Une image contenant texte, Police, logo, Marque

Description générée automatiquement

Rapport de projet tuteuré en Réseaux informatique

**Thème : Application des réseaux neurones à la détection d'intrusions réseaux avec (TensorFlow et Keras)**

Filière Ingénierie des Systèmes informatiques

2er année cycle ingénieur

**Encadré Par** **:** **Pr.** LAYLA WAKRIM

**Rédigé par :**

**SONDE Oubaida**

**KONATE Moussa**

**SANOGO Farima**

**ATABA Keng**

**BIBANG Armand Nang**

2022-2023

Table des matières

[I. Introduction 3](#_Toc138121491)

[II. Concepts de base en détection d'intrusions réseau 4](#_Toc138121492)

[1. Concepts et terminologies liés à la détection d'intrusions réseau 4](#_Toc138121493)

[2. Techniques et méthodes de détection utilisées 4](#_Toc138121494)

[III. Études de l'existant 5](#_Toc138121495)

[1. Les travaux de recherche existants dans le domaine 6](#_Toc138121496)

[**2.** Avantages, des limites et des résultats obtenus dans les études précédentes 7](#_Toc138121497)

[IV. Base de données 8](#_Toc138121498)

[**1.** Données utilisées pour l'entraînement et l'évaluation du modèle 8](#_Toc138121499)

[2. Des critères de sélection du jeu de données 9](#_Toc138121500)

[V. Implémentation 10](#_Toc138121501)

[**1.** Prétraitement des données 10](#_Toc138121502)

[**2.** Conception du modèle et Entrainement du modèle sur le jeu de données KDDTrain+\_2 14](#_Toc138121503)

[VI. Conclusion 19](#_Toc138121504)

# Introduction

La sécurité des réseaux informatiques est devenue une préoccupation majeure dans notre société de plus en plus connectée. Les intrusions réseau représentent une menace constante pour la confidentialité, l'intégrité et la disponibilité des données. Afin de contrer ces attaques et de protéger les systèmes informatiques, la détection d'intrusions réseau joue un rôle crucial en identifiant les activités malveillantes et en prenant des mesures préventives.

Le présent mémoire se focalise sur l'application des réseaux neuronaux à la détection d'intrusions réseau en utilisant TensorFlow et Keras. L'objectif de cette recherche est de développer un modèle de détection d'intrusions performant et précis en exploitant les capacités des réseaux neuronaux, tout en bénéficiant des fonctionnalités et de la facilité d'utilisation offertes par les outils TensorFlow et Keras.

La problématique centrale de cette étude repose sur la question suivante : Comment les réseaux neuronaux peuvent-ils être utilisés efficacement pour détecter les intrusions réseau et renforcer la sécurité des systèmes informatiques ?

Ce mémoire est structuré en plusieurs parties. Après cette introduction, nous aborderons dans la deuxième partie les concepts de base en détection d'intrusions réseau, en mettant l'accent sur les techniques et les méthodes utilisées. Dans la troisième partie, nous passerons en revue les travaux de recherche existants dans le domaine de la détection d'intrusions réseau avec des réseaux neuronaux, en soulignant les avantages et les limites des approches existantes.

Ensuite, nous présenterons dans la quatrième partie la collecte de données. La cinquième partie sera consacrée à l'implémentation, où nous détaillerons les étapes de prétraitement des données et de conception du modèle de détection d'intrusions réseau avec TensorFlow et Keras. Enfin, nous conclurons ce mémoire en récapitulant les résultats obtenus, en discutant des limites et des perspectives d'amélioration.

