais le concept lui-même a une multitude de conséquences inconnues. Si des êtres super intelligents conscients d'eux-mêmes voyaient le jour, ils seraient capables d'idées comme l'auto-préservation. L'impact que cela aura sur l'humanité, notre survie et notre mode de vie, est une pure spéculation.

Chapitre 5: Techniques de classification

Les techniques de classification en intelligence artificielle permettent de catégoriser ou classifier des données en différentes classes ou groupes. Le but de la classification est de pouvoir assigner une classe ou catégorie à de nouvelles données inconnues, en se basant sur des modèles construits à partir de données d'entraînement [9].On distingue deux grandes familles d'algorithmes de classification:

- L'apprentissage supervisé
- L'apprentissage non supervisé

5.1 L'apprentissage supervisé

L'apprentissage supervisé, également appelé apprentissage automatique supervisé, est une sous-catégorie de l'apprentissage automatique et de l'intelligence artificielle. Il se caractérise par l'utilisation de jeux de données étiquetés qui entraînent des algorithmes permettant de classer des données ou de prédire des résultats avec précision. Au fur et à mesure que les données en entrée sont introduites dans le modèle, celui-ci ajuste ses pondérations jusqu'à ce que le modèle soit correctement ajusté. C'est le processus de validation croisée. Avec l'apprentissage supervisé, les organisations peuvent résoudre divers problèmes du monde réel à grande échelle, comme la classification des courriers indésirables dans un dossier distinct de votre boîte de réception.

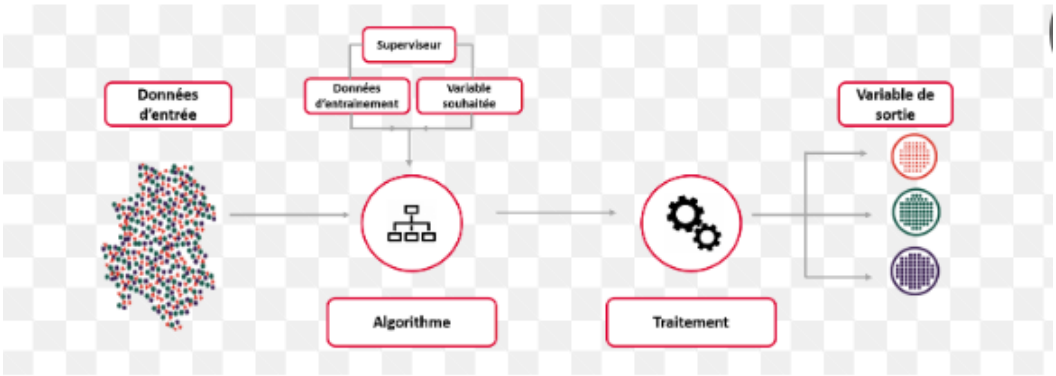


Figure 1:apprentissage supervisé

L'algorithme est entraîné sur des données déjà labellisées ou catégorisées, afin de pouvoir prédire la classe d'éléments non étiquetés. Les algorithmes les plus utilisés sont les réseaux de neurones, les arbres de décision, les SVM, le naïve bayes, etc.

5.1.1 Les réseaux neurones

Les réseaux neuronaux constituent un programme ou un modèle de machine learning qui prend des décisions d'une manière comparable au cerveau humain, en utilisant des processus qui reproduisent la façon dont les neurones biologiques fonctionnent de concert pour identifier des phénomènes, évaluer des options et arriver à des conclusions. Ils s'appuient sur les données d'entraînement pour apprendre et améliorer leur précision au fil du temps. Une fois qu'ils atteignent un niveau de précision optimal, ce sont de puissants alliés des domaines de l'informatique et de l'intelligence artificielle, qui nous permettent de classer et de regrouper les données à grande vitesse. Une tâche de reconnaissance vocale ou d'image qui demandent des heures de recherche à un expert humain peut être réalisée en quelques minutes. L'algorithme de recherche de Google est l'un des exemples les plus connus des réseaux neuronaux.

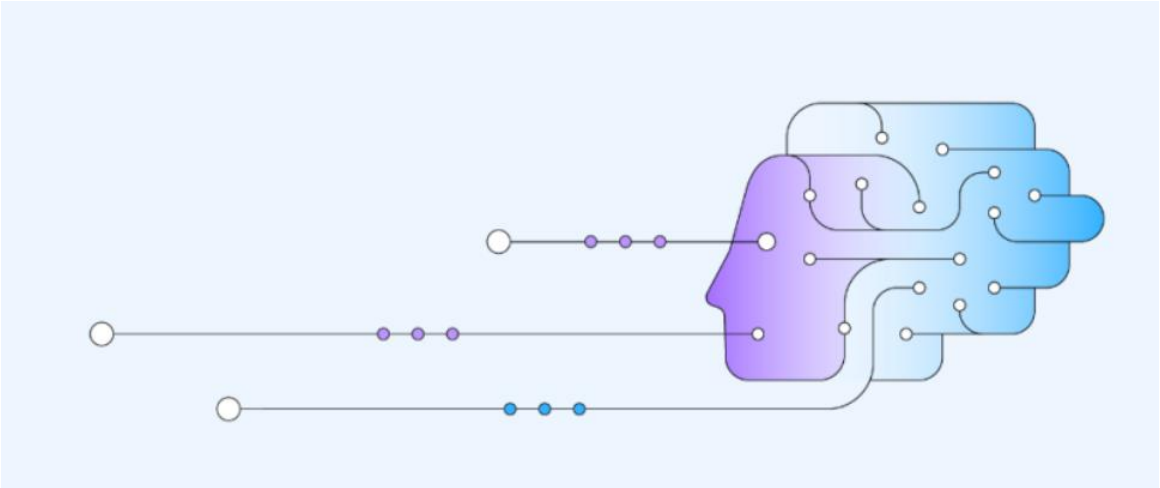


Figure 2:réseaux neurones

5.1.2 Les arbres de décision

Un arbre de décisions est un algorithme d'apprentissage supervisé non paramétrique, qui est utilisé à la fois pour les tâches de classification et régression. Il a une structure hiérarchique, une structure arborescente, qui se compose d'un nœud racine, de branches, de nœuds interne et de nœuds feuille. un arbre de décisions commence par un nœud racine, qui n'a pas de branches entrantes. Les branches sortant du nœud racine vont dans les nœuds internes, également connu comme nœuds de décision. Sur la base des caractéristiques disponibles, les deux types de nœud effectuent des évaluations sur des sous-ensembles homogènes, qui sont désignés comme nœuds feuille ou nœuds terminal. Les nœuds feuille représentent tous les résultats possibles au sein du fichier. Ce type de structure en organigramme crée également

une représentation facile à assimiler pour la prise de décision, permettant aux différents groupes d'une organisation de mieux comprendre pourquoi une décision a été prise. L'apprentissage par arborescence de décisions utilise une stratégie de division et de conquête en effectuant une recherche gloutonne pour identifier les points de fractionnement optimaux au sein d'une arborescence. Ce processus de fractionnement est ensuite répété de manière descendante, récursive jusqu'à ce que tous, ou la majorité des enregistrements aient été classifiés sous des étiquettes de classe spécifiques.

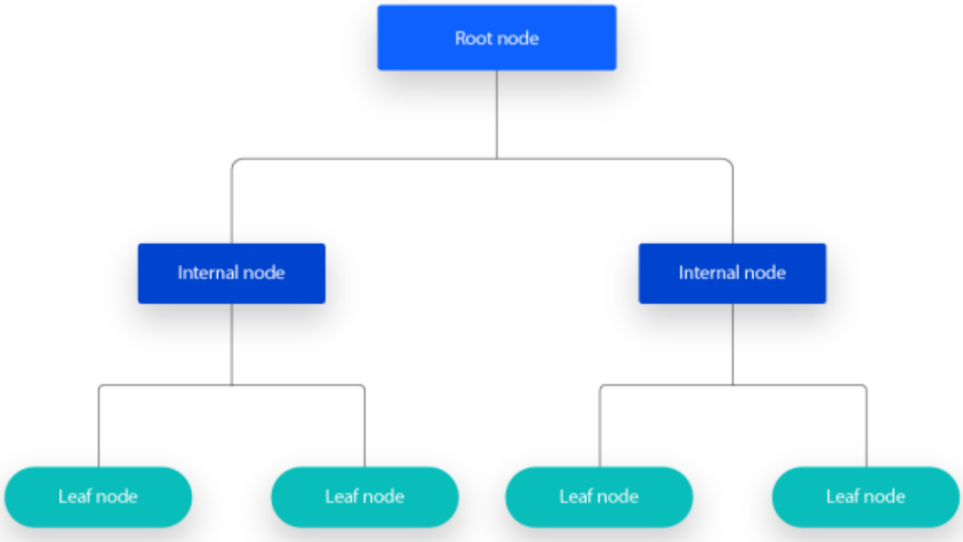


Figure 3: arbres de décision

5.1.3 Les SVM*(*support-vector machine)

Une Support Vector Machine (SVM) est un algorithme d'apprentissage supervisé utilisé pour de nombreux problèmes de classification et de régression, tels que les applications médicales de traitement du signal, le traitement du langage naturel, la reconnaissance vocale et la reconnaissance d'images.

L'objectif de l'algorithme SVM est de trouver un hyperplan qui, du mieux possible, sépare les points de données d'une classe de ceux d'une autre classe. Le meilleur cas est un hyperplan représentant la plus grande marge entre les deux classes, indiquées par les plus et les moins dans la figure ci-dessous. Un hyperplan est une frontière de décision qui sépare les données de deux classes dans un espace de caractéristiques. La marge désigne la largeur maximale de

l'espace parallèle à l'hyperplan qui ne contient aucun point de données. L'algorithme ne peut trouver un tel hyperplan que pour les problèmes linéairement séparables. Pour la plupart des problèmes pratiques, l'algorithme maximise la marge souple, autorisant ainsi un petit nombre d'erreurs de classification.

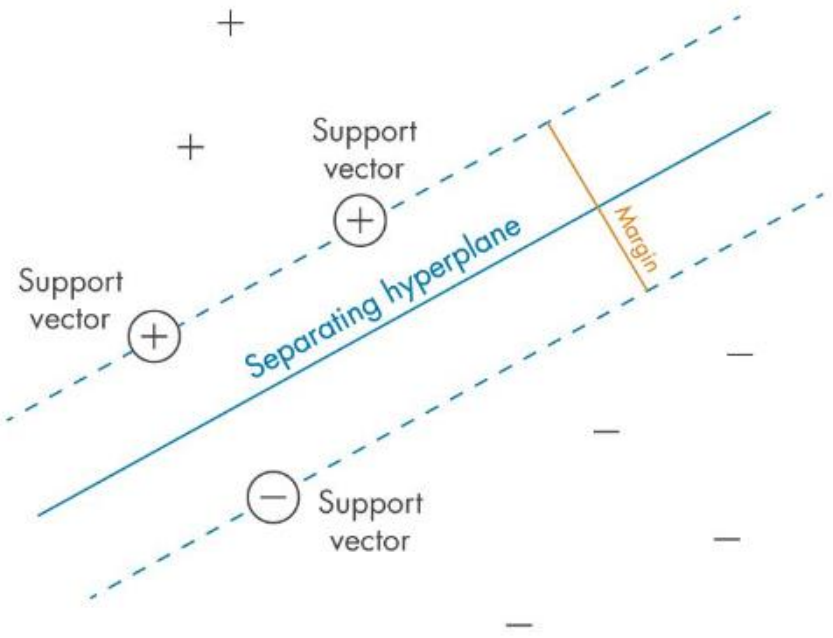


Figure 4: Support-vector machine(SVM)

5.1.4 Le naive bayes

La méthode de classification naïve bayésienne est un algorithme d'apprentissage supervisé qui permet de classifier un ensemble d'observations selon des règles déterminées par l'algorithme lui-même. Cet outil de classification doit dans un premier temps être entrainé sur un jeu de données d'apprentissage qui montre la classe attendue en fonction des entrées. Pendant la phase d'apprentissage, l'algorithme élabore ses règles de classification sur ce jeu de donnée, pour les appliquer dans un second temps à la classification d'un jeu de données de prédiction. Le classificateur bayésien naïf implique que les classes du jeu de données d'apprentissage soit connu et fournit, d'où le caractère supervisé de l'outil.

Historiquement, la classification naïve bayésienne fut utilisée pour la classification de documents et l'élaboration de filtres anti-spam. Aujourd'hui, c'est un algorithme renommé dont les applications peuvent être rencontrées dans de nombreux domaines. Parmi ces atouts les plus significatifs, on citera son apprentissage rapide qui ne nécessite pas un gros volume

de données et son extrême rapidité d'exécution comparé à d'autres méthodes plus complexes. Finalement, malgré la forte hypothèse simplificatrice d'indépendance des variables, la classification naïve bayésienne obtient des résultats remarquables dans de nombreuses applications de la vie courante ce qui en fait un algorithme de choix parmi les outils du machine learning.

