REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

Université «ABBES LAGHROUR» KHENCHELA

Faculté des sciences et de la technologie

Département de Mathématique et Informatique

Mémoire

Présentée pour l'obtention du diplôme de master en informatique

Spécialité Sécurité et Technologie Web

Thème

Détection d'intrusion sur les objets connectes

Réalisé par :

NEMER Mourad

Encadré par :

Dr. ZIANOU Ahmed Seghir

SESSION : Juin 2022


بسم الله الرحمن الرحيم

Remerciements

Tout d'abord je tiens à remercier Allah, le clément et le miséricordieux de m'avoir donné la patience et la force de mener bien ce modeste travail.

Je voudrai exprimer mes vifs remerciements à mon encadreur Dr : ZIANOU Ahmed Seghir pour ses aides, ses encouragements et ses conseils judicieux durant toute la période du travail.

Je remercie chaleureusement tous les enseignants du département math et informatique à université de Khenchela.

Je remercie aussi toutes personnes ayant contribué de près ou de loin à l'aboutissement de ce travail.

Dédicace

Je dédie ce travail

A mon cher père

A ma chère mère

A ma chère tante

A l'âme pure de mon oncle

Qui non jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A mes frères et mes sœurs qui leur souhaitent une bonne santé.

A ma femme qui n'a pas manqué de me mettre dans les meilleurs conditions, de me soutenir pour que je puisse accomplir ce travail.

A mon ange ADEM que dieu protège et prolonge sa vie.

A tous mes amis et surtout qui m'ont soutenu

KHIARI Abdeljalil et BECHIRI Houssem.

A toute ma promotion 2022.

NEMER MOURAD

Résumé

L'ouverture d'Internet a créé la capacité de connecter des appareils, des applications et des services à grande échelle, ce qui change la façon dont nous interagissons avec notre environnement et notre société. L'IoT a un énorme potentiel pour rendre notre monde meilleur, et en même temps des milliards d'appareils, d'applications et de services IoT sont déjà utilisés et de plus en plus en ligne, la sécurité de l'IoT devient de plus en plus importante. Les appareils et services IoT peuvent servir de points d'entrée pour les cyberattaques. À mesure que la quantité de données augmente, les méthodes traditionnelles de détection d'intrusion deviennent inadéquates. Dans ce mémoire , nous proposerons un système de détection d'intrusion IoT basé sur l'apprentissage automatique pour obtenir une sécurité robuste contre les attaques et les intrusions inconnues.

Abstract

The open nature of the Internet has created the ability to connect devices, applications and services on a scale that is transforming the way we interact with our environment and society.

The Internet of Things has enormous potential to change our world for the better, at the same time, with billions of IoT devices, applications and services already in use, and more and more coming online, the IoT security is becoming more important. Poorly secured IoT devices and services can serve as entry points for cyber-attacks.

With the large volume of data, traditional intrusion detection methods become insufficient.

In this thesis, we will propose an intrusion detection system for the Internet of Things based on machine learning, in order to achieve strong security against attacks and unknown intrusions.

ملخص

أعطت الإنترنات المفتوحة القدرة على توصيل الأجهزة والتطبيقات والخدمات على نطاق واسع، فتغير الطريقة التي نتفاعل ونتعامل بها مع البيئة والمجتمع الذي نعيش فيه.

تتمتع إنترنت الأشياء بإمكانيات هائلة لتغيير عالمنا للأفضل، وتواصل مليارات الأجهزة والتطبيقات وخدمات إنترنت الأشياء مع بعضها البعض في نفس الوقت.

أصبح أمان وحماية إنترنت الأشياء جد مهم. لأن أجهزة وخدمات إنترنت الأشياء يمكن أن تعمل على دخول الهجمات الإليكترونية مع الحجم الكبير للبيانات، صارت الطرق التقليدية للكشف عن تسلل الأجهزة غير مجدية.

ضمن هذه الأطروحة، نقترح نظام لاكتشاف تسلل إنترنت الأشياء يعتمد على التعلم الآلي، من أجل تحقيق أمان قوي ضد الهجمات والاقتحامات غير المعروفة.

Table des matières

Introduction générale **11**

Chapitre 01 : Internet des objets **13**

1.1 Introduction **13**

1.2 Internet des objets : définitions **13**

1.3 Domaines d'utilisation (exemples d'application) **14**

1.3.1 L'IoD dans le domaine de la santé **14**

1.3.2 La révolution numérique en réponse aux impératifs énergétiques **15**

1.3.3 La domotique ou maison connectée **15**

1.3.4 L'industrie connectée **15**

1.3.5 L'Internet des objets dans l'agriculture **16**

1.3.6 Les objets connectés dans les élevages **16**

1.3.7 Smart retail : des supermarchés branchés **16**

1.4 Architecture de l'internet des objets **17**

1.5 Conclusion **18**

Chapitre 2 : Système de détection des intrusions **20**

2.1 Définition **20**

2.2 Les composants d'un système de détection d'intrusion **20**

2.2.1 La surveillance du système d'information **20**

2.2.2 Le scannage des sources d'informations **21**

2.2.3 Les décisions prises lors de l'évaluation positive d'une intrusion **21**

2.3 Caractéristiques souhaitées d'un IDS **21**

2.4 Principes de détection **22**

2.4.1 Approche par scénario **22**

2.4.2 Approche comportementale **23**

2.4.3 Systèmes de détection d'intrusion par signatures **24**

2.4.4 Systèmes de détection d'intrusion par anomalies **24**

2.4.5 Hybride **25**

2.4.6 Temporalité de détection **25**

2.4.7 Corrélation des alertes **25**

2.5 Étude et choix d'une solution de détection d'intrusion **26**
2.6 Les défis liés à la détection d'intrusions **27**
2.7 Conclusion **29**
Chapitre 03 : Machine learning (apprentissage automatique) **31**
3.1 Introduction **31**
3.2 Aperçu sur l'apprentissage automatique **31**
3.3 Définition du machine learning **31**
3.4 Ingrédients de le machine learning **32**
3.5 Types d'apprentissage **34**
3.5.1 Apprentissage supervisé **34**
3.5.1.1 Les arbres de décision **34**
3.5.2 Apprentissage non supervisé **36**
3.5.3 Apprentissage semi-supervisé **36**
3.5.4 Apprentissage par renforcement **36**
3.6 Quelques exemples d'application **37**
3.7 Conclusion **37**
Chapitre 4 : Systèmes de détection d'intrusion de l'IoT basés sur le machine learning . **39**
4.1 Obtention des données **39**
4.2 Supervisé ou non supervisé **39**
4.3 Hypothèse **39**
4.4 Implémentation **40**
4.5 Problématiques des modèles **42**
4.6 Optimisation **42**
4.7 Conclusion **42**
Chapitre 5 : Conception et réalisation **44**
5.1 Introduction **44**
5.2 Environnement d'exécution **44**
5.2.1 L'éditeur choisi **44**
5.2.2 Le langage de programmation utilisé **45**
5.3 Dataset **45**
5.4 Algorithme d'apprentissage pour la détection d'intrusion **46**

5.4.1 La classification naïve bayésienne (Gaussian Naive Bayes) **46**
5.4.2 Arbre de décision (Decision Tree) **47**
5.4.3 Forêt d'arbres décisionnels (RANDOM FOREST) **47**
5.4.4 Machine à vecteurs de support (SUPPORT VECTOR MACHINE) **48**
5.4.5 Régression logistique (LOGISTIC REGRESSION) **48**
5.5 Résultat et Discussion **49**
5.5.1 Les mesures d'évaluation des modèles **49**
5.5.1.1 La précision **50**
5.5.1.2 Le taux de détection (Rappel) **50**
5.5.1.3 Le taux de faux positif (FP) **50**
5.5.1.4 Le taux de réussite (Accuracy) **50**
5.5.1.5 Micro-moyenne (Micro-averaged) **50**
5.5.1.6 Macro-moyenne (Macro-averaged) **51**
5.5.1.7 Moyenne pondérée (Weighted-averaged) **51**
5.5.1.8 F1-score: **51**
5.5.2 Discussion et analyse **51**
5.6 Conclusion **55**
Conclusion générale **57**

Table des figures

Figure 1.1 : Architecture de l'IoD **17**

Figure 2.1 : Placement d'un NIDS en amont d'un pare-feu **20**

Figure 2.2 : Placement en aval **20**

Figure 2.3 : Approche par scénario **23**

Figure 2.4 : Approche compos te mentale **24**

Figure 2.5 : Positionnement des sondes de détection d'intrusion dans une architecture réseau simplifié **26**

Figure 2.6 : Le processus de gestion de sécurité **27**

Figure 3.1 : Exemple d'arbre décision **35**

Figure 3.2 : Exemple de phase d'apprentissage **35**

Figure 3.3 : Exemple de phase de classification **36**

Figure 5.1: PyCharm 2022.1.1 ..... **45**

Figure 5.2 : Algorithme classification naïve bayésienne **46**

Figure 5.3 : Algorithme Arbre de décision **47**

Figure 5.4 : Algorithme Forêt d'arbres décisionnels **47**

Figure 5.5 : Algorithme Machine à vecteurs de support **48**

Figure 5.6 : Algorithme Régression logistique **48**

Figure 5.7 : Matrice de confusion **49**

Figure 5.8 : Matrice de confusion: (a) classification naïve bayésienne (b) Arbre de décision(c) Forêt d'arbres décisionnels (d) Machine à vecteurs de support (d) Régression logistique **52**

Figure 5.9 : Le taux de réussite (Accuracy) de (a) la phase de teste (b) la phase d'apprentissage **54**

Figure 5.10 : Le temps d'exécution de (a) la phase de teste (b) la phase d'apprentissage **55**

INTRODUCTION GENERALE

Introduction générale :

Les réseaux et les systèmes informatiques sont devenus des outils essentiels au fonctionnement des entreprises. Ils sont aujourd'hui déployés dans tous les domaines professionnels : universitaires, bancaires, assurantiels ou encore militaires.

Les IT que ces systèmes gèrent sont des objets gourmands. Il peut être attaqué en exploitant des éléments vulnérables des systèmes d'information. Détecter les comportements malveillants est rapidement devenu une nécessité.

Les mesures préventives se sont avérées insuffisantes et ont conduit à la création d'un Système de Détection d'Intrusion (IDS : Intrusion Detection System).

Une intrusion est définie comme toute tentative de compromettre l'intégrité, la confidentialité ou la disponibilité d'un réseau, et toute tentative de contourner les dispositifs de sécurité sur un réseau ou une machine. Ces tentatives d'intrusion peuvent être bénignes ou extrêmement dangereuses et préjudiciables à l'entreprise.

Le domaine de la détection d'intrusion est encore jeune, mais en pleine phase de développement, on compte actuellement une centaine de systèmes de détection d'intrusion (ou IDS pour Intrusion Detection System), qu'il s'agisse de produits commerciaux ou du domaine public, ces systèmes de surveillance de réseau évoluent en raison d'un développement doit être indispensable. Le nombre et le danger des cyberattaques ont augmenté ces dernières années.[1]

Dans ce travail, je suis intéressé aux différents systèmes utilisés dans la détection d'intrusion, généralement des systèmes de data mining ou d'apprentissage automatique et des techniques statiques.

Nos profils sont organisés comme suit : Le premier chapitre est consacré à l'Internet des objets. Le deuxième chapitre se concentre sur le système de détection d'intrusion. Le troisième chapitre est basé sur l'apprentissage automatique. Le chapitre 4 aborde différents systèmes de détection d'intrusion IoT basés sur l'apprentissage automatique. Je termine mes recherches par la conception et la mise en œuvre de l'application.

CHAPITRE 01

Internet Des Objets

Chapitre 01 : Internet des objets

1.1 Introduction

Ces dernières années, un nouveau paradigme appelé l'Internet des objets a rapidement attiré l'attention. L'Internet des objets fait référence à "un réseau mondial d'objets interconnectés adressables de manière unique basés sur des protocoles de communication standard" avec un accent sur Internet. L'IoT est basé sur la présence omniprésente d'objets autour des personnes capables de mesurer, déduire, comprendre et même modifier leur environnement. Il est basé sur des nœuds (objets) intelligents et interconnectés dans une infrastructure de réseau dynamique et globale. Il se caractérise généralement par de petits objets dans le monde réel, largement distribués et une puissance de stockage et de traitement limitée, ce qui implique des problèmes tels que la fiabilité, les performances, la sécurité et la confidentialité. .Il est alimenté par les dernières avancées de divers appareils et technologies de communication. Il s'agit non seulement d'appareils complexes tels que les téléphones portables, mais aussi Il existe également des objets simples pour un usage quotidien, tels que des montres, des thermostats, des vêtements, etc.

Il ne fait aucun doute que la principale conséquence de l'IoT est son impact sur la vie quotidienne des utilisateurs potentiels. L'IoT a un impact significatif à la maison comme au travail, et il jouera un rôle décisif dans un futur proche (santé, transports intelligents, domotique, aide à la vie, etc.). Les entreprises (transport de marchandises, sécurité, logistique, automatisation industrielle, etc.) devraient également en tirer des avantages significatifs. Sur la base de ces considérations, le National Intelligence Council des États-Unis a déclaré l'Internet des objets l'une des six technologies susceptibles d'affecter les intérêts américains.

1.2 Internet des objets : définitions

Il existe également des objets simples pour un usage quotidien, tels que des montres, des thermostats, des vêtements, etc.

En 2011, il y avait plus d'appareils connectés que d'habitants sur la planète. Le nombre d'appareils connectés était estimé à 30 milliards en 2018 et atteindra 50 milliards en 2020, soit 6,58 par habitant. Ces chiffres suggèrent que l'IoT deviendra l'une des principales sources de données volumétriques. [1]

Avant de définir le concept d'Internet des Objets, il est important de définir un objet connecté, qui est un appareil dont la vocation première n'est pas un système informatique ou une interface d'accès au réseau, tel qu'un objet tel qu'une machine à café ou une serrure qui n'est pas conçu avec un système informatique intégré ou connecté à Internet.

- **Définiion1** : L'Internet des objets (IoT) est défini comme un réseau mondial de services interconnectés et de divers objets intelligents conçus pour soutenir les activités humaines quotidiennes grâce à ses capacités de détection, de calcul et de communication. Leur capacité à observer le monde physique et à éclairer les décisions fera partie intégrante de la future architecture Internet.

- **Définition 2 :** Est l'extension d'Internet aux objets et aux lieux du monde physique. Il représente l'échange d'informations et de données depuis des appareils du monde réel vers Internet.
- **Définition 3** : Est un réseau qui permet, grâce à des systèmes d'identification électronique standardisés et sans fil, d'identifier et de communiquer numériquement avec des objets physiques afin de pouvoir mesurer et échanger des données entre les mondes physique et virtuel.