# Concepts de base en détection d'intrusions réseau

## Concepts et terminologies liés à la détection d'intrusions réseau

La détection d'intrusions réseau comprend divers concepts et terminologies qui sont essentiels pour comprendre et analyser les activités malveillantes sur un réseau. Voici quelques-uns des concepts clés :

* **Intrusion** : Une intrusion désigne une tentative d'accès non autorisée ou une activité malveillante visant à compromettre la sécurité d'un réseau ou d'un système informatique.
* **Adresse IP** : Une adresse IP (Internet Protocol) est une étiquette numérique attribuée à chaque appareil connecté à un réseau informatique. Elle permet l'identification et la communication entre les appareils sur le réseau.
* **Port** : Un port est un point d'accès numéroté sur un ordinateur qui est utilisé pour la communication entre les applications. Les ports permettent de différencier les différentes applications et services sur un appareil.
* **Protocole** : Un protocole est un ensemble de règles et de normes qui régissent la communication entre les appareils sur un réseau. Certains des protocoles couramment utilisés sont TCP (Transmission Control Protocol) et UDP (User Datagram Protocol).
* **Paquet** : Un paquet est une unité de données qui est transmise sur un réseau. Il contient des informations telles que l'adresse source, l'adresse de destination et les données elles-mêmes.
* **Flux de données** : Un flux de données est une séquence de paquets qui sont associés à une connexion réseau spécifique. L'analyse des flux de données peut fournir des informations sur les activités et les comportements sur le réseau.
* **Attaquant** : Un attaquant est une personne ou une entité qui tente de compromettre la sécurité d'un réseau ou d'un système informatique en menant des activités malveillantes.
* **Utilisateur légitime** : Un utilisateur légitime est un individu ou un système autorisé à accéder au réseau ou au système informatique de manière légale.
* **Administrateur système** : Un administrateur système est responsable de la gestion, de la configuration et de la sécurité d'un réseau ou d'un système informatique. Il joue un rôle crucial dans la détection et la prévention des intrusions.

## Techniques et méthodes de détection utilisées

Il existe plusieurs techniques et méthodes utilisées pour détecter les intrusions réseau. Voici quelques-unes des approches couramment utilisées :

* **Détection basée sur les signatures** : Cette méthode repose sur la comparaison des schémas d'attaques connus avec les données du réseau. Des signatures préétablies sont utilisées pour identifier les modèles spécifiques associés à des attaques connues.
* **Détection basée sur les anomalies** : Cette approche vise à détecter des comportements anormaux qui ne correspondent pas aux modèles normaux du réseau. Elle utilise des techniques d'apprentissage automatique pour analyser les données et identifier les activités inhabituelles.
* **Analyse des journaux de connexion** : Les journaux de connexion contiennent des enregistrements des activités sur le réseau. L'analyse de ces journaux peut révéler des schémas d'attaques ou des comportements suspects
* **Analyse des flux de données** L'analyse des flux de données réseau peut fournir des informations précieuses pour détecter et prévenir les activités malveillantes dans les réseaux. Cependant, il est important de noter que cette approche nécessite une compréhension approfondie du réseau, des outils appropriés pour l'extraction des caractéristiques et des techniques d'analyse adaptées aux données spécifiques du réseau.

# Études de l'existant

## Les travaux de recherche existants dans le domaine

Il existe de nombreux travaux de recherche sur l'application des réseaux neuronaux à la détection d'intrusions réseau en utilisant TensorFlow et Keras. Les réseaux neuronaux sont devenus populaires dans ce domaine en raison de leur capacité à détecter des schémas complexes et à apprendre à partir de grandes quantités de données.

Voici quelques travaux de recherche pertinents dans ce domaine :

"Deep Learning-Based Intrusion Detection System for IoT Networks" (2018) : Cette recherche utilise un réseau de neurones convolutionnel (CNN) pour détecter les intrusions dans les réseaux Internet des objets (IoT). Ils ont utilisé TensorFlow et Keras pour entraîner et tester leur modèle.

"A Deep Learning Approach for Network Intrusion Detection System" (2019) : Les auteurs de cette recherche ont proposé un modèle de détection d'intrusion basé sur un réseau de neurones récurrent (RNN) utilisant des cellules LSTM. Ils ont utilisé TensorFlow avec Keras pour la mise en œuvre et l'entraînement du modèle.