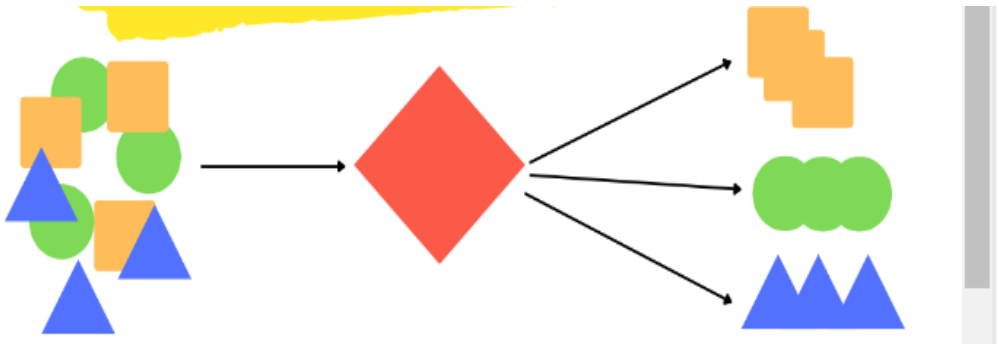


Figure 5: Le naive bayes

5.2 L'apprentissage non supervisé

L'apprentissage non supervisé est une branche de l'apprentissage automatique qui concerne l'analyse et l'inférence de modèles à partir de données qui n'ont pas de labels ou de catégories préalablement définis. Contrairement à l'apprentissage supervisé, où les modèles sont formés sur des ensembles de données étiquetés, l'apprentissage non supervisé cherche à identifier des structures, des motifs ou des relations dans les données brutes. Le système d'association permet de trier et regrouper les données qui peuvent être liées grâce à certaines caractéristiques. Le but est donc de trouver des objets liés les uns aux autres sans qu'il s'agisse néanmoins d'objets identiques. Il est très souvent utilisé dans le domaine de la reconnaissance vocale, comme pour l'utilisation de Siri ou Alexa.

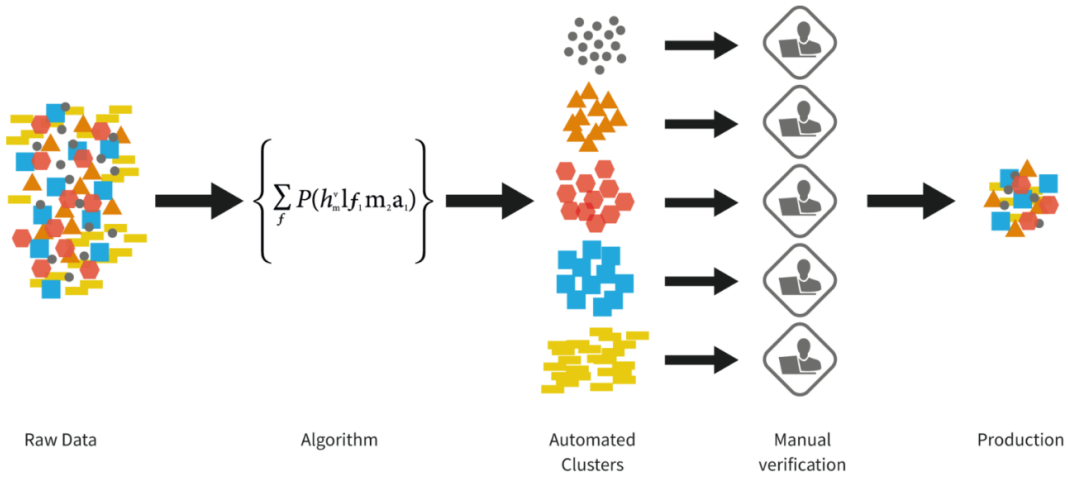


Figure 6:apprentissage non supervisé

Il permet aux organisations de mieux comprendre leurs données et d'en extraire des informations précieuses. À mesure que les volumes de données continuent de croître, l'apprentissage non supervisé devient de plus en plus crucial pour l'analyse de données et la prise de décision basée sur les données. Les algorithmes les plus utilisés sont : K-Means, ACP, GMM, Algorithme Apriori, Algorithme des K-Médoïdes.

5.2.1 K-Means(k-moyennes)

Le K-Means est un algorithme de regroupement non supervisé largement utilisé en apprentissage automatique et en analyse de données pour partitionner un ensemble de données en *k* clusters (groupes) distincts. Chaque cluster est défini par son centre, appelé centroïde. L'objectif de K-Means est de minimiser la variance intra-cluster, c'est-à-dire de rendre les points de chaque cluster aussi proches que possible de leur centroïde. Le clustering est une discipline particulière du Machine Learning ayant pour objectif de séparer vos données en groupes homogènes ayant des caractéristiques communes.

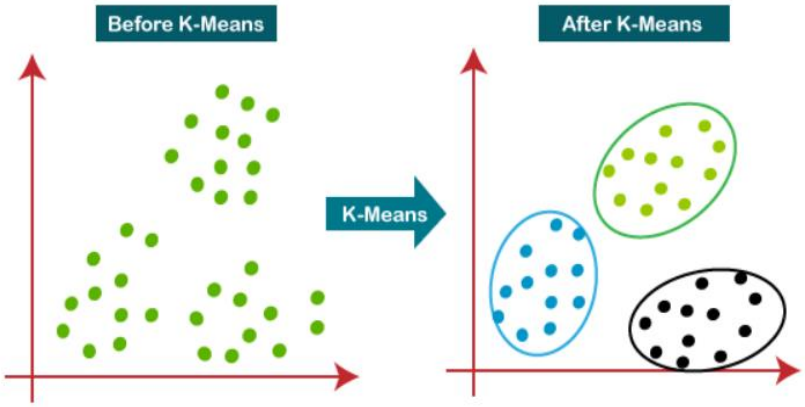


Figure 7:K-Means

5.2.2 ACP(Analyse en Composantes Principales)

L'Analyse en Composantes Principales (ACP) est une technique de réduction de dimensionnalité largement utilisée en statistiques, apprentissage automatique et exploration de données. L'ACP permet de transformer un ensemble de données avec de nombreuses variables corrélées en un ensemble de variables non corrélées appelées composantes principales. Ces composantes principales sont ordonnées de manière que la première composante retienne le maximum de variance des données, la seconde composante retient le maximum de variance restante, et ainsi de suite.

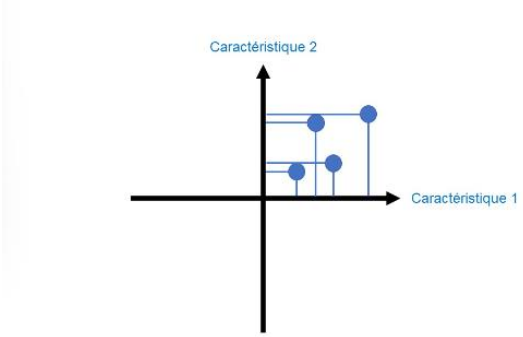


Figure 8 : ACP(Analyse en Composantes Principales)

5.2.3 GMM (Gaussian Mixture Models)

Un modèle de mélange gaussien est un apprentissage machine méthode utilisée pour déterminer probabilité chaque point de données appartient à un cluster donné. Le modèle est un soft méthode de clustering utilisé dans apprentissage non supervisé. Les GMM sont des

modèles probabilistes qui représentent la distribution des données comme une combinaison de plusieurs distributions normales (gaussiennes). Chaque point de données est supposé avoir été généré par l'une des kkk distributions gaussiennes, avec une certaine probabilité.

Clustering est un problème d'apprentissage non supervisé où nous avons l'intention de trouver des groupes de points dans notre ensemble de données qui partagent certaines caractéristiques communes.

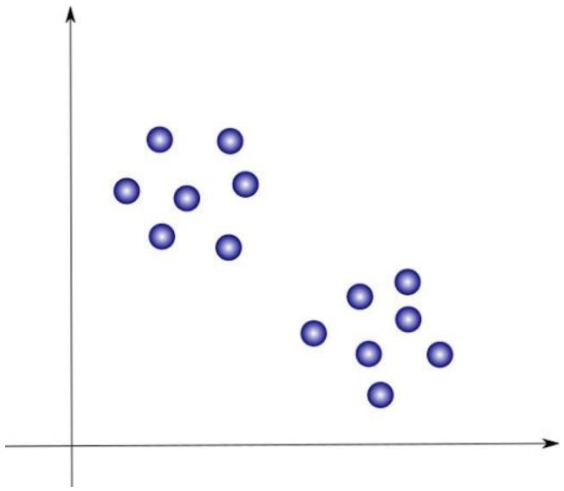


Figure 9: GMM (Gaussian Mixture Models)

5.2.4 Algorithme Apriori

L'algorithme Apriori est un des algorithmes les plus connus et largement utilisés pour la découverte de règles d'association dans les bases de données transactionnelles. Cet algorithme, proposé par Rakesh Agrawal et Ramakrishnan Srikant en 1994, vise à identifier des ensembles fréquents (itemsets) dans les données et à en dériver des règles d'association utiles pour des tâches telles que l'analyse de panier d'achat. Un itemset fréquent est un ensemble d'éléments qui apparaît ensemble dans au moins un certain pourcentage (support minimum) des transactions.

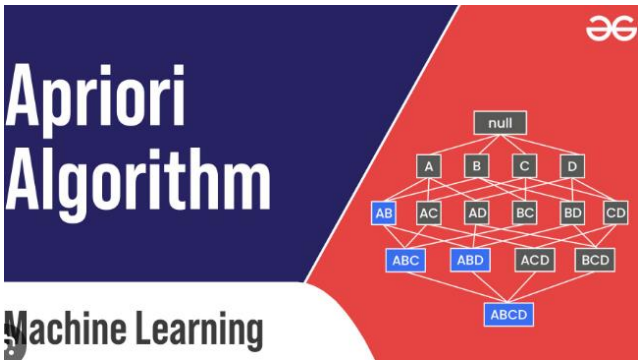


Figure 10 : Algorithme Apriori

5.2.5 Algorithme des K-Médoïdes

L'algorithme des K-Médoïdes est une technique de partitionnement des données similaire à l'algorithme des K-Means, mais il se distingue par son utilisation des médoïdes au lieu des centroides. Le médoïde est un point représentatif d'un cluster qui est lui-même un point de données existant, contrairement au centroïde qui est une moyenne arithmétique des points d'un cluster. Cette caractéristique rend les K-Médoïdes plus robustes aux valeurs aberrantes et aux bruits dans les données.

L'objectif des K-Médoïdes est de partitionner un ensemble de données en kkk clusters en minimisant la somme des dissimilarités entre les points de données et le médoïde de leur cluster respectif.

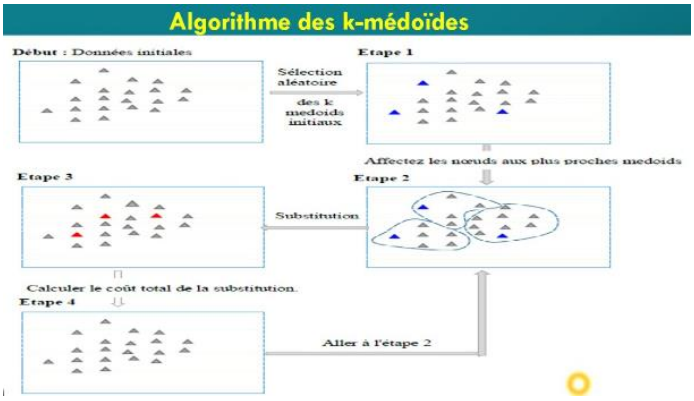


Figure 11: Algorithme des K-Médoïdes

Tableau 1:Comparatif des Techniques de Classification Algorithmes supervisés

Catégorie	Techniques	Avantages	Inconvénients	Applications
Apprentissage Supervisé	Régression Logistique	- Simple à implémenter - Interprétable	Limité aux problèmes linéaires - Sensible aux outliers	Prédiction binaire, analyse marketing
	k-Nearest Neighbors (k-NN)	Facile à comprendre - Pas de phase d'entraînement	Lent pour grands datasets - Sensible aux dimensions élevées	Reconnaissance faciale, recommandation
	SVM (Support Vector Machines)	- Performant pour données complexes - Bonne généralisation	Coûts de calcul élevés - Réglage des kernels difficile	Classification de texte, bioinformatique
	Réseaux Neuronaux Profonds (DNN)	- Capable d'apprentissage complexe - Non linéaire	Exige beaucoup de données - Coûteux en calcul	Reconnaissance d'images et voix
	Forêts Aléatoires	Réduit l'overfitting - Bonne précision	- Moins interprétable - Coûteux en calcul pour grands ensembles	Prédiction de ventes, détection d'anomalies