Toutes ces définitions se traduisent par l'Internet des objets, communément appelé en anglais Internet des objets (IoT), spécifiant une technologie de pointe dans laquelle des objets traditionnellement non connectés autour de nous tels que des lumières, des machines, des vêtements, etc…), à la fois physiques et virtuels, ont désormais la capacité de communiquer entre eux en temps réel. Ce réseau d'objets permet de partager ses données via une plateforme cloud sans intervention humaine. Grâce à l'optimisation de l'interaction entre l'homme et la machine et la multiplication des flux de données, les objets connectés offrent la possibilité de définir les besoins précis de l'individu, lui offrant ainsi un bien ou un service unique. [2]

1.3 Domaines d'utilisation (exemples d'application).

On en entendait à peine parler il y a quelques années, et maintenant ils sont partout. Les objets connectés ont envahi notre quotidien sans même qu'on s'en aperçoive.

Des téléviseurs intelligents aux voitures connectées, ces nouveaux outils ont grandement amélioré notre niveau de confort et facilité nos loisirs et nos déplacements.

Le potentiel de connexion des objets est énorme. Une étude Gartner de 2016 a prédit que d'ici 2020, plus de la moitié des outils et processus commerciaux utiliseront l'IoT. Les applications sont diverses et couvrent de nombreux domaines : industrie, science, santé, etc. Les Jeudis vous propose dix applications IoT qui changent le paysage social. [3]

1.3.1 L'IoD dans le domaine de la santé

Appareils de radiographie et d'imagerie, moniteurs connectés, compteurs d'énergie… 60% des hôpitaux dans le monde utilisent déjà l'IoT pour augmenter leur productivité et améliorer la prise en charge des patients. Les recherches d'Arubanetworks montrent que d'ici 2019, près de 90 % des prestataires de soins de santé auront des objets connectés intégrés dans leurs dispositifs médicaux.

L'objet connecté est utilisé au quotidien pour :

- Surveillance et maintenance dans les établissements de santé
- Opération chirurgicale et télécommande
- Services de géolocalisation

La standardisation de l'IoT dans le domaine de la santé permettra la création de nouveaux modèles opérationnels qui augmenteront la productivité des employés tout en facilitant la collaboration entre soignants et la communication avec les patients.

1.3.2 La révolution numérique en réponse aux impératifs énergétiques

L'intelligence artificielle est une véritable valeur ajoutée dans le secteur de l'énergie et peut représenter un investissement écologique décisif pour l'avenir de notre société dans les années à venir. L'enjeu est aussi économique, et les entreprises l'ont bien compris. L'IoT répond aux principales problématiques énergétiques :

- Épuisement des ressources naturelles.
- Croissance de la demande énergétique mondiale.
- Prix du marché instables.
- Manque de main d'oeuvre.

La révolution numérique entre dans le débat énergétique à travers la gestion des ressources : compteurs électriques, réseaux intel

1.3.3 La domotique ou maison connectée

Aussi connu sous le nom de domotique, les maisons intelligentes se normalisent. Une étude de Juniper Research prédit que d'ici fin 2021, le nombre d'objets sur l'intranet domestique augmentera de 200 %.

Outre les objets de divertissement tels que les smart TV ou les enceintes connectées, la domotique prend également en compte la sécurité et les économies d'énergie à l'intérieur de la maison :

- Cellule familiale : Contrôler et programmer différentes interventions au sein de la famille.
- Capteurs d'information (systèmes d'alarme, changements de température, etc.)
- Actionneurs qui permettent la programmation et le contrôle de divers appareils électroniques dans la maison, même à distance.

1.3.4 L'industrie connectée

L'industrie n'est pas passée inaperçue dans l'utilisation de l'IoT et des avantages qu'elle apporte. Dans le cadre des problématiques rencontrées dans le secteur industriel, l'utilisation des objets de connexion est très spécifique et répond aux besoins :

- Optimisation (supply chain).
- Transformation des processus métiers.
- Augmentation de l'efficacité et de la productivité.
- Traçabilité et sécurité.

La révolution numérique offre également des opportunités à certains types d'industries de se renouveler et d'apporter de la valeur ajoutée à des domaines en perte de popularité. C'est le cas de SNCF Fret, par exemple, qui regagne doucement en crédibilité avec le lancement de locomotives connectées permettant de mieux suivre ses wagons et d'apporter une sécurité accrue aux clients.

1.3.5 L'Internet des objets dans l'agriculture

La croissance rapide de la population mondiale, les changements d'habitudes alimentaires, les perturbations climatiques sont trois grands facteurs, parmi d'autres, qui font de l'agriculture moderne un défi au quotidien.

D'ici 2050, la productivité agricole devra avoir augmenté de 70 % pour répondre à la demande mondiale. Plus qu'un défi technologique, il s'agit d'un enjeu humanitaire. Les céréaliers et maraîchers ont d'ores et déjà mis à profit les drones afin de de récolter en temps réel des informations essentielles à la gestion de l'exploitation :

- Humidité de la terre.
- État des plantations, Climat, etc.

Les données récoltées sont transférées aux tracteurs connectés (parfois autonomes). Cela permet de doser finement le niveau d'engrais et d'arrosage sur telle ou telle parcelle et de réduire les coûts, tant financiers qu'énergétiques.

1.3.6 Les objets connectés dans les élevages

Traceurs GPS pour le bétail, recueillement des habitudes alimentaires des bovins, les objets connectés ne sont pas seulement utiles aux agriculteurs, mais également aux éleveurs qui peuvent surveiller plus finement l'état de santé de leurs bêtes.

Avez-vous déjà entendu parler des vaches connectées ? Fait amusant, il s'agit de l'animal le plus connecté au monde ! Son collier doté de nombreux capteurs permet une meilleure traçabilité mais aussi d'avoir des informations en temps réel sur son état de santé et son comportement.

1.3.7 Smart retail : des supermarchés branchés

Les entreprises de briques et de mortier subissent également des changements à l'ère numérique. Avec la concurrence féroce du e-commerce et du m-commerce, les magasins de détail cherchent à tirer parti de la popularité de l'IoT en combinant le e-commerce avec le commerce de détail traditionnel.

En conséquence, les magasins physiques se tournent également vers la révolution numérique, offrant de plus en plus de fonctionnalités amusantes et interactives pour améliorer l'expérience de vente et augmenter les conversions.

Dans le concept de "smart retail", on retrouve la technologie d'identification par radiofréquence (RFID), qui améliore l'expérience client en proposant un parcours client hyper-personnalisé. En complément des applications mobiles, des concepts de paniers connectés ont été pensés pour faciliter les achats en supermarché :

- Liste de courses complète.
- Cours guidés pour optimiser le temps d'exécution.
- Calculer automatiquement le nombre de paniers, …

Les commerçants investissent également dans des applications mobiles pour fidéliser et attirer les clients dans les magasins physiques. Par exemple, lorsque les clients traversent le magasin, les promotions/ventes en cours sont notifiées. [1]

1.4 Architecture de l'internet des objets

L'architecture d'un système IoT se compose de plusieurs couches qui communiquent entre elles, connectant le monde physique des objets au monde virtuel des réseaux et des clouds. Tous les projets n'adoptent pas une architecture formellement identique, mais les chemins des données peuvent être schématisés. Précisons les rôles des différents processus présentés dans ce schéma :

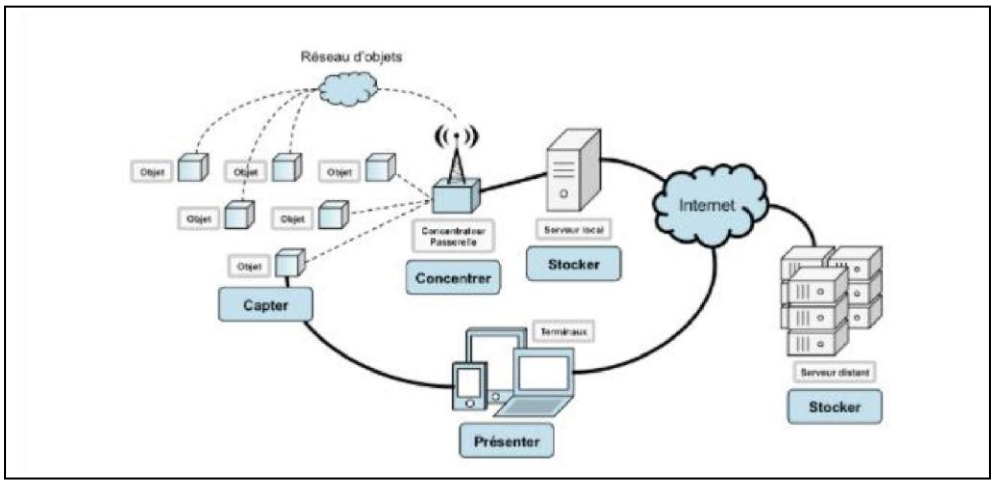


Figure 1.1 architicture de l'IoT

Précisons le rôle des différents processus présentés sur ce schéma :

- **Capter** fait référence à l'action de convertir des grandeurs physiques analogiques en signaux numériques.
- **Concentrer** peuvent connecter des réseaux d'objets spécialisés à des réseaux IP standard (par exemple WiFi) ou à des appareils grand public.
- **Stocker** limite le fait que les données brutes agrégées, générées en temps réel, méta-balisées, arrivent de manière imprévisible.

Enfin, la présentation démontre la capacité de présenter des informations d'une manière que les humains peuvent comprendre, tout en leur donnant un moyen d'agir et/ou d'interagir.

Les deux autres processus n'apparaissent pas sur la figure, arce qu'ils sont à la fois horizontaux et omniprésents :

- **Le traitement des données :** est un processus qui peut intervenir à tous les niveaux de la chaîne, de l'acquisition de l'information à la récupération de l'information. Lorsque l'on parle d'IoT, une stratégie connexe courante consiste à stocker les informations sous une forme monolithique. Nous collectons les "big data" de manière

exhaustive sans préjuger de la manière dont les données seront traitées. Cette stratégie est aujourd'hui possible grâce à une architecture distribuée de type NoSQL, capable de stocker de grandes quantités d'informations tout en offrant la possibilité d'y effectuer des traitements complexes (e.g. Map/Reduce).

- **La transmission des données :** est un processus qui se produit à tous les niveaux de la chaîne. Deux types de réseaux, supports de transmission, coexistent généralement :
  - **Réseau local de concentration :** Ensuite on utilise des technologies comme ANT, NFC ou Bluetooth.
  - **Réseau WAN :** Permet à un réseau dédié de s'interconnecter et de s'interfacer avec des fermes de serveurs. Utilisez ensuite le WiFi, les réseaux cellulaires (GSM, UMTS, LTE) ou encore les connexions physiques standards (Ethernet, Fibre). Ces réseaux sont généralement connectés à Internet.

La technologie de transport utilisée dépend largement de l'application et du contexte. Par exemple, le transport peut utiliser Push over Comet ou WebSocket. Si l'application autorise la rétroaction, le canal peut être bidirectionnel.

Dans certains cas, ces canaux doivent transmettre des données en temps réel, dans d'autres cas, le temps ne sera pas le facteur décisif. [3]

1.5 Conclusion

Nous pouvons affirmer que l'intégration de nombreux objets du monde réel sur Internet nécessitera la création de nouvelles interactions intuitives de haut niveau avec le monde physique et sera au cœur de l'Internet des objets. Du fait de leur complexité croissante, intégrer un maximum d'automatisation dans les architectures IoT ne peut que bénéficier à leur utilisation. Si des objets de natures différentes sont amenés à coopérer entre eux, nous devons faire face à des problèmes d'hétérogénéité, d'interopérabilité et de sécurité. Aujourd'hui, les applications conçues pour l'IoT sont trop monolithiques et trop adaptables à un contexte spécifique, ce qui entrave toute personnalisation et réutilisation.

Concernant la sécurité, nous notons que le paysage IoT ouvert, hétérogène et mobile est vulnérable. Il pose des risques de sécurité importants. Les frontières du système sont plus perméables à mesure que le système s'est étendu : Des objets intelligents aux passerelles vers le cloud. De plus, par exemple, les applications IoT peuvent générer des informations qui peuvent être facturées ou le fait que certains objets doivent être vérifiés pour leur intégrité, ce qui nous oblige à fournir un environnement d'exécution sécurisé et fiable pour exécuter des applications de haute sécurité. [4]

CHAPITRE 02

Système de détection des intrusions

Chapitre 2 : Système de détection des intrusions

2.1 Introduction :

Un système de détection d'intrusions (« Intrusion Detection Systems » ou IDS) est un appareil ou une application qui alerte l'administrateur en cas de faille de sécurité, de violation de règles ou d'autres problèmes susceptibles de compromettre son réseau informatique.

Les systèmes de détection d'intrusions surveillent et analysent les activités d'un réseau, analysent ses configurations et ses vulnérabilités, et vérifient l'intégrité des fichiers. Ils peuvent reconnaître des schémas d'attaque classiques. Pour ce faire, ils analysent les comportements anormaux et suivent les violations de règles par les utilisateurs. Certains systèmes industriels de détection d'intrusions peuvent également réagir à des menaces détectées.

Un système IDS est en général à double détente. La première étape, que l'on peut qualifier de passive, intervient sur la machine. Il s'agit de l'inspection des fichiers de configuration du réseau, notamment pour détecter les paramètres déconseillés et les violations de règles. La seconde étape, que l'on peut qualifier d'active, intervient sur le réseau. Ici, les mécanismes réutilisent des méthodes d'attaque identifiées et enregistrent les réactions.[5]

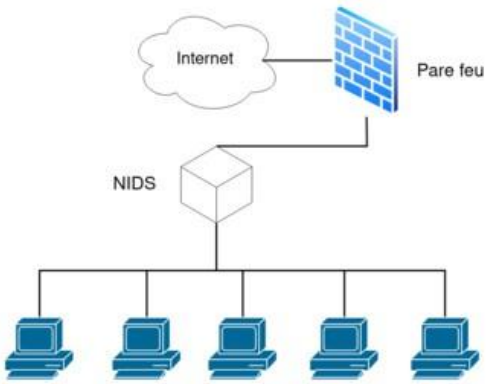


Figure 2.1 : Placement d'un NIDS en amont d'un pare-feu

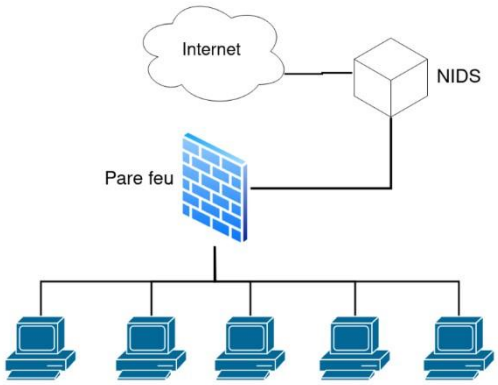


Figure 2.2 : Placement d'un NIDS en aval d'un pare-feu.

