PUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE ECHAHID CHEIKH LARBI TEBESSI - TEBESSA-

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Département d'Informatique

Spécialité : Réseaux et Sécurité Informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de MASTER

Réalisée Par : DJABRI Noussaiba

Thème

La détection des attaques Botnet Dans l'internet des objets (IoT)

Devant le jury :

• *Président*	*GATTAL Abdeldjalil*	*Pr*	*Université de Tébessa*
• *Examinateur*	*ABBAS Faycel*	*MCA*	*Université de Tébessa*
• *Encadreur*	*MERZOUG Soltane*	*MCA*	*Université de Tébessa*

Date de soutenance :
09/06/2024

Année universitaire :
2023-2024

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Remerciment

Tout d'abord, Alhamdulillah qui a rendu cela possible pour moi, De ma donné la volonté et la patience pour mener à terme ma formation de Master et pouvoir réaliser ce travail.

Je tiens à exprimer mes remerciements et mon respect à mon encadreur Docteur **MERZOUG Soltane** pour la qualité de son encadrement, sa disponibilité, et la confiance qu'il ma témoignés tout au long de ce travail et qu'il d'avoir dirigé mes travaux de thème.

Je tiens à gratifier les membres de jury d'avoir accepté d'examiner et d'évaluer mon travail.

Je voudrais exprimer mes sincères remerciements et ma gratitude à **M.Nouar SAADANE**, **M.Salim FAID**, **Mahfoudh** et SH. **Imad** merci pour tout le soutien et les conseils que vous nous avez donnés pendant notre séjour dans **JVGas**. J'ai vécu une expérience extraordinaire et unique, au cours de laquelle j'ai beaucoup appris et acquis des compétences précieuses.

Abderrahim, merci tout particulièrement pour votre aide inlassable et votre soutien constant. Tu as joué un rôle important en facilitant mon travail et en surmontant les obstacles auxquels j'ai été confronté.

Cette expérience restera gravée dans ma mémoire pour toujours, ce sera une incitation pour moi à avancer vers la réalisation de mes objectifs.

A la fin, je veux me remercier de ne pas avoir abandonné et d'être resté forte.

Dédicace

Je dédie ce modeste travail

A ma très chère mère,

A mon cher père,

A ma Sœur Amina,

A mon frère Khaled,

A ma grand-mère,

A mes oncles et mes tantes,

Ils ont toujours été présents, ils m'ont donné le soutien, l'amour et le réconfort pour mon continuer mon voyage jusqu'au bout,

A Meriem et Rayen,

A toutes mes amies,

A mes amis que je n'ai pas vus et qui m'ont soutenu Hama-Mulay Abdallah, Adnan et Abdelbasset,

A Abderrahim, Houssam, Belkhir pour leur hospitalité et le bon traitement,

J'avais l'impression d'ils connaitre depuis de nombreuses années, ce qui a rendu mon expérience dans JVGas encore plus agréable et enrichissante,

A toute l'équipe de JVGas.

Résumé

L'Internet des objets (IoT) révolutionne notre mode de vie en connectant des appareils intelligents au Web. Cependant, cette connectivité accrue s'accompagne de risques pour la sécurité, notamment les attaques botnets. Les botnets IoT peuvent prendre le contrôle d'appareils compromis et les utiliser pour lancer des attaques Distriuted Denial of service (DDoS) ou voler des données.

Ce mémoire propose une nouvelle approche pour détecter les attaques botnet dans les environnements IoT a une étape précoce. L'approche s'appuie sur un modèle hybride combinant Auto-Ecoder (AE) et Gated Recurrent Unit (GRU), permet d'extraire des caractéristiques pertinentes du trafic réseau, tandis que le GRU capture les dépendances temporelles entre les paquets.

Le modèle proposé est évalué sur un ensemble de données réel de trafic IoT, MedBIoT, qui contient des traces de trafic provenant de 83 appareils. Les résultats montrent que le modèle hybride AE-GRU surpasse les méthodes existantes en termes de précision, de rappel et de F1-score.

Mots-clés :

Internet des objets, sécurité IoT, attaques botnet, détection d'anomalies, apprentissage en profondeur.

Abstract

The Internet of Things (IoT) is revolutionizing our way of life by connecting smart devices to the web. However, this increased connectivity comes with security risks, including botnet attacks. IoT botnet can take control of compromised devices and use them to launch distributed denial-of-service (DDoS) attacks or steal data.

This thesis proposes a novel approach to detect botnet attacks in IoT environments. The approach is based on a hybrid model that combines an autoencoder (AE) and a gated recurrent unit (GRU) neural network. The AE is used to extract relevant features from network traffic, while the GRU captures the temporal dependencies between packets.

The proposed model is evaluated on a real IoT traffic dataset, MedBIoT, which contains traffic traces from 83 devices. The results show that the hybrid AE-GRU model outperforms existing methods in terms of accuracy, recall, and F1 score.

Keywords:

Internet of Things, IoT security, botnet attacks, anomaly detection, Deep learning.

ملخص

تقوم ثورة إنترنت الأشياء (IoT) بإحداث ثورة في طريقة حياتنا من خلال ربط الأجهزة الذكية بالويب. ومع ذلك، فإن هذا الاتصال المتزايد مصحوب بمخاطر أمنية، بما في ذلك هجمات بوت نت يمكن لبوت نت IoT السيطرة على الأجهزة المخترقة واستخدامها لشن هجمات رفض الخدمة الموزع (DDoS) أو سرقة البيانات.

تقترح هذه المذكرة طريقة جديدة لاكتشاف هجمات بوت نت في بيئات IoT, تعتمد هذه الطريقة على نموذج هجين يجمع بين مُشفِّر ذاتي (AE) وشبكة عصبية اصطناعية ذات وحدات ذاكرة بوابة متكررة (GRU). يتم استخدام المُشفِّر ذاتي لاستخراج الميزات ذات الصلة من حركة مرور الشبكة، بينما يلتقط GRU التبعيات الزمنية بين الحزم.

يتم تقييم النموذج المقترح على مجموعة بيانات حركة مرور IoT، MedBIoT، والتي تحتوي على مسارات حركة مرور من 83 جهازًا تُظهر النتائج أن النموذج الهجين AE-GRU يتفوق على الأساليب الموجودة من حيث الدقة والاستدعاء ومقياس F1.

الكلمات المفتاحية:

إنترنت الأشياء، أمان IoT، هجمات بوت نت، اكتشاف حركة مرور ضارة، التعلم العميق.

Table de matière

Liste des figures

Liste des tableaux

Liste des abréviations

IoT : Internet of Thing

DoS : Denial of service

DDoS : Distributed Denial of Service

RFID : Radio Frequency Identification

ZigBee : Zonal intercommunication Global-standard

LAN : Local Area Network

LTE : Long Term Evolution

C&C : Command and control server

P2P: Peer to Peer

SHAP: SHapley Additive exPlanations

BHO-RF: black hole optimized random forest

SGAN: Semi-supervised GAN

LGANet: Local Graph Attention Network

DNN: Dynamic Neural Network

CNN: convolutional Neural Network

RNN: Recurrent Neural Network

LSTM: Long Short-Term-Memory

RL: Reinforcement Learning

GRU: Gated Recurrent Unit

AE: Auto- Encoder

TCP: Transmission Control Protocol

IDS: Intrusion Detection System

PCAP: Packet Capture

IP: Internet Protocol

Introduction Générale

Introduction

Nous vivons à l'ère d'Internet, où les objets du monde réel sont devenus plus intelligents, plus sophistiqués et capables de communiquer avec les autres sans avoir besoin de communication humaine pour rendre l'existence humaine plus facile et beaucoup plus confortable. L'IoT est un paradigme de communication dans lequel le vaste spectre des objets du quotidien est interconnecté sur le Web. Les appareils intelligents sont entrés dans nos moyens de subsistance en raison de la croissance de l'IoT. De nombreuses utilisations inventives des technologies modernes, y compris "la maison intelligente, le bureau intelligent, le réseau intelligent, les soins de santé intelligents, l'agriculture intelligente, les transports intelligents, la ville intelligente", etc, ont en effet été transformées par l'IoT. De telles technologies auront également un impact significatif sur la vie des gens, et les humains deviendraient de plus en plus dépendants de ces technologies.

Le manque de sécurité et de normes efficaces pour les appareils IoT a conduit à l'émergence de vulnérabilités que les cybercriminels peuvent exploiter dans divers types d'attaques. L'un des principaux scénarios d'attaque est un attaquant qui expose les appareils IoT à un risque d'utilisation dans le cadre de botnets IoT. Une fois qu'un appareil IoT est infecté et compromis, l'attaquant prend le contrôle de l'appareil infecté. Cette dernière étape intervient après l'achèvement du processus de prise de contrôle par l'attaquant d'autant d'appareils IoT que possible. De cette façon, l'attaquant crée et étend son propre botnet IoT.

Le nombre d'appareils de l'IoT continuera de croître dans le monde entier dans les années à venir. Cette croissance des appareils connectés entraînera une augmentation des surfaces d'attaque et, plus important encore, une augmentation du nombre de botnets existants. Lorsque les appareils IoT sont compromis, les attaquants peuvent utiliser ces appareils comme points de lancement et d'entrée dans de plus grands réseaux d'entreprise ou personnels. De plus, ils peuvent être réorganisés en botnets pour lancer des attaques DDoS sur les réseaux publics et privés. Par exemple, le botnet Mirai a infecté avec succès 2,5 millions d'appareils au cours du dernier trimestre de 2016.

Le domaine de la recherche sur la détection des iiiiii est un domaine vaste pour sécuriser les appareils IoT. Les technologies actuelles de pare-feu sont assez matures ; cependant, ils sont insuffisants pour les environnements IoT en raison des modèles de trafic polyvalents, des protocoles de communication, etc. Par conséquent, il est urgent de développer une solution et de l'intégrer à l'infrastructure de sécurité existante pour mieux protéger les outils IoT contre les attaques de botnet.

Problématique

Étant donné que les attaques Botnets automatisés mènent leurs activités en deux étapes principales, l'étape précoce et l'étape tardive, la plupart des études précédentes, dont nous parlerons des plus importantes, se sont concentrées sur la détection de ces attaques à une étape avancée, qui se produisent rapidement, alors qu'il est logique de se concentrer sur les premières étapes, car les bots se forment et se propagent sur une longue période de temps. C'est le principal problème de ce travail, et notre objectif était de répondre aux questions suivantes :

- *Comment peut-on garantir la sécurité dans les environnements IoT ?*
- *Quelles sont les différentes phases de la formation d'un botnet IoT ?*
- *Quelles méthodes et techniques sont utilisées pour détecter les botnets IoT ?*
- *Comment peut-on améliorer la détection des attaques botnets dans l'IoT pour une efficacité accrue ?*

Objective

L'objectif principal de ce travail est de sécuriser et de protéger les réseaux IoT contre diverses activités malveillantes des botnets. La plupart des mécanismes de détection de botnets dans les réseaux IoT proposés dans la littérature abordent généralement la détection de ces attaques à un stade avancé, et ne reposent que sur des réseaux simulés. Par conséquent, le système de détection de botnet a été testé sur le trafic réel de l'ensemble de données MedBIoT qui est configuré avec 83 appareils et s'est concentré uniquement sur les étapes initiales des attaques.

Plan de du mémoire

Ce mémoire est structuré en quatre chapitres, chacun couvrant divers aspects de notre étude sur la détection des attaques botnets dans les environnements IoT.

- *Chapitre 01 : Introduction aux environnements IoT et les attaques Botnets.*
- *Chapitre 02 : Etat de l'art des méthodes et techniques utilisées pour la détection des attaques botnets dans l'IOT.*
- *Chapitre 03 : Proposition d'un modèle hybride AE-GRU pour la détection des attaques Botnet dans l'IOT.*
- *Chapitre 04 : Présentation de l'environnement et Simulation*

Chapitre 01

Chapitre 01

Introduction aux environnements IoT et les attaques Botnets

1. Introduction

L'Internet des objets (IoT) représente une convergence technologique où divers objets physiques sont connectés à Internet, permettant ainsi l'échange d'informations entre ces objets. Au cours des dernières années, l'IOT est devenu la nouvelle génération d'Internet la plus populaire et l'une des tendances technologiques les plus répandues du XXIe siècle. Toutefois, malgré cette popularité, les préoccupations liées aux vulnérabilités de ces dispositifs face à des menaces telles que les Botnets suscitent une attention croissante.

Ce chapitre est introductif dédié aux environnements IOT et aux attaques Botnets qui ouvre la voie au sujet de la détection des attaques Botnets dans l'IOT. On commence par présenter les concepts fondamentaux des environnements de l'IoT, puis aborde les attaques de botnets.

2. Les environnements IOT

Dans cette section on a expliqué les concepts de l'IoT, la croissance de l'IOT, ses architectures, ainsi que la sécurité et les types d'attaques dans ses environnements.

2.1. Historique :

L'Internet des objets (IoT) trouve ses racines dès 1982 avec la première connexion Internet établie par une machine à soda modifiée à l'Université Carnegie Mellon. En 1999, le terme "Internet des objets" est officiellement introduit par Kevin Ashton d'Auto-ID Labs au MIT, marquant le début de la popularité de l'IoT. Les années suivantes voient une croissance significative avec des déploiements massifs de technologies comme le RFID [1] , l'adoption par des entreprises majeures, et le déclenchement d'une expansion majeure avec le lancement de l'IPv6 en 2008 [2], [3].

2.2. Définition :

L'Internet des objets (IoT) est un réseau qui interconnecte des dispositifs physiques au cyberespace, incluant des objets tels que sonnettes intelligentes, ampoules connectées, caméras de surveillance, thermostats, dispositifs médicaux intelligents et systèmes de contrôle du trafic [4]. Cette interconnexion permet l'automatisation de divers appareils, fonctionnant comme un système de communication interconnecté via des technologies de communication sans fil ou filaire, telles que Zigbee et Bluetooth. Cette infrastructure IoT comprend également des composants tels que des passerelles, des capteurs, des systèmes de stockage de données, des services cloud, des utilisateurs et des dispositifs IoT eux-mêmes [5] comme montre la figure suivante:

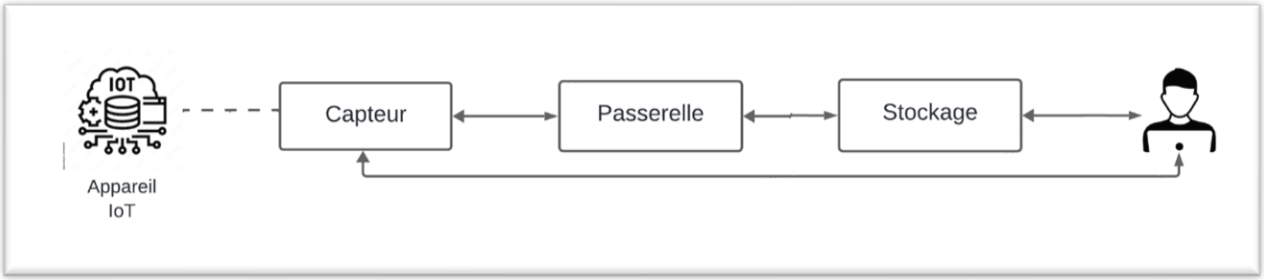


Figure 1.1. L'infrastructure de l'IOT [5]

2.3. Architecture de l'IOT :

Il existe différentes architectures IoT proposées. La vision partagée par la plupart des chercheurs suggère que l'architecture de l'IoT se compose généralement de trois couches [2][6], [7], [8], [9], [10], [11] :

- Couche de perception : couche de dispositifs physiques et de communication composée de capteurs et d'actionneurs qui agrègent, détectent et traitent les données, puis transmettent les données à la couche réseau. Cette couche contient des objets physiques tels que des caméras, des RFID, etc [9].
- Couche réseau et transport : transmet et achemine les données agrégées de la couche perception vers la couche application à l'aide de différents périphériques, tels que des passerelles, des commutateurs et des routeurs [11].
- Couche d'application : est une couche de messagerie contenant l'application qui interagit avec les utilisateurs. Les domaines tels que la santé électronique et le transport intelligent utilisent tous de telles applications [11].