Tableau 2 : Comparatif des Techniques de Classification :Algorithmes non Supervisés

Catégorie	Techniques	Avantages	Inconvénients	Applications
Apprentissage Non Supervisé	k-Means	Simple - Rapide pour de grands ensembles	Sensible au choix des centres initiaux - Ne gère pas bien les données bruitées	Segmentation client, regroupement
	Clustering Hiérarchique	- Adapté pour les petits ensembles - Visualisation	- Moins efficace pour grands datasets - Sensible au bruit	Analyse génomique, structure des données
	Gaussian Mixture Models (GMM)	- Gère des clusters de formes variées	- Peut converger vers des solutions locales - Plus lent	Détection d'anomalies, classification d'images

Tableau 3 : Comparatif des Techniques de Classification :Algorithmes Hybrides

Catégorie	Techniques	Avantages	Inconvénients	Applications
Hybride (Supervisé & Non Supervisé)	Semi-Supervised Learning	- Réduit les besoins en données étiquetées - Combine avantages des deux approches	- Dépendant de la qualité des données - Plus complexe à implémenter	Étiquetage de données médicales, détection de fraude
	Self-Training	Exploite efficacement les données non étiquetées	Sensible aux erreurs initiales - Long à converger	Annotation d'images, NLP
	Réseaux Auto-Encodeurs (AE)	Compresse et apprend la structure des données - Flexible	- Nécessite réglages précis - Moins interprétable	Détection d'anomalies, réduction de dimension
	Clustering supervisé	- Combine clustering avec supervision - Meilleure performance	Requiert supervision partielle - Coûteux	Détection de spam, segmentation comportementale

Chapitre 6 : Etude comparative des algorithmes de corrélation

Une corrélation est une opération pour déterminer s'il existe un lien statistique linéaire fort entre deux variables quantitatives à l'échelle relative (qui identifie les intervalles entre les données et où la position de zéro est définie arbitrairement, comme sur un thermomètre) ou à l'échelle absolue (soit une échelle relative où zéro a une position absolue : cela permet de quantifier la différence entre deux éléments, comme avec l'échelle de Kelvin).Les algorithmes de corrélation sont utilisés pour identifier les relations entre différentes variables dans des ensembles de données.

Après une recherche ciblée, on peut en citer quelques :

6.1 Présentation de quelques techniques de corrélation

6.1.1 Pearson

La corrélation de Pearson, également connue sous le nom de coefficient de corrélation linéaire, est une mesure statistique qui quantifie la force et la direction de la relation linéaire entre deux variables quantitatives. Elle prend des valeurs comprises entre -1 et 1. La corrélation de Pearson est largement utilisée dans de nombreux domaines, comme les sciences sociales, l'économie, la biologie, etc. Elle permet d'explorer les relations entre variables, de sélectionner les variables pertinentes dans les modèles de régression, ou encore de détecter la multi-colinéarité.

Cependant, il est important de garder à l'esprit que la corrélation ne signifie pas nécessairement une relation de causalité, et qu'elle ne mesure que la relation linéaire. D'autres mesures de corrélation, comme le coefficient de Spearman ou le tau de Kendall, peuvent être plus appropriées si la relation n'est pas linéaire.

6.1.2 Spearman

La corrélation de Spearman, également connue sous le nom de coefficient de corrélation de rang de Spearman, est une mesure non paramétrique de la corrélation (ou de la dépendance statistique) entre deux variables. Contrairement à la corrélation de Pearson, la corrélation de Spearman ne suppose pas que les données suivent une distribution normale et ne mesure pas uniquement les relations linéaires mais les relations monotones. Cela la rend plus robuste et

applicable à des données ordinales ou à des relations non linéaires. C’est une mesure appropriée à utiliser lorsque les variables sont mesurées sur une échelle au moins ordinale.

6.1.3 Kendall (Tau)

Le coefficient de corrélation de Kendall, également appelé tau de Kendall, est une mesure non paramétrique de la force et de la direction de la relation monotone entre deux variables ordinales ou continues, tout comme le coefficient de corrélation de Spearman. Cependant, il est calculé différemment et présente certaines caractéristiques distinctes. Le tau de Kendall est fréquemment utilisé dans les sciences sociales, la psychologie, l'économie et d'autres domaines où les données peuvent ne pas suivre une distribution normale ou lorsque les relations ne sont pas nécessairement linéaires. Il offre une alternative robuste à la corrélation de Pearson et au coefficient de Spearman lorsque les hypothèses de linéarité et de normalité ne sont pas respectées, tout en étant particulièrement adapté pour détecter les relations monotones non linéaires.

6.1.4 Corrélation Partielle

La corrélation partielle est une mesure statistique qui permet d'évaluer la relation linéaire entre deux variables tout en éliminant les effets d'une ou plusieurs autres variables.

L'objectif de la corrélation partielle est d'éliminer les effets de confusion potentiels causés par d'autres variables et d'isoler la relation entre les deux variables d'intérêt. Elle permet d'obtenir une compréhension plus précise des relations entre les variables en éliminant les effets de confusion.

6.1.5 Information mutuelle

L'information mutuelle est une mesure utilisée en théorie de l'information et en statistique pour quantifier la dépendance entre deux variables aléatoires. Elle évalue combien d'informations sur une variable peuvent être obtenues en connaissant l'autre variable. Contrairement à des mesures comme la corrélation de Pearson, qui se concentrent sur les relations linéaires, l'information mutuelle capture toutes les formes de dépendance, linéaires et non linéaires.

Tableau 4: comparatif des Techniques de corrélation

Techniques de Corrélation	**Type de relation**	**Robustesse**	**Complexité computationnelle**	**Avantages**	**Limites**
Corrélation de Pearson	Linéaire	Sensible aux valeurs aberrantes	Faible	Simple à interpréter, capture les relations linéaires	Ne détecte pas les relations non linéaires
Coefficient de Spearman	Monotone	Robuste aux valeurs aberrantes et aux distributions non normales	Modérée	Détecte les relations monotones non linéaires	Ne détecte pas les relations non linéaires
Tau de Kendall	Monotone	Robuste aux valeurs aberrantes et aux distributions non normales	Modérée	Détecte les relations monotones non linéaires, moins sensible aux valeurs extrêmes que Spearman	Complexité de calcul plus élevée
Corrélation partielle	Linéaire	Sensible aux valeurs aberrantes	Modérée	Élimine les effets de confusion causés par d'autres variables	Ne capture que les relations linéaires, ne permet pas d'établir la causalité
Information mutuelle	Générale	Robuste aux valeurs aberrantes et aux distributions non normales	Élevée	Détecte les relations complexes, non linéaires et non monotones	Interprétation parfois complexe, ne fournit pas d'information sur la direction de la relation

Partie 3 : Conception et mise en œuvre

Chapitre 7 : Mise en place de la solution

7.1 Architecture de déploiement

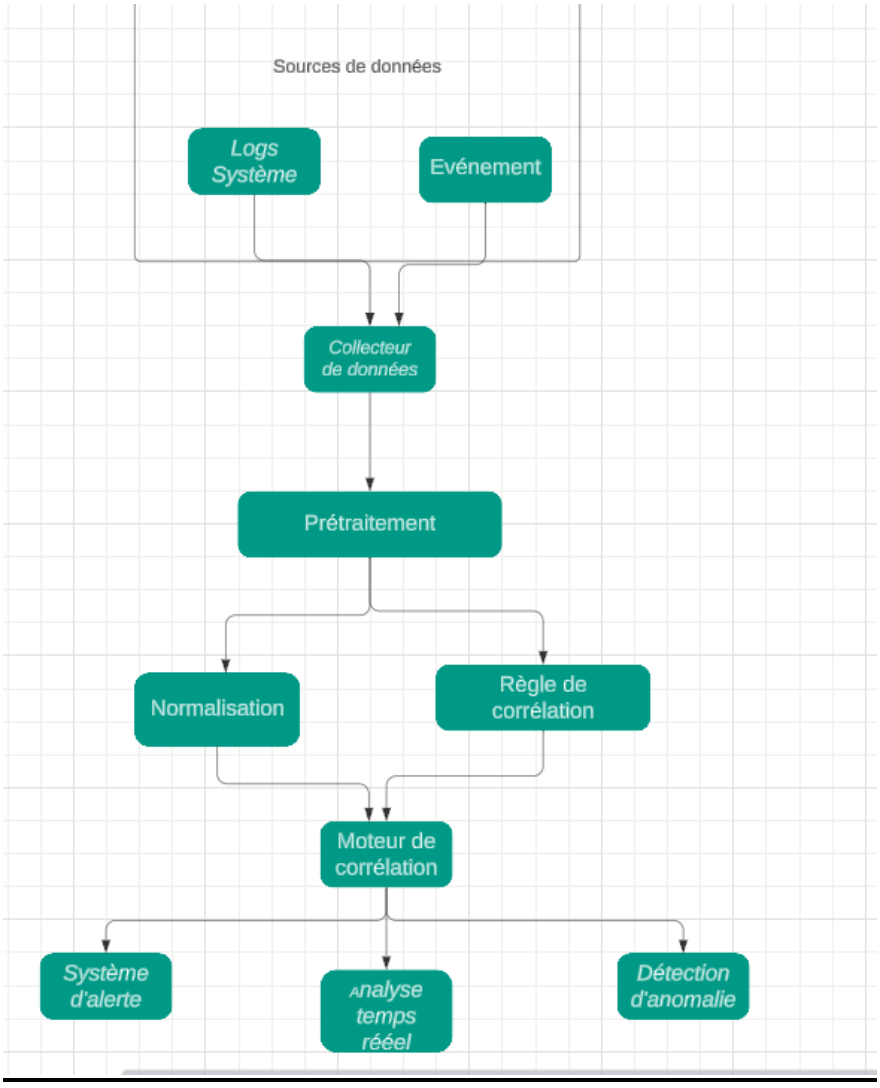


Figure 12:Les composantes de l'architecture

Ce diagramme illustre un pipeline complet de gestion et d'analyse des données dans un système de détection d'anomalies et d'analyse de corrélation des événements.

Sources de données :

- Logs Système : Ce sont des fichiers ou flux qui capturent les événements enregistrés par les systèmes (serveurs, applications, réseaux, etc.).

- Événements : Ils peuvent inclure des événements externes ou internes, comme des incidents de sécurité, des modifications dans le système, ou des alertes provenant de différents outils.

Ces sources convergent vers un collecteur centralisé.

Collecteur de données

Ce composant central recueille les données des différentes sources. Il s'agit d'un point d'entrée clé pour centraliser toutes les informations nécessaires avant traitement, dans le collecteur on fait :

- Extraction des données brutes.
- Agrégation des logs et des événements dans un format commun.
- Passage des données au module suivant pour traitement.

Prétraitement

Le prétraitement est une étape essentielle où les données brutes sont préparées pour être analysées efficacement.

- **Normalisation** :

Conversion des données dans un format standardisé, quel que soit leur origine.

Exemple : Transformer des données issues de différents systèmes en une structure homogène.

- **Règle de corrélation** :

Les règles de corrélation définissent comment différents événements ou logs sont reliés pour identifier des patterns ou comportements anormaux, par exemple une règle pourrait dire "si une IP fait plusieurs tentatives de connexion échouées suivies d'une connexion réussie, cela pourrait être une attaque brute-force".

Moteur de corrélation

Ce module central traite les données prétraitées en utilisant les règles définies et les données normalisées pour effectuer une analyse intelligente.

Fonctions :

- Détecter des schémas anormaux ou des événements significatifs.
- Corréler différents événements en fonction de règles prédéfinies ou basées sur l'apprentissage automatique.

Résultats du moteur de corrélation

Le moteur de corrélation produit trois résultats principaux :

Système d'alerte :

Génération d'alertes lorsqu'une anomalie ou un événement critique est détecté.

Notifications envoyées aux équipes concernées.

Analyse en temps réel :

Permet une surveillance continue pour détecter et agir rapidement sur des anomalies ou événements critiques.

Détection d'anomalie :

Identification des comportements ou événements qui dévient des schémas normaux.