2.2 Les composants d'un système de détection d'intrusion :

On peut distinguer trois couches nécessaires pour un IDS :

2.2.1 La surveillance du système d'information :

La couche de surveillance du système d'information est chargée d'enregistrer toutes les actions. Les actions viennent soit :

- Du trafic réseau (NIDS).
- Du Système d'exploitation (HIDS).

Ces actions peuvent être de différentes sortes (lectures, écriture de fichiers, authentification...) et sont ensuite enregistrées en logs, qui sont envoyés à la couche surveillance afin d'être

2.2.2 Le scannage des sources d'informations :

La couche de monitoring va analyser les logs. Son but est d'évaluer les menaces pesant sur le système d'information en recherchant des événements notables. Les logs sont regroupés afin de distinguer des motifs permettant de distinguer une action connue, qui est alors évaluée. Ces événements sont alors classés en événements notables ou non.

Dans le cas où les événements sont notables, l'IDS va évaluer la menace du système d'information.

Après avoir identifié la menace de notre Système d'Information, il va alors indiquer son jugement sur la sécurité du réseau ou des systèmes et transmettre sa décision à la couche décision [4].

2.2.3 Les décisions prises lors de l'évaluation positive d'une intrusion :

Si la couche surveillance juge que le système a été compromis, l'IDS peut prendre plusieurs décisions :

- La première est de prévenir l'administrateur du système d'information afin qu'il puisse prendre les décisions qui s'imposent.
- La deuxième est d'effectuer une action définie à l'avance par l'administrateur système afin d'empêcher l'intrusion (ou la ralentir).

Remarque :

Selon les choix effectués lors de la conception d'un IDS, certaines parties de l'architecture peuvent changer. Les systèmes de détection d'intrusion distribués (DIDS) combinent les contraintes des systèmes distribués et des IDS.

La technique d'analyse des données peut éventuellement modifier l'architecture en ajoutant la couche de traitement des données propre à la technologie utilisée. Par exemple, lors de l'utilisation de réseaux neuronaux, les données doivent être traduites en données compréhensible pour le réseau et le résultat doit être traduit.

On peut ensuite considérer que le cloud computing pose de nouvelles contraintes sur l'architecture d'un IDS, notamment pour la détection d'intrusion. [5]

2.3 Caractéristiques souhaitées d'un IDS :

- Il doit fonctionner de manière continue avec une présence humaine minimum.
- Il doit être tolérant aux fautes c'est-à-dire qu'il doit être capable de retrouver son état initial de fonctionnement après un crash causé soit par une manipulation accidentelle soit par des activités émanant de personnes malintentionnées.

- Il doit résister à la subversion. L'IDS doit être capable de se contrôler lui-même et de détecter s'il a été modifié par un attaquant.
- Il doit imposer une supervision minimale du système sur lequel il tourne afin de ne pas interférer avec ses opérations normales.
- Il doit être configurable d'après les politiques de sécurité du système qu'il supervise.
- Il doit également être capable de s'adapter aux changements des systèmes et des comportements des utilisateurs au cours du temps (par exemple installation de nouvelles applications, transfert des utilisateurs d'une activité vers une autre et du coup transfert des ressources du système).

  Lorsque le nombre de systèmes à superviser augmente et donc que les attaques potentielles augmentent également, nous pouvons alors attendre de l'IDS les caractéristiques suivantes :

- Il doit être capable de superviser un nombre important de stations tout en fournissant des résultats de manière rapide et précise.
- Il doit fournir "un service minimum de crise" c'est-à-dire que si certains composants de l'IDS cessent de fonctionner, les autres composants doivent être affectés le moins possible par cet état de dégradation.
- Il doit autoriser des reconfigurations dynamiques. Si un grand nombre de stations est supervisé, il devient pratiquement impossible de redémarrer l'IDS sur tous les hôtes lorsque l'on doit effectuer un changement. [4]

2.4 Principes de détection :

Nous classons les IDS en deux grandes catégories de principe de détection d'intrusion :

2.4.1 Approche par scénario

Les systèmes à base de signatures qui consistent à rechercher dans l'activité de l'élément surveillé les signatures (empreintes) d'attaques répertoriées et donc connues. Ce principe de détection d'intrusion est réactif et pose plusieurs contraintes, en effet il ne détecte que les attaques répertoriées dont il possède l'empreinte. De ce fait il nécessite des mises à jour fréquentes. Ce principe de détection implique aussi que les pirates peuvent contourner celui-ci en maquillant leurs attaques, il modifie en fait la signature connue par les IDS et de ce fait l'attaque devient invisible par l'IDS.

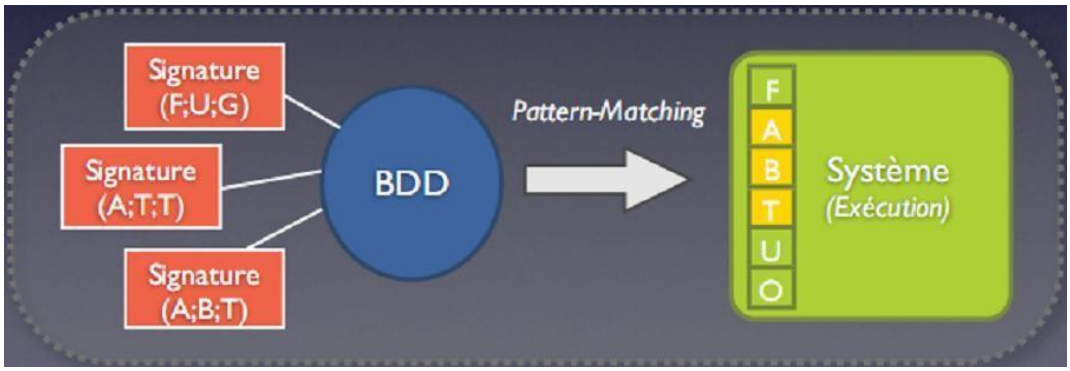
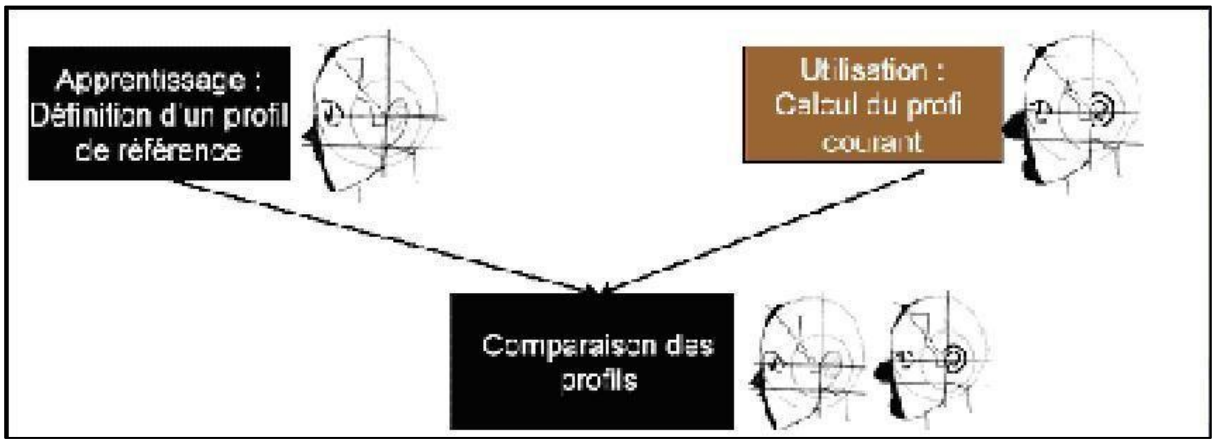


Figure 2.3: Approche par scénario

Cette approche fournit un diagnostic clair, il est donc possible de réagir et de contre attaquer, si la politique de sécurité est celle-là. Par contre ils ne peuvent détecter que les attaques contenues dans la base de connaissances. Il faut en permanence maintenir à jour cette base. Il est possible de rendre inactif un IDS utilisant cette approche par une attaque en déni de service.

2.4.2 Approche comportementale :

Les systèmes à approche comportementale consistent à détecter les différentes anomalies sur le réseau. C'est l'administrateur qui définira le fonctionnement "normal" des éléments surveillés, il y a donc une phase d'apprentissage pour fixer ce niveau. Par la suite l'IDS sera en mesure de signaler à l'administrateur toute situation qui divergera du niveau de fonctionnement de référence. Le fonctionnement de référence peut être élaboré par différentes analyses statistiques de l'élément à surveiller. Ce système de détection présente un avantage par rapport au précèdent : il détecte les nouveaux types d'attaques. Cependant il faudra faire parfois des ajustements afin que le fonctionnement de référence corresponde au mieux à l'activité normale des utilisateurs et ainsi réduire les fausses alertes qui en découleraient.



.

Figure 2.4 Approche comportementale

Cette approche a l'avantage de ne pas avoir besoin d'une base de signature. Elle permet donc, en théorie, de détecter des attaques inconnues. Cependant, elle a des inconvénients assez importants.

Que se passe-t-il s'il y a une attaque pendant l'apprentissage ? Celle-ci est considérée comme un comportement normal, et ne sera jamais détectée. De même il n'y a aucune interprétation de l'attaque, on ne sait pas quelle attaque a été déclenchée. On ne sait donc pas réagir. Il est très difficile d'effectuer un apprentissage complet.[6]

Méthodologie de détection :

Les systèmes de détection d'intrusion sont généralement classifiés en deux catégories, les systèmes de détection d'intrusion par signatures et les systèmes de détection d'intrusion par anomalies.

2.4.3 Systèmes de détection d'intrusion par signatures :

Les systèmes de détection d'intrusion par signature (ou SIDS : Signature-based Intrusion Detection System), reposent sur des bibliothèques de description des attaques (appelées signatures). Au cours de l'analyse du flux réseau, le système de détection d'intrusion analysera chaque événement et une alerte sera émise dès lors qu'une signature sera détectée. Cette signature peut référencer un seul paquet, ou un ensemble. Cette méthodologie de détection se révèle être efficace uniquement si la base de signatures est maintenue à jour de manière régulière. Dans ce cas, la détection par signatures produit peu de faux-positifs. Cependant, une bonne connaissance des différentes attaques est nécessaire pour les décrire dans la base de signature8. Dans le cas d'attaques inconnues de la base, ce modèle de détection s'avérera inefficace et ne générera donc pas d'alertes. La base de signature est donc très dépendante de l'environnement (système d'exploitation, version, applications déployées, …).pour effectuer une détection par signature, on peut utiliser, Les arbres ou les systèmes de transition d'états.

2.4.4 Systèmes de détection d'intrusion par anomalies :

Contrairement aux SIDS, les systèmes de détection d'intrusion par anomalies (ou AIDS : Anomaly-based Intrusion Détection System) ne se reposent pas sur des bibliothèques de description des attaques. Ils vont se charger de détecter des comportements anormaux lors de l'analyse du flux réseau. Pour cela, le système va reposer sur deux phases :

- Une phase d'apprentissage, au cours de laquelle ce dernier va étudier des comportements normaux de flux réseau.
- Une phase de détection, le système analyse le trafic et va chercher à identifier les événements anormaux en se basant sur ses connaissances.

Cette méthode de détection repose sur de nombreuses techniques d'apprentissage supervisé, telles que :

- Les réseaux de neurones artificiels.
- Le modèle de Markov caché.

- Les machines à vecteurs de support.

En 2019, la détection d'intrusion par anomalies est reconnue par la communauté comme étant très efficace. En effet, selon les méthodes d'apprentissage implémentées, l'exactitude des résultats peut rapidement atteindre plus de 90% de détection.

2.4.5 Hybride :

Cette méthodologie de détection consiste à reposer à la fois sur un système de détection par signatures et sur un système de détection par anomalies. Pour cela, les deux modules de détection, en plus de déclencher des alertes si une intrusion est détectée, peuvent communiquer leurs résultats d'analyse à un système de décision qui pourra lui-même déclencher des alertes.

2.4.6 Temporalité de détection :

Il existe deux types de temporalité dans les systèmes de détection d'intrusion. La détection en temps réel (système temps réel), et la détection post-mortem (analyse forensique). Le plus souvent, l'objectif est de remonter les alertes d'intrusion le plus rapidement possible à l'administrateur système. La détection en temps réel sera donc privilégiée. Cette temporalité de détection présente des défis de conception pour s'assurer que le système puisse analyser le flux de données aussi rapidement qu'il est généré. il est aussi envisageable d'utiliser un système de détection d'intrusion dans le cadre d'analyse post-mortem. Dans ce cas, ce dernier permettra de comprendre le mécanisme d'attaque pour aider à réparer les dommages subis et réduire le risque qu'une attaque du même genre se reproduise.

2.4.7 Corrélation des alertes :

La corrélation des alertes a pour objectif de produire un rapport de sécurité de la cible surveillée, Ce rapport sera basé sur l'ensemble des alertes produites par les différentes sondes de détection d'intrusion disséminées sur l'infrastructure. Pour cela, il est nécessaire de différencier deux composants :

- Les sondes : chargées de récupérer les données depuis les sources concernant leurs cibles (fichiers de logs, paquets réseaux,…) et de générer, si nécessaire, des alertes.
- Les composants d'agrégation et de corrélation : chargés de récolter les données des sondes et des autres composants d'agrégation et de corrélation afin de les corréler et produire le rapport de sécurité transmis à l'administrateur.

Les corrélations peuvent être décrites en deux types :

- Les corrélations explicites : ces corrélations sont utilisées lorsque l'administrateur peut exprimer une connexion entre des événements connus.
- Les corrélations implicites : celles-ci sont utilisées lorsque les données ont des relations entre elles et que des opérations sont nécessaires pour mettre en valeur certains événements. [5]

2.5 Étude et choix d'une solution de détection d'intrusion :

Concernant la source de données, la première chose à considérer lors de l'étude d'une solution de détection d'intrusion est le choix d'une sonde de détection : HIDS et NIDS (Network-based Intrusion Detection System). L'évaluation des sondes pourra s'appuyer sur une grille de notation intégrant les critères suivants :

- Méthodes et capacités de détection.
- Performance en conditions de charge élevées.
- Résistance aux techniques d'évasion.
- Exploitation des données à traiter.
- Ergonomie des interfaces d'administration et d'exploitation.
- Coûts de la solution.

Un HIDS aura un impact sur le serveur en termes de performance, car il va consommer une partie des ressources de ce serveur. Concernant l'installation d'un ou plusieurs NIDS, il faudra tenir compte de la disponibilité de points de raccordements permettant d'écouter le réseau. Le positionnement d'une sonde de détection dépend des contraintes propres à l'architecture. À l'extérieur du pare-feu (côté WAN), la sonde NIDS est plus proche des attaquants, mais va lever un volume important d'alertes en raison d'attaques classiques (e.g. balayage de port) qui seront très certainement bloquées par le pare-feu. Une sonde NIDS à l'intérieur d'un pare-feu (côté LAN), sera moins exposée aux bruits de fond résiduels et aux faux-positifs qui en résultent.