"An Intrusion Detection System for Network Traffic Using Deep Learning" (2020) : Cette étude propose une méthode de détection d'intrusion basée sur un modèle hybride qui combine des couches convolutionnels et récurrentes. Ils ont utilisé TensorFlow et Keras pour entraîner leur modèle à détecter les anomalies dans le trafic réseau.

"A Hybrid Approach Using Deep Learning and Rule-Based Systems for Network Intrusion Detection" (2021) : Cette recherche combine les techniques de Deep Learning avec des systèmes basés sur des règles pour améliorer la détection d'intrusions. Ils ont utilisé TensorFlow avec Keras pour implémenter leur modèle.

Ces travaux de recherche illustrent différentes approches utilisant TensorFlow et Keras pour la détection d'intrusions réseau. Il convient de noter que le domaine de la détection d'intrusions évolue rapidement, de nouveaux travaux de recherche sont publiés régulièrement, et il peut être utile de consulter les dernières publications pour obtenir les informations les plus récentes dans ce domaine.

## Avantages, des limites et des résultats obtenus dans les études précédentes

Les travaux de recherche sur l'application des réseaux neuronaux à la détection d'intrusions réseau avec TensorFlow et Keras présentent plusieurs avantages, des limites et des résultats obtenus. Voici un aperçu de ces aspects :

Avantages :

Détection de schémas complexes : Les réseaux neuronaux peuvent capturer et apprendre des schémas complexes dans les données réseau, ce qui peut améliorer la détection des intrusions par rapport aux méthodes traditionnelles basées sur des règles.

Adaptabilité : Les réseaux neuronaux peuvent s'adapter à des environnements réseau dynamiques et apprendre à partir de nouvelles données, ce qui permet une détection plus précise des intrusions.

Traitement des données non structurées : Les réseaux neuronaux peuvent traiter des données réseau non structurées, telles que les paquets de réseau bruts, sans nécessiter une extraction manuelle de fonctionnalités.

Amélioration de la précision : Les études antérieures ont montré que l'utilisation de réseaux neuronaux avec TensorFlow et Keras peut conduire à une meilleure précision dans la détection d'intrusions par rapport à d'autres méthodes.

Limites :

Besoin de données étiquetées : L'entraînement de réseaux neuronaux nécessite des données étiquetées, c'est-à-dire des exemples d'intrusions et de trafic normal annotés. L'obtention de ces données étiquetées peut être coûteuse et chronophage.

Besoin de ressources de calcul importantes : Les réseaux neuronaux sont des modèles complexes qui nécessitent des ressources de calcul significatives, en particulier lorsqu'ils sont entraînés sur de grandes quantités de données. Cela peut nécessiter des infrastructures informatiques puissantes.

Sensibilité aux attaques adversaires : Les réseaux neuronaux peuvent être vulnérables aux attaques adversaires, où des intrus cherchent à manipuler les données ou les modèles pour tromper le système de détection.

Interprétabilité limitée : Les réseaux neuronaux sont souvent considérés comme des "boîtes noires", ce qui signifie qu'il peut être difficile d'expliquer comment et pourquoi ils prennent leurs décisions, ce qui peut être un défi en termes de confiance.

Résultats obtenus :

Les études précédentes ont montré des résultats prometteurs dans l'application des réseaux neuronaux à la détection d'intrusions réseau avec TensorFlow et Keras. Ces résultats incluent une amélioration de la précision de détection par rapport aux méthodes traditionnelles, une réduction des faux positifs, une capacité à détecter des attaques complexes et une adaptabilité à des environnements réseau changeants. Cependant, il convient de noter que les résultats peuvent varier en fonction des données, des architectures de réseau, des méthodes d'apprentissage et des paramètres utilisés. Il est important de considérer ces facteurs lors de l'application de ces techniques à des scénarios spécifiques de détection d'intrusions. la partie technique des conteneurs.

# [Base de données](#_bookmark14)

## Données utilisées pour l'entraînement et l'évaluation du modèle

Les données utilisées pour l'entraînement et l'évaluation des modèles est un dataset (KDDTrain+2) qui dérive du jeu de données KDD Cup 1999 : créé dans le but de développer des systèmes de détection d'intrusions efficaces. Le jeu de données KDDTrain+2 est un ensemble de données étiquetées qui contient des données de trafic réseau avec différents types d'attaques et d'activités normales.