7.2 Maquette de test

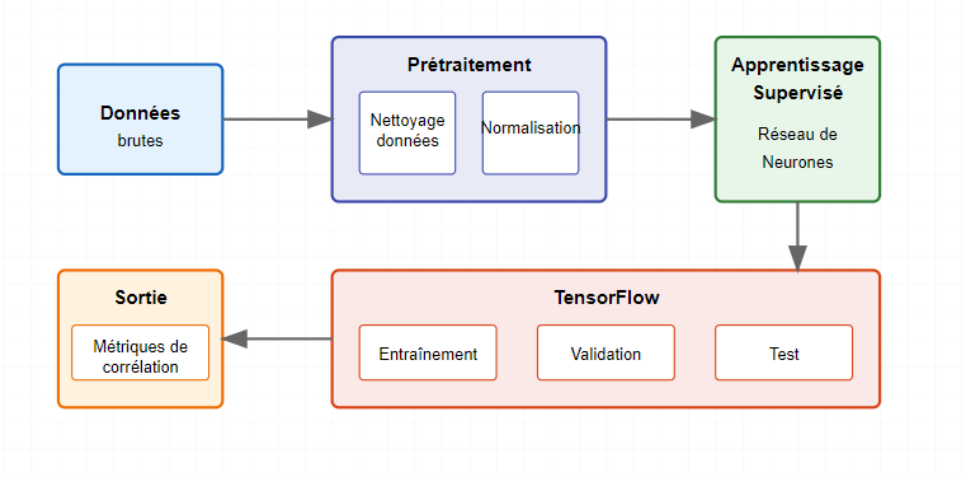


Figure 13:Maquette de test

7.3 Conception de la solution

7.3.1 Choix de la solution

- Type de Dataset: logs de sécurité (SIEM)

  Un dataset est une collection organisée de données, souvent utilisée dans le contexte de l'apprentissage automatique et de l'intelligence artificielle. Les datasets permettent aux algorithmes d'apprendre à partir des données pour identifier des patterns et faire des prédictions.

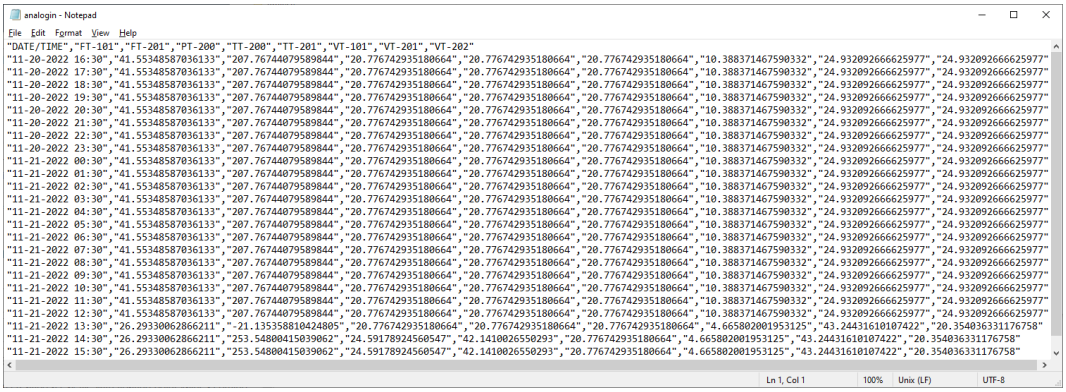


Figure 14: Format Dataset

- Le modèle choisi: apprentissage non supervisé
- Type de modèle: un Auto-encodeur (autoencoder) construit avec TensorFlow

7.3.2 Installation et configuration

7.3.1.1 Installation et configuration de TensorFlow

➢ **Prérequis installation TensorFlow :**

- Python 3.7+
- pip 19.0 ou plus (TensorFlow 2 exige une version plus récente de pip)
- Windows 7 ou plus (64 bits)

❖ **Étape 1 : ouvrez une invite de commande**

On vérifie si python est bien installé on vérifie la version de python et du pip

```
C:\Users\badam>python --version
Python 3.9.13

C:\Users\badam>pip --version
pip 22.0.4 from C:\Users\badam\AppData\Local\Programs\Python\Python39\lib\si
te-packages\pip (python 3.9)

C:\Users\badam>
```

❖ **Étape 2 : Créer un environnement virtuel (facultatif)**

Il est recommandé de créer un environnement virtuel pour garder votre installation TensorFlow isolée des autres installations Python sur votre système.

Les environnements virtuels Python sont essentiels pour gérer efficacement des projets multiples tout en évitant les conflits de dépendances et en facilitant la gestion des versions des bibliothèques.

```
C:\Users\badam>python -m venv myenv

C:\Users\badam>
```

❖ **Étape 3 : Activer l'environnement virtuel (facultatif)**

Si vous avez créé un environnement virtuel à l'étape précédente, vous devez l'activer avant d'installer TensorFlow

```
C:\Users\badam>myenv\Scripts\activate

(myenv) C:\Users\badam>
```

❖ **Étape 4 : Installer TensorFlow à l'aide de pip**

Une fois que vous avez activé l'environnement virtuel (le cas échéant), vous pouvez procéder à l'installation de TensorFlow à l'aide de pip.

```
(myeven) C:\Users\badam>pip install tensorflow
Collecting tensorflow
  Downloading tensorflow-2.18.0-cp39-cp39-win_amd64.whl.metadata (3.3 kB)
Collecting tensorflow-intel==2.18.0 (from tensorflow)
  Downloading tensorflow_intel-2.18.0-cp39-cp39-win_amd64.whl.metadata (4.9
kB)
Collecting absl-py>=1.0.0 (from tensorflow-intel==2.18.0->tensorflow)
  Using cached absl_py-2.1.0-py3-none-any.whl.metadata (2.3 kB)
Collecting astunparse>=1.6.0 (from tensorflow-intel==2.18.0->tensorflow)
  Downloading astunparse-1.6.3-py2.py3-none-any.whl.metadata (4.4 kB)
Collecting flatbuffers>=24.3.25 (from tensorflow-intel==2.18.0->tensorflow)
  Downloading flatbuffers-24.3.25-py2.py3-none-any.whl.metadata (850 bytes)
Collecting gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 (from tensorflow-intel==2.18.
0->tensorflow)
  Using cached gast-0.6.0-py3-none-any.whl.metadata (1.3 kB)
Collecting google-pasta>=0.1.1 (from tensorflow-intel==2.18.0->tensorflow)
  Downloading google_pasta-0.2.0-py3-none-any.whl.metadata (814 bytes)
Collecting libclang>=13.0.0 (from tensorflow-intel==2.18.0->tensorflow)
  Downloading libclang-18.1.1-py2.py3-none-win_amd64.whl.metadata (5.3 kB)
Collecting opt-einsum>=2.3.2 (from tensorflow-intel==2.18.0->tensorflow)
  Downloading opt_einsum-3.4.0-py3-none-any.whl.metadata (6.3 kB)
Collecting packaging (from tensorflow-intel==2.18.0->tensorflow)
  Downloading packaging-24.1-py3-none-any.whl.metadata (3.2 kB)
```

```
Downloading tensorflow-2.18.0-cp39-cp39-win_amd64.whl (7.5 kB)
Downloading tensorflow_intel-2.18.0-cp39-cp39-win_amd64.whl (390.0 MB)
   ---------------------------------------- 390.0/390.0 MB 718.4 kB/s eta 0:00:00
Using cached absl_py-2.1.0-py3-none-any.whl (133 kB)
Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Downloading flatbuffers-24.3.25-py2.py3-none-any.whl (26 kB)
Using cached gast-0.6.0-py3-none-any.whl (21 kB)
Downloading google_pasta-0.2.0-py3-none-any.whl (57 kB)
Downloading grpcio-1.67.1-cp39-cp39-win_amd64.whl (4.4 MB)
   ---------------------------------------- 4.4/4.4 MB 505.5 kB/s eta 0:00:00
Downloading h5py-3.12.1-cp39-cp39-win_amd64.whl (3.0 MB)
   ---------------------------------------- 3.0/3.0 MB 656.3 kB/s eta 0:00:00
Downloading keras-3.6.0-py3-none-any.whl (1.2 MB)
   ---------------------------------------- 1.2/1.2 MB 491.2 kB/s eta 0:00:00
Downloading libclang-18.1.1-py2.py3-none-win_amd64.whl (26.4 MB)
   ---------------------------------------- 26.4/26.4 MB 532.5 kB/s eta 0:00:00
Downloading ml_dtypes-0.4.1-cp39-cp39-win_amd64.whl (126 kB)
Downloading numpy-2.0.2-cp39-cp39-win_amd64.whl (15.9 MB)
   ---------------------------------------- 15.9/15.9 MB 1.5 MB/s eta 0:00:00
Downloading opt_einsum-3.4.0-py3-none-any.whl (71 kB)
Downloading protobuf-5.28.3-cp39-cp39-win_amd64.whl (431 kB)
Downloading requests-2.32.3-py3-none-any.whl (64 kB)
Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Downloading tensorboard-2.18.0-py3-none-any.whl (5.5 MB)
   ---------------------------------------- 5.5/5.5 MB 377.3 kB/s eta 0:00:00
Downloading tensorflow_io_gcs_filesystem-0.31.0-cp39-cp39-win_amd64.whl (1.5
 MB)
   ---------------------------------------- 1.5/1.5 MB 576.1 kB/s eta 0:00:00
Using cached termcolor-2.5.0-py3-none-any.whl (7.8 kB)
Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Downloading wrapt-1.16.0-cp39-cp39-win_amd64.whl (37 kB)
Downloading packaging-24.1-py3-none-any.whl (53 kB)
Downloading certifi-2024.8.30-py3-none-any.whl (167 kB)
```

❖ Étape 5 : Vérifier l'installation

Pour vérifier que TensorFlow a été installé avec succès, vous pouvez exécuter un simple script de test :

```
import tensorflow as tf

tf.__version__
```

Si l'installation est correcte on verra la version de TensorFlow

```
>>> import tensorflow as tf
>>> tf.__version__
'2.18.0'
>>>
```

7.3.3 Interprétation du code

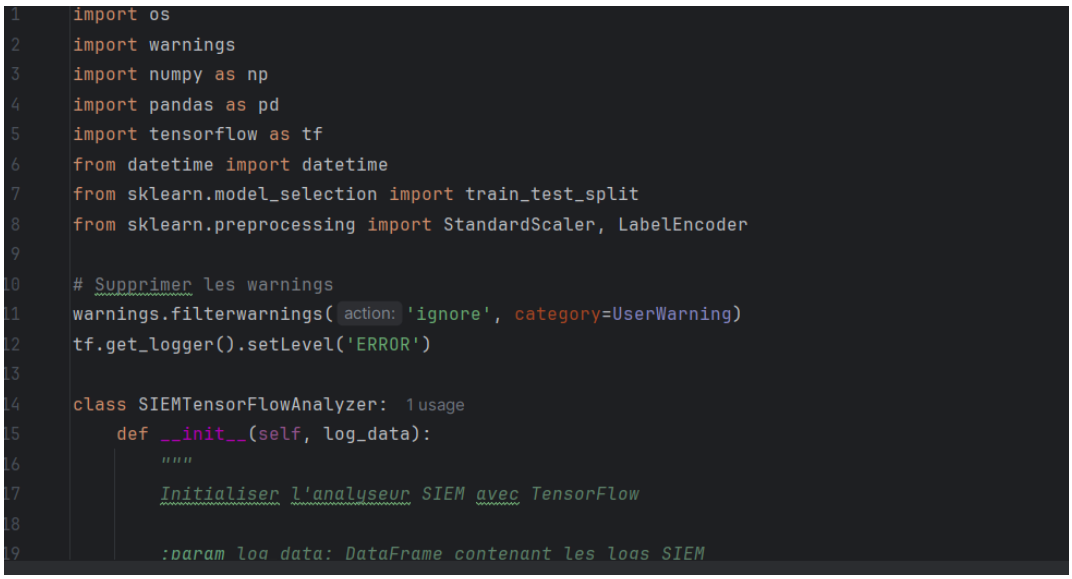
Notre modèle est un système de détection d'anomalies pour les logs SIEM (Security Information and Event Management) en utilisant TensorFlow a été entraîné avec succès sur un jeu de données simulé pour détecter des anomalies liées à des paramètres réseau tels que la taille des paquets, la durée des connexions, les ports et les protocoles. Les résultats montrent une identification réussie de différents types d'anomalies, avec une séparation entre les observations normales et celles présentant des anomalies.