Par ailleurs, il peut être nécessaire de configurer la sonde NIDS avec deux interfaces réseaux. La première effectuera une surveillance en mode « promiscuous ». Dans ce mode, on capture tous les paquets qui passent par le lien réseau, qu'ils soient ou non adressés à la sonde. La seconde sera placée sur un VLAN (Virtual Local Area Network) dédié pour communiquer avec le système de gestion des événements et la console de gestion. En ce qui concerne les sondes HIDS, elles devront être déployées sur les serveurs critiques. [7]

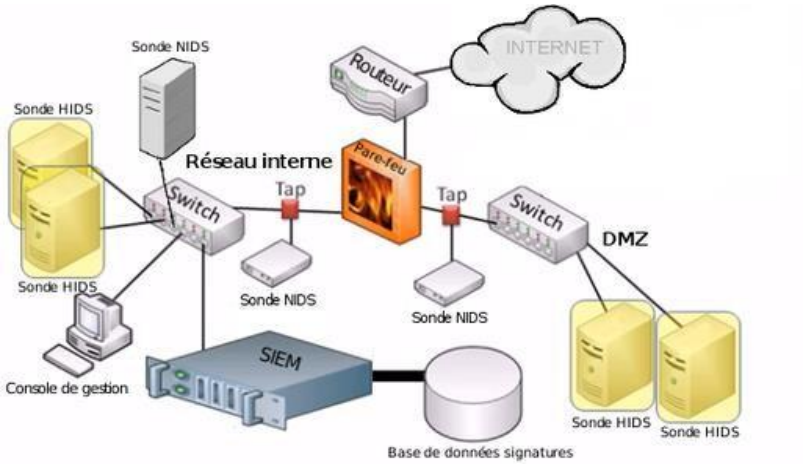


Figure.2.5 : Positionnement des sondes de détection d'intrusion dans une architecture réseau simplifiée.[7]

2.6 Les défis liés à la détection d'intrusions :

La sécurité de l'information ne se limite plus à une approche purement technique.

Aujourd'hui, certaines entreprises ont conscience que quelques briques technologiques, logicielles ou matérielles, ne suffisent plus à protéger leurs informations critiques. Désormais, ces entreprises s'orientent vers le management de la sécurité, dans une approche globale, aussi bien organisationnelle, technologique que juridique.

L'étude du cabinet d'audit Price Waterhouse Coopers auprès de 3 877 entreprises dans 78 pays « fait ressortir que plus d'une entreprise sur deux déclare que le Directeur des Systèmes d'Information est, in fine, propriétaire des risques de cybercriminalité. Seulement, une entreprise sur cinq (5 % en France) déclare ainsi que cette responsabilité est, in fine, du ressort de la Direction Générale ou du Conseil d'Administration » .Or, un facteur déterminant dans la réussite d'un projet de sécurité de l'information, s'intégrant ou non dans un système de gestion de la sécurité de l'information (ISMS ou Information Security Management System), est l'engagement réel et affiché de la structure dirigeante et des managers intermédiaires de l'organisation.

La prise en compte de la gestion des risques au sein du SI permet de considérer la sécurité de l'information comme un processus métier transverse et une réelle composante de la stratégie d'entreprise.

Ainsi, un projet de détection des intrusions qui s'inscrit dans un contexte de gestion des risques, dépasse la seule compétence des équipes techniques, du RSSI. [8]

Le processus de la gestion des risques de sécurité des SI peut être résumé en six phases principales :

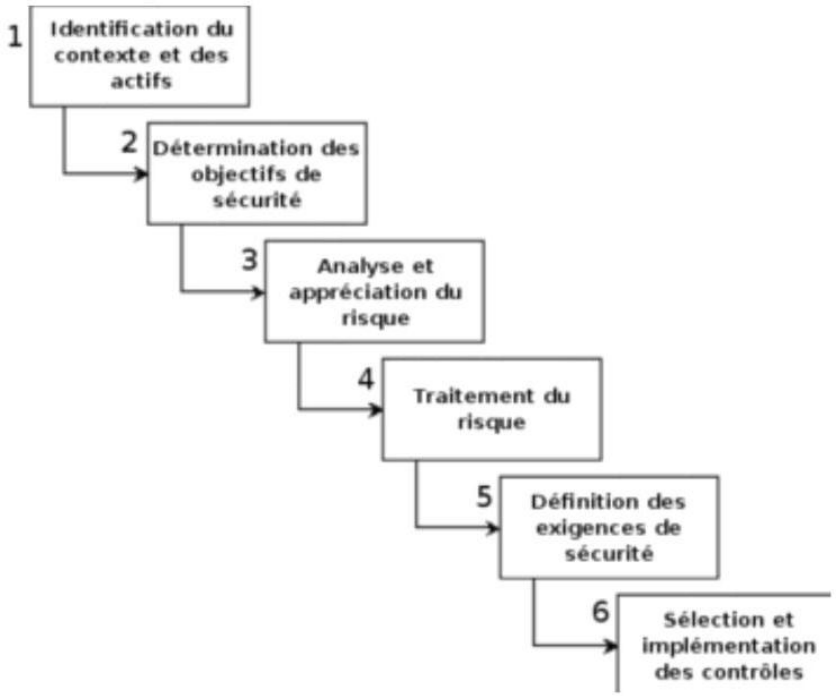


Figure 2.6 : Le processus de gestion des risques de sécurité.

Les trois premières phases conduisent à identifier, analyser et apprécier les risques de sécurité.

En phase 4, nous pouvons les accepter et ne rien faire, ou les transférer en externalisant une activité par exemple, voire les réduire en agissant sur leurs origines ou leurs conséquences.

En phase 5, des exigences de sécurité peuvent alors être déterminées afin de réduire le risque. Nous choisissons des mesures à mettre en œuvre pour atteindre ces objectifs de sécurité et elles vont être le fondement de la politique de sécurité. Par exemple, des contrôles techniques peuvent être choisis comme la détection des intrusions sur le réseau du SI.

Un système de détection d'intrusion qui s'inscrit dans ce contexte de gestion du risque, devra être perçu et accompagné par des moyens humains et financiers nécessaires à sa mise en œuvre.

En premier lieu, l'approche purement financière dans le choix de la conduite ou non d'un projet de détection des intrusions est un élément déterminant. Beaucoup de projets n'aboutissent pas, car aucune évaluation du retour sur investissement en sécurité informatique (ROSI ou Return On Security Investment) par les DSI ou les RSSI n'est réalisée. Selon les secteurs d'activités, la ligne budgétaire consacrée à la sécurité de l'information est assimilée à un centre de coût et non de profit. Dans un second lieu, l'approche « évaluation des risques » moins financière et plus qualitative est un argument important, mais non suffisant. La mise en place de solutions de détection des intrusions doit s'accompagner d'une évaluation du ROSI.

Le CLUSIF propose un état de l'art autour d'un modèle de coût et de la notion de ROSI. Mais il n'y a pas de consensus clair autour de la définition de ROSI. La définition orientée incidents de sécurité est celle que nous retiendrons.

Toute la difficulté est de pouvoir estimer la valeur exacte de chaque incident de sécurité lié à une intrusion et sa probabilité d'occurrence. La réalisation préalable d'une analyse des risques de sécurité des systèmes d'information facilitera ce travail, car elle intègre une évaluation des actifs du SI, du facteur d'exposition du système (EF ou Exposure Factor) et des vulnérabilités d'une infrastructure globale pouvant être exploitées par une ou plusieurs menaces connues ou inconnues.

Enfin, réagir de manière appropriée à une intrusion est surtout une question d'organisation et de procédures qui doivent être définies et appliquées par les équipes informatiques en réponse aux incidents.

Comment traiter l'alerte, comprendre l'incident et le clôturer ? Quelles sont les mesures conservatoires à prendre ? Qui alerter (le responsable sécurité, le CERT, la direction métier, la direction générale, la gendarmerie, etc.) ?

Les activités relatives à la détection d'intrusion doivent s'appuyer sur des actions précises et des rôles alloués à chacun, des outils dont l'informaticien a besoin et les résultats qu'il doit produire sous conditions de présentation, de contraintes de réactivité ou d'astreinte.

Les éléments collectés lors de la phase de détection et la qualification de l'intrusion ne suffisent pas eux seuls. Il est également nécessaire d'évaluer une stratégie de réponse en fonction de la criticité du système compromis et de l'impact d'une interruption de service du système.

- Activation d'une cellule de crise. □ délai de remise en production.
- Implication des relations publiques et communication appropriée.
- Volonté de répondre à une attaque.
- Volonté de poursuivre légalement l'attaquant. [6]

2.7 Conclusion

Les outils de détection d'intrusions sont apparus depuis quelques années maintenant et leur usage se répand dans les systèmes d'information et les réseaux. Ils sont sortis du domaine militaire et commencent à être intégrés à la définition des architectures de systèmes d'information commerciaux. Ces systèmes font pour la plupart de l'analyse de trafic (réseau, requêtes) envoyé à un système d'information, et recherchent dans leurs bases de connaissances des éléments identifiant ce trafic comme dangereux. L'évolution naturelle de ces systèmes conduit à prendre en compte des descriptions génériques des mécanismes d'attaque, plutôt que la détection d'attaques spécifiques sur des vulnérabilités connues. Dans un deuxième temps, il pourra apparaître sur le marché des systèmes de détection d'intrusions utilisant des notions de politique de sécurité pour détecter des actions non conformes à celle-ci, même si l'attaque sousjacente n'est pas explicitement identifiée. [5]

CHAPITRE 03

Machine learning (apprentissage automatique)

Chapitre 03 : Machine learning (apprentissage automatique)

3.1 Introduction

Le machine learning est un domaine captivant. Issu de nombreuses disciplines comme les statistiques, l'optimisation, l'algorithmique ou le traitement du signal, c'est un champ d'études en mutation constante qui s'est maintenant imposé dans notre société. Déjà utilisé depuis des décennies dans la reconnaissance automatique de caractères ou les filtres anti-spam, il sert maintenant à protéger contre la fraude bancaire, recommander des livres, films, ou autres produits adaptés à nos goûts, identifier les visages dans le viseur de notre appareil photo, ou traduire automatiquement des textes d'une langue vers une autre. Dans les années à venir, le machine learning nous permettra vraisemblablement d'améliorer la sécurité routière (y compris grâce aux véhicules autonomes), la réponse d'urgence aux catastrophes naturelles, le développement de nouveaux médicaments, ou l'efficacité énergétique de nos bâtiments et industries. Le but de ce chapitre est d'établir plus clairement ce qui relève ou non du machine learning, ainsi que des branches de ce domaine dont ce chapitre traitera. [9]

3.2 Aperçu sur l'apprentissage automatique

Un bref historique dans le domaine d'apprentissage automatique aussi communément appelé (Machine Learning, en anglais), nous amènent à parler des trois grandes époques de l'ordinateur, plus précisément, au tout début de l'informatique, de son évolution, au fil du temps et enfin au monde d'aujourd'hui et de demain.

De nos jours, nous pouvons constater et ce n'est qu'un point de vue, que l'évolution de l'informatique s'est faite principalement sur deux axes :

- Gain en capacité à cumuler de l'information et à sa diffusion dans des domaines tels que les fouilles de données (Data Mining), les entrepôts de données, les réseaux et services web, sans oublié leurs applications sous-jacentes sous Smartphones.
- Gain en intelligence des systèmes informatique, en particuliers, les domaines liés à l'intelligence artificielle

La discipline de l'apprentissage automatique (AA) possède de riches fondements théoriques. On sait, désormais, répondre à des questions comme :

- Quelles méthodes d'apprentissage sont les plus efficaces pour résoudre tel ou tel types de problèmes ?
- Combien d'exemples d'entraînement faut-il fournir à un programme d'apprentissage pour être certain qu'il apprenne avec une efficacité donnée ? [10]

3.3 Définition du machine learning

Qu'est-ce qu'apprendre, comment apprend-on, et que cela signifie-t-il pour une machine ? La question de l'apprentissage fascine les spécialistes de l'informatique et des mathématiques tout autant que neurologues, pédagogues, philosophes ou artistes.

- Une définition qui s'applique à un programme informatique comme à un robot, un animal de compagnie ou un être humain est celle proposée par Fabien Benureau (2015): «L'apprentissage est une modification d'un comportement sur la base d'une expérience».
- Dans le cas d'un programme informatique, qui est celui qui nous intéresse dans ce chapitre, on parle d'apprentissage automatique, ou machine learning, quand ce programme a la capacité d'apprendre sans que cette modification ne soit explicitement programmée.
- **L'apprentissage automatique** (en anglais Machine Learning) est un type d'intelligence artificielle qui confère aux ordinateurs la capacité d'apprendre sans être explicitement programmés. Il consiste à la mise en place d'algorithmes ayant pour objectif d'obtenir **une analyse prédictive** à partir de données, dans un but précis. [11]

Exemple : Supposons qu'une entreprise veuille connaître le montant total dépensé par un client ou une cliente à partir de ses factures. Il suffit d'appliquer un algorithme classique, à savoir une simple addition : un algorithme d'apprentissage n'est pas nécessaire. Supposons maintenant que l'on veuille utiliser ces factures pour déterminer quels produits le client est le plus susceptible d'acheter dans un mois. Bien que cela soit vraisemblablement lié, nous n'avons manifestement pas toutes les informations nécessaires pour ce faire. Cependant, si nous disposons de l'historique d'achat d'un grand nombre d'individus, il devient possible d'utiliser un algorithme de machine learning pour qu'il en tire un modèle prédictif nous permettant d'apporter une réponse à notre question. [12]

3.4 Ingrédients de le machine learning

Le machine learning repose sur deux piliers fondamentaux :

- D'une part, les données, qui sont les exemples à partir duquel l'algorithme va apprendre.
- D'autre part, l'algorithme d'apprentissage, qui est la procédure que l'on fait tourner sur ces données pour produire un modèle. On appelle entraînement le fait de faire tourner un algorithme d'apprentissage sur un jeu de données.

Ces deux piliers sont aussi importants l'un que l'autre. D'une part, aucun algorithme d'apprentissage ne pourra créer un bon modèle à partir de données qui ne sont pas pertinentes c'est le concept garbage in, garbage out qui stipule qu'un algorithme d'apprentissage auquel on fournit des données de mauvaise qualité ne pourra rien en faire d'autre que des prédictions de mauvaise qualité. D'autre part, un modèle appris avec un algorithme inadapté sur des données pertinentes ne pourra pas être de bonne qualité.

Remarque : Bien que l'usage soit souvent d'appeler les deux du même nom, il faut distinguer l'algorithme d'apprentissage automatique du modèle appris : le premier utilise les données pour produire le second, qui peut ensuite être appliqué comme un programme classique.

Un algorithme d'apprentissage permet donc de modéliser un phénomène à partir d'exemples. Nous considérons ici qu'il faut pour ce faire définir et optimiser un objectif. Il peut

par exemple s'agir de minimiser le nombre d'erreurs faites par le modèle sur les exemples d'apprentissage.