* **Origine** : Le jeu de données KDDTrain+\_2 est un sous-ensemble du jeu de données original KDD Cup 1999, créé à l'aide d'une simulation d'un environnement réseau militaire.
* **Caractéristiques** : Le jeu de données se compose d'un large éventail de caractéristiques extraites du trafic réseau, notamment le type de protocole, le service, les adresses IP source et destination, les indicateurs, les compteurs de paquets, etc.
* **Étiquettes** : Chaque enregistrement du jeu de données est étiqueté comme étant soit un type spécifique d'attaque, soit une activité normale. Les attaques sont catégorisées en différentes classes, telles que les attaques par déni de service (DoS), les sondes (Probe), les accès non autorisés à distance (R2L), les élévations de privilèges non autorisées (U2R) et les activités normales.
* **Déséquilibre** : Le jeu de données est connu pour être déséquilibré, avec un nombre nettement plus élevé d'instances normales par rapport aux instances avec des attaques. Cela peut affecter l'entraînement et l'évaluation des modèles de détection d'intrusions.
* **Utilisation** : Le jeu de données KDDTrain+\_2 est couramment utilisé dans la recherche et le développement de systèmes de détection d'intrusions. Il sert de jeu de données de référence pour évaluer les performances de différentes algorithmes et techniques de détection d'intrusions réseau.

## Des critères de sélection du jeu de données

Lors de la sélection d'un jeu de données pour l'application des réseaux neuronaux à la détection d'intrusions réseau avec TensorFlow et Keras, certains critères importants peuvent être pris en compte. Voici quelques critères de sélection courants :

* Représentativité : Le jeu de données doit être représentatif des différents types d'intrusions et du trafic réseau normal que l'on souhaite détecter. Il devrait inclure une variété d'attaques et de scénarios réalistes pour garantir que le modèle est capable de généraliser à des situations réelles.
* Volume de données : Le jeu de données doit être suffisamment volumineux pour permettre l'apprentissage efficace d'un modèle de réseau neuronal. Des ensembles de données plus importants peuvent aider à améliorer la précision et la généralisation du modèle.
* Qualité des données : Les données du jeu de données doivent être fiables, précises et exemptes de bruit excessif. Les erreurs ou les incohérences dans les données peuvent affecter la capacité du modèle à apprendre correctement et à prendre des décisions précises.
* Étiquetage des données : Les exemples d'intrusions et de trafic normal doivent être correctement étiquetés pour permettre à l'algorithme d'apprentissage de distinguer entre eux. Un étiquetage précis est crucial pour l'entraînement et l'évaluation du modèle.
* Variabilité des données : Il est préférable d'avoir une variabilité suffisante dans le jeu de données, ce qui signifie que les exemples d'intrusions et de trafic normal devraient couvrir un large éventail de scénarios et de caractéristiques. Cela permettra au modèle d'apprendre à détecter des motifs variés et d'être plus robuste face à de nouvelles situations.
* Confidentialité et conformité : Il est important de prendre en compte les aspects de confidentialité et de conformité lors de la sélection d'un jeu de données. Assurez-vous que les données utilisées sont conformes aux réglementations en matière de confidentialité et qu'elles ont été anonymisées si nécessaire.

Il est également recommandé de consulter la littérature existante et les jeux de données couramment utilisés dans la communauté de recherche sur la détection d'intrusions réseau. Ces jeux de données ont souvent été préparés et validés par des experts du domaine, ce qui facilite leur utilisation et leur comparaison avec d'autres travaux de recherche.