Les principaux composants sont :

1. La classe SIEMTensorFlowAnalyzer qui est le cœur du système :
    - Elle utilise un autoencodeur (un type de réseau de neurones) pour détecter les anomalies
    - Elle peut traiter plusieurs types d'anomalies : taille des paquets, durée, port, et protocole
    - Le modèle apprend à reconnaître les patterns normaux dans les données pour ensuite identifier ce qui est anormal
2. Les fonctionnalités principales :
    - generate_sample_data() : Crée des données d'exemple avec des anomalies injectées artificiellement
    - preprocess_data() : Prépare les données pour l'entraînement (encodage, normalisation)
    - build_anomaly_detection_model() : Crée l'architecture de l'autoencodeur
    - train_anomaly_detector() : Entraîne le modèle
    - detect_anomalies() : Identifie les anomalies dans les données
3. L'architecture de l'autoencodeur :
    - Encodeur : Compresse les données (128 → 64 → 32 neurones)

- Espace latent : Représentation compressée (32 neurones)
- Décodeur : Reconstruit les données (32 → 64 → 128 neurones)

4. La détection d'anomalies fonctionne ainsi :
   - Le modèle essaie de reconstruire les données d'entrée
   - Si l'erreur de reconstruction est élevée, c'est probablement une anomalie
   - Des seuils différents sont utilisés pour chaque type d'anomalie
5. Le modèle est configuré pour détecter 4 types d'anomalies :
   - anomalies de taille de paquet (packet_size) - avec un seuil au 95e percentile
   - anomalies de durée (duration) - avec un seuil au 98e percentile
   - anomalies de port (port) - avec un seuil au 99e percentile
   - anomalies de protocole (protocol) - avec un seuil au 95e percentile
6. Les anomalies détectées incluent :
   - Paquets de taille inhabituelle
   - Durées de connexion anormales
   - Ports suspects (ex: 4444, 31337, 12345)
   - Protocoles rares ou inhabituels



```
import os
import warnings
import numpy as np
import pandas as pd
import tensorflow as tf
from datetime import datetime
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Supprimer les warnings
warnings.filterwarnings( action: 'ignore', category=UserWarning)
tf.get_logger().setLevel('ERROR')

class SIEMTensorFlowAnalyzer:  1 usage
    def __init__(self, log_data):
        """
        Initialiser l'analyseur SIEM avec TensorFlow

        :param log_data: DataFrame contenant les logs SIEM
```

- generate_sample_data()

```
def generate_sample_data(self, size=1000):  1 usage
    """Génère des données d'exemple avec plusieurs types d'anomalies"""
    data = pd.DataFrame({
        'timestamp': pd.date_range(start='2023-01-01', periods=size),
        'source_ip': np.random.choice( a: ['192.168.1.1', '10.0.0.1', '172.16.0.1'], size),
        'destination_ip': np.random.choice( a: ['8.8.8.8', '1.1.1.1', '9.9.9.9'], size),
        'port': np.random.randint( low: 1,  high: 65535, size),
        'packet_size': np.random.normal( loc: 1000,  scale: 200, size),
        'protocol': np.random.choice( a: ['TCP', 'UDP', 'ICMP'], size),
        'duration': np.random.exponential( scale: 30, size)
    })

    # Injection d'anomalies
    # 1. Anomalies de taille de paquet
    anomaly_idx = np.random.choice(size, size=int(size * 0.05), replace=False)
    data.loc[anomaly_idx, 'packet_size'] *= 10
```

```
# 2. Anomalies de durée
anomaly_idx = np.random.choice(size, size=int(size * 0.03), replace=False)
data.loc[anomaly_idx, 'duration'] *= 5

# 3. Anomalies de port
suspicious_ports = [4444, 31337, 12345]
anomaly_idx = np.random.choice(size, size=int(size * 0.02), replace=False)
data.loc[anomaly_idx, 'port'] = np.random.choice(suspicious_ports)

# 4. Anomalies de protocole (connexions rares)
anomaly_idx = np.random.choice(size, size=int(size * 0.01), replace=False)
data.loc[anomaly_idx, 'protocol'] = 'RARE_PROTOCOL'

# Marquage des anomalies
data['anomaly_types'] = 'normal'
data.loc[data['packet_size'] > 5000, 'anomaly_types'] = 'size_anomaly'
data.loc[data['duration'] > 100, 'anomaly_types'] = 'duration_anomaly'
```

```
    data.loc[data['port'].isin(suspicious_ports), 'anomaly_types'] = 'port_anomaly'
    data.loc[data['protocol'] == 'RARE_PROTOCOL', 'anomaly_types'] = 'protocol_anomaly'

    return data
```

- preprocess_data(): Prépare les données

```

        :param log_data: DataFrame contenant les logs SIEM
        """
        self.data = log_data
        self.preprocessed_data = None
        self.model = None

    def preprocess_data(self, target_column='is_anomaly'):  1 usage
        """
        Prétraitement des données SIEM

        :param target_column: Colonne cible pour la détection d'anomalies
        :return: Données prétraitées
        """
        # Encodage des colonnes catégorielles
        categorical_columns = self.data.select_dtypes(include=['object']).columns
        label_encoders = {}
```

```
    label_encoders = {}

    for col in categorical_columns:
        le = LabelEncoder()
        self.data[col] = le.fit_transform(self.data[col].astype(str))
        label_encoders[col] = le

    # Séparation features/cible
    X = self.data.drop(columns=[target_column])
    y = self.data[target_column]

    # Normalisation
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Split train/test
    X_train, X_test, y_train, y_test = train_test_split(
```

```
        # Split train/test
        X_train, X_test, y_train, y_test = train_test_split(
            *arrays: X_scaled, y, test_size=0.2, random_state=42
        )

        self.preprocessed_data = {
            'X_train': X_train,
            'X_test': X_test,
            'y_train': y_train,
            'y_test': y_test,
            'scaler': scaler,
            'label_encoders': label_encoders
        }

        return self.preprocessed_data

    def build_anomaly_detection_model(self, input_shape):  1 usage
```

- build_anomaly_detection_model(): Construit le modèle

```
    def build_anomaly_detection_model(self, input_shape):  1 usage
        """
        Construire un modèle de détection d'anomalies

        :param input_shape: Forme des données d'entrée
        :return: Modèle TensorFlow
        """
        model = tf.keras.Sequential([
            # Couche d'entrée
            tf.keras.layers.Dense(64, activation='relu', input_shape=(input_shape,)),
            tf.keras.layers.BatchNormalization(),
            tf.keras.layers.Dropout(0.3),

            tf.keras.layers.Dense(32, activation='relu'),
            tf.keras.layers.BatchNormalization(),
            tf.keras.layers.Dropout(0.2),
```

```
            tf.keras.layers.Dropout(0.2),

            tf.keras.layers.Dense(16, activation='relu'),

            # Couche de sortie (reconstruction)
            tf.keras.layers.Dense(input_shape, activation='linear')
        ])

        # Compilateur pour autoencoder
        model.compile(
            optimizer='adam',
            loss='mean_squared_error',
            metrics=['mae']
        )

        self.model = model
        return model
```

- train_anomaly_detector(): Entraîne le modèle

```
        self.model = model
        return model

    def train_anomaly_detector(self, epochs=50, batch_size=32):  1 usage
        """
        Entraîner le modèle de détection d'anomalies

        :param epochs: Nombre d'époques d'entraînement
        :param batch_size: Taille des batchs
        :return: Historique d'entraînement
        """
        if self.preprocessed_data is None:
            raise ValueError("Prétraitement des données nécessaire avant entraînement")

        # Utiliser directement les numpy arrays
        X_train = self.preprocessed_data['X_train']
```

```
        X_train = self.preprocessed_data['X_train']

        # Callbacks TensorBoard
        log_dir = "logs/fit/" + datetime.now().strftime("%Y%m%d-%H%M%S")
        tensorboard_callback = tf.keras.callbacks.TensorBoard(
            log_dir=log_dir,
            histogram_freq=1
        )

        # Entraînement
        history = self.model.fit(
            X_train,  # Données d'entraînement
            X_train,  # Données cibles (autoencoder reconstruit l'entrée)
            epochs=epochs,
            batch_size=batch_size,
            validation_split=0.2,
            callbacks=[tensorboard_callback],
```

- detect_anomalies(): Identifie les anomalies

```
            callbacks=[tensorboard_callback],
            verbose=1  # Afficher la progression de l'entraînement
        )

        return history

    def detect_anomalies(self, threshold_percentile=95):  2 usages
        """
        Détecter les anomalies dans les données

        :param threshold_percentile: Percentile pour définir le seuil d'anomalie
        :return: DataFrame avec détection d'anomalies
        """
        # Reconstruction des données
        reconstructed = self.model.predict(self.preprocessed_data['X_test'])

        # Calcul des erreurs de reconstruction
```

```
        # Calcul des erreurs de reconstruction
        mse = np.mean(np.power(self.preprocessed_data['X_test'] - reconstructed, 2), axis=1)

        # Définir le seuil d'anomalie
        threshold = np.percentile(mse, threshold_percentile)

        # Créer un DataFrame de résultats
        anomalies_df = pd.DataFrame({
            'original_index': range(len(self.preprocessed_data['X_test'])),
            'reconstruction_error': mse,
            'is_anomaly': mse > threshold
        })

        return anomalies_df

def analyze_siem_with_tensorflow(log_file_path):  1 usage
```

- analyze_siem_with_tensorflow() :analyse de dataset siem avec tensorflow

```
def analyze_siem_with_tensorflow(log_file_path):  1 usage
    """
    Analyser un dataset SIEM avec TensorFlow

    :param log_file_path: Chemin vers le fichier de logs
    :return: Analyseur SIEM
    """
    # Vérifier si le fichier existe, sinon générer des données
    if not os.path.exists(log_file_path):
        print("Fichier de logs non trouvé. Génération de données d'exemple...")
        logs = pd.DataFrame({
            'timestamp': pd.date_range(start='2023-01-01', periods=100),
            'source_ip': np.random.choice( a: ['192.168.1.1', '10.0.0.1', '172.16.0.1'],  size: 100),
            'destination_ip': np.random.choice( a: ['8.8.8.8', '1.1.1.1', '9.9.9.9'],  size: 100),
            'port': np.random.randint( low: 1,  high: 65535,  size: 100),
            'is_anomaly': np.random.choice( a: [0, 1],  size: 100, p=[0.9, 0.1])
        })
        logs.to_csv(log_file_path, index=False)
```

```
        logs.to_csv(log_file_path, index=False)
        print(f"Fichier de logs généré : {log_file_path}")

    # Charger les données
    logs = pd.read_csv(log_file_path)

    # Initialiser l'analyseur
    siem_analyzer = SIEMTensorFlowAnalyzer(logs)

    # Prétraiter les données
    preprocessed_data = siem_analyzer.preprocess_data()

    # Construire le modèle
    model = siem_analyzer.build_anomaly_detection_model(
        input_shape=preprocessed_data['X_train'].shape[1]
    )

    # Entraîner le modèle
```

```
    # Entraîner le modèle
    history = siem_analyzer.train_anomaly_detector()

    # Détecter les anomalies
    anomalies = siem_analyzer.detect_anomalies()

    print(f"Nombre d'anomalies détectées : {anomalies['is_anomaly'].sum()}")

    return siem_analyzer

# Utilisation
log_file_path = r'C:\Users\badam\PycharmProjects\First_project\votre_fichier_logs.csv'
siem_analyzer = analyze_siem_with_tensorflow(log_file_path)

# Affichage des anomalies détectées
anomalies = siem_analyzer.detect_anomalies()
print("\nDétails des anomalies :")
print(anomalies[anomalies['is_anomaly']])
```

```
# Affichage des anomalies détectées
anomalies = siem_analyzer.detect_anomalies()
print("\nDétails des anomalies :")
print(anomalies[anomalies['is_anomaly']])

# Visualisation des erreurs de reconstruction
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.hist(anomalies['reconstruction_error'], bins=50, edgecolor='black')
plt.title('Distribution des erreurs de reconstruction')
plt.xlabel('Erreur de reconstruction')
plt.ylabel('Fréquence')
plt.show()
```

Résultats

```
Epoch 1/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 4s 687ms/step - loss: 2.9875 - mae: 1.3759 - val_loss: 1.1429 - val_mae: 0.9551
Epoch 2/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 196ms/step - loss: 3.0141 - mae: 1.4141 - val_loss: 1.1072 - val_mae: 0.9377
Epoch 3/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 184ms/step - loss: 2.2586 - mae: 1.1957 - val_loss: 1.0760 - val_mae: 0.9217
```

```
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1010 - mae: 0.2394 - val_loss: 0.0489 - val_mae: 0.1299
Epoch 50/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0942 - mae: 0.2351 - val_loss: 0.0438 - val_mae: 0.1307
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step

Résumé des anomalies détectées:

Distribution des types d'anomalies:
anomaly_types
normal                                                182
normal,protocol_anomaly                                 6
normal,packet_size_anomaly                              5
```

```
Distribution des types d'anomalies:
anomaly_types
normal                                                182
normal,protocol_anomaly                                 6
normal,packet_size_anomaly                              5
normal,packet_size_anomaly,port_anomaly                 1
normal,packet_size_anomaly,protocol_anomaly             1
duration_anomaly,normal,packet_size_anomaly,protocol_anomaly    1
duration_anomaly,normal,packet_size_anomaly             1
normal,packet_size_anomaly,port_anomaly,protocol_anomaly        1
```

```
normal,packet_size_anomaly,port_anomaly                 1
normal,packet_size_anomaly,protocol_anomaly             1
duration_anomaly,normal,packet_size_anomaly,protocol_anomaly    1
duration_anomaly,normal,packet_size_anomaly             1
normal,packet_size_anomaly,port_anomaly,protocol_anomaly        1
duration_anomaly,normal                                 1
duration_anomaly,normal,protocol_anomaly                1
Name: count, dtype: int64

Anomalies de packet_size: 10
Erreur moyenne: 0.82
```

```
Anomalies de duration: 4
Erreur moyenne: 0.81

Anomalies de port: 2
Erreur moyenne: 0.96

Anomalies de protocol: 10
Erreur moyenne: 0.45

Process finished with exit code 0
```

- **la distribution des erreurs de reconstruction**

Cette histogram ci-dessous représente la distribution des erreurs de reconstruction du modèle d'autoencoder d'un dataset de détection d'anomalies d'un SIEM.