Quelques exemples :

- Un vendeur en ligne peut chercher à modéliser des types représentatifs de clientèle, à partir des transactions passées, en maximisant la proximité entre clients et clientes affectés à un même type.
- Une compagnie automobile peut chercher à modéliser la trajectoire d'un véhicule dans son environnement, à partir d'enregistrements vidéo de voitures, en minimisant le nombre d'accidents
- Des chercheurs en génétique peuvent vouloir modéliser l'impact d'une mutation sur une maladie, à partir de données patientes, en maximisant la cohérence de leur modèle avec les connaissances de l'état de l'art
- Une banque peut vouloir modéliser les comportements à risque, à partir de son historique, en maximisant le taux de détection de non solvabilité. [11] **L'intelligence artificielle et le machine learning**

Le machine learning peut être vu comme une branche de l'intelligence artificielle. En effet, un système incapable d'apprendre peut difficilement être considéré comme intelligent. La capacité à apprendre et à tirer parti de ses expériences est en effet essentielle à un système conçu pour s'adapter à un environnement changeant. L'intelligence artificielle, définie comme l'ensemble des techniques mises en œuvre afin de construire des machines capables de faire preuve d'un comportement que l'on peut qualifier d'intelligent, fait aussi appel aux sciences cognitives, à la neurobiologie, à la logique, à l'électronique, à l'ingénierie et bien plus encore.

Pourquoi utiliser l'apprentissage automatique ?

La machine learning peut servir à résoudre des problèmes que l'on ne sait pas résoudre (comme dans l'exemple de la prédiction d'achats ci-dessus), que l'on sait résoudre, mais dont on ne sait formaliser en termes algorithmiques comment nous les résolvons (c'est le cas par exemple de la reconnaissance d'images ou de la compréhension du langage naturel), que l'on sait résoudre, mais avec des procédures beaucoup trop gourmandes en ressources informatiques (c'est le cas par exemple de la prédiction d'interactions entre molécules de grande taille, pour lesquelles les simulations sont très lourdes).

Le machine learning est donc utilisé quand les données sont abondantes (relativement), mais les connaissances peu accessibles ou peu développées. Ainsi, le machine learning peut aussi aider les humains à apprendre : les modèles créés par des algorithmes d'apprentissage peuvent révéler l'importance relative de certaines informations ou la façon dont elles interagissent entre elles pour résoudre un problème particulier. Dans l'exemple de la prédiction d'achats, comprendre le modèle peut nous permettre d'analyser quelles caractéristiques des achats passés permettent de prédire ceux à venir. Cet aspect de le machine learning est très utilisé dans la recherche scientifique : quels gènes sont impliqués dans le développement d'un certain type

de tumeur, et comment ? Quelles régions d'une image cérébrale permettent de prédire un comportement ? Quelles caractéristiques d'une molécule en font un bon médicament pour une indication particulière ? Quels aspects d'une image de télescope permettent d'y identifier un objet astronomique particulier ? [10]

3.5 Types d'apprentissage :

Les algorithmes d'apprentissage peuvent se caractériser selon le mode d'apprentissage qu'ils emploient :

3.5.1 Apprentissage supervisé :

- Dans ce type d'apprentissage, on cherche à définir une règle de prédiction $R : \mathcal{X} \rightarrow \mathcal{Y}$ d'une variable à prédire Y en fonction de variables prédictives X. On dispose pour cela de données pour lesquelles à la fois X et Y sont observés et on cherche, parmi une famille de règles possibles, celle qui optimise un critère de qualité à définir. Le but est ensuite de pouvoir appliquer $\mathcal{R}$ à de nouvelles données pour lesquelles seules X est connu afin d'en déduire une prédiction $Ypred$ = (X).
- On dispose d'un ensemble d'objets et pour chaque objet une valeur cible associée, il faut apprendre un modèle capable de prédire la bonne valeur cible d'un objet nouveau. Il existe deux types de sous-problèmes en apprentissage supervisé numérique :
  - Régression : lorsque la valeur cible à prédire est continue.
  - Classement, classification ou catégorisation : lorsque la valeur cible à prédire est discrète.

Exemple : un diagnostic médical est une règle d'apprentissage supervisé (X sont les symptômes, Y le diagnostic). Parmi les méthodes d'apprentissage supervisé on trouve :

3.5.1.1 Les arbres de décision :

Les arbres de décision, un moyen **d'apprentissage supervisé** qui permet de séparer des individus dans des groupes selon des règles ou de prévoir la valeur d'une variable (cible) à partir de variables en entrée. Un arbre de décision est composé de :

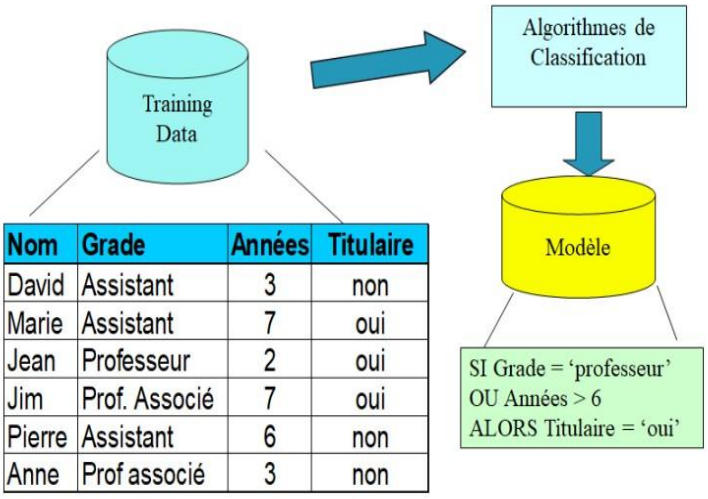
- Un ensemble de nœuds internes : un nœud interne correspond à un test sur un attribut.
- Un ensemble de branches : une branche correspond à un résultat d'un test (la valeur de l'attribut).
- Un ensemble de Feuilles : une feuille correspond à une classe ou bien à une valeur de la variable cible.

Figure 3.1: Exemple d'arbre de décision. [11]

Le fonctionnement de la classification par arbre de décision se décompose en deux phases : la **phase d'apprentissage** et la **phase de classification** :

- **La phase d'apprentissage :** Dans cette phase, les approches de classification utilisent un jeu d'apprentissage (Training Data) dans lequel tous les objets sont déjà associés aux classes de références connues.

  L'algorithme de classification apprend du jeu d'apprentissage et construit un modèle.



Nom	Grade	Années	Titulaire
David	Assistant	3	non
Marie	Assistant	7	oui
Jean	Professeur	2	oui
Jim	Prof. Associé	7	oui
Pierre	Assistant	6	non
Anne	Prof associé	3	non

Figure 3.2 : Exemple de phase d'apprentissage.

- **La phase de classification :**
  - Tester le modèle appris (arbre générer) sur un échantillon d'instances classées (jeu de teste).
  - Appliquer le modèle sur des instances non classées.

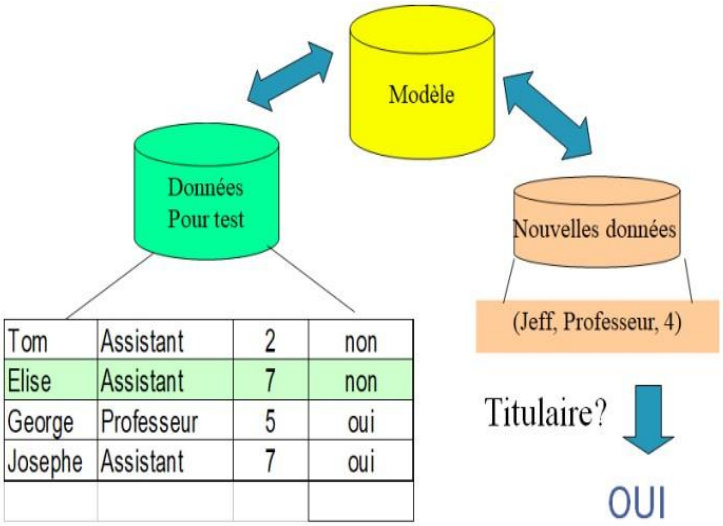


Figure 3.3: Exemple de phase de classification.

3.5.2 Apprentissage non supervisé :

Dans ce second type d'apprentissage, on dispose de variables observées X dont on souhaite apprendre une caractéristique structurelle. Le but n'est alors pas de prédire une autre variable Y mais une caractéristique inconnue de la matrice X. La famille la plus couramment utilisée d'algorithmes d'apprentissage supervisé est celle des algorithmes de classification (clustering) dont le but est de créer des groupes d'individus rassemblés sur la base de la proximité de leurs valeurs de X.

On dispose d'un ensemble d'objets sans aucune valeur cible associée, il faut apprendre un modèle capable d'extraire les régularités présentes au sein des objets pour mieux visualiser ou appréhender la structure de l'ensemble des données.

Exemple : Identifier des groupes de documents ayant un sujet similaire, sans les avoir au préalable étiqueté par sujet. Cela permet d'organiser de larges banques de textes.

3.5.3 Apprentissage semi-supervisé :

On dispose d'un petit ensemble d'objets avec pour chacun une valeur cible associée et d'un plus grand ensemble d'objets sans valeur cible ; il faut tirer profit à la fois des données avec et sans valeurs cibles pour résoudre des tâches d'apprentissage supervisé ou non-supervisé.

3.5.4 Apprentissage par renforcement :

On dispose d'un ensemble de séquences de décisions (politiques ou stratégiques) dans un environnement dynamique, et pour chaque action de chaque séquence une valeur de récompense (la valeur de récompense de la séquence est alors la somme des valeurs des

récompenses des actions qu'elle met en œuvre) ; il faut apprendre un modèle capable de prédire la meilleure décision à prendre étant donné un état de l'environnement. [10]

3.6 Quelques exemples d'application.

Exemples de problèmes de régression :

- Prédiction du montant des ventes d'une entreprise compte tenu du contexte économique.

  Prédiction du prix de vente d'une maison en fonction de plusieurs critères.

- Prédiction de la consommation électrique dans une ville étant donnée des conditions météorologiques…

Exemples de problèmes de catégorisation :

- Prédiction de l'état sain/malade d'un patient par rapport à une maladie et compte tenu de différents facteurs.
- Prédiction de l'accord ou du refus d'un crédit à un client d'une banque en fonction de ses caractéristiques.

Prédiction du chiffre correct à partir d'une image scannée d'un chiffre écrit à la main. [11]

3.7 Conclusion

Nous avons défini dans ce chapitre la notion de L'apprentissage automatique (en anglais machine learning) ou apprentissage statistique est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches statistiques pour donner aux ordinateurs la capacité d'apprendre à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune. Plus largement, cela concerne la conception, l'analyse, le développement et l'implémentation de telles méthodes…

CHAPITRE 04

Systèmes de détection d'intrusion de l'IoT basés sur la machine learning

Chapitre 4 : Systèmes de détection d'intrusion de l'IoT basés sur le machine learning

4.1 Obtention des données

La première étape consiste à obtenir les données. Lors de la phase d'apprentissage, ces informations peuvent en apprendre davantage sur les habitudes de l'utilisateur ou sur différents types d'attaques. En phase d'exécution, ils permettent de détecter les attaques. La question, cependant, est de savoir s'il existe une mesure de la quantité de données requises pour un modèle correct, qui respecte un taux d'erreur acceptable. Cette question reste en suspens à ce jour. Cependant, le choix des informations à considérer dépendra du type d'étude.[12]

4.2 Supervisé ou non supervisé

L'apprentissage supervisé utilise généralement des ensembles de données qui représentent des attaques connues. L'apprentissage non supervisé repose uniquement sur le comportement de l'utilisateur pour détecter les changements de comportement normal qui représentent une attaque. Les algorithmes supervisés obtiennent d'excellents résultats contre les intrusions connues, ils ont surpassé les algorithmes non supervisés. À l'inverse, pour les attaques inconnues, l'efficacité des algorithmes supervisés chute considérablement, contrairement aux algorithmes non supervisés. Cela peut s'expliquer par le fait que puisque les algorithmes non supervisés ne font que partitionner les données, ils sont toujours inconscients de l'attaque. En effet, comme ces derniers ne font que regrouper des données similaires, ils ne savent pas quand il s'agit d'une attaque. Bien que les algorithmes supervisés et non supervisés obtiennent des résultats similaires dans les attaques inconnues, les modèles non supervisés sont privilégiés pour leur robustesse. En fait, ces modèles ne changeaient pas significativement leur taux de réussite selon que l'intrusion était connue ou non. De plus, ils n'ont pas d'oracle leur indiquant à quelle classe appartiennent les données. Ils vont seuls en classe. Par conséquent, elles sont plus indépendantes que les techniques supervisées

Un oracle est nécessaire. Leur point fort est la classification des attaques appartenant à des labels inconnus. En fait, ils ne devraient pas rendre toutes les étiquettes. De plus, il est parfois difficile de catégoriser sans ambiguïté les informations au sein d'une classe. C'est pourquoi cette approche est parfois mieux adaptée à l'IDS.

4.3 Hypothèse

Dans le cas des algorithmes supervisés, l'hypothèse difficile à respecter lors du processus d'apprentissage est qu'aucune attaque n'est incluse dans les informations qui modélisent le comportement normal du programme, mais le comportement normal du programme peut présenter de nombreuses irrégularités. En fait, dans le cas contraire, on insérerait les propriétés de l'attaque comme comportement normal. Par conséquent, ce dernier ne sera pas détecté.

Au lieu de cela, comme décrit dans ce chapitre, des algorithmes non supervisés peuvent démanteler cette hypothèse en nettoyant les données pour supprimer les informations d'attaque. Pour ce faire, on recherche un ensemble de séquences (appelées patterns) d'appels au système qui sont fortement présents dans le système et on les classe selon leur danger

d'attaquer le système. Ensuite, nous examinons toutes les séquences de système d'appel existantes et les comparons avec des modèles découverts précédemment, ce qui donne une sorte d'empreinte de différentes séquences de système d'appel par rapport aux modèles qu'elles contiennent. Après cela, nous regardons la distance de toutes les empreintes les unes par rapport aux autres afin de générer un seul graphique pour l'ensemble. Cela nous permet de mieux visualiser ceux qui ont les mêmes propriétés (et donc sont très proches les uns des autres) par rapport à ceux qui sont très différents. Ainsi l'espacement entre les deux ensembles d'empreintes est très grand, l'un en contient beaucoup et l'autre en contient très peu Supposons que l'un soit une attaque et l'autre non. De plus, étant donné que les traces d'attaque sont supposées être inférieures aux traces de non-attaque, il est facile de trouver toutes les traces d'attaque. C'est pourquoi nous utilisons des algorithmes hors ligne non supervisés pour détecter les anomalies de données.