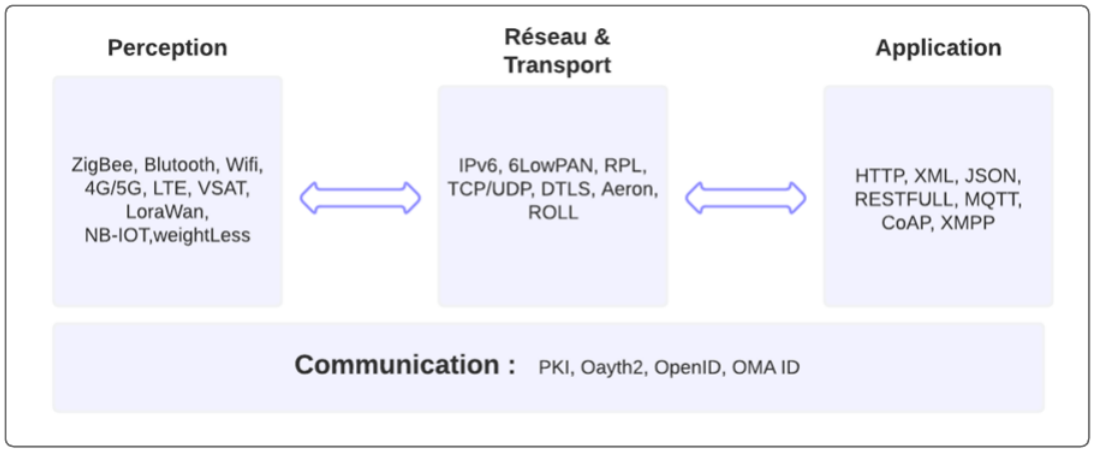


Figure 1.2. Les protocoles et les normes de l'IoT [11]

Bien que l'architecture à trois couches définisse l'idée principale de l'IoT, elle n'est pas suffisante pour la recherche sur l'IoT, car la recherche se concentre souvent sur les aspects plus fins de l'IoT. Par conséquent, une architecture à cinq couches [1], [2], [6], [8], [12] est définie :

- *Couche de perception : fonctionne de la même manière que précédemment décrit dans l'architecture à 3 couches. Il est utilisé pour prendre des informations des capteurs et les mettre en œuvre [2].*
- *Couche de transport : prend les données de la perception couche et passez ces données à la couche suivante qui est couche de traitement et vice versa. Cela se fera avec l'aide de réseaux tels que LAN, technologie sans fil, 3G, 4G, LTE, RFID, etc [6].*
- *Couche de traitement : doit effectuer la tâche principale car elle traitera toutes les informations recueillies par la couche de perception. Il y a une énorme quantité de données qui seront stockées à l'aide de certaines techniques comme le cloud computing ou n'importe quel SGBD. Ensuite, il analysera comment récupérer les données chaque fois que nécessaire pour terminer la tâche souhaitée [8].*
- *Couche d'application : implémente le fonctionnement de l'IoT. Pour cela, une application est requise avec l'appareil correspondant afin d'accomplir la tâche souhaitée [2].*
- *Couche de Métier : gère le fonctionnement de l'ensemble du système avec de nombreuses autres fonctionnalités, l'une d'elles est la confidentialité [8].*

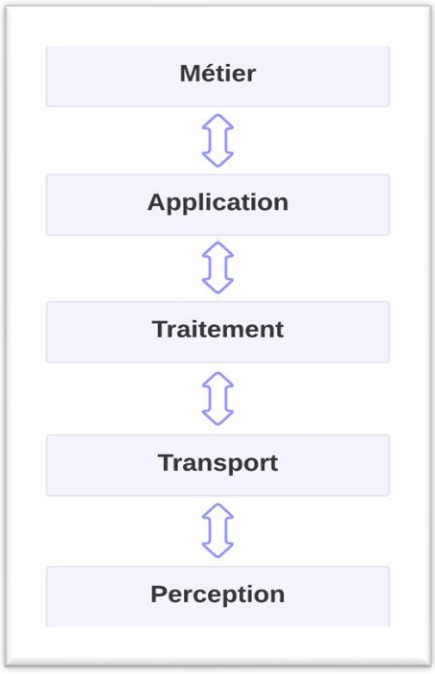


Figure 1.3. L'architecture de l'IOT de cinq couches [8]

2.4. La sécurité dans l'IoT :

- *Confidentialité : garantit que seuls les utilisateurs autorisés peuvent accéder aux données, prévenant ainsi toute interférence ou suppression non autorisée. Ce principe revêt une importance cruciale en raison de l'intégration de divers dispositifs de mesure, tels que les RFID et les capteurs [9].*
- *Intégrité : Garantir l'intégrité des données dans l'IoT est essentiel pour maintenir l'exactitude et prévenir la manipulation. Cela implique l'utilisation de diverses approches de sécurité comme l'authentification et la cryptographie pour contrer les attaques potentielles et surmonter les limitations des ressources [10], [13].*
- *Accessibilité : est cruciale pour répondre à la demande de services en temps réel. Les attaques de type déni de service (DoS) sont des menaces sérieuses, nécessitant l'étude et l'application de techniques avancées telles que des protocoles de routage sécurisés pour garantir l'accessibilité de l'IoT [8].*
- *Identification & Authentification : assurer que seuls les dispositifs ou applications autorisés peuvent se connecter à l'IoT, tandis que l'authentification garantit que les données transmises dans les réseaux sont légitimes et que les dispositifs ou applications demandant les données sont également légitimes [14].*
- *La vie privée des données : La vie privée des données dans l'IoT garantit le contrôle exclusif de l'utilisateur sur ses données, limitant l'accès et le traitement par d'autres. Cela est essentiel pour prévenir les déductions d'informations sensibles dans un environnement où de multiples appareils et utilisateurs partagent le même réseau de communication [8], [13].*
- *La sécurité physique des périphériques IoT : est cruciale pour prévenir le vol, la destruction physique et les attaques malveillantes. Cela implique la mise en place de mesures de sécurité robustes, telles que la résistance aux conditions environnementales, la prévention des accès non autorisés, et la défense contre les attaques par injection de code malveillant[15].*

2.5. Les types d'attaques dans l'IOT :

Les attaques contre l'IOT comprennent une gamme de menaces, notamment les attaques par déni de service distribués (DDoS), où des appareils compromis submergent un système de demandes [16], les attaques par phishing qui visent à obtenir des informations personnelles via

des e-mails frauduleux [13][17], le spoofing où les attaquants usurpent des identités pour accéder au système [8], les attaques de wormhole qui manipulent les données de routage [18], le sniffing où des données réseau sont interceptées, et les virus/malwares qui infectent les applications IOT pour voler ou altérer des données [8].

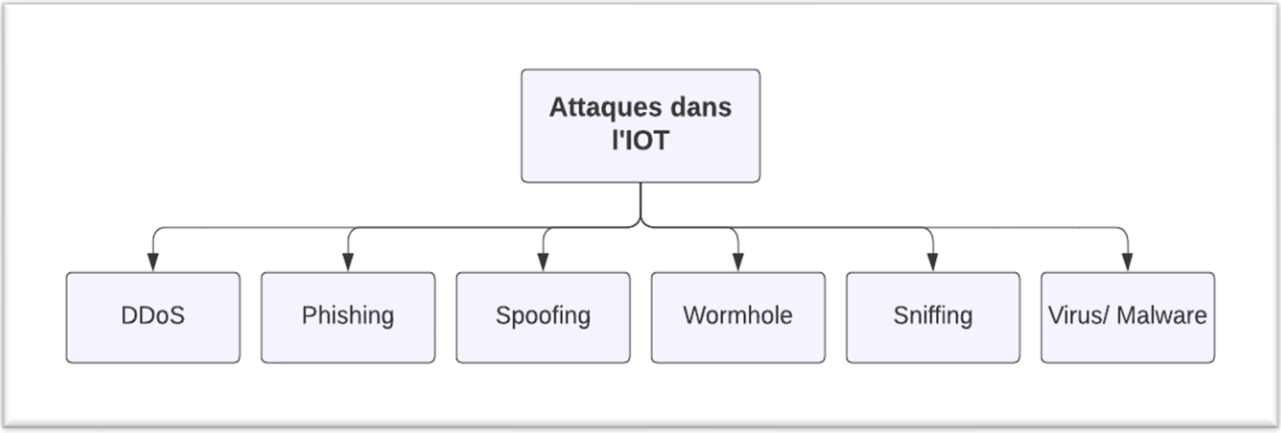


Figure 1.4. Les principales attaques dans l'IOT [9]

3. Les attaques Botnets

L'un des objectifs principaux des cybercriminels lorsqu'ils compromettent des appareils IoT est de les intégrer dans un botnet. Dans cette section, nous exposons les fondements de ce type d'attaques, l'évolution, l'architecture, phases de cycle de vie et quelques exemples des attaques botnets.

3.1. Définition du malware :

Programme conçu pour causer des dommages à un système informatique sans le consentement de l'utilisateur infecté. Les pirates utilisent diverses méthodes telles que le phishing par e-mail ou le téléchargement automatique de fichiers pour infecter les ordinateurs. Le terme "virus" est souvent utilisé à tort pour désigner tous les types de logiciels malveillants, y compris les virus, les vers et les chevaux de Troie [16].

3.2. Définition des Botnets :

Des réseaux de dispositifs connectés à Internet, tels que des caméras de surveillance, des thermostats intelligents ou d'autres appareils IoT, qui ont été compromis par des logiciels

malveillants. Ces dispositifs infectés agissent comme des bots contrôlés à distance par un Botmaster. Les botnets IoT sont souvent utilisés pour des activités malveillantes telles que les attaques DDoS [17].

3.3. Composition et Structure de Botnets :

Les botnets IoT se composent principalement de quatre parties : le Bot, le Serveur C&C, les serveurs de chargement et de Rapport, la figure [5] présente ces composants [18].

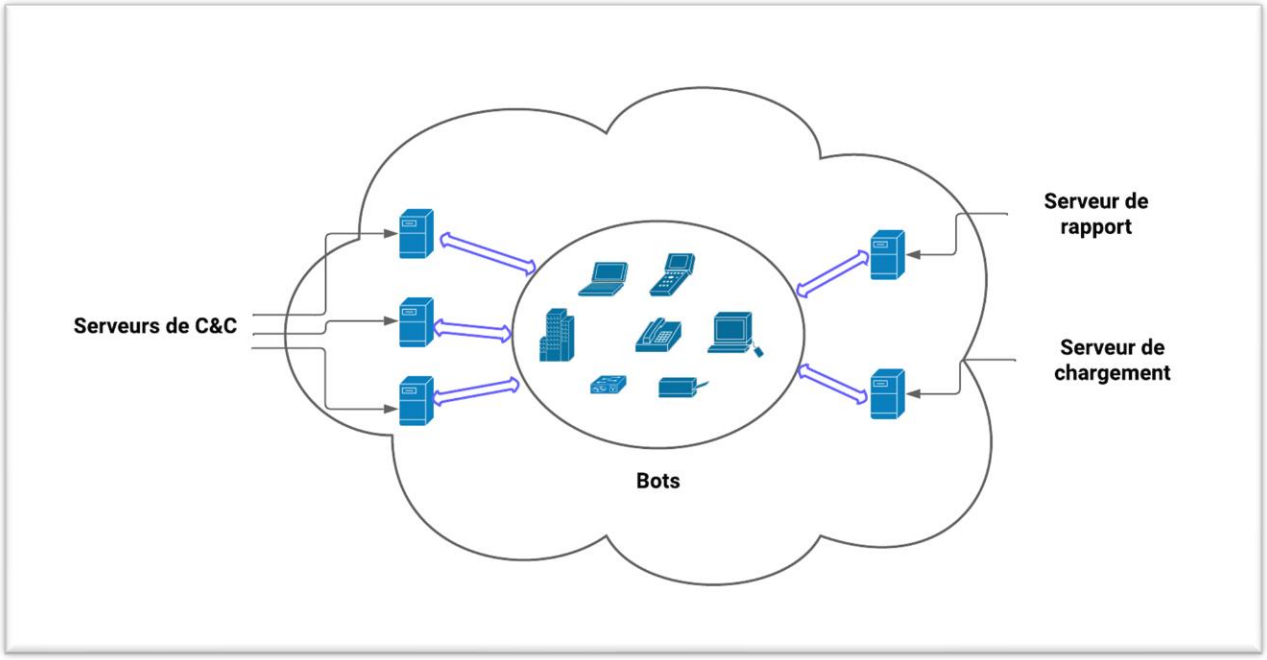


Figure 1.5. Composition de Botnets IOT [18]

3.4. Architecture de Botnets :

La variété des composants au sein de la structure, de la capacité et de la mise en œuvre technique des Botnets les distingue presque comme étant uniques, mais en général les architectures du botnet IoT sont généralement divisés en trois parties, Centralisés, Décentralisés et Hybride :

Botnets Centralisés : Le Botmaster gère et surveille l'ensemble des bots à partir d'un serveur central unifié, ce qui réduit la latence ; c'est-à-dire que tous les bots reçoivent des instructions et font des rapports à un serveur command and control (C&C). Le serveur utilise des protocoles tels que HTTP et IRC. L'une des familles de botnets IoT centralisés les plus connues

est la famille Mirai [18], qui est un malware qui infecte les appareils intelligents qui fonctionnent sur des processeurs IRC, les transformant en un réseau de robots ou de zombies contrôlés à distance.

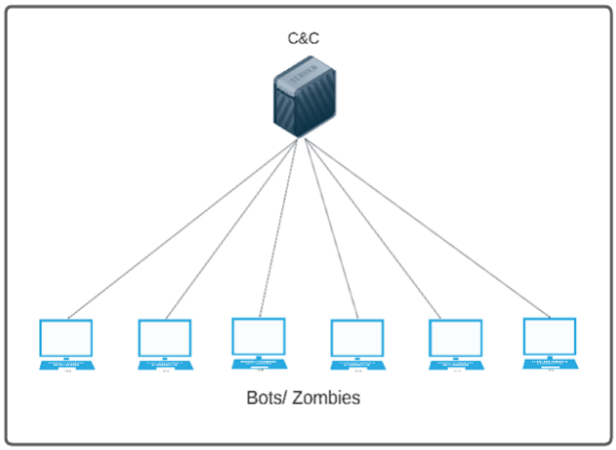


Figure 1.6. Les botnets centralisés [19]

- *Botnets Décentralisés : également appelés botnets P2P, où chaque bot fonctionne à la fois en tant que client et serveur. Chaque bot est connecté à au moins un autre bot, et les commandes ne parviennent à chaque bot que si tous les bots sont interconnectés. Dans cette architecture, la coordination entre les bots peut être difficile, mais elle est plus sophistiquée et moins facile à détecter en raison des communications différentes entre pairs. Cette architecture de botnet IoT utilise un protocole P2P pour la communication. Un exemple bien connu de botnet IoT décentralisé est Hajime [18].*

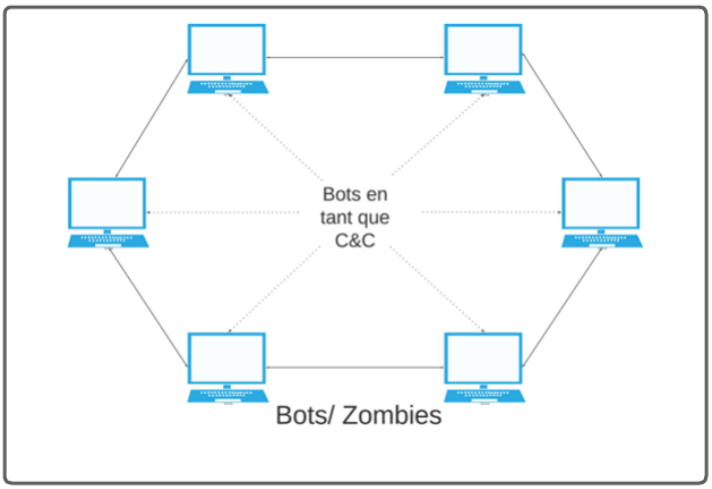


Figure 1.7. Les botnets décentralisés [19]

• *Botnets Hybride : combinent les avantages des architectures centralisées et décentralisées. Ils présentent une structure de C&C décentralisée, mais les bots se connectent aux ces serveurs selon le modèle client-serveur classique. Cette approche réduit considérablement le risque d'identification et d'élimination de l'ensemble du botnet. En cas de disqualification d'un serveur C&C, seuls les bots contrôlés par ce serveur sont impactés [20].*

3.5. Phases de cycle de vie de Botnets :

Précédemment, on a abordé les quatre principaux composants des attaques botnets. Dans cette sous section, on discute l'infection et la prolifération des Botnets IOT [11], [19], [21] telles qu'elles sont présentées dans la figure [8] :