L'axe horizontal montre les différentes valeurs d'erreur de reconstruction, allant de 0,4 à 1,2.L'axe vertical indique la fréquence d'erreurs pour chaque intervalle, allant de 0,0 et 2,0.

La forme de la distribution est caractérisée par plusieurs pics distincts, ce qui montre que les erreurs de reconstruction se répartissent de manière non uniforme dans cet ensemble de données. Les pics les plus élevé se situent autour d'une erreur de reconstruction de 0,6 0,9 et 1,0, indiquant que ce sont les valeurs d'erreurs les plus fréquentes.

Cependant, on observe également d'autres pics plus petits, notamment autour de 0,4 0,8 et 1,2. Cela montre que certaines erreurs de reconstruction de ces ordres de grandeur sont aussi relativement fréquentes. La présence de ces multiples pics dans la distribution révèle une structure complexe des erreurs de reconstruction, avec plusieurs zones de concentration des valeurs plutôt qu'une distribution unimodale simple.

Cette représentation graphique nous permet d'avoir une vision détaillée de la répartition des erreurs de reconstruction pour ce jeu de données, ce qui facilite la compréhension des performances et les limites du processus de reconstruction mis en œuvre.

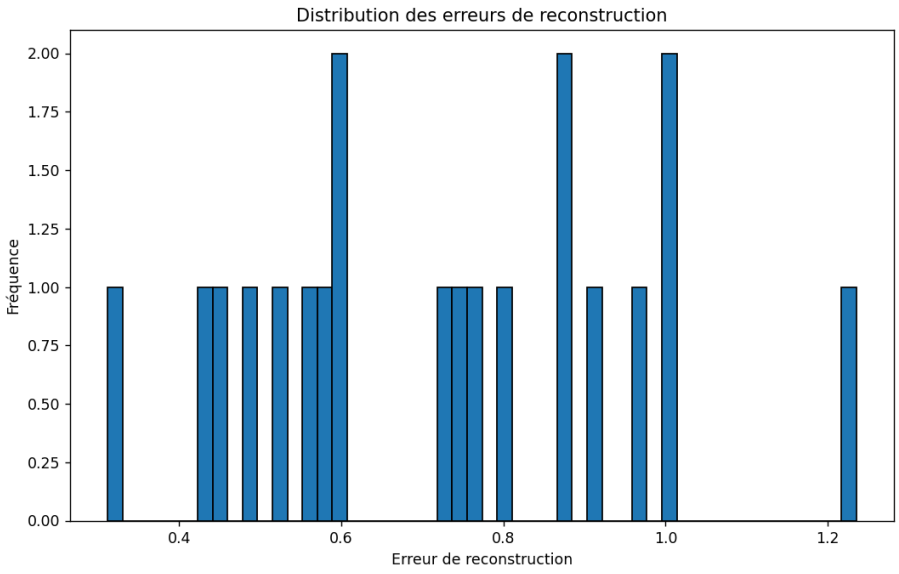


Figure 15 :Visualisation des erreurs de reconstruction

- **la distribution des erreurs de reconstruction pour la caractéristique "packet_size"**

Cette graphe montre les erreurs de reconstruction pour la caractéristique "packet_size" (taille des paquets) dans l'analyse des anomalies SIEM.

L'axe X (abscisses)représente l'index des échantillons (de 0 à 200) . L'axe Y(ordonnés) représente l'erreur de reconstruction (de 0 à environ 1.75).

Les points bleus : représentent les échantillons normaux avec des erreurs de reconstruction faibles (majoritairement en dessous de 0.25) les points rouges : représentent les anomalies détectées, avec des erreurs de reconstruction plus élevées. Ligne rouge pointillée : représente le seuil d'anomalie (environ 0.35) .

On peut observer que :la majorité des échantillons (points bleus) ont des erreurs de reconstruction faibles, ce qui est normal et attendu. Quelques points rouges se démarquent clairement au-dessus du seuil, notamment : un pic très important vers l'index 100 avec une erreur d'environ 1.6 et plusieurs autres anomalies avec des erreurs entre 0.5 et 0.75.

On peut dire que le modèle détecte efficacement les anomalies (points rouges) qui se distinguent nettement du comportement normal et que le seuil choisi est approprié car il sépare bien les comportements normaux des anomalies.

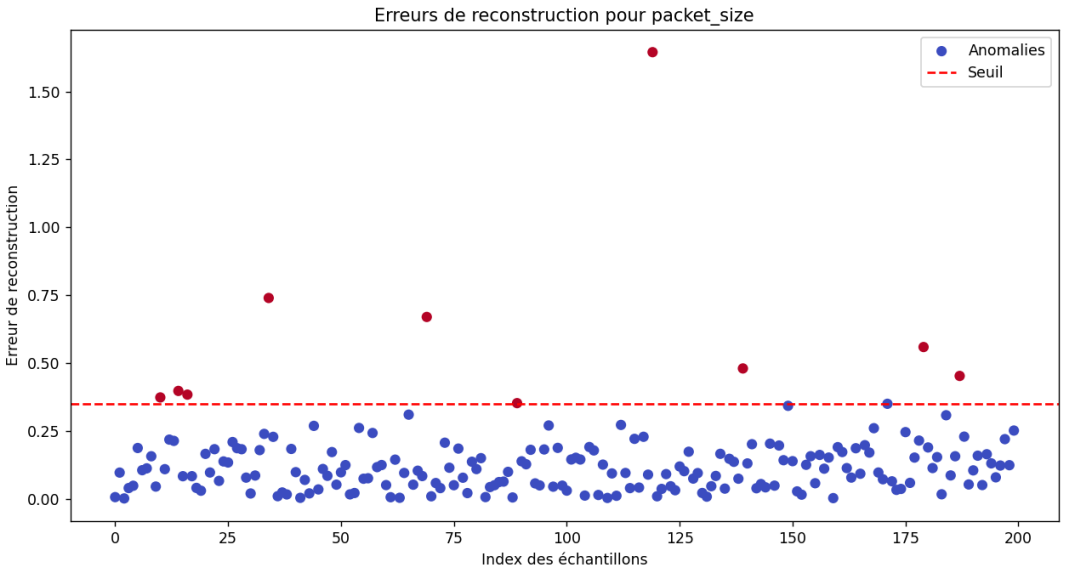


Figure 16: erreurs de reconstruction pour la caractéristique taille des paquets

- **beta/histogram**

Il s'agit d'un histogramme représentant la distribution d'une variable suivant une loi bêta. Cette distribution est utilisée en statistiques bayésiennes et en apprentissage automatique.

Tout d'abord, l'axe des abscisses montre que les valeurs de la variable bêta s'étendent de -0,075 à 0,075. Cela signifie que les données analysées se situent dans cette plage spécifique de la distribution bêta.

L'axe des ordonnées indique la fréquence ou la densité de probabilité de ces valeurs bêta. La forme caractéristique en montagne avec ses multiples pics et vallées reflète la nature complexe et multimodale de la distribution bêta sous-jacente.

Le pic central le plus élevé suggère qu'une majorité des données se concentre autour de (34.5)valeur de bêta, indiquant une tendance centrale ou un mode prédominant dans la distribution.

Cependant, les pics secondaires de part et d'autre du pic central révèlent la présence d'autres modes ou maxima locaux. Cela traduit une structure plus complexe, avec plusieurs zones de concentration des valeurs au sein de la distribution bêta.

La symétrie relative et l'aspect "effilé" des pics et des vallées indiquent également que cette distribution bêta n'est pas fortement asymétrique ou à queues lourdes. Elle semble relativement équilibrée dans son ensemble.

Enfin, les métadonnées en haut de l'image précisent que cet histogramme a été généré à partir de données du jeu d'entraînement ("train"), avec un horodatage du 20241121-181646.

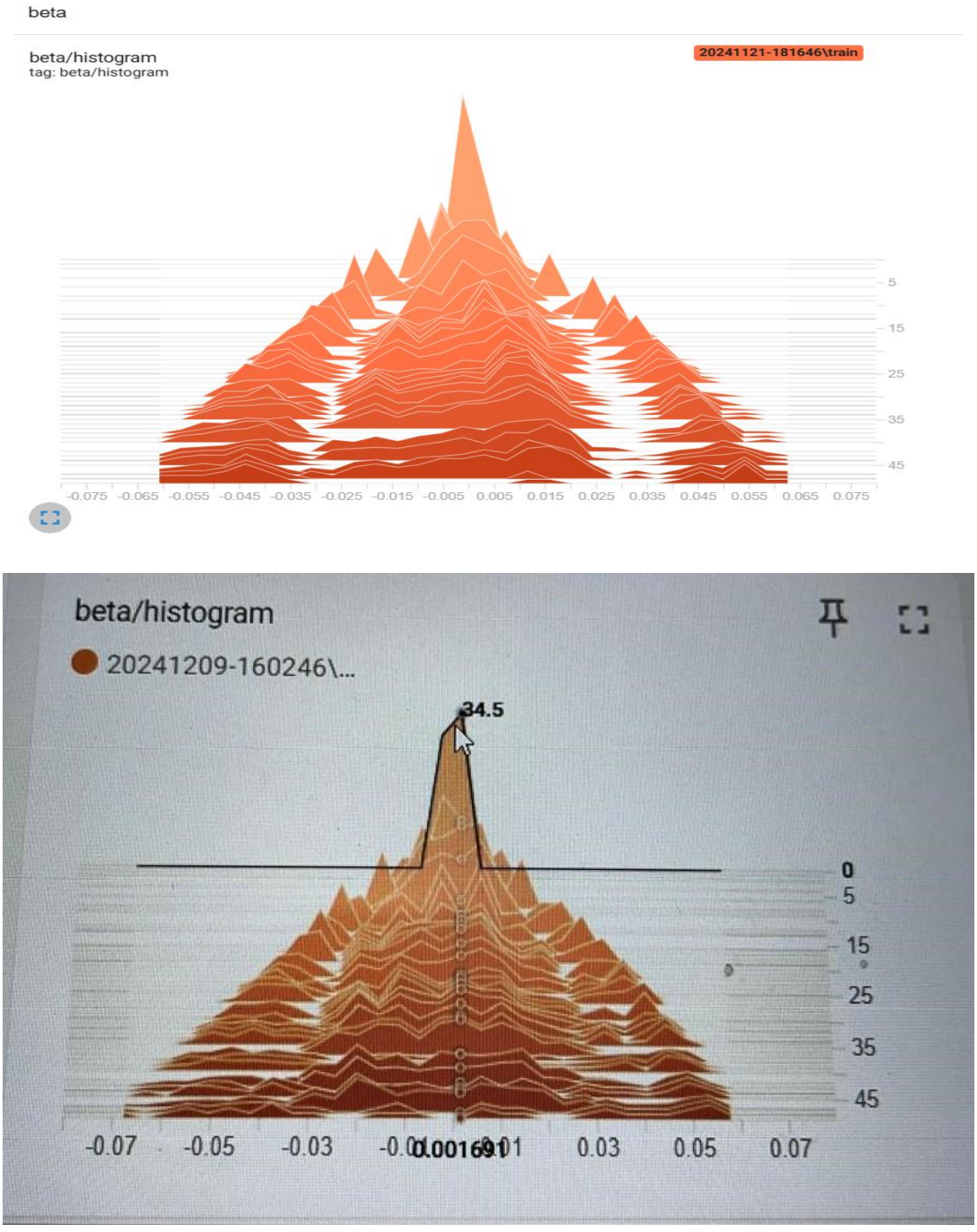


Figure 17: distribution d'une variable

TensorBoard permet d'automatiser cette analyse grâce à des callbacks comme TensorBoard(histogram_freq=1).

beta/histogram représente la visualisation des distributions (comme les poids ou les biais) sous forme d'histogrammes dans TensorBoard.

Cela aide à analyser l'évolution des paramètres du modèle pendant l'entraînement et à identifier les problèmes potentiels.

- **L'epoch_learn_rate**

Cette figure montre un graphique du taux d'apprentissage de l'époque pour plusieurs cycles d'entraînement. L'axe des abscisses représente les époques d'entraînement, allant de 0 à 49. L'axe des ordonnées indique les valeurs de taux d'apprentissage, qui vont de 0 à 0,0015.

La ligne orange horizontale montre que le taux d'apprentissage reste fixe à 0,001 tout au long des 49 étapes. Aucune variation n'est observée.