Ensuite, des algorithmes supervisés peuvent utiliser ces données nettoyées pour concevoir et implémenter des modèles.

4.4 Implémentation

Une approche courante basée sur des approches comportementales consiste à prendre les empreintes digitales des utilisateurs, c'est-à-dire leur comportement, et à observer s'il y a une incompatibilité sur le système. Ainsi, un comportement anormal et donc d'éventuelles attaques peuvent être détectés. Au lieu de cela, nous pouvons les détecter par les empreintes digitales de certains pirates connus. Ce dernier peut être appris grâce à l'apprentissage automatique.

Cependant, une autre approche basée sur des scénarios consiste à utiliser des données représentant des attaques. Pour y parvenir, l'IDS doit être suffisamment rapide, efficace et flexible pour s'adapter aux petits changements normaux d'utilisateurs, mais ne pas autoriser les écarts par rapport aux situations d'attaque. Sinon, soit il ne sera pas détecté, soit le nombre de faux positifs risque de se multiplier. En général, un tel IDS est réalisé en trois étapes. Nous modélisons tous les comportements normaux ou les caractéristiques d'agression de chaque utilisateur, nous le transmettons à l'apprentissage automatique pour apprendre, puis nous voyons si le comportement s'écarte du comportement habituel ou s'approche d'une situation agressive.

Dans certains cas, il est intéressant de connaître le type d'attaque, pas seulement si elle s'est produite. Pour cela, on peut demander au réseau de neurones de nous donner par exemple [0,0,0] s'il n'y a pas d'attaque, [0,0,1] s'il y a une attaque de type 1, [0,1,0] s'il y a est une attaque de type 2, s'il y a une attaque de type 2 Pour les attaques de type 3, c'est [1,0,0]. Si le réseau nous donne [0,1,1], nous pouvons vérifier s'il y a deux attaques simultanées, ou si le réseau a correctement déterminé ce qui s'est passé.

La dernière implémentation consiste à gérer le volume élevé d'alertes d'IDS via l'apprentissage automatique. Cette approche est donc une sorte de filtre pour ces derniers, pour se concentrer sur les alertes les plus importantes. En fait, IDS peut générer un grand nombre de fausses alarmes, ce qui empêche les administrateurs système d'accomplir la tâche. Par

conséquent, il s'agit d'un complément à IDS, pas d'un remplacement pour eux. Comme mentionné précédemment, il existe NIDS et HIDS.

HIDS Pour modéliser le comportement d'un utilisateur, on peut regarder toutes les commandes qu'il utilise sur une période de temps, comme fait dans l'article [12], qui donne le HIDS offline. Cette approche est logique car la plupart des gens n'utiliseront pas le système dans le même but ou de la même manière.

Pour les HIDS similaires mais en ligne, l'apprentissage automatique identifie les futures commandes en fonction du dernier k utilisé. Cependant, leur ordre n'indique pas si une attaque a eu lieu. Il semble plus logique de regarder toutes les commandes utilisées sur une période de temps. Le principal inconvénient est que les paramètres d'appel système ne sont pas pris en compte. Par conséquent, il est intéressant de considérer la valeur de retour, le statut d'erreur et d'autres paramètres pour détecter l'attaque.

En pratique, prenons les appels système suivants effectués par un simple utilisateur : open, read, write. Ces trois appels système peuvent sembler inoffensifs car il s'agit d'un simple fichier d'ouverture, de lecture et d'écriture. Cependant, si nous regardons les paramètres de ces appels système, la situation change, c'est-à-dire le passage de fichiers. Une autre façon est d'apprendre le programme au lieu de l'utilisateur. Par conséquent, le fonctionnement normal du logiciel sur la machine d'apprentissage automatique.

NIDS En parlant de NIDS, une compréhension globale de toutes les variables d'un paquet, ainsi que du protocole, est nécessaire pour comprendre le fonctionnement normal du système. Essayons de partitionner les paquets TCP 8. Pour ce faire, nous pouvons d'abord voir des propriétés entre eux qui ne changeront peut-être jamais : version du protocole + paramètres réservés. Des propriétés supplémentaires permettent de les partitionner : adresse source/destination + protocole utilisé.

Ces deux propriétés sont souvent utilisées par les pare-feux pour filtrer les paquets. Enfin, certains attributs peuvent être différents dans une même partition : taille d'en-tête, identifiant, TTL, ... Ce sont ceux qui sont généralement utilisés pour détecter des anomalies, et regarder les valeurs de ces attributs pour déterminer des anomalies n'est pas une bonne façon de faire.

Nous regardons donc la moyenne de certaines valeurs, le pourcentage d'événements en fonction de la valeur de l'attribut, le pourcentage de paquets avec une telle valeur.

Par conséquent, il est nécessaire de s'assurer qu'il y a suffisamment de paquets pendant la phase d'apprentissage pour s'assurer que les comportements normaux qui ne sont pas présents dans ces paquets ne sont pas oubliés. [13]

4.5 Problématiques des modèles

- **Non périodicité** : Le caractère non périodique de certains phénomènes peut avoir des effets problématiques. Prenez la rédaction d'un mémoire par exemple. Il y a une période où les étudiants font beaucoup de recherches et une autre période où les étudiants rédigent un mémoire. Dans les deux phases, le même étudiant travaille toujours sur la

même machine. Cependant, les besoins en ressources ne sont pas égaux, ce qui peut faire croire à l'apprentissage automatique qu'il y a quelque chose d'inhabituel. De même, une augmentation du trafic sur le site Web ne signifie pas nécessairement une attaque sur ce site Web. La mise en place d'un détecteur d'intrusion pouvant ralentir cette croissance aurait des conséquences négatives. Pour y remédier, il est conseillé de mélanger un grand nombre de paramètres pour créer le profil de l'utilisateur et permettre à l'administrateur de considérer s'il s'agit d'une situation indésirable.

- **Espacement des attaques** : Une attaque difficile à voir est une attaque qui se produit à intervalles réguliers. traitez avec eux.

4.6 Optimisation

Il y a une forte volonté de réduire le temps d'étude pour pouvoir mettre en place des solutions commerciales. Une méthode pour réduire le temps d'apprentissage et la taille de la structure est proposée. Pour être plus efficace, nous pouvons définir la structure de tous les réseaux de neurones utilisés dans IDS. Voici quelques-uns :

La première case contient le filtre d'information, puis une autre case contient le réseau de neurones.

La première boîte contient un filtre d'informations, puis n boîtes de réseau de neurones, et la dernière boîte agira comme un arbitre, déterminant ce qui se passe sur le réseau en fonction des informations reçues par les n boîtes de réseau de neurones. Chacun d'eux identifie un type d'attaque (DOS, U2R, R2L, scan...). Ceci est également connu sous le nom de méthode Boo Sting.

Les trois premières cases, puis la dernière case. La formation est la suivante :

La première case est entraînée avec un certain nombre d'informations. Nous acquérons au hasard de nouvelles informations et en formons une seconde. - Nous acquérons de nouvelles informations et observons la réaction des deux premiers. Si les deux boîtes ne convergent pas vers la même idée, nous utilisons cette information comme entraînement pour la troisième. [11].

4.7 Conclusion

Dans ce chapitre, nous avons présenté différentes techniques de détection des intrusions IoD basées sur l'apprentissage automatique en fonction de leur mise en œuvre et de leur utilisation.

CHAPITRE 05

Conception et réalisation

Chapitre 5 : Conception et réalisation

5.1 Introduction

Ces dernières années, la diffusion des appareils IoT dans le monde a progressé rapidement. Les objets connectés sont désormais déployés dans tous les domaines tels que la santé, les villes intelligentes, l'éducation, etc. Pour intégrer ce flux de commercialisation rapide, peu d'attention a été accordée à la sûreté et à la sécurité des appareils et des réseaux IoT qui mettent en danger les utilisateurs IoT et à leur tour perturbe l'ensemble de l'écosystème connecté à Internet, y compris les sites Web, les applications, les réseaux sociaux et les serveurs. De plus, les vecteurs d'attaques de sécurité ont évolué dans les deux sens, en termes de complexité et de diversité. Par conséquent, une plus grande attention doit être accordée à l'analyse de ces attaques, à leur détection, ainsi qu'à la prévention des infections et à la récupération du système après les attaques.

Dans cette thèse, nous avons étudié et proposé un système de détection et de prévention des intrusions (IDPS) basé sur le Machine Learning (ML) pour l'écosystème IoT afin de détecter et de répondre immédiatement aux menaces potentielles dès qu'elles se produisent. Il représente un cadre cohérent avec un workflow de sécurité complet, de la collecte de données à détection des menaces et activation des actions appropriées.

5.2 Environnement d'exécution

5.2.1 L'éditeur choisi :

PyCharm Community Edition 2022.1.1

Pycharm est un environnement de développement intégré utilisé pour programmer en Python. IL permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.

Développé par l'entreprise tchèque JetBrains, c'est un logiciel multiplateforme qui fonctionne sous Windows, Mac OS X et Linux. Il est décliné en édition professionnelle, diffusé sous licence propriétaire, et en édition communautaire diffusé sous licence Apache.

PyCharm fournit la saisie automatique de code intelligente, des inspections de code, la mise en évidence d'erreur à la volée et des correctifs rapides, en plus de refacturations de code automatisées et de riches capacités de navigation.

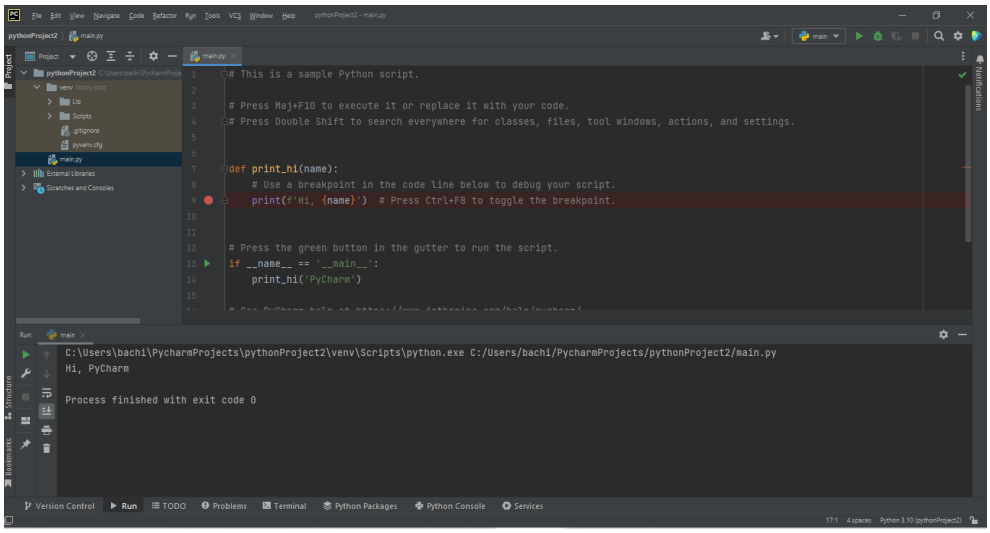


Figure 5.1: PyCharm 2022.1.1

5.2.2 Le langage de programmation utilisé :

Python 3.8.0 :

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

L'interpréteur Python et sa vaste bibliothèque standard sont disponibles librement, sous forme de sources ou de binaires, pour toutes les plateformes majeures depuis le site Internet https://www.python.org/ et peuvent être librement redistribués. Ce même site distribue et pointe vers des modules, des programmes et des outils tiers. Enfin, il constitue une source de documentation.

5.3 Dataset

L'ensemble de données d'intrusion choisi pour cette étude c'est IOT-2019 data-set, fourni par laboratoire de recherche sur le piratage et les contre-mesures (HCR Lab) de Corée du Sud [17], Les divers types d'attaques réseau ont été créés dans l'environnement Internet des objets (IoT) à des fins académiques. Deux appareils domestiques intelligents typiques - SKT NUGU (NU 100) et caméra Wi-Fi EZVIZ (C2C Mini O Plus 1080P) - ont été utilisés. Tous les appareils, y compris certains ordinateurs portables ou téléphones intelligents, étaient connectés au même réseau sans fil.

L'ensemble de données se compose de 42 fichiers de paquets réseau bruts (pcap) à différents moments.

- Les fichiers de paquets sont capturés en utilisant le mode moniteur de l'adaptateur réseau sans fil.

- Toutes les attaques à l'exception la catégorie Mirai Botnet sont des paquets capturés lors de la simulation d'attaques à l'aide d'outils tels que Nmap. Dans le cas de la catégorie Mirai Botnet, les paquets d'attaque ont été générés sur un ordinateur portable puis manipulés pour le faire apparaître comme s'ils provenaient de l'appareil IoT.

CATEGORIE	SOUS-CATEGORIE	NOMBRE DE PAQUETS
Normal	Normal	1756276
Scanning	Host Discovery	2454
Scanning	Port Scanning	20939
Scanning	OS/Version Detection	1817
Man in the Middle (MITM)	ARP Spoofing	101885
Denial of Service (DoS)	SYN Flooding	64646
Mirai Botnet	Host Discovery	673
Mirai Botnet	Telnet Bruteforce	1924
Mirai Botnet	UDP Flooding	949284
Mirai Botnet	ACK Flooding	75632
Mirai Botnet	HTTP Flooding	10464

5.4 Algorithme d'apprentissage pour la détection d'intrusion

Dans cette étude quatre algorithme d'apprentissage sont utilisés

5.4.1 La classification naïve bayésienne (Gaussian Naive Bayes)

- from sklearn.naive_bayes import GaussianNB model1 = GaussianNB()