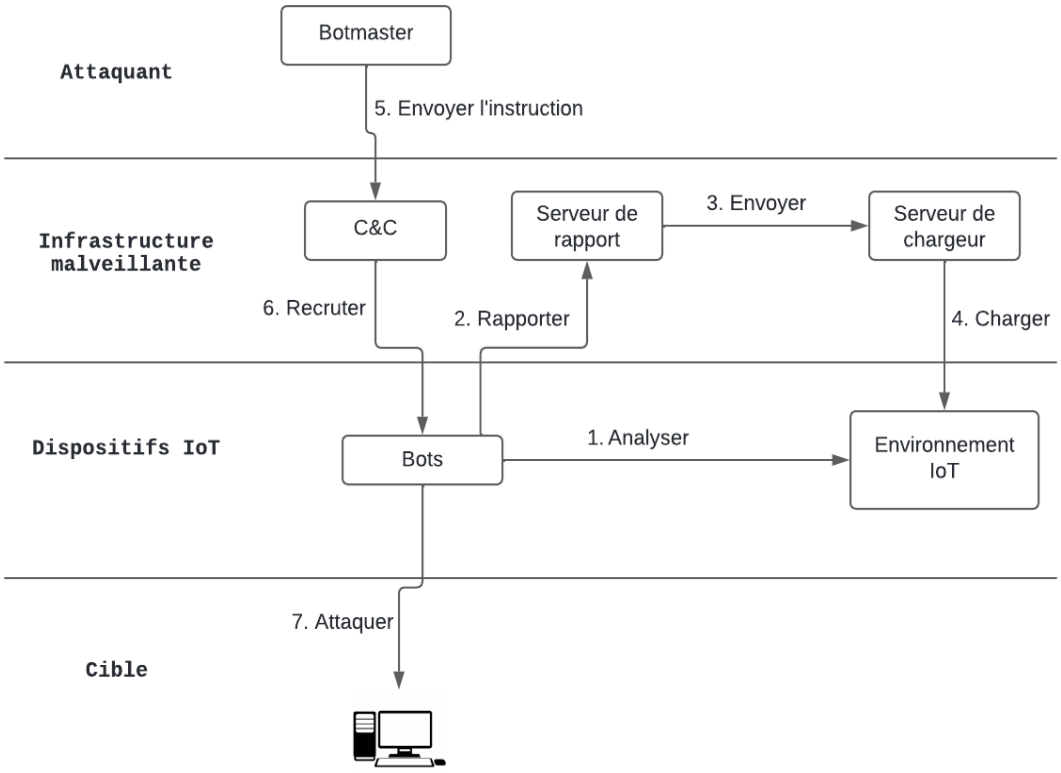


Figure 1.8. Infection et prolifération des Botnets IOT [11]

• *Les bots recherchent dans l'espace d'adresses IP des périphériques exécutant Telnet ou SSH et tentent de se connecter à l'aide d'un dictionnaire d'informations d'identification IoT*

• *En cas de succès, un bot rapporte l'adresse IP de la cible et les identifiants associés à un serveur de rapport. Le serveur de rapport envoie ces informations au serveur de chargement.*

- *Le serveur de chargement force simultanément l'infection du dispositif d'une manière dépendante de l'architecture de la cible.*
- *L'attaquant envoie une commande au serveur C&C spécifiant la cible et les détails requis pour lancer l'attaque.*
- *Le serveur C&C informe les bots en préparation de l'attaque.*
- *Les bots déclenchent l'attaque contre la cible spécifiée.*

3.6. Les menaces des attaques Botnets :

Dans le domaine complexe de la cybersécurité, les attaques Botnets exploitent différentes vulnérabilités pour compromettre la sécurité des dispositifs et des réseaux IOT. Les menaces d'attaques de botnet comprennent :

- *Attaque par courrier électronique : Les attaques d'ingénierie sociale par courrier électronique incitent l'utilisateur à ouvrir une pièce jointe ou à suivre un lien non sollicité, infectant directement le système avec des logiciels malveillants [18].*
- *Attaque sur le client Web : consiste à propager des logiciels malveillants en attirant la victime vers des sites hébergés sur des systèmes contrôlés par l'attaquant. En cas de réussite, le logiciel malveillant s'installe discrètement sans que l'utilisateur en ait connaissance [18].*
- *Attaques par messagerie instantanée : impliquent l'envoi de messages non sollicités à partir d'un compte compromis de messagerie instantanée de l'utilisateur. Bien que ces messages semblent légitimes, ils dirigent l'utilisateur vers des sites malveillants ou déclenchent le téléchargement et l'installation de fichiers malveillants [18].*
- *Fraude au clic : La fraude au clic se produit lorsque des visites sont effectuées sur une publicité en ligne ou une autre ressource facturée au sponsor sur une base par clic, de manière illégitime. Les bots sont couramment utilisés pour exécuter la fraude au clic en envoyant des demandes Web représentant des "clics" sur les publicités Internet de certains affiliés [18].*
- *Enregistrement des touches (Key-logging) : Les enregistreurs de frappe logiciels capturent les événements du clavier et enregistrent les données de frappe avant qu'elles ne soient envoyées à l'application prévue pour le traitement [18].*

3.7. Exemples des Botnets dans l'IOT :

Le développement des botnets a évolué au fil du temps, passant d'une simple machine à une distribution à l'échelle du réseau. Certains des botnets IoT remarquables identifiés sont répertoriés dans le tableau 1 où N / A indiqué que l'information particulière n'a pas été fournie par ses auteurs :

Botnet	Description	Nombre estimé d'appareils	Année
Linux/ Hydra [22]	Considéré comme le premier botnet ciblant les appareils IoT, avec un mécanisme de propagation.	N/A	2008
Psybot [23]	En utilisant le dynamitage des noms d'utilisateur et des mots de passe, il a contrôlé avec succès environ 100 000 routeurs et modems DSL, qui ont été fermés par le concepteur.	100 000	2009
Bashlight [24]	Basé sur IRC ciblant les appareils IoT basés sur Linux (BusyBox) comme les caméras et les DVR. Il exécute la force brute avec des informations d'identification par défaut avec un port TelNet ouvert.	120,000	2014/ 2016
Miraï [25]	Centralisée ciblant les caméras de télévision en circuit fermé, les routeurs et les enregistreurs vidéo numériques (DVR). Il utilise un port TelNet et des vecteurs d'attaque prédéfinis Attaque par dictionnaire basée sur 62 entrées.	145,000+	2016
Brickerbot [26]	Forcer brutalement le mot de passe TelNet puis exécutez la commande pour corrompre le stockage, supprimer tous les fichiers et rendre le périphérique inutilisable.	10,000,000+	2017

Hajime [27]	*Identique à Mirai, mais il a un modèle d'architecture P2P. Récemment, Hajime a évolué pour utiliser différents ports et différents exploits. Jusqu'à présent, il se contentait de scanner et d'infecter les appareils vulnérables mais ne lançait aucune attaque DDoS*	*300 000*	*2016/ 2018*
Wirex [28]	*Travailler sur des appareils Android et proliférer grâce à l'application dans Google Play.*	*100,000*	*2017*
Reaper [29]	*Exploiter les vulnérabilités des appareils IoT tels que les routeurs de LinkSys, DLink et les caméras connectées.*	*1,000,000*	*2018*
Torii [30]	*Architecture modulaire, fonctions plus riches, peut collecter des informations, obtenir des commandes d'exécution grâce à une communication cryptée multicouche*	*N/A*	*2018*
Mozi [31]	*Une combinaison d'au moins trois codes malveillants (Gafgyt, Mirai, IoT Reaper), et construire un P2P réseau de zombies*	*N/A*	*2019*

Tableau 1.1. Exemples des attaques botnets

3.8. Les attaques Déni de service distribué :

- *Définition : Attaque coordonnée exploitant de nombreux hôtes compromis pour perturber un réseau. L'attaquant identifie les vulnérabilités, compromet des machines pour les contrôler, puis les utilise pour envoyer des paquets d'attaque à la victime sans que ces machines en soient conscientes. Cela peut causer des dommages considérables à la victime en fonction de l'intensité des attaques et du nombre d'hôtes impliqués [32][33].*
- *Processus d'Attaque DDoS :*

L'objectif d'un attaquant DDoS est de perturber un réseau afin qu'il ne puisse fournir aucun service aux utilisateurs légitimes. Pour lancer une attaque, un attaquant suit généralement quatre étapes de base :

i. *Collecte d'informations pour scanner un réseau afin de trouver des hôtes vulnérables à utiliser ultérieurement pour lancer une attaque.*
ii. *Compromission des hôtes pour installer des logiciels malveillants dans les hôtes compromis (appelés zombies) afin qu'ils ne puissent être contrôlés que par l'attaquant.*
iii. *Lancement de l'attaque pour commander aux zombies d'envoyer des paquets d'attaque avec des intensités spécifiées à la victime, et nettoyage pour supprimer tous les enregistrements ou fichiers d'historique de la mémoire [5].*

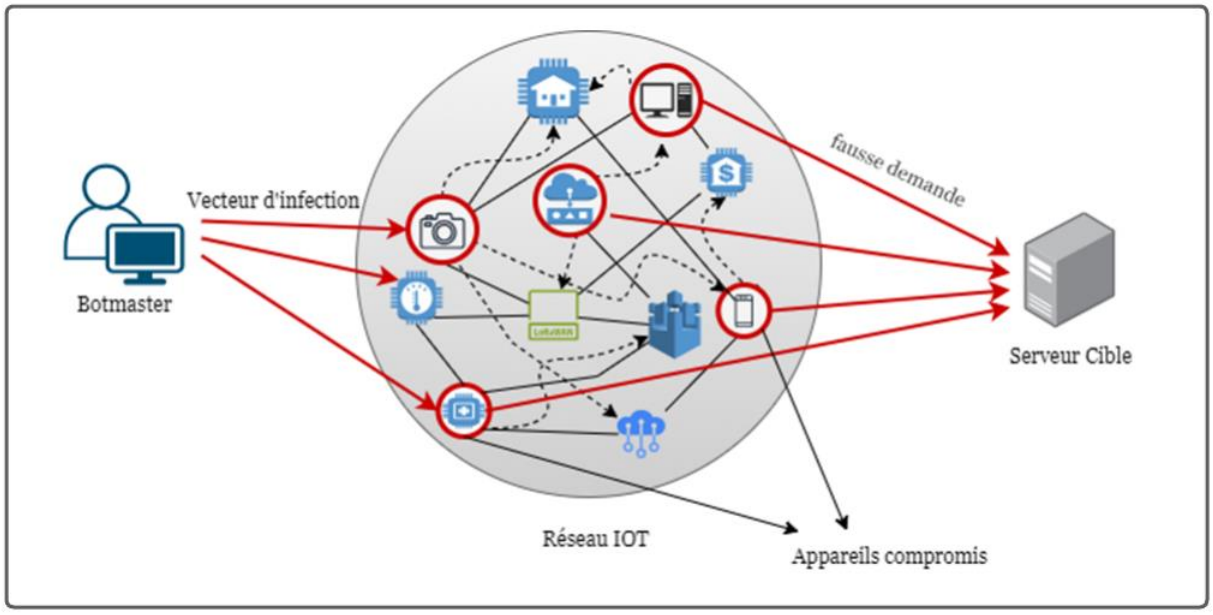


Figure 1.9. Mécanisme de DDoS [5]

4. Conclusion

En Conclusion, ce chapitre met en lumière la complexité croissante des environnements IoT, caractérisés par la prolifération rapide de dispositifs IOT. Alors que cette évolution offre des opportunités innovantes, elle expose également des vulnérabilités majeures. Les attaques de botnets, en particulier, sont identifiées comme une menace sérieuse, pouvant avoir des conséquences dévastatrices sur la sécurité des dispositifs IoT et des réseaux associés

Comprendre les mécanismes des attaques Botnets, ainsi que les différentes techniques utilisées, devient impératif pour élaborer des stratégies de défense et de la détection.

Dans le deuxième chapitre, on présentera les méthodes les plus importantes utilisées pour détecter ces attaques en identifiant les études connexes et en travaillant à les comparer.

Chapitre 02

Chapitre 02

État de l'art des techniques de détection des attaques botnets

1. Introduction

L'internet des objets fait référence à l'interconnexion de plusieurs appareils informatiques, leur permettant d'échanger des données. Cette technologie joue un rôle essentiel dans de nombreux aspects de la vie moderne. Toutefois, cette communication expose également à de nombreux risques potentiels et vulnérabilités. Les attaques botnets représentent une menace significative pour la sécurité de l'IOT, où les appareils connectés sont compromis par des logiciels malveillants.

Pour détecter ces attaques, des méthodes de détection ont été développées, de plus en plus basées sur l'intelligence artificielle, ces techniques sont utilisées pour analyser de grandes quantités de données réseau et identifier les menaces potentielles en temps réel. En ce sens, ce chapitre permet d'identifier et de fournir une revue littérature des travaux connexes de recherche pertinents pour la détection des botnets IOT et les comparer.

2. Les méthodes de détection des attaques Botnets

Les méthodes basées sur l'intelligence artificielle (IA) sont de plus en plus utilisées pour détecter les attaques botnets en raison de leur pertinence croissante dans ce domaine. On a choisi de concentrer sur la revue de littérature pour la période de 2021 à 2024, car ces années représentent les plus récentes dans le domaine de la détection des botnets IoT. Pendant cette période, un grand nombre d'articles ont été publiés chaque année, ce qui souligne l'intérêt croissant pour ce sujet.

Dans la section suivante, on discute les différentes méthodes et techniques utilisées dans ces études, en résumant les résultats et les conclusions en fonction de ces catégories. Ces catégories incluent divers algorithmes et techniques de ML et DL tels que les CNN, RNN, DNN et l'apprentissage par renforcement (RL) et d'autres.

Pour l'analyse, on a classé les études sélectionnées en deux catégories d'IA principales selon les méthodes proposées. Chaque catégorie comprend des algorithmes utilisés et peut être combinés dans des approches différentes comme présenté dans la figure suivante :

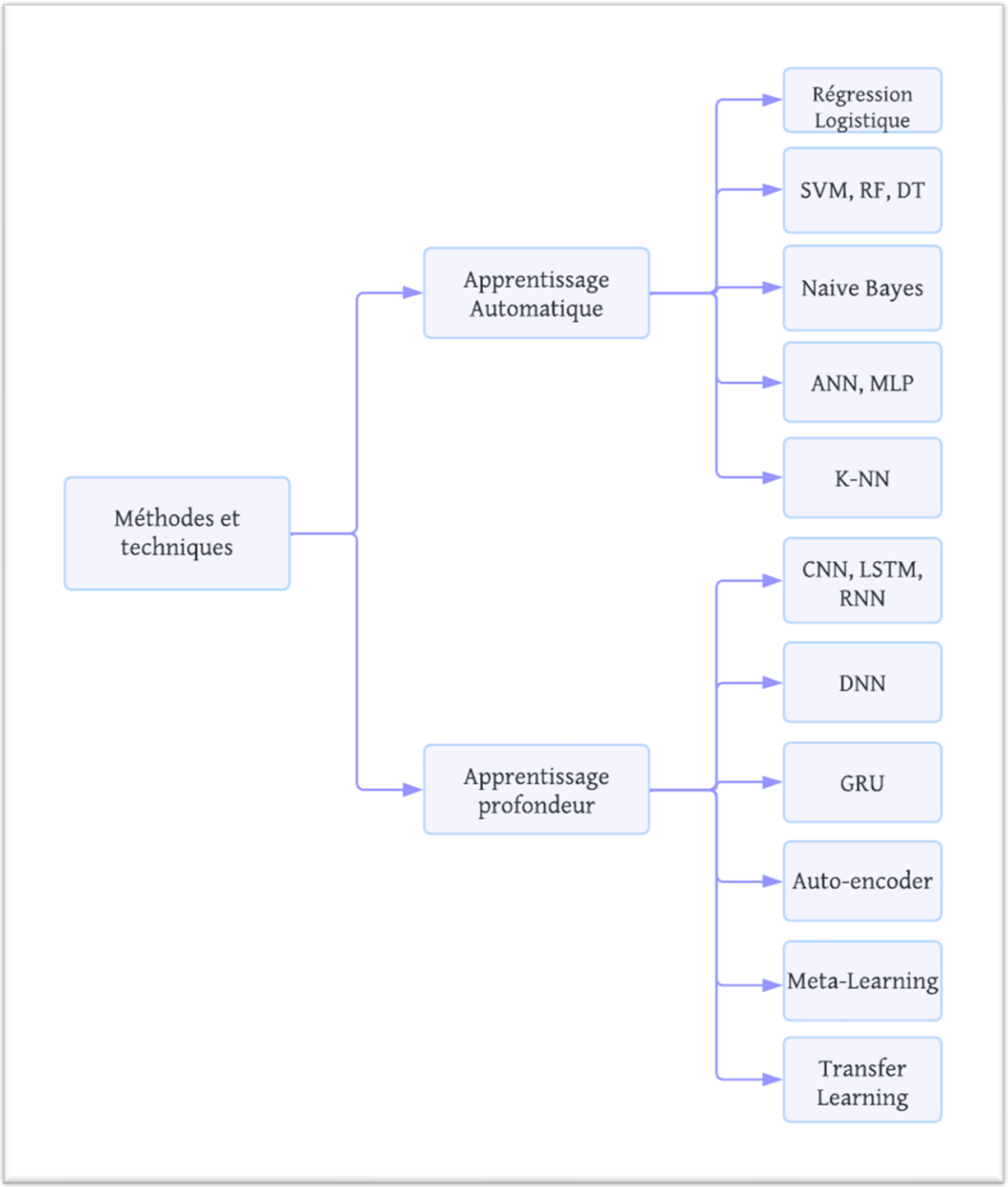


Figure 2.1. Méthodes et techniques d'IA utilisées pour la détection de botnets

Dans la section suivante, on examine chaque méthode en détail, mettant en évidence les principes fondamentaux de chaque approche, les ensembles de données utilisés, les résultats obtenus ainsi que les limites identifiées dans les études. Cette revue littérature permet d'offrir un aperçu complet des méthodes utilisées pour détecter les botnets IoT à l'aide de l'apprentissage automatique et de l'apprentissage en profondeur.