# Implémentation

## Prétraitement des données

Le prétraitement des données est une étape importante dans l'application des réseaux neuronaux à la détection d'intrusions réseau avec TensorFlow et Keras. Voici quelques étapes courantes de prétraitement des données dans ce contexte :

* **Importation des librairies** : Pour implémenter notre modèle de détection d'intrusions réseau, nous avons importé les librairies nécessaires pour le traitement des données et la construction du modèle. Les librairies utilisées comprennent TensorFlow, Keras et les librairies courantes de manipulation de données en Python telles que Pandas et NumPy.
* **Chargement des données :** Nous avons procédé au chargement des données de notre jeu de données KDDTrain+\_2. Les données ont été importées à partir du fichier CSV contenant les caractéristiques et les étiquettes des intrusions réseau.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

* **Analyse des données** : Avant de procéder à l'implémentation du modèle, nous avons réalisé une analyse exploratoire des données. Cela comprenait l'examen des statistiques descriptives des caractéristiques, la visualisation de la distribution des étiquettes d'intrusions et l'identification de tout déséquilibre de classe potentiel.

**Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement**

**Une image contenant capture d’écran, texte, ligne, logiciel

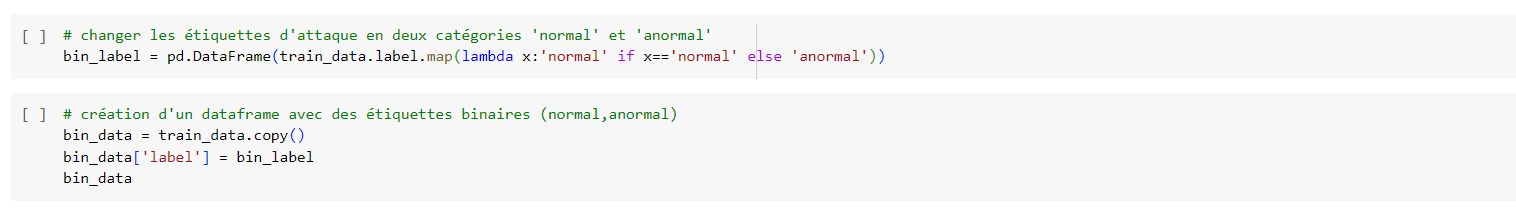
Description générée automatiquement**

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

* **Préparation des données (encodage one-hot)** : Les données ont été préparées en convertissant les variables catégorielles en variables binaires à l'aide de l'encodage one-hot. Cela a permis de représenter les caractéristiques catégorielles de manière appropriée pour l'entraînement du modèle.

Sélection de feature, nous le faisons parce que nous préférons former des modèles avec des fonctionnalités significatives et obtenir un modèle fiable pour classer l'ensemble de test



Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

* **Normalisation des données** : Avant de fournir les données au modèle, nous avons effectué une étape de normalisation des données pour mettre toutes les caractéristiques à la même échelle. Cela a été réalisé en utilisant la normalisation z-score, qui a transformé les valeurs des caractéristiques en les centrant autour de zéro avec une variance unitaire.

Une image contenant texte, Police, logiciel, capture d’écran

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

## Conception du modèle et Entrainement du modèle sur le jeu de données KDDTrain+\_2

Dans cette section, nous présenterons l'implémentation des différents modèles de classification que nous avons choisis pour notre étude. Nous explorerons à la fois la classification binaire et la classification multiple.

* 1. Classification Binaire
* **KNN (K plus proches voisins)**

Le modèle KNN est une méthode de classification basée sur les voisins les plus proches. Nous avons utilisé la classe KNeighborsClassifier du module sklearn.neighbors pour implémenter ce modèle.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

* **LSTM (Long Short-Term Memory)**

Le modèle LSTM est un type de réseau de neurones récurrents (RNN) utilisé pour les séquences de données. Nous avons utilisé les classes nécessaires du module tensorflow.keras.layers pour construire le modèle LSTM.