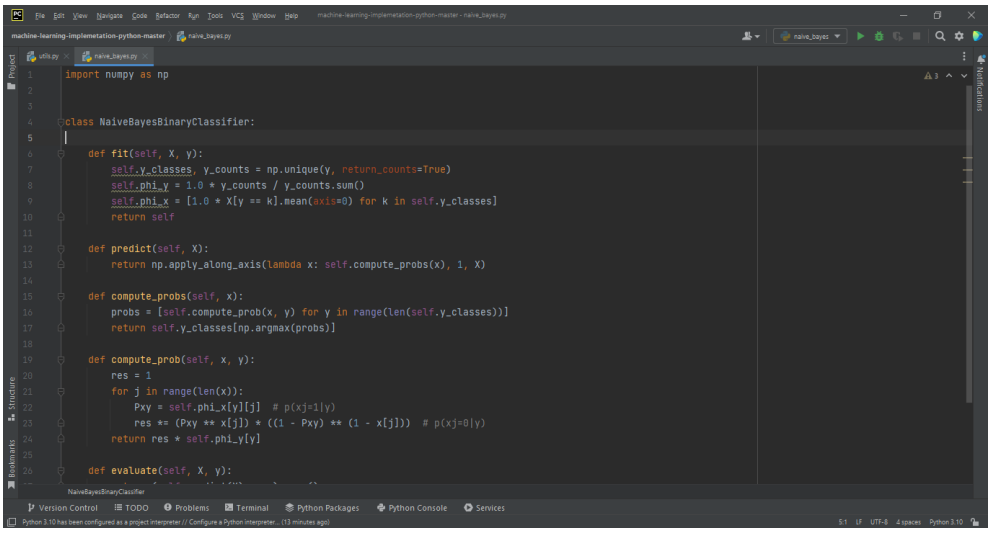
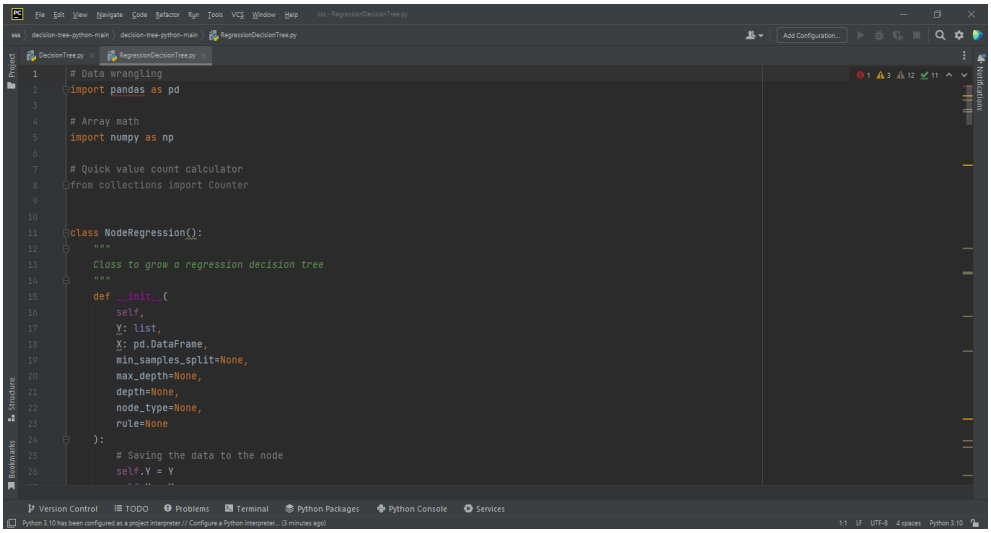


Figure 5.2 : Algorithme classification naïve bayésienne

5.4.2 Arbre de décision (Decision Tree)

- from sklearn.tree import DecisionTreeClassifier
- model2 = DecisionTreeClassifier(criterion="entropy", max_depth=4)

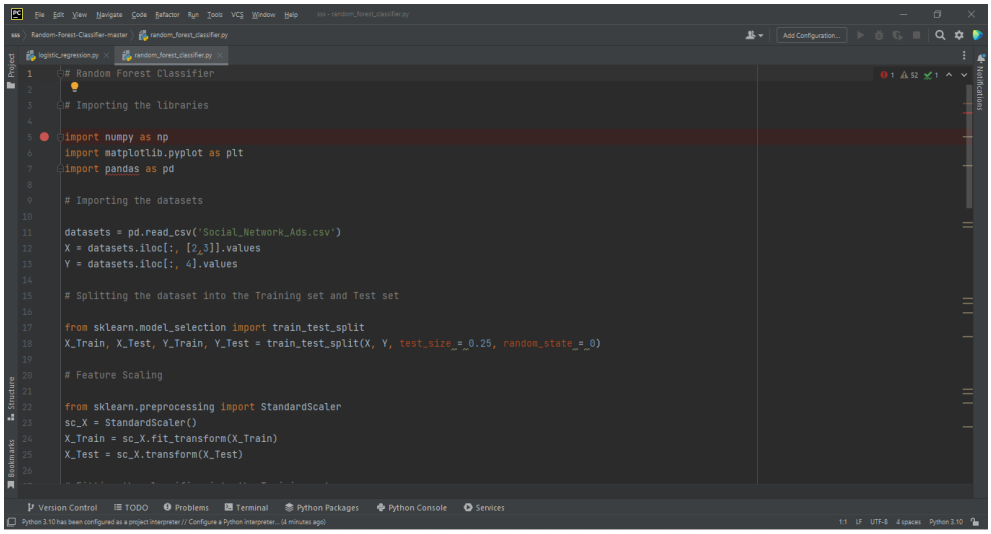


```
# Data wrangling
import pandas as pd

# Array math
import numpy as np

# Quick value count calculator
from collections import Counter


class NodeRegression():
    """
    Class to grow a regression decision tree
    """
    def __init__(
        self,
        Y: list,
        X: pd.DataFrame,
        min_samples_split=None,
        max_depth=None,
        depth=None,
        node_type=None,
        rule=None
    ):
        # Saving the data to the node
        self.Y = Y
```

Figure 5.3 : Algorithme Arbre de décision

5.4.3 Forêt d'arbres décisionnels (RANDOM FOREST)

- from sklearn.ensemble import RandomForestClassifier model3 = RandomForestClassifier(n_estimators=30)

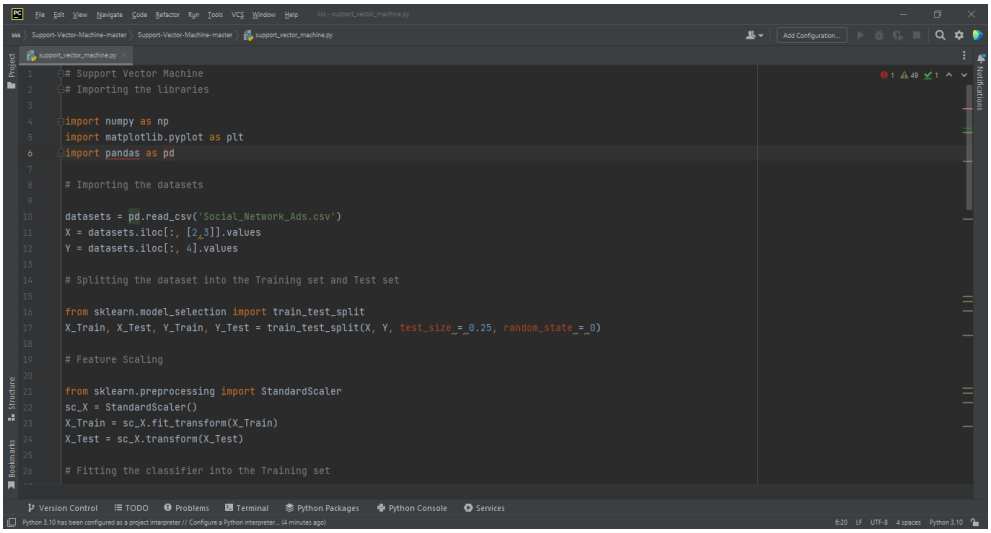


```
# Random Forest Classifier

# Importing the libraries

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the datasets

datasets = pd.read_csv('Social_Network_Ads.csv')
X = datasets.iloc[:, [2,3]].values
Y = datasets.iloc[:, 4].values

# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_Train, X_Test, Y_Train, Y_Test = train_test_split(X, Y, test_size = 0.25, random_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_Train = sc_X.fit_transform(X_Train)
X_Test = sc_X.transform(X_Test)
```

Figure 5.4 : Algorithme Forêt d'arbres décisionnels

5.4.4 Machine à vecteurs de support (SUPPORT VECTOR MACHINE)

- from sklearn.svm import SVC model4 = SVC(gamma='scale')

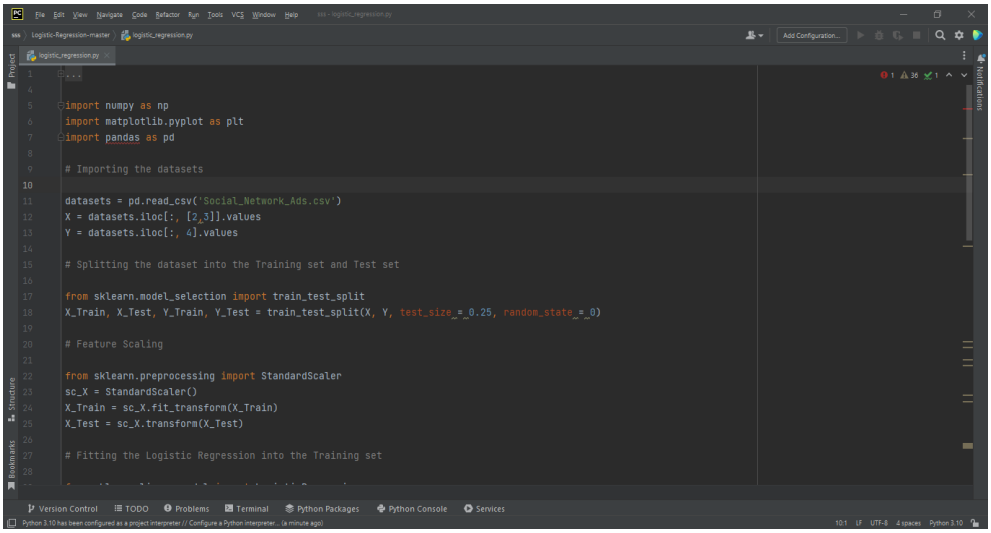


```
# Support Vector Machine
# Importing the libraries

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the datasets

datasets = pd.read_csv('Social_Network_Ads.csv')
X = datasets.iloc[:, [2,3]].values
Y = datasets.iloc[:, 4].values

# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_Train, X_Test, Y_Train, Y_Test = train_test_split(X, Y, test_size = 0.25, random_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_Train = sc_X.fit_transform(X_Train)
X_Test = sc_X.transform(X_Test)

# Fitting the classifier into the Training set
```

Figure 5.5 : Algorithme Machine à vecteurs de support

5.4.5 Régression logistique (LOGISTIC REGRESSION)

- from sklearn.linear_model import LogisticRegression model5 = LogisticRegression(max_iter=1200000)



```
...

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the datasets

datasets = pd.read_csv('Social_Network_Ads.csv')
X = datasets.iloc[:, [2,3]].values
Y = datasets.iloc[:, 4].values

# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_Train, X_Test, Y_Train, Y_Test = train_test_split(X, Y, test_size = 0.25, random_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_Train = sc_X.fit_transform(X_Train)
X_Test = sc_X.transform(X_Test)

# Fitting the Logistic Regression into the Training set
```

Figure 5.6 : Algorithme Régression logistique

5.5 Résultat et Discussion

5.5.1 Les mesures d'évaluation des modèles

La matrice de confusion est utilisée pour mesurer les performances des algorithmes précédents. Ceci fournit une visualisation de la performance du classificateur sur le jeu de données en entrée. Un certain nombre de mesures de performance différentes, y compris le rappel et la précision, sont dérivées de la matrice de confusion. La figure 4.1 montre la structure de cette matrice. Les 4 cas possibles sont :

- **Vrai positif (VP)** : une attaque correctement détectée par le test.
- **Faux positif (FP)** : une activité normale détectée comme attaque par le test.
- **Vrai négatif (VN)** : une activité normale correctement détectée par le test.
- **Faux négatif (FN)** : une attaque détectée comme activité normale par le test.
  - **TP (True Positives)** : les cas où la prédiction est positive, et où la valeur réelle est effectivement positive. Exemple : le médecin vous annonce que vous êtes malade, et vous êtes malade.
  - **TN (True Negatives)** : les cas où la prédiction est négative, et où la valeur réelle est effectivement négative. Exemple : le médecin vous annonce que vous n'êtes pas malade, et vous n'êtes effectivement pas malade.
  - **FP (False Positive)** : les cas où la prédiction est positive, mais où la valeur réelle est négative.

    Exemple : le médecin vous annonce que vous êtes malade, mais vous n'êtes pas malade.
  - **FN (False Negative)** : les cas où la prédiction est négative, mais où la valeur réelle est positive. Exemple : le médecin vous annonce que vous n'êtes pas malade, mais vous êtes malade.

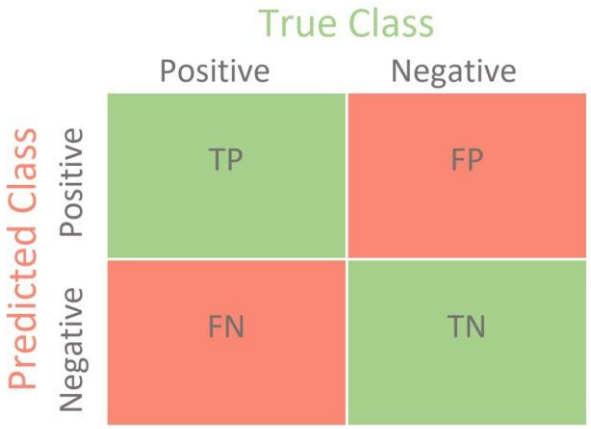


Figure 5.7 : Matrice de confusion.

5.5.1.1 La précision

Cette métrique, également relative à chaque catégorie, renseigne sur la probabilité qu'une prédiction d'une catégorie donnée soit correcte. **Précision** ***= TP/(TP+FP)*100%***

5.5.1.2 Le taux de détection (Rappel)

C'est le rapport entre le nombre d'intrusions correctement détectées et le nombre total d'intrusions. Et décrit par la formule :

Rappel ***=TP/(TP+FN)*100%***

5.5.1.3 Le taux de faux positif (FP)

Le taux des fausses alertes est calculé comme le rapport entre les nombres de trafic normal qui sont incorrectement classés comme intrusions et le nombre total de trafic normal.