3. Travaux connexes

Cette section identifie les études pertinentes pour ce sujet. Jusqu'à présent, plusieurs travaux ont abordé la question de la détection des botnets IoT de différentes manières. Ces techniques ont été abordées dans diverses études, souvent en se concentrant sur l'apprentissage automatique (ML) et l'apprentissage profond (DL). Chacune de ces méthodes a ses propres points forts et limites.

3.1. Méthodes basées sur des algorithmes d'apprentissage automatique :

De nombreuses études ont été réalisées par des chercheurs dan7s le domaine de la détection des attaques de botnet en utilisant une combinaison de honeypot et de techniques d'apprentissage automatique, Méta-Apprentissage, systèmes multicouches, machine d'Apprentissage Extrême, ainsi que d'autres techniques. On mentionne dans cette sous-section les travaux qui ont été proposés récemment :

- *Système d'apprentissage en ensemble (KDR) : Selon [34], les auteurs ont proposé un système d'apprentissage en ensemble basé sur les algorithmes K-neighbors, Decision tree, and Random forest (KDR) sur bechmark CTU-13 dataset qui détecte ces attaques avec une précision remarquable de 99,7% en seulement 12,99 secondes.*
- *Modèle de Détection de Botnet de Méta-Apprentissage: Selon [35], les auteurs ont proposés des méthodes légères basées sur l'apprentissage méta avec des scores de performance élevés. Les ensembles de données Aposemat IoT-23, UC Irvine KDD99 et UNSW TON ont été utilisés pour l'entraînement et les évaluations de performance. Les résultats du modèle étudié montrent qu'il a atteint un taux de précision de 97,9%.*
- *Pots de miel combinés à l'apprentissage automatique: Selon [36], l'approche visait à résoudre le problème de la détection des attaques de botnet dans les usines intelligentes de l'IoT, les auteurs ont proposés une méthode de leurre et d'apprentissage automatique. Ils ont mis en*

place un environnement industriel intelligent basé sur l'architecture matérielle des dispositifs IoT. Ils ont utilisé le programme d'apprentissage automatique Weka avec l'algorithme de forêt aléatoire pour atteindre un score de précision élevé supérieur à 96%.

- *Une Double Approche D'Apprentissage Automatique: Selon [37], les auteurs ont fournis les derniers développements utilisant les techniques ML et DL. Une approche d'apprentissage automatique en deux volets est proposée. Deux modèles ResNet-18 ont été entraînés sur les vecteurs de caractéristiques résultants. Les résultats expérimentaux prouvent que cette approche peut efficacement prévenir et détecter les attaques de botnet par rapport aux autres modèles formés avec une précision de 99%.*
- *Un système multicouche : Est une approche qui combine plusieurs techniques et couches pour détecter et prévenir les attaques de botnets de manière robuste. Voici les recherches qui sont proposées cette approche :*

En 2021, selon [38], les auteurs ont proposés cette approche d'identifier les botnets du serveur C&C dissimulés par des techniques telles que l'obscurcissement ou le chiffrement. Ils ont proposé un intervalle de temps court (1s) pour agréger le comportement des botnets pour deux couches. Les deux couches du système affichent une précision supérieure à 90%.

Et en 2023, selon [39], les auteurs ont présentés un système basé sur (ML) et optimisé par la méthode du Black Hole (BHO-RF). Les résultats démontrent que cette technique surpasse les techniques conventionnelles, offrant un résultat plus élevé dans l'identification des botnets.

En 2024, selon [40], les auteurs ont proposé un système de détection et de classification des botnets à couches multiples, utilisant des algorithmes d'apprentissage automatique explicables. Le système comprend trois couches distinctes. La technique SHAP est utilisée pour fournir de la transparence au processus de prise de décision. L'évaluation sur l'ensemble de données NCC-2 avec une validation croisée à 10 plis montre des performances exceptionnelles, avec une exactitude moyenne de 99,98% dans la détection des botnets et de 99,34% dans la classification des familles de botnets.

- *Un modèle intelligent composé d'un classifieur RF avec une Analyse en Composantes Indépendantes (ICA) : Selon [41], les auteurs ont présentés un modèle en utilisant des*

algorithmes d'apprentissage automatique. Ce modèle intelligent combine un classifieur Random Forest avec une Analyse en Composantes Indépendantes (ICA). Les résultats expérimentaux ont démontré des performances exceptionnelles sur trois ensembles de données différents, avec une précision atteignant jusqu'à 99,99% et un temps de prédiction minimal de 0,12s.

- *Un système léger basée sur l'Extreme Learning Machine (ELM) : Cette proposition est présentée dans [42] en 2022. Les activités réseau des bots sont détectées et les bots sont identifiés en utilisant des états de connexion extraits et une modélisation de chaînes de Markov. Les résultats expérimentaux montrent une efficacité élevée et une amélioration significative par rapport aux méthodes existantes, avec un temps d'entraînement réduit.*
- *Approche en temps réel basée sur un arbre de décision: En 2023, les auteurs ont développés une approche [43] qui a fourni une méthode de détection des botnets en temps réel à l'aide d'un arbre de décision et d'une collection minimale de quatre fonctionnalités en effectuant une analyse du réseau. Ils considéraient que le réseau pouvait être surchargé dans la vraie vie. Cependant, en raison d'une surcharge, les routeurs ou autres périphériques peuvent abandonner certains paquets.*
- *Un Algorithme de Classificateur d'Ensemble avec Processus d'Empilement: Selon [44], les auteurs ont proposés l'utilisation d'un algorithme de classification par ensemble avec un processus d'empilement ''Stacking process'' (ECASP) pour sélectionner les caractéristiques optimales, afin d'évaluer les performances de détection des botnets. Les résultats montrent une précision de 94%.*

3.2. Méthodes basées sur des algorithmes d'apprentissage profond :

L'utilisation de l'apprentissage profond et de ses techniques, ainsi que des méthodes telles que le transfert d'apprentissage et la méthode des Black-Holes, est essentielle pour améliorer la détection des attaques de botnets.

- *Apprentissage profond hybride LSTM-CNN: En 2021, selon [45] les auteurs ont proposés la méthode hybride CNN-LSTM pour les applications de l'IoT, qui combine CNN et LSTM. Les résultats expérimentaux démontrent l'efficacité du modèle CNN-LSTM dans la détection des attaques de botnet à partir de dispositifs de sonnette (marques Danminin et Ennio) avec des taux de précision de 90,88% et 88,61%. De même, l'algorithme proposé atteint un taux de précision acceptable de 88,53% pour identifier les attaques de botnet à partir de dispositifs de thermostat.*

- *Prétraitement des données réseau par Vision par Ordinateur : Une Approche Utilisant le Traitement d'images est présentée dans [46] en 2023. Les données prétraitées sont ensuite envoyées à un réseau neuronal pour classification, explorant deux modèles : séquentiel et auto-encodeur. L'avantage de cette approche est sa capacité à être déployée au niveau des appareils IOT, avec un réseau neuronal moins complexe pour réduire le temps de détection.*
- *Réseau Antagoniste Génératif Tabulaire Conditionnel (CTGAN) : Selon [5], les auteurs étaient de déployer le modèle CTGAN pour modéliser et générer des données tabulaires. Les chercheurs ont travaillé avec un ensemble de données déséquilibré ''Bot-IOT'' et ont développé des techniques pour résoudre ce problème. Les résultats montrent des résultats prometteurs, avec le Perceptron Multicouche (MLP) atteignant une précision de 98,93%.*
- *Un système multicouches basé sur des algorithmes d'apprentissage automatique explicables: Selon [47], les auteurs ont présenté un modèle intelligent utilisant une technique de classification hybride optimisée. Ce modèle utilise un classificateur hybride construit en intégrant un Bi-GRU optimisé dans un RNN. Cette recherche propose une nouvelle approche d'optimisation hybride appelée SMIE (Moisissure visqueuse avec Évolution de l'immunité), qui combine deux méthodes d'optimisation. Remarquablement, ce modèle atteint une précision de détection de 97%.*
- *Deux modèles de réseau neuronal profond (DNN): Selon [48], les auteurs ont proposés cette nouvelle approche qui tire parti des avantages de l'ACP pour la collecte de caractéristiques et des classificateurs neuronaux profonds pour améliorer la détection efficace et précise des botnets dans les environnements IoT. Deux méthodes hybrides novatrices, DNNBoT1 et DNNBoT2, utilisant l'ACP, ont démontrées une précision 91% à la fois lors de la phase d'entraînement et de validation avec moins de variance pour la classification multi-classe afin de détecter les attaques de botnet.*
- *Proposition de MalBoT-DRL: Selon [49], les auteurs ont proposé un modèle de détection d'intrusions novateur utilisant une combinaison inédite de statistiques incrémentielles amorties et de mécanisme de récompense d'attention pour s'adapter dynamiquement aux logiciels malveillants IoT, il atteint des taux de détection exceptionnels de 99,80 % et 99,40 % dans les phases de détection précoce et tardive. Ce modèle explore l'efficacité de l'apprentissage par renforcement pour améliorer la généralisabilité des systèmes de détection d'intrusions.*
- *Une approche de l'apprentissage approfondi semi-supervisée: Est présentée dans [50] en 2022, utilisant le Réseau Contradictoire Génératif Semi-supervisé (SGAN) pour surmonter le*

manque de données étiquetées. Cette méthode combine une petite quantité de données étiquetées avec une grande quantité de données non étiquetées pour former un modèle. Les résultats montrent une efficacité accrue avec une précision de 99,89% en classification binaire et de 59% en classification multiple.

- *Modèle économique basé sur l'apprentissage approfondi : Selon [51], les auteurs ont proposés un modèle économique basé sur l'apprentissage approfondi, combinant le GRU pour apprendre les informations temporelles et CNN pour capturer les informations spatiales. Le modèle GRU-CNN montre une précision supérieure de 99%, tout en nécessitant moins de temps et de ressources d'implémentation en raison de sa taille architecturale réduite.*
- *Système de détection d'intrusion (IDS) basé sur le transfert d'apprentissage : Ce système est proposé dans [52]. En comparant les modèles d'apprentissage profond (ANN, RNN avec LSTM) et de transfert d'apprentissage, les résultats montrent que la solution proposée atteint une précision jusqu'à 99%, surpassant les modèles d'apprentissage profond pour les deux algorithmes.*
- *Un système LGANet (Local Graph Attention Network): Selon [53], LGANet est présenté pour la détection précise des bots P2P en combinant des caractéristiques basées sur le trafic réseau et des caractéristiques topologiques. Les résultats expérimentaux montrent une efficacité notable de LGANet, avec une réduction significative du score F1 en cas de suppression de la LGA-Layer. Cela souligne l'importance de considérer les informations topologiques pour améliorer la capacité de représentation des nœuds et booster la détection des botnets P2P.*
- *Détection Précoce des Botnets IoT (Cross CNN_LSTM) : Les botnets passent par différentes phases avant de pouvoir lancer des attaques, et ils peuvent être détectés à une phase précoce. Selon [54], Les auteurs ont proposés un système pour répondre à ce besoin, basé sur une étude empirique du comportement des botnets à une phase précoce. Ils ont élaboré un modèle appelée Cross CNN_LSTM, qui fusionne des modèles de CNN et de LSTM, atteignant une précision de 99,7%. Le modèle n'a pas de technique implantée pour empêcher le surajustement qui peut se produire lors de l'entraînement du modèle.*

Ces travaux sont résumés et comparés sur les plus importantes caractéristiques dans le tableau suivant :

4. Etude comparative

Article	Approche proposée	Méthodes/ Classificateurs utilisées	Ensemble de données	Précision (%)
Seungjin Lee et AL 2021 [36]	Pots de miel combinés à l'apprentissage automatique	SVM, RF, DT, HoneyPots	10 Caractécritiques de Botnets	96
WAN NUR HIDAYAH IBRAHIM et AL 2021 [38]	Système multi-couche	LSTM, SVM, K-NN, MLP	CTU-13	92
FAISAL HUSSAIN et AL 2021 [39]	Une Double Approche D'Apprentissage Automatique	Logistic Regression, ResNet-18	CICIDS-19, CICIDS-17, Bot-IoT, ScanLab	99
H. Alkahtani et AL 2021 [45]	Apprentissage profond hybride LSTM-CNN	CNN, LSTM	N-BaIOT	90

Yunyi Yang et Liming Wang 2021 [53]	*LGANet (Local Graph Attention Network), un système novateur pour détecter précisément les bots P2P*	*LGANet, AGF*	*PeerRush, CTU-23*	*95*
Nazmus Sakib Akash et AL 2022 [41]	*Un modèle novateur intelligent composé d'un classifieur RF avec une Analyse en Composantes Indépendantes (ICA)*	*Random Forest, ICA, k-Nearest Neighbor, SVM. Naïve Bayes*	*N-BaIOT, Aposemat IOT-23*	*99*
M. Wazzan et AL 2022 [54]	*Détection Précoce des Botnets IoT - Cross CNN_LSTM -*	*CNN, LSTM*	*MedBIoT*	*99.7*
Mohd Anul Haq et Mohd Abdul Rahim Khan 2022 [48]	*Deux modèles de réseau neuronal profond (DNN), DNNBoT1 et DNNBoT2*	*PCA, DNN*	*N-BaIOT*	*91*
Cut Alna Fadhilla et AL 2022 [35]	*Modèle de Détection de Botnet de Méta-Apprentissage*	*Random Forest, MLP, Naive Bayes, DT*	*Aposemat IoT-23, UC Irvine KDD99, UNSW TON*	*97*

Nasimul Hasan et AL 2022 [42]	*Un système léger basée sur une méthode d'apprentissage "Extreme Learning Machine"*	*Markov Chains, DT, MLP, LR*	*MedBIOT ETF*	*97*
Mohammad Al-Fawa'reh et AL 2022 [49]	*Détection de Botnets malveillante à l'aide de l'Apprentissage par renforcement approfondi (MalBoT-DRL)*	*RL, DNN, DQN, MDP*	*MedBIoT, N-BaIoT*	*99,8*
Kumar Saurabh et AL 2022 [50]	*Approche de l'apprentissage profond semi-supervisée*	*SCAN*	*Bot-IOT*	*99.89*
Aurélien Agniel et AL 2023 [46]	*Technique prétraite les données réseau à l'aide de la vision par ordinateur et du traitement d'images*	*Auto-Encoder*	*N-BaIoT*	*99*
Javier Velasco-Mata et AL 2023 [43]	*Approche en temps réel*	*Decision tree*	*CTU-13*	*92*

Omar Habibi et AL 2023 [5]	*Réseau Antagoniste Génératif Tabulaire Conditionnel (CTGAN)*	*CTGAN, MLP*	*Bot-IOT*	*98*
Amina Arshad et AL 2023 [34]	*Système d'apprentissage en ensemble (KDR)*	*K-neighbors, Decision tree, Random Forest*	*CTU-13*	*99,7*
Aditee Mattoo et AL 2023 [39]	*Forêt aléatoire optimisée pour les trous noirs (BHORF)*	*Random Forest*	*CTU-13*	*96*
Balaganesh Bojarajulu et AL 2023 [47]	*Un système multicouches basé sur des algorithmes d'apprentissage automatique explicables.*	*RNN, Bi-GRU, SMIE,*	*IoT-botnet*	*97*
Nelly Elsayed et AL 2023 [51]	*Modèle économique basé sur l'apprentissage approfondi*	*GRU, LSTM, CNN*	*UNSW 2018 IoT Botnet*	*99*
Sana Rabhi et AL 2023 [52]	*Système de détection d'intrusion (IDS) basé sur le transfert d'apprentissage*	*ANN, LSTM, RNN,*	*N-BaIoT, IoT-23*	*98*

S. Srinivasan et D. P 2023 [44]	*Un Algorithme de Classificateur d'Ensemble avec Processus d'Empilement*	*ELM, CNN, SVM, ECASP*	*Cyber Clean Center (CCC)*	*94*
Balaganesh Bojarajulu et AL 2024 [40]	*Système à couches multiples basé sur des algorithmes d'apprentissage automatique explicables.*	*SHAP (SHapley Additive exPlanations).*	*NCC-2*	*99*

Tableau 2.1. Tableau de comparaison les travaux connexes.

5. Synthèse

Les articles présentés dans le tableau précédent mettent en lumière la diversité des approches adoptées pour la détection des attaques de botnets. Ces études démontrent une forte performance dans la détection des botnets, avec des taux de précision qui sont généralement entre 90% et 99,8%. Cette performance est principalement attribuable à l'utilisation généralisée de techniques d'apprentissage automatique et profond, telles que SVM, RF, LSTM et CNN. Ces techniques permettent d'extraire des caractéristiques à partir de données brutes, offrant ainsi une capacité de détection avancée.