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

* **ANN (Artificial Neural Network)**

Le modèle ANN est un réseau de neurones artificiels composé de plusieurs couches cachées. Nous avons utilisé les classes nécessaires du module tensorflow.keras.layers pour construire le modèle ANN.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

* 1. Classification Multiple
* **KNN (K plus proches voisins)**

Le modèle KNN est également applicable à la classification multiple. Nous avons utilisé la classe KNeighborsClassifier du module sklearn.neighbors pour implémenter ce modèle.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

* **MLPC (Multilayer Perceptron Classifier)**

Le modèle MLPC est un classificateur de perceptron multicouche, également connu sous le nom de réseau de neurones artificiels à couches multiples. Nous avons utilisé la classe MLPClassifier du module sklearn.neural\_network pour implémenter ce modèle.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, ligne, affichage

Description générée automatiquement

* **ANN (Artificial Neural Network)**

Le modèle ANN peut également être utilisé pour la classification multiple. Nous avons utilisé les classes nécessaires du module tensorflow.keras.layers pour construire le modèle ANN.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

# Conclusion

Les applications des réseaux neuronaux à la détection d'intrusions réseau avec TensorFlow et Keras offrent de nombreux avantages et opportunités. Ces outils et Framework de Deep Learning permettent de développer des modèles de détection d'intrusions puissants et sophistiqués. Voici quelques points clés à retenir :

1. Avantages :

- Les réseaux neuronaux permettent d'extraire des caractéristiques complexes et non linéaires à partir des données, ce qui peut améliorer la précision de la détection d'intrusions.

- TensorFlow et Keras offrent des fonctionnalités avancées pour la conception, l'entraînement et l'évaluation des modèles de détection d'intrusions.

- Les modèles basés sur les réseaux neuronaux peuvent s'adapter à des données de différentes natures, telles que les flux de données réseau, les journaux d'événements, etc.

- Les réseaux neuronaux peuvent détecter des schémas d'attaques complexes et émergentes, ce qui les rend utiles pour la détection d'intrusions en temps réel.

2. Limites :

- Les modèles basés sur les réseaux neuronaux peuvent nécessiter des quantités importantes de données d'entraînement pour obtenir des performances optimales, ce qui peut être un défi dans le domaine de la détection d'intrusions où les données d'attaques sont souvent rares.

- L'interprétabilité des modèles basés sur les réseaux neuronaux peut être limitée, ce qui peut rendre difficile la compréhension des motifs de détection et l'explication des décisions prises par le modèle.

- L'entraînement et le réglage des modèles de détection d'intrusions basés sur les réseaux neuronaux peuvent être complexes et nécessiter une expertise en Deep Learning.

3. Résultats obtenus :

- Les recherches et les études précédentes ont montré que l'utilisation de réseaux neuronaux avec TensorFlow et Keras dans la détection d'intrusions réseau peut conduire à de bons résultats, avec des taux de détection élevés et de fausses alarmes réduites.

- Les modèles basés sur les réseaux neuronaux peuvent être compétitifs voire surpasser les approches traditionnelles de détection d'intrusions, en particulier pour la détection d'attaques complexes et en constante évolution.

En conclusion, l'utilisation de réseaux neuronaux avec TensorFlow et Keras offre des opportunités prometteuses pour la détection d'intrusions réseau. Cependant, il est important de prendre en compte les défis associés à l'entraînement, à l'interprétabilité et à l'adaptation des modèles aux spécificités des données et aux besoins de sécurité. Les recherches futures dans ce domaine peuvent contribuer à améliorer encore davantage les performances des modèles de détection d'intrusions basés sur les réseaux neuronaux.

Références :

-Dataset : Kaggle (kdd cup 1999)

-Traitements de données : Tan, P. N., Steinbach, M., & Kumar, V. (2019). Introduction to Data Mining. Pearson.

-Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

-Data Preparation: Han, J., & Kamber, M. (2006). Data Mining: Concepts and Techniques. Morgan Kaufmann.

Brownlee, J. (2020). One Hot Encoding for Machine Learning: Techniques for Handling Categorical Data. Machine Learning Mastery.

-Data Standardization: Hastie, T., Tibshirani, R., & Friedman, J. (2009). \* The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

\* Scikit-learn documentation. Preprocessing data. Available at: <https://scikit-learn.org/stable/modules/preprocessing.html>

-Modélisation:

\* Chollet, F. (2018). Deep Learning with Python. Manning Publications.

\* Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.