FP*=FP/(TN+FP)*100%*

5.5.1.4 Le taux de réussite (Accuracy)

Nous avons évalué cet algorithme en utilisant le taux de réusite comme métrique de performance. L'exactitude, dans ce cas, représente le taux de précision globale de la classification de l'ensemble de données de testes. Elle traduit le rapport entre les détections correctes et les détections totales obtenues. Elle est donné par :

Accuracy ***=(TP+TN)/(TP+TN+FP+FN)*100%***

5.5.1.5 Micro-moyenne (Micro-averaged) :

Agrégera les contributions de toutes les échantillons pour calculer la métrique moyenne. Afin de montrer comment calculer le micro-moyenne, on prend l'exemple de la table suivante :

	TP	FP	FN	Precision	Number of samples
bird	1	0	1	1	2
cat	4	1	0	0.8	4
dog	2	1	1	0.667	3
TOTAL	7	2	2		

Tableau 5.1 – exemple

La formule suivante calcule le micro-moyenne de la précision

$$\text{Micro-averaged Precision} = \frac{TP_{total}}{TP_{total} + FP_{total}} = \frac{7}{7+2} = 0.7777$$

5.5.1.6 Macro-moyenne (Macro-averaged) :

Calcule la métrique indépendamment pour chaque classe, puis prend la moyenne (donc toutes les classes sont traitées de manière égale) voir la formule suivante :

$$\text{Macro-averaged Precision} = \frac{1}{3}Precision_{birds} + Precision_{cats} + Precision_{dogs} = \frac{1}{3}(1 + 0.8 + 0.6666) = 0.8222$$

5.5.1.7 Moyenne pondérée (Weighted-averaged) :

La contribution de chaque classe à la moyenne est pondérée par sa taille (moyenne pondérée est la moyenne d'un certain nombre de valeurs affectées de coefficients). Cette formule permet de calculer la moyenne pondérée de la précision de l'exemple précédent

$$\text{Weighted-averaged Precision} = \frac{Precision_{birds} * N_{birds} + Precision_{cats} * N_{birds} + Precision_{dogs} * N_{birds}}{\text{Total number of samples}}$$

$$= \frac{1 * 2 + 0.8 * 4 + 0.6666 * 3}{2 + 4 + 3} = 0.8$$

5.5.1.8 F1-score :

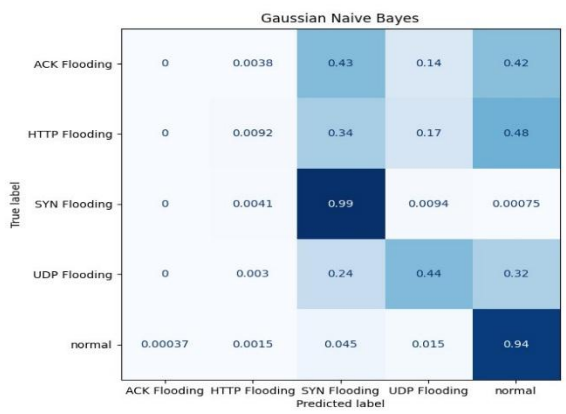
Combine la précision et le rappel en une seule mesure. Mathématiquement, c'est la moyenne harmonique de la précision et du rappel. Il peut être calculé comme suit :

$$F_1\text{−score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

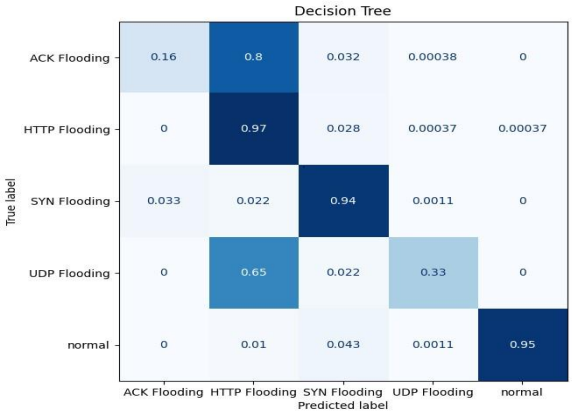
5.5.2 Discussion et analyse

Les modèles de classification naïve bayésienne (NB), Arbre de décision (DT), Forêt d'arbres décisionnels (RF), machine à vecteurs de support (SVM) et régression logistique (LR) ont été testés pour la classification des 4 classes d'attaques avec la classe **normal**. Les tableaux 4.2, 4.3, 4.4, 4.5 et 4.6 donne les résultats obtenus. La meilleure précision obtenue est 82% du modèle arbre de décision. La matrice de confusion normalisée de figure 4.2 ci-dessous montre que la classe **normal** a été bien prédite avec d'autres classes d'attaques qui ont été prédites convenablement. Cependant, certains types d'attaques comme : ACK Flooding et UDP Flooding ont été mal classé, cela signifie que ces attaques ayant des comportements similaires et partagent certaines propriétés entre eux ce qui rendent la tache de reconnaissance de ces derniers plus difficiles.

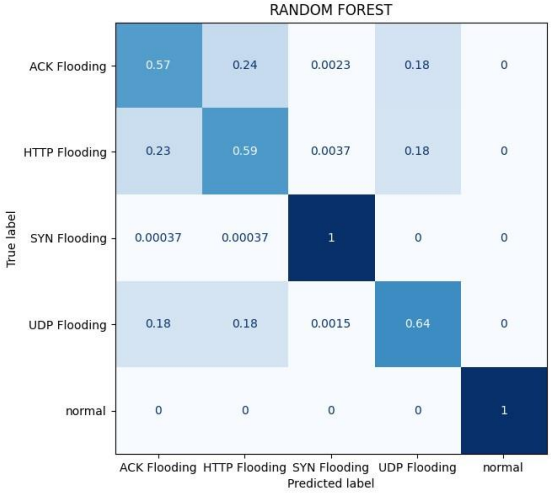
Le taux de réussite (Accuracy) et le temps d'exécution de la phase de teste et d'apprentissage respectivement schématisés dans les figures 4.3 et 4.4 montre que le modèle machine à vecteurs de support a le meilleur taux de réussite et plus grand temps d'exécution.



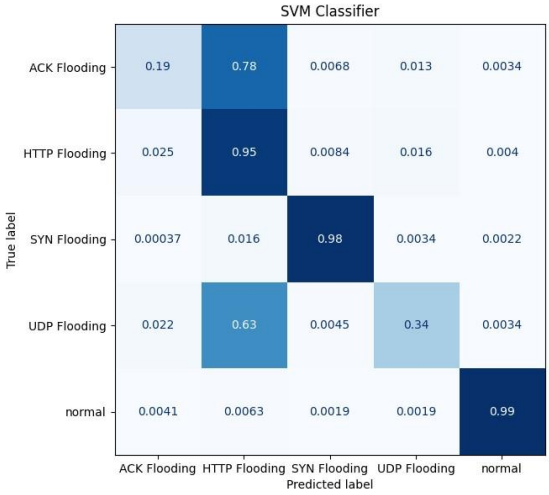
(a)



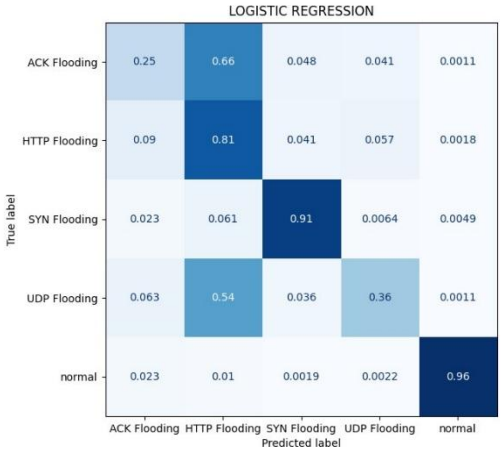
(b)



(c)



(d)



(e)

Figure 5.8 : Matrice de confusion: (a) classification naïve bayésienne (b) Arbre de décision(c) Forêt d'arbres décisionnels (d) Machine à vecteurs de support (d) Régression logistique

	PRECISION	RECALL	F1-SCORE
ACK Flooding	0.00	0.00	0.00
HTTP Flooding	0.43	0.01	0.02
SYN Flooding	0.48	0.99	0.65
UDP Flooding	0.57	0.44	0.49
Normal	0.44	0.94	0.59
Accuracy	/	/	0.47
Macro avg	0.38	0.47	0.35
Weighted avg	0.38	0.47	0.35

Tableau 4.2 : Le rapport de classification avec la classification naïve la classification naïve

	PRECISION	RECALL	F1-SCORE
ACK Flooding	0.83	0.16	0.27
HTTP Flooding	0.40	0.97	0.57
SYN Flooding	0.88	0.94	0.91
UDP Flooding	0.99	0.33	0.49
Normal	1.00	0.95	0.97
Accuracy	/	/	0.67
Macro avg	0.82	0.67	0.64
Weighted avg	0.82	0.67	0.65

Tableau 4.3 : Le rapport de classification avec l'Arbre de décision

	PRECISION	RECALL	F1-SCORE
ACK Flooding	0.58	0.57	0.58
HTTP Flooding	0.59	0.59	0.59
SYN Flooding	0.99	1.00	1.00
UDP Flooding	0.64	0.64	0.64
Normal	1.00	1.00	1.00
Accuracy	/	/	0.76
Macro avg	0.76	0.76	0.76
Weighted avg	0.76	0.76	0.76

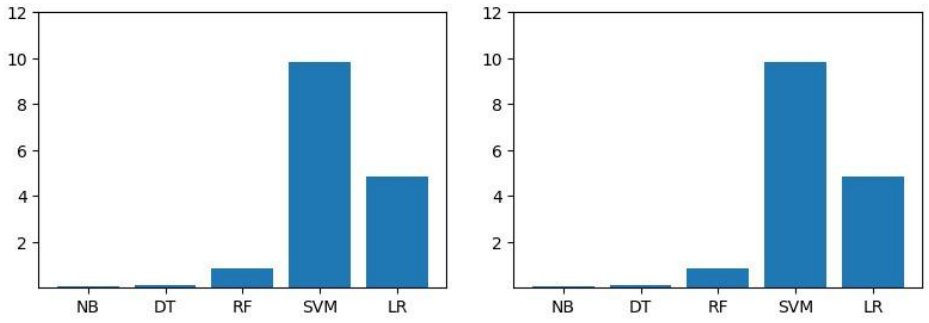
Tableau 4.4 : Le rapport de classification avec la Forêt d'arbres décisionnels

	PRECISION	RECALL	F1-SCORE
ACK Flooding	0.79	0.19	0.31
HTTP Flooding	0.40	0.95	0.57
SYN Flooding	0.98	0.98	0.98
UDP Flooding	0.91	0.34	0.50
Normal	0.99	0.99	0.99
Accuracy	/	/	0.69
Macro avg	0.81	0.69	0.67
Weighted avg	0.81	0.69	0.67

Tableau 4.5 : Le rapport de classification avec la Machine à vecteurs de support

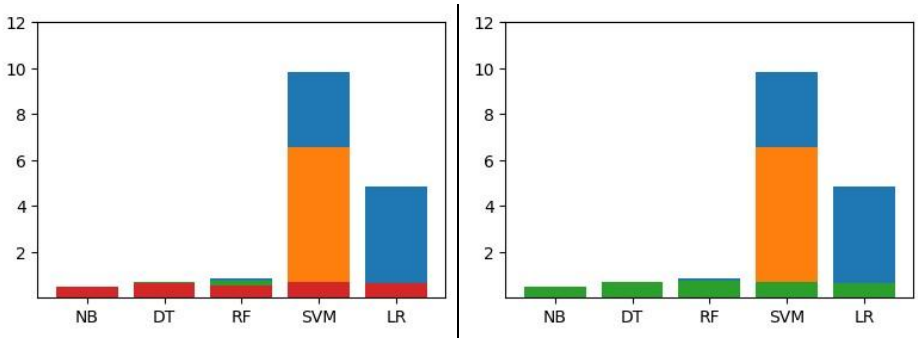
	PRECISION	RECALL	F1-SCORE
ACK Flooding	0.55	0.25	0.34
HTTP Flooding	0.39	0.81	0.53
SYN Flooding	0.88	0.91	0.89
UDP Flooding	0.77	0.36	0.49
Normal	0.99	0.96	0.98
Accuracy	/	/	0.66
Macro avg	0.72	0.66	0.65
Weighted avg	0.72	0.66	0.65

Tableau 4.6 : Le rapport de classification avec la Régression logistique



(a) (b)

Figure 5.9 : Le taux de réussite (Accuracy) de (a) la phase de teste (b) la phase d'apprentissage



(a) (b)

Figure 5.10 : Le temps d'exécution de (a) la phase de teste (b) la phase d'apprentissage.

5.6 Conclusion

Dans ce chapitre, nous avons testé la performance de quelques algorithmes d'apprentissage sur la base IOT-2019 data-set, fourni par laboratoire de recherche sur le piratage et les contremesures (HCR Lab) de Corée du Sud. Le taux de réussite obtenu par l'algorithme machine à vecteurs de support en phase d'apprentissage est le meilleur en aux autres comparant aux algorithmes, d'autre part, son temps d'exécution est le plus grand.

CONCLUSION GENERALE

Conclusion générale

L'apprentissage automatique est utilisé pour identifier les attaques difficiles à représenter par de simples signatures ou pour découvrir des attaques cachées. Pour ce faire, il faut d'abord obtenir des données détectables.

Dans ce travail, nous avons téléchargé la base de données de l'ensemble de données IOT-2019, qui se compose de 42 paquets, pour ouvrir ces paquets, nous avons utilisé Wireshark, qui nous intéresse dans les techniques de détection d'intrusion IoT, nous avons spécifié sur les techniques d'apprentissage automatique et notre approche de ce dernier était par rapport. Cette comparaison est basée sur la précision par rapport à la base de données. Nous avons utilisé des algorithmes : Gaussian Naive Bayes, Decision Tree Algorithm', Random Forest', SVM Classifier', 'Logistic Regression'.

Nous utilisons Python pour implémenter ces algorithmes, le code Python est généralement très facile à lire, et python dispose d'une plateforme PyPi pour les bibliothèques très active (plus de 70 000 bibliothèques répertoriées).

Bien que le temps ait été court, nous avons appris la programmation Python et les techniques de détection d'intrusion, et avons également réussi à utiliser la bibliothèque tkinter pour créer une interface graphique.

Références

Références :

- [1]:Yassine haddab, introduction à l'internet des objets, 2014.
- [2]:Imed Saleh, Internet of things, Laboratoire Paragraphe, Université Paris8, imad.saleh@univ-paris8.fr, 2017.
- [3]:Mekriou Ryma, Mazari Walid, Introduction à l'internet de l'objet et réalisation D'un système domotique, Université de Bejaïa 2016.
- [4]:https://www.fondation-mines-telecom.org/wp-content/uploads/2016/01/2011-Linternethttps://www.fondation-mines-telecom.org/wp-content/uploads/2016/01/2011-Linternet-des-objetsdes-objets.
- [5]:Tarek ABBES, Doctorat de l'université Henri Poincaré - Nancy 1, Laboratoire Lorrain de Recherche en Informatique et ses Applications, le 14 décembre 2004.
- [6]:M. Tran Van Tay, LE SYSTÈME DE DÉTECTION DES INTRUSIONS ET LE SYSTÈME D'EMPÊCHEMENT DES INTRUSIONS (ZERO DAY), Montréal, Février 2005.
- [7]:Melle BEN BRAHIM EMBARKA, Melle AMICHE SELYNA, Mise en place d'une solution de détection d'intrusion,2017.
- [8]:https://connect.ed-diamond.com/MISC/MISC-072/La-detection-d-intrusion-unehttps://connect.ed-diamond.com/MISC/MISC-072/La-detection-d-intrusion-une-approche-globaleapproche-globale
- [9]:https://moodle.insa-rouen.fr/course/view.php?id=92.
- [10]:Eloïse Berthier, comment les machines apprennent, une introduction au machine learning, vendredi 8 mars 2019.
- [11]:Julien Ah-Pine, (julien.ah-pine@univ-lyon2.fr), apprentissage automatique, université lyon 2, 2019/2020
- [12]:Chloé-Agathe Azencott, Introduction au Machine Learning, 2003.
- [13]:Nadia Chaabouni, Intrusion detection and prevention for IoT systems using machine learning, submitted on 29 Sep 2020.