Les recherches s'intéressent de plus en plus à l'intégration de méthodes innovantes. Par exemple, l'utilisation de transfert d'apprentissage, Méta-Appprentissage, systèmes multicouches, machine d'Apprentissage Extrême. De même, l'utilisation des applications de techniques telles que les réseaux antagonistes génératifs (GAN) et l'apprentissage par renforcement approfondi (DRL).

Les ensembles de données utilisés dans ces études sont également variés, couvrant des scénarios de détection des botnets tels que N-BaIOT, CTU-13 et Bot-IoT. Cette diversité d'ensembles de données permet une évaluation approfondie des approches proposées dans des contextes variés.

Les travaux les plus récents mettent en évidence une tendance vers des approches multicouches, combinant plusieurs méthodes et modèles pour renforcer la précision et la robustesse des systèmes de détection. De plus, l'intégration de techniques d'apprentissage explicables gagne en importance, permettant une meilleure compréhension des décisions prises par les modèles de détection.

Les articles couvrent une période allant de 2021 à 2024, montrant une évolution dans les techniques de détection au fil du temps. On observe notamment l'émergence des systèmes multicouches en 2022, ainsi qu'une concentration sur la détection dans la phase finale de l'attaque botnet dans les dernières années.

D'après cette étude, La majorité des études se concentrent sur le développement de techniques de détection des botnets IoT à une phase avancée, lorsque ces botnets sont déjà actifs et lancent des attaques contre des cibles. Cependant, Il est essentiel de concentrer sur la détection des botnets IoT dès leurs premières phases de développement, avant même qu'ils ne lancent des attaques. Cette approche permettrait de limiter l'utilisation illégale des ressources des appareils et de prévenir les interruptions ou refus de service sur les réseaux IoT.

6. Conclusion

En conclusion de ce chapitre, on a résumé plusieurs articles pertinents dans le domaine la détection des attaques de botnets, en concentrant sur les méthodes et les performances de détection qu'ils présentent. Une étude comparative de ces articles montre une diversité de techniques utilisées, notamment l'apprentissage profond, le transfert d'apprentissage, les méthodes d'ensemble, ainsi que l'utilisation de données et de modèles spécifiques à chaque étude.

Dans le chapitre suivant, on travaillera à développer un modèle qui détecte les attaques botnet dans leurs premières phases avant de déclencher des attaques, en utilisant des techniques de l'intelligence artificielle.

Chapitre 03

Chapitre 03

Proposition d'un modèle hybride AE-GRU

1. Introduction

Les botnets est devenu un risque grave pour les systèmes de l'IOT en raison du manque de sensibilisation des utilisateurs finaux a la sécurité. Par défaut, plusieurs ports sont ouverts et les informations d'identification des utilisateurs ne sont pas modifiées. Heureusement, les botnets passent par plusieurs étapes pour lancer des attaques, et chaque étape a des activités malveillantes différentes ce qu'ils peuvent être détectés à une étape précoce, cela signifie que cette détection diffère de la détection à l'étape avancée.

Dans ce chapitre, on présente une approche de la détection des Botnet dans une étape précoce, en développant un modèle d'apprentissage en profondeur en combinant Auto-Encodeur et GRU. Avant de décrire la méthodologie de cette approche, on aborde d'abord les deux principales étapes de ces attaques, puis on présente les algorithmes de DL utilisés.

2. Proposition

Dans le premier chapitre, on a abordé les étapes de la formation des attaques Botnets en particulier, mais maintenant on va les rassembler en deux étapes principales, précoce et tardive, et chacune comporte leurs activités malveillantes qui les distinguent :

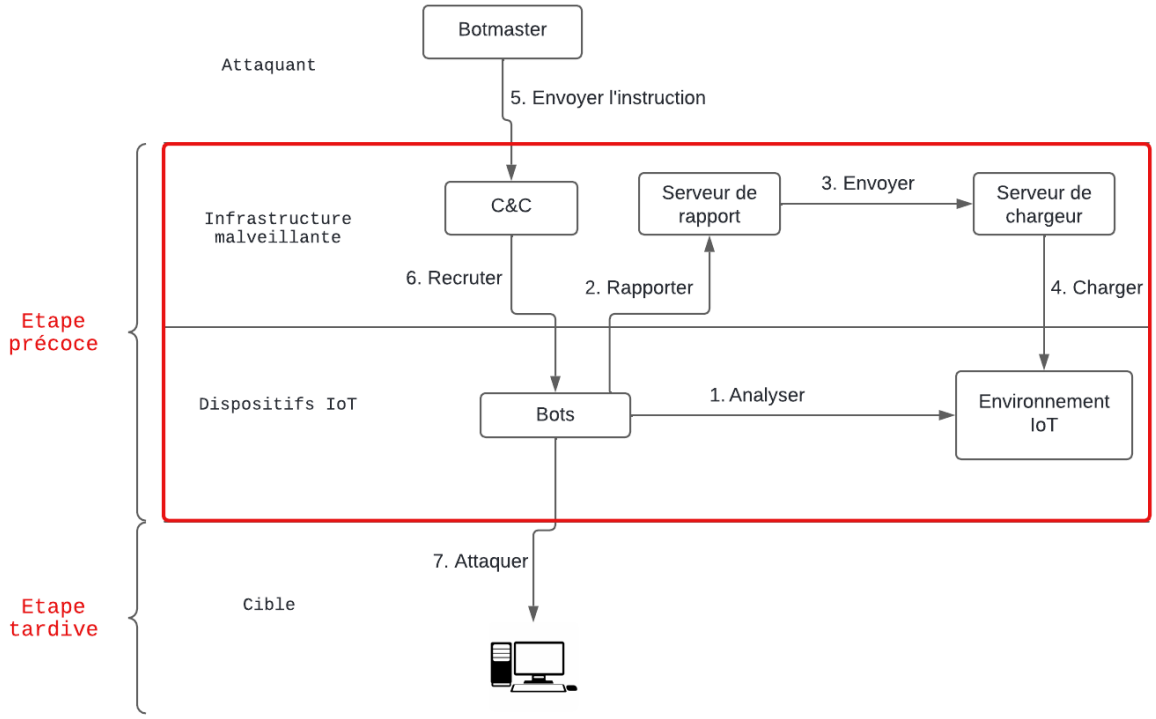


Figure 3.1. L'étape de la détection des attaques botnets dans l'IoT

L'étape précoce des attaques de botnets IoT se concentre sur la formation et l'établissement du botnet, où les activités sont souvent subtiles et moins détectables. En revanche, l'étape tardive se caractérise par des activités plus agressives, telles que les attaques DDoS, une fois le botnet pleinement opérationnel.

Donc, on propose une approche hybride combinant les auto-encodeurs et le GRU pour répondre à ces défis. Par conséquent, l'utilisation de l'ensemble de données MedBIoT pourrait fournir une base solide pour l'entrainement et l'évaluation de ce modèle.

Les deux algorithmes d'apprentissage en profondeur ont été sélectionnés utilisés pour la détection d'anomalies dans le trafic réseau :

- *Auto-Encodeur (Auto Encoder -AE-) :*

Un auto-encodeur est un type de réseau de neurones artificiels utilisé pour apprendre des codages efficaces de données non étiquetées, généralement dans le but de réduire la dimensionnalité des données. Il se compose d'un encodeur qui comprime les données en une représentation latente et d'un décodeur qui reconstruit les données à partir de cette représentation, d'où son nom "auto-encodeur". L'entraînement d'un auto-encodeur vise à minimiser les erreurs de reconstruction, souvent en utilisant une fonction de perte comme l'erreur quadratique moyenne (MSE), et implique la présentation des données d'entrée non étiquetées pour la compression et la reconstruction.[4]

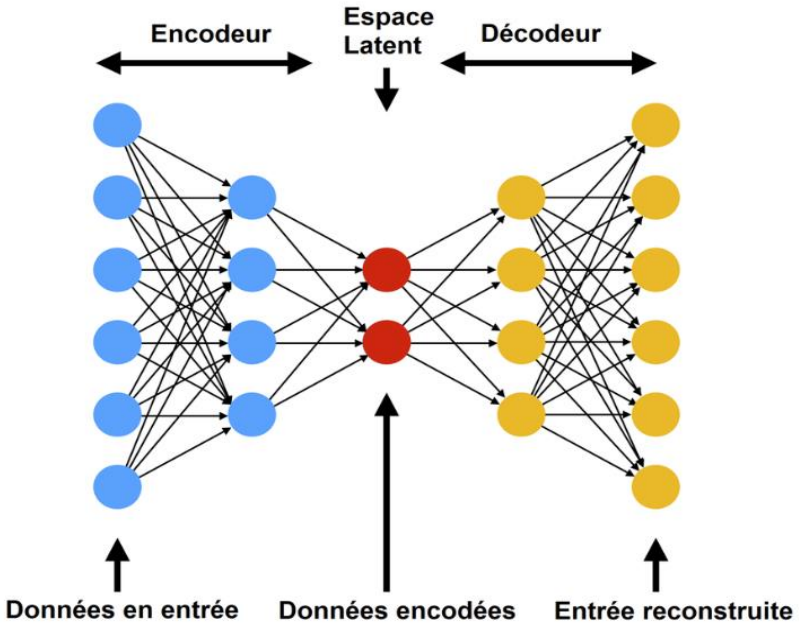


Figure 3.2. Architecture de base de l'auto-encodeur [55]

- *Les Gated Recurent Unit (GRU) :*

Les Unités Récurrentes à Portes GRU sont une architecture de réseaux de neurones récurrents (RNN) conçue pour résoudre des problèmes comme le vanishing gradient rencontré par les RNN traditionnels. Le GRU utilise des portes pour contrôler le flux d'informations, permettant ainsi de mémoriser des dépendances à long terme dans les données. Il intègre deux portes internes, la Reset

Gate et l'Update Gate, pour décider quelles informations passées conserver et intégrer avec les entrées actuelles, facilitant ainsi l'apprentissage des dépendances à long terme. [46]

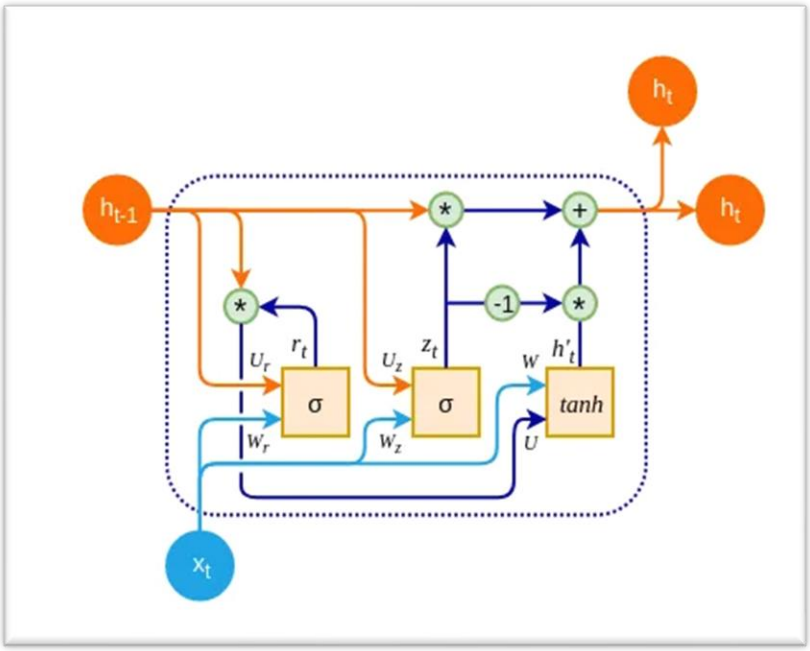


Figure 3.3. Architecture de base de GRU [56]

2.1. Méthodologie :

Pour détecter les attaques de botnet dans une étape précoce en utilisant une combinaison de GRU et d'auto-encodeurs avec l'ensemble de données MedBIoT de fichiers PCAP comme mentionne dans la figure 12. Tout d'abord, il est crucial de comprendre l'ensemble de données MedBIoT, notamment la structure des fichiers PCAP et les caractéristiques du trafic réseau qu'ils contiennent, y compris les schémas d'activité normaux et les comportements malveillants associés aux botnets. Ensuite, le processus de prétraitement des données doit être effectué avec soin, en extrayant les caractéristiques pertinentes du trafic réseau et en divisant les données en ensembles d'entraînement, de validation et de test équilibrés.

La construction d'un auto-encodeur, en utilisant des couches GRU pour capturer les séquences temporelles, est essentielle pour extraire les caractéristiques discriminantes des données normales. Entraîner l'auto-encodeur avec les données normales permet d'obtenir une représentation compacte des schémas normaux du trafic. Ensuite, un modèle GRU est construit pour détecter les

anomalies en utilisant les caractéristiques extraites par l'auto-encodeur comme entrée. Ce modèle est entraîné avec un ensemble de données qui comprend à la fois des données normales et des données d'attaques de botnets. Après l'entraînement, le modèle est évalué sur un ensemble de test pour mesurer ses performances de détecter les attaques de botnets dans une étape précoce.

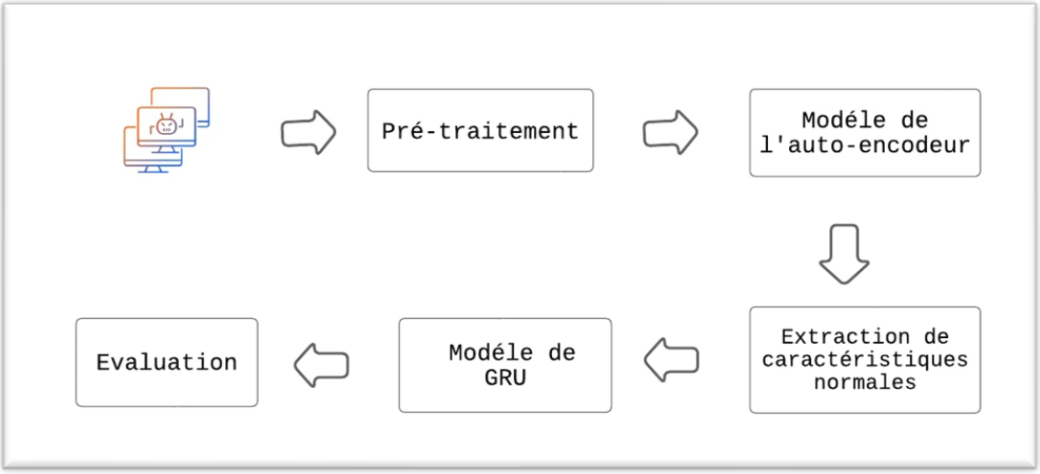


Figure 3.4. Modèle hybride proposé pour la détection des botnet

Algorithme 1 : Algorithme pour le modèle hybride proposé AE-GRU
Entrée: Données Malware/ Normales
Sortie: Précision, perte, précision, rappel, score F1
1: Filtrage de données (Données prétraitées)
2: Extraction de caractéristiques (Données prétraitées)
3: Normalisation (Données prétraitées)
4: Fractionnement basé sur 80:20 (données d'entraînement, données de test)
5: Appliquer le modèle Auto-Encodeur basé sur les couches GRU
6: Appliquer dense
7: Utilisez l'optimiseur Adam
8: Aplatir les données de prédiction
9: Appliquer le modèle GRU
10: évaluer la perte
11: évaluer la précision

12: Utiliser les données de test pour calculer la précision, le rappel, le score F1 15: Calculer la perte, la précision

2.1.1. Sélection de l'ensemble de données :

La qualité et la taille de l'ensemble de données ont un impact significatif sur les performances des modèles d'apprentissage en profondeur. Certains chercheurs en détection de botnets utilisent des ensembles de données généraux qui ne sont pas spécifiquement conçus pour un problème ou un domaine particulier tel que UNSW-NB15 qui est un ensemble de données crée pour la détection d'intrusion dans les réseaux. En conséquence, cette recherche souffre d'un manque d'ensemble de données de références. Répondant à ce besoin, des ensembles de données IoT spécifiques ont été générés, comme Bot-IoT. Cependant, ces ensembles de données présentent parfois des lacunes, notamment des problèmes de déséquilibre des classes, qui peuvent affecter les performances des modèles proposés. Par conséquent, il y a des critères spécifiques pour sélectionner l'ensemble de données pour cette étude :

- *L'ensemble de données doit être généré à l'aide de différents types d'appareils IoT.*
- *Plus d'un malware IoT doit être utilisé.*
- *L'ensemble de données doit se concentrer sur les premières étapes du déploiement du botnet IoT.*

L'ensemble de données MedBIoT comble le manque d'ensembles de données IoT générés dans la détection des botnets IoT. Il a été généré à l'aide d'un réseau d'appareils IoT de taille moyenne composé de 83 appareils IoT. Ces appareils sont une combinaison d'appareils IoT physiques et émulés. Il fournit de vraies données réseau en déployant de vrais logiciels malveillants (Mirai, Bashlite et Torii). Cet ensemble de données se concentre sur l'étape de propagation (propagation et communication C&C). L'ensemble de données se compose de 23 340 359 paquets réseau répartis en différentes classes [57].