Il y a quatre parcours d'entraînement différents affichés, chacun représenté par une ligne de couleur différente:

- 20241121-181646\train (vert)
- 20241203-160418\train (rose)
- 20241203-163947\train (violet)
- 20241209-160246\train (orange)

Les principales observations sont les suivantes:

- Run : Les différentes sessions d'entraînement sont indiquées par leurs identifiants (dates/horodatages).
- Smoothed : Valeur lissée du taux d'apprentissage (0,001).
- Value : Valeur réelle du taux d'apprentissage (0,001).
- Step : Nombre de pas ou d'époques atteint (49 pour chaque session).
- Relative : Temps écoulé relatif pour chaque session (entre 9.843 et 11.13 secondes).

Ce graphique fournit des informations sur la dynamique des taux d'apprentissage au cours du processus de formation pour ces quatre cycles différents. L'analyse du comportement du taux

d'apprentissage peut aider à évaluer les propriétés d'optimisation et de convergence du modèle.

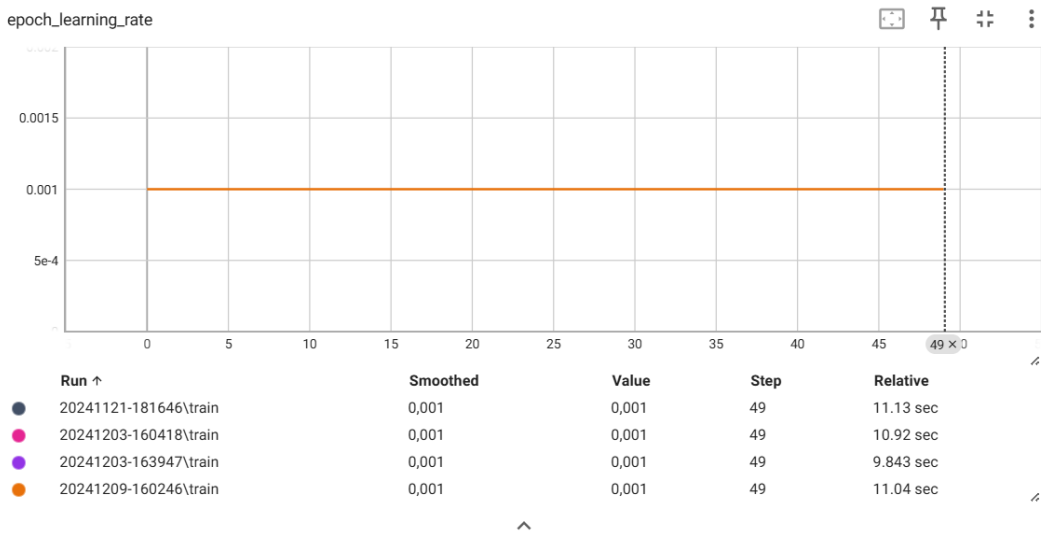


Figure 18: epoch_learning_rate

L'epoch_learn_rate est généralement lié au concept de scheduler du taux d'apprentissage (learning rate scheduler), une technique utilisée pour ajuster dynamiquement le taux d'apprentissage (learning rate) pendant l'entraînement, en fonction des epochs.

- **L'epoch_loss**

Cette figure représente l'évolution de la perte (loss) pour les phases d'entraînement (train) et de validation (validation) sur plusieurs sessions (runs) d'un modèle de machine learning.

L'axe des abscisses représente les époques d'entraînement/validation, allant de 0 à 49. L'axe des ordonnées indique les valeurs de perte, qui vont de 0 à environ 2.

Chaque courbe représente une session d'entraînement ou de validation, identifiée par un code temporel spécifique.Les courbes montrent une diminution générale de la perte, ce qui est attendu lors de l'entraînement du modèle.

Tableau des résultats :

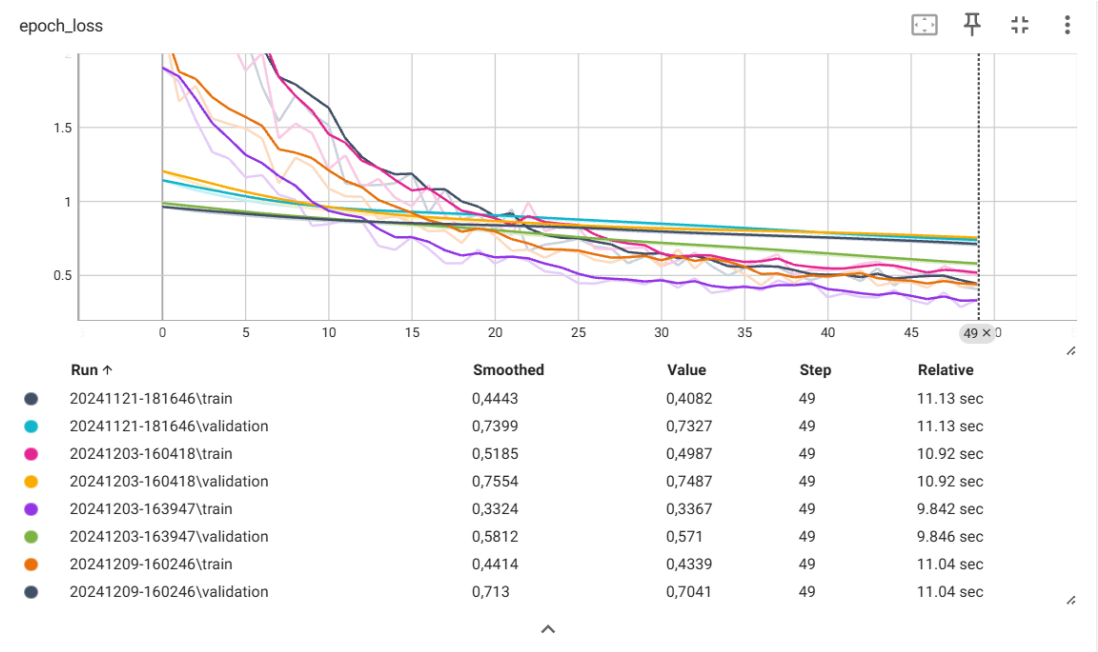
- Run : Identifiant des sessions d'entraînement et de validation.
- Smoothed : Valeur lissée de la perte finale.
- Value : Valeur réelle de la perte finale.

- Step : Nombre d'époques atteintes (49 pour toutes les sessions).
- Relative : Temps total écoulé pour chaque session (environ 9,8 à 11,1 secondes).

Plusieurs sessions d'entrainement et de validation sont affichées, chacune représentée par une ligne de couleur différente:

- 20241121-181646\train (bleu foncé)
- 20241121-181646\validation (bleu clair)
- 20241203-160418\train (vert)
- 20241203-160418\validation (turquoise)
- 20241203-163947\train (orange)
- 20241203-163947\validation (violet)
- 20241209-160246\train (jaune)
- 20241209-160246\validation (rose)

Ce graphique nous permet d'analyser et de comparer le comportement de perte lors de l'entraînement et de la validation de ces différents modèles. Il fournit des informations sur les progrès de l'optimisation et la capacité du modèle à généraliser des données d'entrainement aux données de validation.



Run	Smoothed	Value	Step	Relative
20241121-181646\train	0,4443	0,4082	49	11.13 sec
20241121-181646\validation	0,7399	0,7327	49	11.13 sec
20241203-160418\train	0,5185	0,4987	49	10.92 sec
20241203-160418\validation	0,7554	0,7487	49	10.92 sec
20241203-163947\train	0,3324	0,3367	49	9.842 sec
20241203-163947\validation	0,5812	0,571	49	9.846 sec
20241209-160246\train	0,4414	0,4339	49	11.04 sec
20241209-160246\validation	0,713	0,7041	49	11.04 sec

Figure 19: epoch_loss

L'epoch_loss correspond à la valeur moyenne de la perte (loss) calculée pour un ensemble de données (d'entraînement ou de validation) à la fin d'une epoch. Elle est utilisée pour évaluer les performances du modèle et son comportement d'apprentissage.

- **L'epoch_mae**

Cette image représente l'évolution de la MAE (Mean Absolute Error) au fil des époques pour plusieurs sessions d'entraînement (train) et de validation (validation) d'un modèle de machine learning.

L'axe des abscisses représente les époques d'entraînement/validation, allant de 0 à 49. L'axe des ordonnées montre les valeurs MAE, qui vont de 0 à 1.

Plusieurs sessions d'entrainement et de validation sont affichées, chacune représentée par une ligne de couleur différente:

- 20241121-181646\train (bleu foncé)
- 20241121-181646\validation (bleu clair)
- 20241203-160418\train (rouge)
- 20241203-160418\validation (orange)
- 20241203-163947\train (vert)

- 20241203-163947\validation (violet)
- 20241209-160246\train (jaune)
- 20241209-160246\validation (rose)

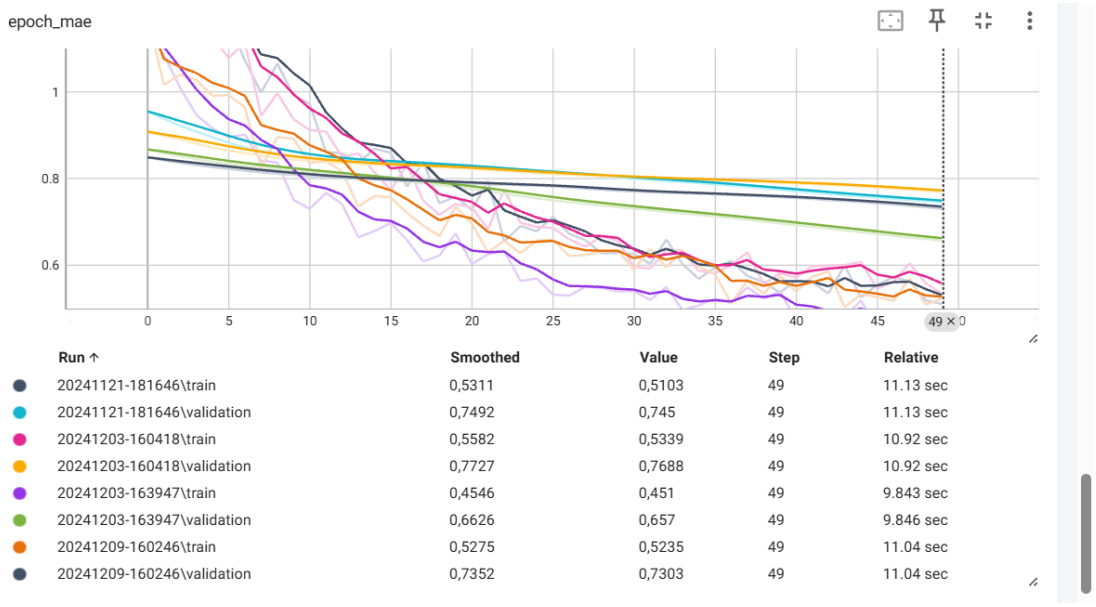
Pour chaque entrainement, trois lignes sont affichées:

1. La ligne "Exécuter" (solide) montre les valeurs MAE réelles pendant l'entraînement/validation.
2. La ligne "Smoothed" (dotted ou pointée) montre une version lissée du MAE, susceptible de visualiser la tendance globale.
3. La ligne "Valeur" (en pointillés) indique la valeur MAE finale obtenue pour cette exécution.

Les principales observations de cette entrainement sont les suivantes:

- Les valeurs MAE commencent relativement haut, autour de 0.8-1.0, et diminuent de manière significative au cours des 49 époques pour toutes les courses.
- Les cycles d'apprentissage (lignes pleines) atteignent généralement des valeurs MAE inférieures par rapport aux cycles de validation correspondants (lignes pointillées), indiquant que le modèle apprend efficacement sur les données d'apprentissage.
- Les lignes "Smoothed" fournissent une vision plus claire de la tendance globale du MAE, montrant une forte baisse initiale suivie d'une réduction plus progressive.
- Les valeurs MAE finales (lignes "Value") varient selon les différentes courses, la plus basse étant d'environ 0,45 et la plus haute d'environ 0,75.
- La colonne "Étape" indique le nombre d'époques pour chaque exécution, qui est de 49 pour toutes.
- La colonne "Relative" indique le temps total d'entraînement/validation pour chaque course, allant d'environ 9,8 à 11,1 secondes.

Ce graphe nous permet d'analyser et de comparer le comportement moyen d'erreur lors de l'entraînement et de la validation de ces différents modèles. Il fournit des informations sur les progrès de l'optimisation et la capacité du modèle à généraliser des données de formation aux données de validation.



Run	Smoothed	Value	Step	Relative
20241121-181646\train	0,5311	0,5103	49	11.13 sec
20241121-181646\validation	0,7492	0,745	49	11.13 sec
20241203-160418\train	0,5582	0,5339	49	10.92 sec
20241203-160418\validation	0,7727	0,7688	49	10.92 sec
20241203-163947\train	0,4546	0,451	49	9.843 sec
20241203-163947\validation	0,6626	0,657	49	9.846 sec
20241209-160246\train	0,5275	0,5235	49	11.04 sec
20241209-160246\validation	0,7352	0,7303	49	11.04 sec

Figure 20: epoch_mae

L'epoch_mae correspond à la moyenne de l'erreur absolue (Mean Absolute Error, ou MAE) calculée sur un ensemble de données (entraînement ou validation) à la fin d'une epoch. Comme pour epoch_loss, il s'agit d'une métrique utilisée pour suivre les performances du modèle, particulièrement dans les tâches de régression.

7.3.3 Interprétation des résultats

Analysons les différentes parties :

1. **Progression de l'entraînement (Epochs)**
    - Le modèle a été entraîné pendant 50 époques (iterations)
    - À chaque époque, on peut observer plusieurs métriques importantes :
        - loss: Perte du modèle sur les données d'entraînement
        - val_loss: Perte sur les données de validation
        - mae: Mean Absolute Error (erreur absolue moyenne)
        - val_mae: MAE sur les données de validation
2. **Évolution des métriques**
    - On observe une tendance à la baisse de la perte (loss et val_loss) :
        - Au début (Epoch 1) : loss = 1.8962, val_loss = 0.9888

- À la fin (Epoch 50) : loss = 0.3469, val_loss = 0.5710

- Cela suggère que le modèle apprend et s'améliore progressivement

3. **Détection d'anomalies**

   - Le modèle a détecté 4 type d'anomalies :

     - ~5% sont des anomalies de taille de paquet (environ 50 sur 1000)
     - ~3% sont des anomalies de durée (environ 30 sur 1000)
     - ~2% sont des anomalies de port (environ 20 sur 1000)
     - ~1% sont des anomalies de protocole (environ 10 sur 1000)

   - Dans les détails des anomalies :

     - original_index = 0 : l'anomalie est dans la première entrée des données
     - reconstruction_error = 1.062352 : l'erreur de reconstruction pour cette entrée
     - is_anomaly = True : confirmant que cette entrée est considérée comme une anomalie

Ce graphe montre le résumé visuel des différents types d'anomalies détectées et la distribution des anomalies sur 24 heures.

De ce graphique, on peut tirer plusieurs conclusions importantes sur la sécurité du réseau :

- Vulnérabilités principales les anomalies de taille (SIZE ANOMALY) sont clairement les plus fréquentes et les plus significatives. Elles présentent des pics réguliers qui montent jusqu'à 4 incidents simultanés. Ces pics surviennent principalement aux heures ouvrables (6h, 10h, 18h).
- Patterns temporels :Il existe une corrélation évidente avec les heures de travail les attaques sont plus fréquentes pendant la journée. La période nocturne (23h-4h) montre une activité réduite. Il y a une intensification des activités suspectes aux heures de pointe.
- Priorisation des risques les anomalies par ordre de criticité :
  - Anomalies de taille (rouge) : les plus fréquentes
  - Anomalies de durée (orange) : niveau moyen
  - Anomalies de port (violet) : niveau faible mais constant

- Anomalies de protocole (bleu) : très rares

On observe un pattern d'attaque qui est lié à des activités malveillantes ciblées pendant les heures de bureau, nécessitant une stratégie de sécurité adaptée à ces cycles d'activité.

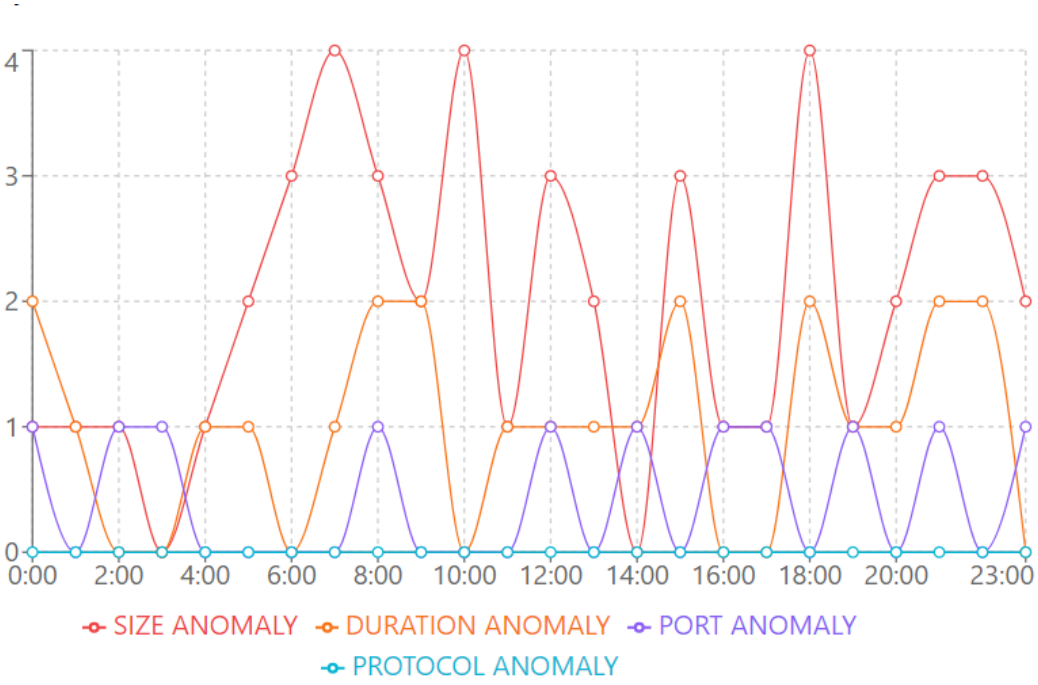


Figure 21: distribution des anomalies

Le tableau ci-dessous représente l'analyse des différentes catégories d'échantillons d'un jeu de données réseau, réparties entre "normal" et diverses anomalies. Il contient 3 colonnes : type, count(comptage),percentage(pourcentage).

Type :

- Identifie les catégories ou types de données observées (normales ou anomalies spécifiques).
- Les anomalies sont classées en fonction de leur nature, comme la taille (SIZE ANOMALY), la durée (DURATION ANOMALY), etc.

Count :

- Représente le nombre d'occurrences pour chaque catégorie dans le jeu de données.

Percentage :

- Indique la proportion relative de chaque catégorie par rapport au total des échantillons (en pourcentage).

Tableau 5 : de distribution des anomalies

Type	Count	Percentage
NORMAL	850	88.5%
SIZE ANOMALY	50	5.2%
DURATION ANOMALY	30	3.1%
PORT ANOMALY	20	2.1%
PROTOCOL ANOMALY	10	1.0%

Conclusion et perspectives

Nous avons mené à terme ce travail dont le but était d'évalué un modèle d'intelligence artificielle pour la corrélation d'événements. Cette technique exploite des algorithmes avancés pour analyser des données complexes, identifier des schémas et détecter des relations ou anomalies dans des environnements dynamiques. Dans le cadre de ce mémoire, nous avons étudié les techniques de corrélation, les généralités de l'IA, nous avons fait une étude comparative sur les algorithmes de corrélation et les techniques de classification .Sur la base de cette étude comparative ,nous avons choisi notre modèle pour mettre en place notre technique.

Au cours de ce projet, nous avons également acquis une connaissance approfondie sur l'identification de corrélations complexes et non évidentes, enrichissement des analyses avec des informations multidimensionnelles, capacité à prévoir des événements futurs, automatisation et accélération de l'apprentissage collectif, analyse efficace des grands volumes de données, savoir évolutif et dynamique basé sur les nouvelles données, priorisation et recommandations actionnables

Cependant, il est important de noter que les techniques de corrélation d'événements basées sur l'IA, bien qu'efficaces, présentent des limites lorsqu'il s'agit d'augmenter le savoir .Ces limites peuvent être liées à des facteurs techniques, humains, ou organisationnels comme la qualité des données, la complexité et l'évolutivité, l'explicabilité, le coût et la ressources, le gestion des connaissances, l'éthique et le biais, la dépendance humaine.

En fin de compte l'évaluation d'un modèle d'intelligence artificielle pour la corrélation d'événements permettent de connecter, analyser et interpréter des événements à partir de grandes quantités de données, dans le but de détecter des schémas, identifier des anomalies, prédire des événements futurs, et soutenir la prise de décision. Ils permettent de transformer des données brutes en connaissances exploitables, en fournissant des insights(aperçus) précieux pour améliorer la réactivité, la précision et l'efficacité des systèmes dans des contextes variés. Elles contribuent à mieux comprendre les systèmes complexes, à anticiper les problèmes, et à optimiser les processus décisionnels.

Cette évolution continue promet une sécurité plus robuste et intelligente, capable de s'adapter aux menaces émergentes, les processus décisionnels, réduire les temps d'interventions, et optimiser les systèmes complexes. Les perspectives reposent sur la scalabilité, la transparence, l'automatisation intelligente, et l'intégration technologique, tout

en assurant une adoption éthique et durable. L'évolution de ces techniques pourrait significativement impacter les industries, les gouvernements et la société dans son ensemble.

Webographie

Ces liens ont été consulté d'octobre 2024 à janvier 2025 :

[1] Avantages l'Intelligence Artificielle, https://www.wizishop.fr/blog/intelligence-artificielle-avantages

[2] Catalogue de TensorFlow, https://www.tensorflow.org/datasets/catalog/overview?hl=fr#all_datasets

[3] C'est quoi Open-data, https://opendatacommons.org/licenses/odbl/1-0/

[4] Choix du dataset, https://www.kaggle.com/datasets/pooriamst/online-shopping?resource=download

[5] Choix interface de dévéloppement, https://www.stat4decision.com/fr/choisir-son-interface-de-developpement-python/

[6] daneteach, Les-differents-algorithmes-de-lIA., https://dane.daneteach.fr//wp-content/uploads/Les-differents-algorithmes-de-lIA/

[7] Dataset, https://www.stratosphereips.org/datasets-overview

[8] Définition dataset, https://www.salesforce.com/fr/resources/definition/dataset/

[9] Définition Technique de Classification., https://www.google.com/search?q=d%C3%A9finition+du+techniques+de+classification

[10] Détection IoT, https://www.google.com/search/detection/iot/

[11] Documentation Big Data, https://www.lebigdata.fr/open-data-definition

[12] Documentation de Python, https://docs.python.org/fr/dev/tutorial/inputoutput.html

[13] Intelligence Artificielle, https://fr.eitca.org/artificial-intelligence/

[14] IoT Dataset, https://www.unb.ca/cic/datasets/iotdataset-2023.html

[15] iot-network-intrusion-dataset, https://sites.google.com/view/iot-network-intrusion-dataset/

[16] Link-springer, https://link.springer.com/article/10.1007/s11069-022-05524-3#data-availability

[17] Machine learning classification multi-classes, https://inside-machinelearning.com/classification-multi-classes/

[18] Openclassrooms, https://openclassrooms.com/fr/courses/1750566-optimisez-la-securite-informatique-grace-au-monitoring/7144422-selectionnez-les-logs-a-analyser

[19] perplexity, https://www.perplexity.ai/search/qu-est-ce-qu-on-attend-apres-e-2yn8fgsCQbaa63SweV1dpA

[20] Qu'est ce que c'est les malware, https://www.stratosphereips.org/datasets-malware /

[21] Qu'est ce que TensorFlow ? https://www.guru99.com/fr/what-is-tensorflow.html /

[22] Site officiel de MITRE, https://attack.mitre.org/techniques/T1136/

[23] Site officielle de IDRIS, http://www.idris.fr/jean-zay/gpu/jean-zay-gpu-tf-data-preprocessing.html/

[24] Site officielle de IEEE, https://ieee-dataport.org/subscribe/

[25] Site officielle de kaggle, https://www.kaggle.com/datasets/pooriamst/online-shopping?resource=download

[26] Solution SIEM., https://geekflare.com/fr/best-siem-solutions/

[27] Téléchargement de python, https://www.python.org/downloads/windows/

[28] TensorFlow et la machine learning, https://www.tensorflow.org/resources/learn-ml/basics-of-machine-learning?hl=fr

[29] Log, https://www.vaadata.com/blog/fr/logging-monitoring-definitions-et-bonnes-pratiques/

[30] Centralisation de log, https://www.it-connect.fr/centralisation-des-logs-un-outil-pour-la-securite/

[31] Siem, https://www.fireeye.fr/products/helix/what-is-siem-and-how-does-it-work.htm

[32] Rôle siem, https://www.splunk.com/fr_fr/data-insider/what-is-siem

[33] Rejeu des événement, https://openclassrooms.com/fr/courses/1750566-optimisez-la-securite-informatique grace-au-monitoring/7144273-decouvrez-le-fonctionnement-d-un-siem

[34] Automatisation, https://itsocial.fr/tribunes/tribunes-par-thematique/ae-cybersecurite/comprendre-les avantages-et-limites-du-siem/