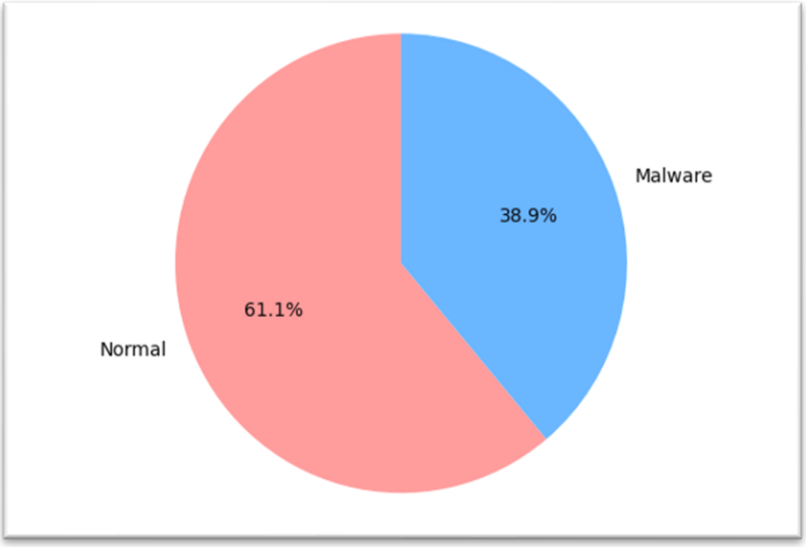


Figure 3.5. Répartition des échantillons normaux et malwares

2.1.2. Pré-traitement des données :

Le prétraitement des données est une étape essentielle dans le processus de développement d'un modèle de détection de botnet à partir de données PCAP de l'ensemble de données MedBIoT. Cette étape comprend plusieurs sous-étapes, comme présenté dans la figure 16 :

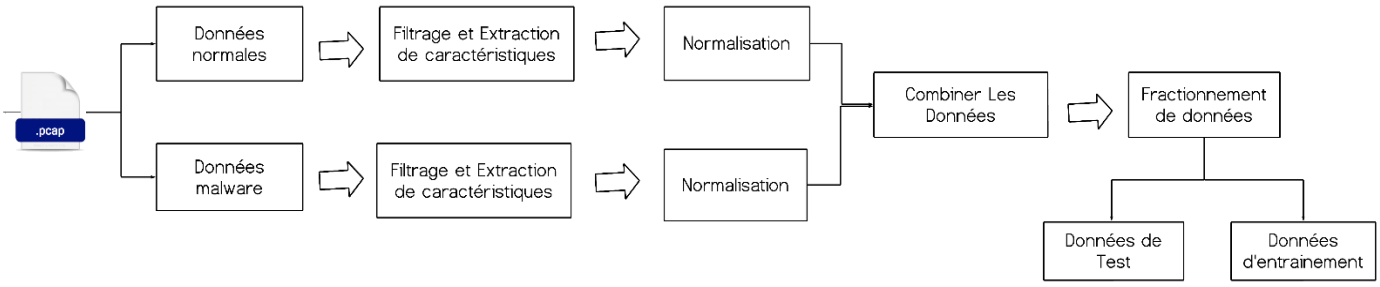


Figure 3.6. Processus de pré-traitement des données

Dans cette phase de prétraitement des données, les deux premières étapes, à savoir le filtrage des données et l'extraction des caractéristiques, sont essentielles pour préparer les données à l'analyse et à la modélisation. L'ensemble de données, composé de fichiers PCAP, contient deux types de données distincts : les données malware (botnets) et les données normales. Chaque type de données est associé à trois types d'attaques différents : Torii, Mirai et Bashlite.

Dans un premier temps, on applique le filtrage des données pour sélectionner uniquement les sessions normales dans la partie des données normales et uniquement les sessions malware dans les données malwares. Cela permet de segmenter efficacement les données en fonction de leur nature, facilitant ainsi le traitement ultérieur.

Ensuite, on procède à l'extraction des caractéristiques à partir de chaque type de données, qu'il s'agisse de données normales ou de données malware. Pour chaque type de donnée, on extraie des caractéristiques qui sont indiquées dans le tableau 3.

Après avoir réalisé les deux premières étapes, on passe à la normalisation des caractéristiques qu'on a extraites précédemment. Cette étape est essentielle pour mettre toutes les caractéristiques à une échelle commune, ce qui facilite l'entraînement de modèle. Une fois les caractéristiques extraites normalisées, on combine les données malware et les données normales pour former un ensemble de données complet et équilibré.

Caractéristique	*Définition*
ip_source	*L'adresse IP source du paquet*
sport	*Le port source du paquet*
ip_des	*L'adresse IP de destination du paquet*
dport	*Le port de destination du paquet*
protocol	*Dans notre cas, toujours TCP car seuls les paquets TCP sont pris en compte*
duration	*La durée de la connexion, calculée comme la différence entre le temps du dernier paquet et le temps du début de la connexion.*
average_packet_size	*La taille moyenne des paquets dans la connexion*
average_packet_interval	*Les intervalles entre les paquets dans la connexion, calculés comme la différence de temps entre les paquets consécutifs.*
label	*Un label indiquant que si c'est une session normale (0) ou anormale (1)*

Tableau 3.1. Caractéristiques extraites pour l'analyse des paquets

2.1.3. Entrainement du modèle proposé :

Le modèle hybride proposé qui est présenté dans la figure 17 se concentre sur l'utilisation d'un auto-encodeur basé sur les couches GRU pour extraire les schémas temporels du trafic réseau normal est entraîné sur des données représentatives. Par la suite, un modèle GRU est mis en place pour détecter les anomalies en se basant sur ces caractéristiques extraites :

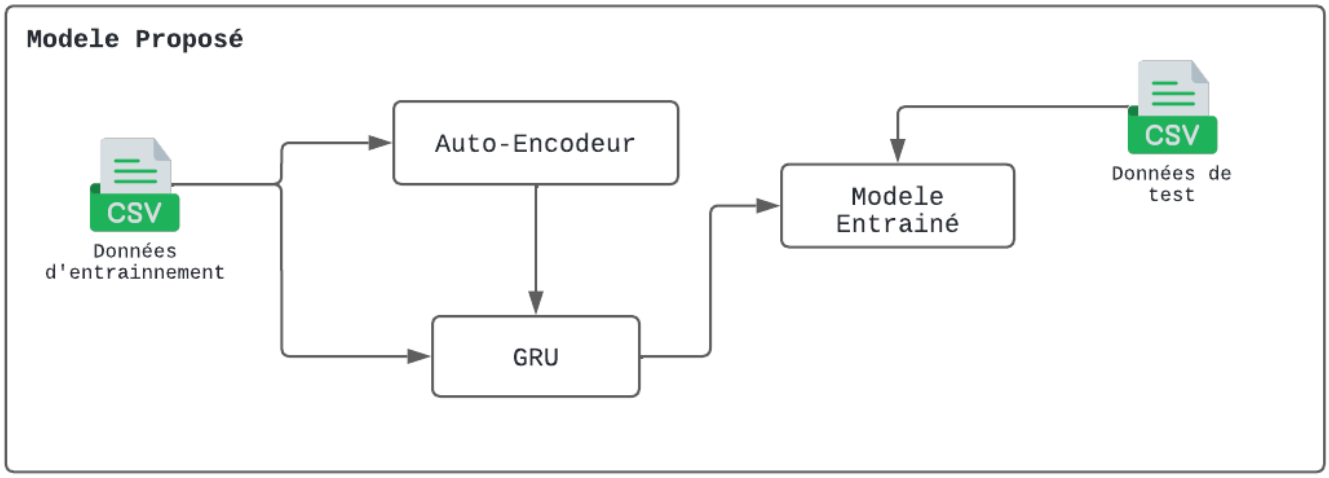


Figure 3.7 : Processus de modèle hybride AE-GRU proposé

En entraînant l'auto-encodeur avec des données normales, on parvient à obtenir une représentation compacte des schémas de trafic normaux. Cette représentation permet de capturer les structures sous-jacentes des données, en filtrant les caractéristiques non pertinentes et en mettant en évidence les schémas récurrents du trafic normal.

Les couches GRU sont particulièrement adaptées à cette tâche car elles sont capables de modéliser des dépendances temporelles à long terme dans les données, ce qui est crucial pour détecter les anomalies danses séquences de données complexes.

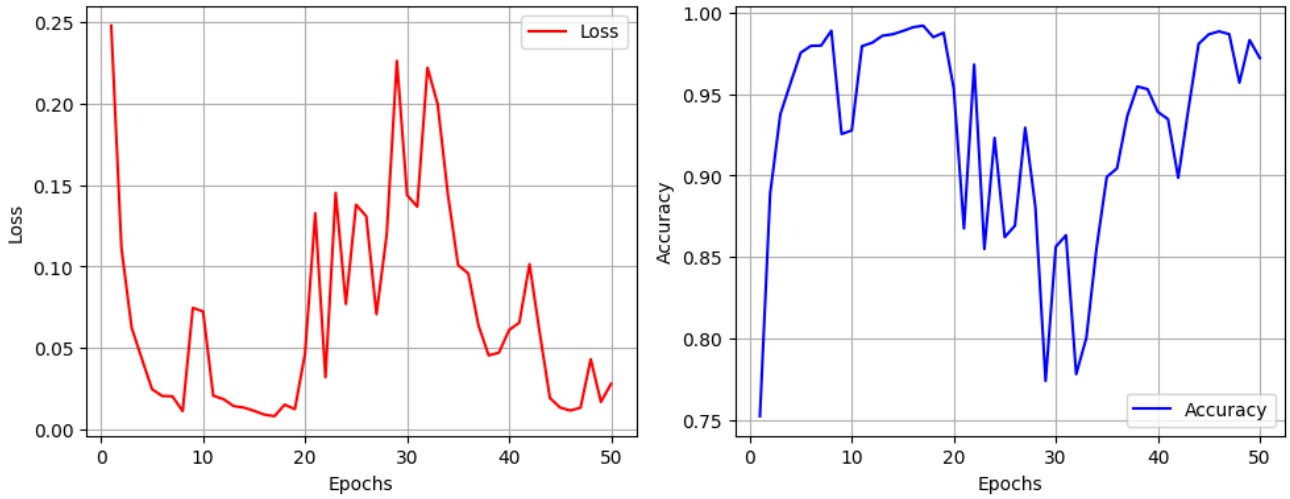


Figure 3.8. Courbes de précision et de perte des modèle Auto-encodeur proposé en fonction des époques d'entrainement.

Ensuite, les caractéristiques extraites par l'auto-encodeur à partir des données normales sont utilisées comme entrée pour le GRU. Le GRU est ensuite entraîné sur un ensemble de données qui comprend à la fois du trafic normal et du trafic contenant des attaques de botnets.

On a utilisé une méthode d'optimisation appelée Grid Search pour ajuster les hyperparamètres de ce modèle, garantissant ainsi des performances optimales. Les résultats de cette optimisation, comprenant les hyperparamètres choisis pour chaque algorithme, sont présentés dans le tableau suivant :

Hyperparamètres	*Auto-Encodeur*	*GRU*
Nombre d'époques ('epochs')	*50*	*10*
Taille du lot ('batch_size')	*32*	*64*
Taille de la couche GRU ('GRU units')	*128, 64*	*64*
Fonction d'activation	*ReLU*	*ReLU, sigmoid*

Optimiseur ('optimiser')	adam	adam
Fontion de perte ('loss')	mse	Binary_crossentropy
Métrique ('metrics')	-	Accuracy, Recall, F1 score

Tableau 3.2. Les hyperparamètres utilises dans le modèle Auto-encodeur et GRU

L'objectif du GRU est d'apprendre à distinguer les données normales des données anormales. En d'autres termes, le GRU doit apprendre à prédire les séquences de données suivantes en se basant sur les séquences d'entrée. Si les données d'entrée sont anormales et correspondent à une attaque de botnet, la prédiction du GRU sera également anormale

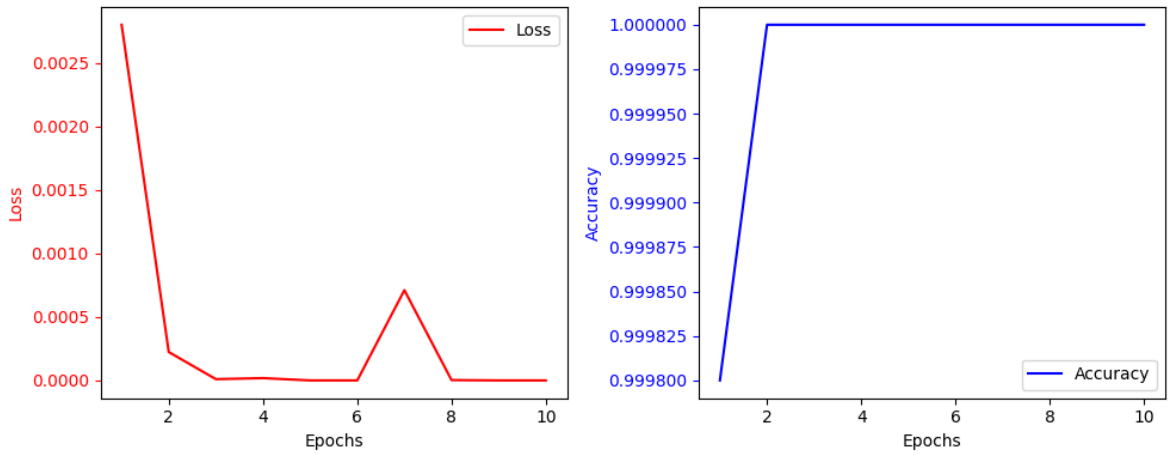


Figure 3.9. Courbes de précision et de perte des modèle GRU proposé en fonction des époques d'entrainement.

2.2. Résultats et Discussion :

Le modèle AE-GRU qu'on a proposé pour détecter les attaques de botnet dans une étape précoce a donné des résultats remarquables lors de son évaluation. Les résultats de l'évaluation de ce modèle sont résumés comme suit :

Une précision de 0.99, ce modèle parvient à classifier correctement les attaques de botnets, réduisant ainsi les faux positifs et garantissant une détection précise des menaces.

De plus, le rappel de 0.99 indique que le modèle réussit à retrouver presque toutes les attaques de botnet présentes dans les données, minimisant ainsi les faux négatifs et assurant une détection exhaustive des menaces.

Le score F1 de 0.99 souligne l'équilibre remarquable entre la précision et le rappel de notre modèle, ce qui renforce sa fiabilité dans la détection des attaques de botnet.

Malgré ces performances exceptionnelles, il reste une légère marge d'erreur, comme en témoigne le taux de faux positifs de 0.01, nécessitant ainsi des améliorations pour réduire davantage ce taux.

Cependant, le taux de vrais négatifs élevé de 0.98 démontre la capacité de notre modèle à distinguer efficacement les activités légitimes des attaques de botnet, ce qui souligne son utilité dans la protection des systèmes contre les menaces de sécurité informatique.

Une précision de 0.997 est un indicateur fort que le modèle AE-GRU est fiable pour classer correctement le trafic réseau. Cela signifie que le modèle a peu de faux positifs (trafic normal identifié à tort comme malveillant) et peu de faux négatifs (trafic malveillant manqué). Ces résultats sont présentés dans la figure 21 :

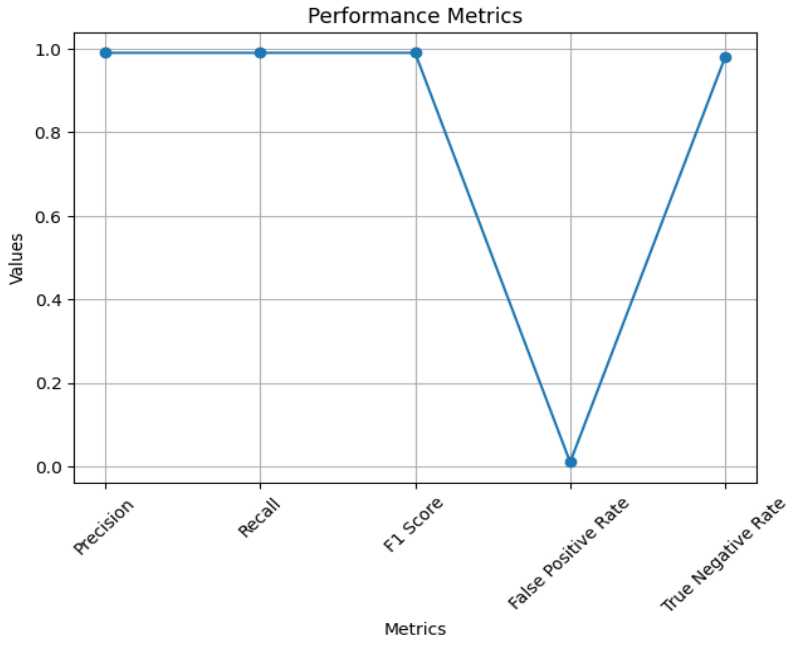


Figure 3.10. L'évaluation de modèle hybride AE-GRU avec les métriques

2.3. Comparaison avec l'autre approche

On a mentionné précédemment l'article [48] qui utilisait un modèle hybride pour détecter les attaques à une étape précoce. Le Cross CNN_LSTM, basé sur des modèles d'apprentissage en profondeur fusionnant un CNN et une LSTM. Les résultats montrent que ce modèle est précis et surpassent certains des méthodes les plus avancées, avec une précision de 99,7 %. Bien que cette approche puisse être plus simple à mettre en œuvre, le manque de détails sur son architecture et ses algorithmes complique une évaluation complète de sa robustesse et de sa généralisabilité.

D'autre part, notre approche basée sur AE et GRU offre une précision de détection exceptionnelle de 99,99% et capture efficacement les séquences temporelles dans le trafic réseau. Cependant, sa complexité due à la combinaison de ces deux modèles nécessite des données d'entraînement de haute qualité et peut entraîner un risque de sur-apprentissage.

3. Conclusion

Dans ce chapitre, on a proposé une approche de détection des botnet dans les réseaux IoT dans une étape précoce. Le modèle GRU détecte les anomalies avec une haute précision de 99,99%, témoignant de sa capacité à identifier de manière fiable les attaques de botnet dans une étape précoce. Ces résultats confirment l'utilité et l'efficacité du modèle AE-GRU dans la lutte contre les menaces émergentes dans les réseaux IoT, fournissant ainsi une contribution significative à la sécurité des systèmes connectés. Finalement, on a présenté et discuté les performances de notre approche obtenues et on a donné une comparaison entre ce model et un autre.

Chapitre 04

Chapitre 04

Présentation de l'environnement et Simulation

1. Introduction

Dans ce dernier chapitre, on a concentré sur la mise en œuvre pratique de notre système de détection d'intrusion basé sur le modèle AE-GRU dans un environnement de simulation. Notre objectif est de déployer notre modèle en tant que IDS dans une machine virtuelle, afin de pouvoir surveiller et détecter les activités anormales sur un réseau simulé.

On débute par une présentation détaillée de l'environnement de travail, décrivant l'architecture de la topologie de l'environnement IOT qui contient la machine virtuelle destinée à héberger notre IDS, ainsi que les outils et les logiciels essentiels pour son fonctionnement et son évaluation. Ensuite, on détaille le processus de déploiement de notre modèle AE-GRU au sein de cette machine virtuelle, en explicitant les étapes nécessaires pour son intégration au sein de l'infrastructure du réseau simulé.

Finalement, on aborde les avantages et les limites de notre approche, tout en envisageant les perspectives d'amélioration et les futures avenues de recherche pour notre IDS.

2. Environnement de Travail

On présente dans cette section l'environnement logiciel et le langage utilisé pour le développement et la mise en place de notre scénario.

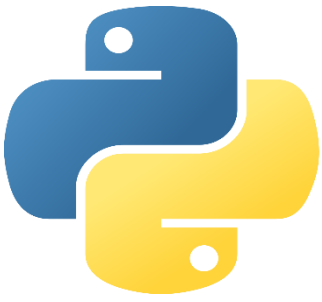
2.1. Environnement logiciel :

Dans ce travail les logiciels utilisés sont :

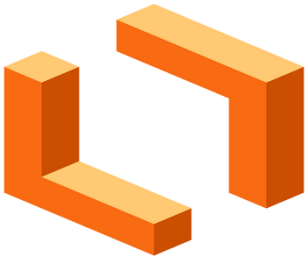
- *VMware Workstation Pro : représente le produit phare de VMware pour exécuter des charges de travail de machines virtuelles et divers systèmes d'exploitation sur des bureaux Windows ou Linux. Il propose une application de virtualisation de bureau simplifiée pour l'hôte, nécessitant simplement des ressources supplémentaires sur la machine hôte. Ce logiciel, pris en charge tant pour un usage professionnel que personnel, est une plateforme de virtualisation de premier plan adaptée à un hyperviseur de type 2 [58].*

- *Ubuntu : est un système d'exploitation open-source basé sur Linux, largement utilisé pour les ordinateurs personnels, les serveurs et les appareils IoT. Il offre une interface conviviale avec une grande variété de logiciels préinstallés, ainsi qu'un gestionnaire de paquets pour faciliter l'installation de nouveaux logiciels. Ubuntu est soutenu par une communauté active qui fournit des mises à jour régulières et un support technique via des forums et des canaux de communication en ligne. [59].*

- *Python : est un langage de programmation polyvalent et convivial, connu pour sa syntaxe claire et lisible. Il est utilisé dans divers domaines tels que le développement web, l'analyse de données, l'intelligence artificielle et l'automatisation des tâches. Grâce à sa vaste bibliothèque standard et à sa communauté active, Python est un outil puissant pour résoudre une variété de problèmes informatiques. Il prend en charge différents paradigmes de programmation, ce qui le rend flexible et adapté à une grande diversité de projets [60].*



• *Google Colab : est un service de bloc-notes Jupyter hébergé qui ne nécessite aucune configuration et offre un accès gratuit aux ressources informatiques, y compris les GPU et les TPU. Colab est particulièrement bien adapté à l'apprentissage automatique, à la science des données et à l'éducation [61].*



• *Lucidchart : est un espace de travail visuel qui combine la création de diagrammes, la visualisation de données et la collaboration pour accélérer la compréhension et stimuler l'innovation [62].*

• *TShark : est un analyseur de protocole réseau. Il vous permet de capturer des paquets de données à partir d'un réseau en direct ou de lire des paquets à partir d'un fichier de capture précédemment enregistré, soit en imprimant une forme décodée de ces paquets sur la sortie standard, soit en écrivant les paquets dans un fichier [63].*

2.2. Bibliothèques du python :

Dans ce travail les bibliothèques utilisés sont :

- *numpy : Une bibliothèque Python utilisée pour effectuer des calculs numériques efficaces, notamment pour les tableaux multidimensionnels [64].*
- *pandas : Une bibliothèque Python utilisée pour la manipulation et l'analyse des données, offrant des structures de données flexibles et des outils pour les opérations de données [65].*
- *scapy : Une bibliothèque Python utilisée pour la manipulation d'interactions au niveau de la couche réseau, notamment pour la capture, l'analyse et la génération de paquets réseau [66].*
- *Scikit-learn : est une bibliothèque Python très utilisée pour l'apprentissage automatique (machine learning), elle offre une variété d'outils pour l'apprentissage supervisé et non supervisé, ainsi que pour l'évaluation des modèles [67].*

- *tensorflow : est une bibliothèque open-source d'apprentissage automatique développée par Google, principalement utilisée pour la création, l'entraînement et le déploiement de modèles d'apprentissage profond (deep learning) [68].*
- *keras : est une bibliothèque haut niveau pour l'apprentissage profond, intégrée à TensorFlow, facilitant la création et la formation de modèles de réseaux de neurones [69].*
- *matplotlib : est une bibliothèque Python largement utilisée pour la création de visualisations graphiques, notamment des graphiques, des diagrammes, des histogrammes, des parcelles de contour, des parcelles en 3D, etc [70].*

3. Architecture du réseau de l'environnement expérimental

L'objectif principal de cette expérience était d'analyser le malware du botnet IoT et d'étudier son comportement en surveillant et en collectant les paquets de trafic. Dans cette expérience, nous nous sommes concentrés sur l'étude de la propagation des botnets IoT,

Dans cette section, on présente une topologie de réseau expérimentale utilisée pour évaluer les performances d'un IDs basé sur un modèle AE-GRU. Le réseau simule un environnement réel avec différents types d'appareils et de trafic réseau, ce qui permet de tester l'efficacité des identifiants pour détecter le trafic malveillant.

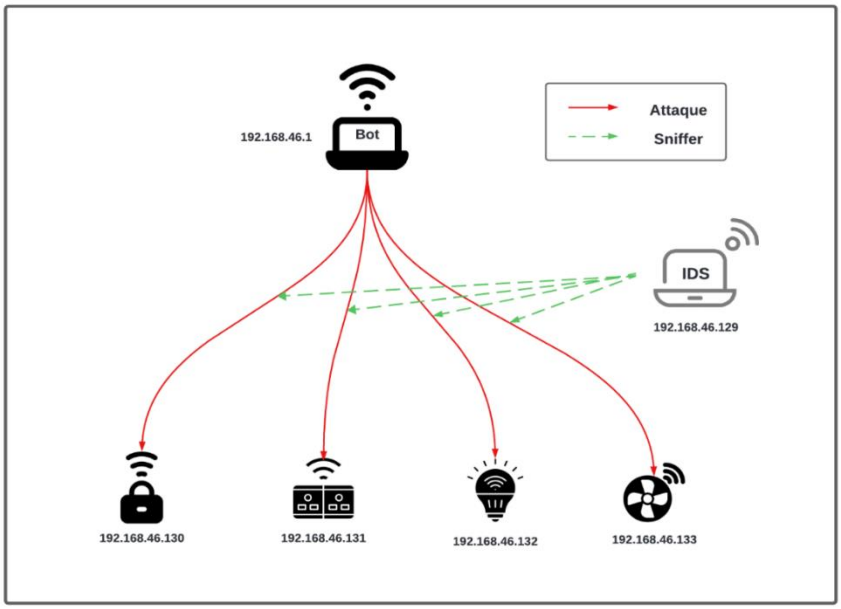


Figure 4.1. Architecture du réseau de l'environnement expérimental

Cette architecture représente un réseau expérimental utilisé pour tester un IDS basé sur un sniffer et conçu pour identifier le trafic malveillant ciblant les appareils IoT. Les éléments clés du réseau et leur rôle sont les suivants :

- *Bot malveillant : Représenté par un ordinateur, le bot joue le rôle d'un appareil compromis qui envoie du trafic malveillant vers les appareils IoT vulnérables du réseau. Ce trafic malveillant peut inclure des paquets de données infectés, des tentatives d'exploitation de failles de sécurité ou des commandes visant à contrôler les appareils IoT.*
- *Appareils IoT : Représentent les cibles potentielles des attaques du bot malveillant. Ces appareils sont un éventail, serrure, lumière et commutateur, ou peuvent être d'autres dispositifs intégrés au réseau domestique ou professionnel.*
- *Sniffer : Représenté par un autre ordinateur. Il utilise Tshark, pour capturer et analyser le trafic circulant sur le réseau.*
- *Système de détection d'intrusion (IDS) : Représenté par un code source exécutable, Il utilise notre modèle proposé, pour identifier les paquets malveillants et les distinguer du trafic réseau normal.*

4. Scenario d'expérimentation

Dans cette section, on définisse les étapes principales de cette expérience :

- *Le bot cible les appareils IoT du réseau en envoyant du trafic malveillant.*
- *Le sniffer surveille le trafic réseau, en capturant les paquets envoyés par le bot ainsi que tout autre trafic à l'aide de Tshark.*
- *Les données collectées, comprenant les paquets malveillants, sont enregistrées sous forme de fichiers 'pcap' pour une analyse ultérieure.*
- *L'IDS examine ces fichiers en utilisant le modèle AE-GRU, identifiant les signes d'activité malveillante tout en standardisant et évaluant les données.*
- *Enfin, l'IDS confirme la présence d'attaques et génère des alertes pour signaler toute activité suspecte détectée.*

5. Déploiement d'un IDS basé sur AE-GRU

Dans cette section, on a exploré la mise en œuvre pratique de notre solution de détection des attaques Botnet basée sur le modèle AE-GRU, en tant qu'IDS dans un environnement simulé. L'objectif est d'évaluer les performances de notre modèle dans des conditions proches de la réalité, ce qui est important pour garantir son efficacité dans un déploiement réel.

On a modifié le code de notre modèle AE-GRU proposé, en supprimant la partie d'entrainement. Les étapes de déployer un IDS basé sur notre modèle peuvent être résumées comme suit :

- *Capture du Trafic : On configure un sniffer ''Tshark'' pour capturer le trafic réseau en temps réel. Puis on enregistre les paquets capturés dans des fichiers au format pcap.*
- *Analyse et Normalisation des Données : Dans cette étape, on charge les fichiers pcap capturés et extraire les caractéristiques pertinentes et on utilise le code de prétraitement pour normaliser ces caractéristiques.*
- *Déploiement du Modèle AE-GRU : On charge le modèle AE-GRU pré-entraîné et le configure pour qu'il analyse les données normalisées en temps réel.*
- *Évaluation en Temps Réel : On fait fonctionner le modèle en continu pour analyser les paquets normalisés qui est détermine si le trafic est normal ou suspect en fonction des sorties du modèle.*

- *Génération d'Alerte et de Résultats : Si la sortie du modèle indique que le trafic est normal (la probabilité de normalité est supérieure à un seuil défini), consignez que le trafic est normal, et si la sortie du modèle indique une anomalie, générez une alerte.*

6. Création du l'environnement de Simulation

L'objectif principal de cette expérience est de tester le fonctionnement du programme dans l'IoT et d'étudier son comportement en surveillant et en collectant les paquets de trafic et en évaluant s'ils sont normaux ou malwares.

Dans cette expérience, l'utilisation d'appareils IoT réels s'est avérée difficile dans le cadre de la réalité. Par conséquent, ces appareils ont été remplacés par des machines virtuelles pour garantir la faisabilité et la cohérence de l'environnement de simulation. De même, le trafic malware a été substitué par du trafic normal afin de démontrer que l'IDS fonctionne correctement. Les trois machines virtuelles et une machine physique ont été configurées avec le système d'exploitation Ubuntu pour assurer une compatibilité et une homogénéité dans le déroulement de l'expérience.

Après cela, on a implémenté la base de test, déployé et démarré toutes les machines virtuelles de l'environnement, et on a envoyé le trafic de test de la machine physique comme un bot vers une autre virtuelle comme une appareil IoT, et exécuté les commandes nécessaires pour surveiller le trafic. On a utilisé la capacité intégrée de Tshark pour collecter le trafic et créer des fichiers PCAP. En conséquence, les fichiers PCAP ont été stockés pour analyse. Ces ajustements ont été mis en place en suivant les étapes suivantes :

1) Configuration de la connexion réseau personnalisée : Dans les paramètres réseau de toutes les machines virtuelles, on a sélectionné "Custom network connection" comme type de connexion, et on a spécifié le nom du réseau personnalisé 'VMnet1 Host-only', correspondant au réseau défini sur la machine physique.

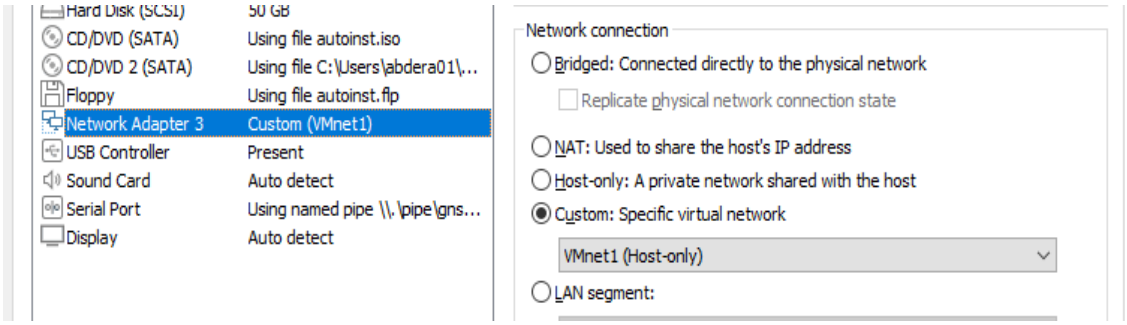


Figure 4.2. Sélection le type de connexion

2) Attribution des adresses IP : Les machines virtuelles obtiennent des adresses IP attribuées par le serveur DHCP de la machine physique. Cela assure une cohérence et une prévisibilité des adresses IP dans le réseau, facilitant la communication entre les machines.

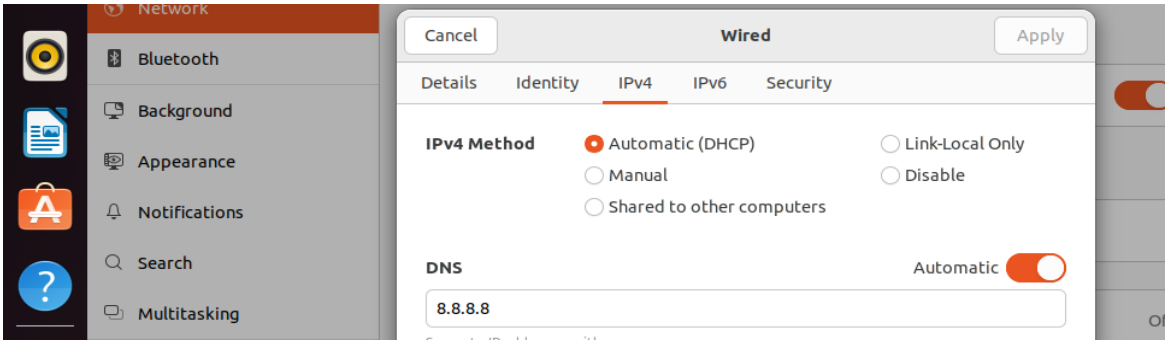


Figure 4.3. Attribution des adresses IP par le serveur DHCP

Ces adresses IP sont attribuées comme indiqué dans le tableau ci-dessous :

Machine	*Adresse IP*
Machine physique -Bot-	*192.168.46.1*
VM sniffer / IDS	*192.168.46.129*
VM 1 -Appareil IoT 1-	*192.168.46.130*
VM 2 -Appareil IoT 2-	*192.168.46.131*

Tableau 4.1. Attribution des adresses IP

3) On a envoyé des paquets de données à une machine virtuelle de manière répétée jusqu'à ce qu'on décidait de l'arrêter.

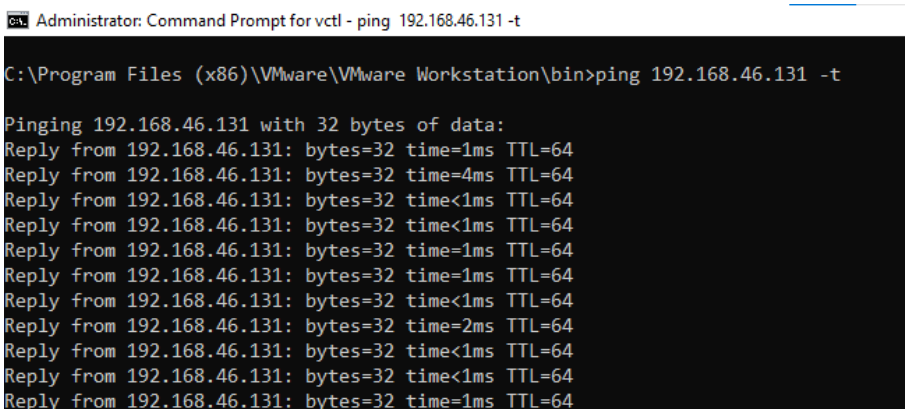


Figure 4.4. Envoie des paquets ICMP à une machine virtuelle

4) On a utilisé cette commande pour capturer le trafic réseau à l'aide de l'outil Tshark. Il fait partie de la suite d'outils d'analyse de réseau Wireshark.

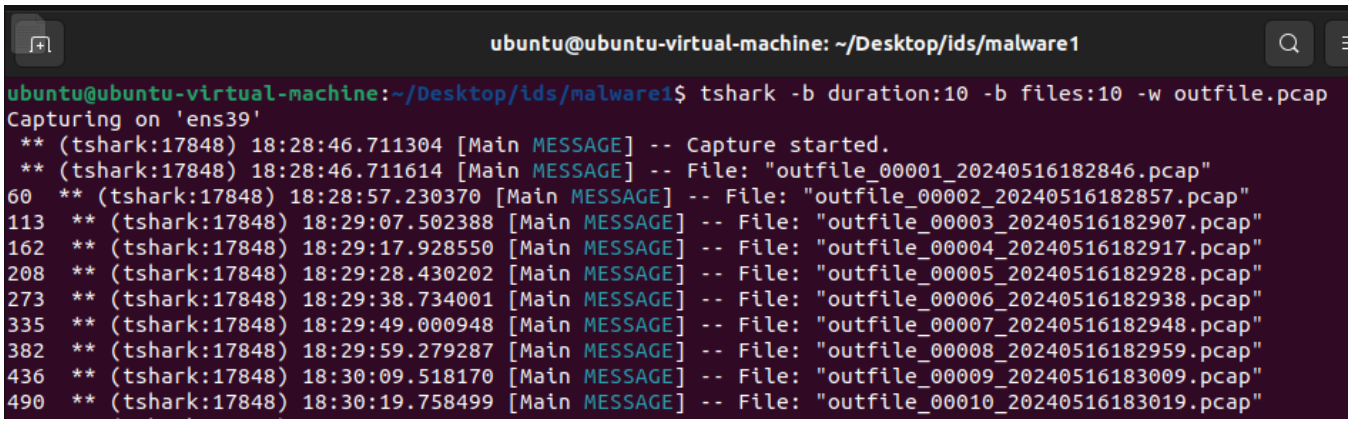


Figure 4.5. Capturer le trafic avec Tshark

Cette commande capturera 10 secondes de trafic réseau et l'enregistrera dans 10 fichiers PCAP distincts, avec une taille de fichier maximale de 100 Mo chacun.

➔*Décomposer la commande :*

- *tshark: C'est le nom de l'outil de ligne de commande utilisé pour capturer le trafic réseau.*
- *-b duration:10 : Cette option spécifie que la capture doit s'arrêter après 10 secondes.*
- *-b files: 10: Cette option spécifie que la capture doit être divisée en plusieurs fichiers, avec un maximum de 10 fichiers. Chaque fichier aura une taille d'environ 100 Mo.*

- *-w sortie.pcap: Cette option spécifie que le trafic capturé doit être enregistré dans un fichier nommé 'outfile.pcap' Ce fichier est au format PCAP (Portable Capture), qui peut être ouvert et analysé par Wireshark ou d'autres outils d'analyse de réseau.*

5) On a exécuté le script d'IDS qui est base sur le modèle AE-GRU. Une fois le code exécuté, le résultat de chaque fichier est affiché séparément. Le trafic généré pour les tests comprenait des pings normaux, utilisés pour démontrer la capacité du modèle à distinguer le trafic légitime, comme indiqué dans la sortie des résultats :

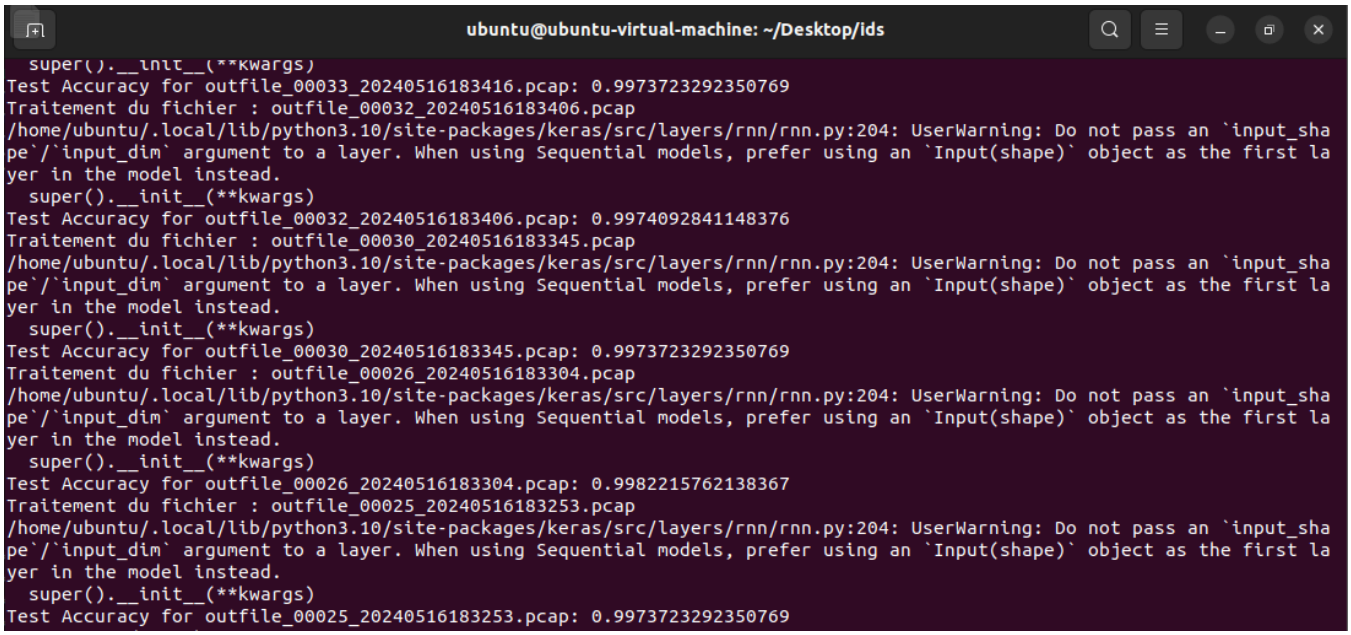


Figure 4.6. Exécution du script python -IDS-

Par exemple, la précision de 0.997 signifie que 99.7% des prédictions faites par le modèle étaient correctes. Cette haute précision indique que le modèle AE-GRU une excellente capacité à identifier correctement le trafic normal dans les conditions de test au moins dans les scénarios de test avec du trafic normal.

Notre modèle AE-GRU, après avoir été entraîné, génère des probabilités ou des scores pour chaque échantillon de trafic réseau. Ces scores indiquent à quel point l'échantillon est normal ou anormal.

Pour chaque échantillon de trafic réseau capturé et prétraité, le modèle produit un score. Un seuil est ensuite appliqué à ce score pour décider si le trafic est normal ou anormal. Si le score est

supérieur ou égal au seuil, l'échantillon est classé comme anormal, et si le score est inférieur au seuil, l'échantillon est classé comme normal.

Les résultats finaux de trafic pour notre expérience sont affichés ci-dessous :

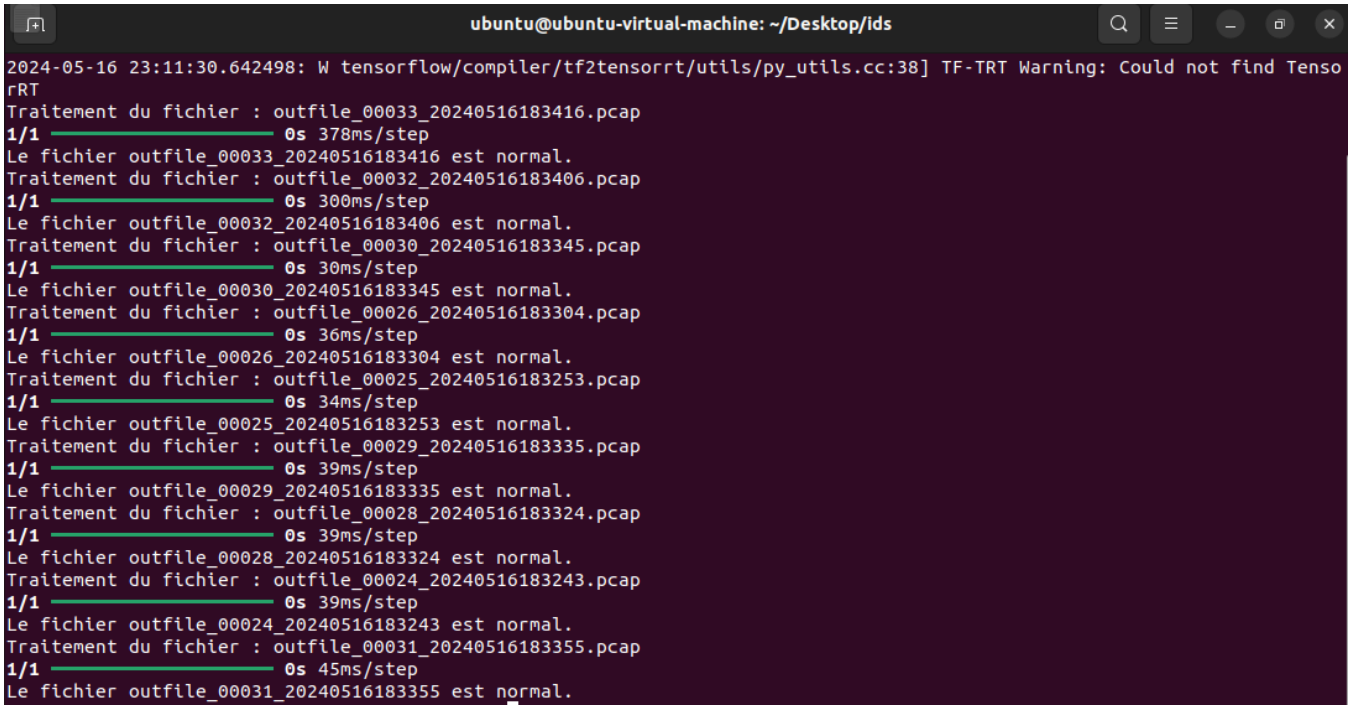


Figure 4.7. Exemples des résultats des fichiers

7. Limitations et Perspectives

Dans cette étude, nous avons fait face à différents défis, donc cette étude présente les limites suivantes qui devraient être surmontées pour améliorer le développement de la méthodologie proposée :

- *Pour installer et exécuter le programme Mirai [71], il est important de respecter les spécifications système recommandées par les développeurs du programme pour garantir une installation et une utilisation réussies. Cependant, notre ordinateur ne répond pas à ces exigences, il était donc préférable d'utiliser un trafic régulier*
- *Dans cette expérience, on ne pouvait pas utiliser d'appareils IoT physiques avec le trafic malware, on a donc implémenté un environnement virtuel. En fait, le coût aurait été très élevé si on avait*

utilisé de vrais appareils IoT physiques, car la répétition de l'expérience peut nécessiter de remplacer l'appareil affecté par un nouveau à chaque fois que nous répétons l'expérience.

Pour les travaux futurs, on espère tester notre modèle proposé avec divers ensembles de données IoT pour l'évaluer et améliorer également l'expérience pour créer un ensemble de données en capturant le trafic IoT sur les appareils IoT à l'aide d'un trafic normal et malveillant pour prouver l'exactitude de la prédiction du modèle.

8. Conclusion

Dans ce chapitre, nous avons mis en œuvre notre IDS basé sur le modèle AE-GRU dans un environnement de simulation. On a déployé cet IDS dans une machine virtuelle pour surveiller et détecter les activités anormales sur un réseau simulé et discuté des avantages et des limites de notre étude, tout en identifiant les perspectives d'amélioration et les futures pistes de recherche pour notre IDS.

Conclusion Générale

Conclusion Générale

Les cyberattaques augmentent de jour en jour et constituent une réelle menace pour les entreprises et les systèmes, et les développeurs de systèmes de cybersécurité modernes s'efforcent de développer de nouvelles technologies de détection des intrusions basées sur de nouvelles méthodes. L'une des plus dangereuses de ces attaques est celle des botnet, qui est l'objet de notre mémoire.

Dans ce mémoire, on a fourni les quelques informations générales sur les environnements IoT et ces technologies, ensuite on a abordé les attaques Botnets et ses phases de cycle de vie en détails, ces phases ont été divisées en deux étapes principales précoce et tardive. Dans le deuxième chapitre on a vu que la plupart des chercheurs et les travailles se sont concentrés sur la détection dans la dernière étape et c'est ce qu'on a fait travailler sur la détection à une étape précoce.

Dans le troisième chapitre, on a essayé de réaliser un système de détection des attaques de botnet afin de protéger les systèmes de ce type d'attaques, lors de la mise en œuvre de ce projet, on a un nouvel ensemble de données appelé ''MedBIoT'' qui se concentre sur les étapes initiales (propagation et communication) de ces attaques et après avoir traité les données de manière à les rendre faciles à utiliser et à obtenir de bons résultats, on a proposé et développé un modèle hybride AE-GRU d'apprentissage en profondeur qui a donné des résultats élevés : le précision, le rappel, Le score F1 de 99% et le taux de faux positifs de 1%.

Dans le dernier chapitre, on a développé le modèle proposé à un IDS et élaboré un scénario composé d'un ensemble d'appareils IoT jouant les rôles de bot, de sniffer, d'IDS et d'autres jouant le rôle de l'environnement IoT cible.

On a essayé ce scénario dans un environnement virtuel et en utilisant uniquement le trafic normal pour prouver son efficacité dans la précision. Après tout cela, on a obtenu de bons résultats qui ont été discutés dans les chapitres précédent

Bibliographie

Références :

[1] R. Parashar and A. Khan, "A SURVEY: THE INTERNET OF THINGS," vol. 4, no. 3, 2016.

[2] K. K. Goyal, A. Garg, A. Rastogi, and S. Singhal, "A Literature Survey on Internet of Things (IoT)," vol. 09, no. 06, 2018.

[3] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment," in 2014 International Conference on Science Engineering and Management Research (ICSEMR), Chennai, India: IEEE, Nov. 2014, pp. 1–8. doi: 10.1109/ICSEMR.2014.7043637.

[4] A. Agniel, D. Arnold, and J. Saniie, "Image Processing for Detecting Botnet Attacks: A Novel Approach for Flexibility and Scalability," in 2022 IEEE International Conference and Expo on Real Time Communications at IIT (RTC), Chicago, IL, USA: IEEE, Oct. 2022, pp. 8–12. doi: 10.1109/RTC56148.2022.9945055.

[5] O. Habibi, M. Chemmakha, and M. Lazaar, "Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection," Engineering Applications of Artificial Intelligence, vol. 118, p. 105669, Feb. 2023, doi: 10.1016/j.engappai.2022.105669.

[6] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, and Hui-Ying Du, "Research on the architecture of Internet of Things," in 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, China: IEEE, Aug. 2010, pp. V5-484-V5-487. doi: 10.1109/ICACTE.2010.5579493.

[7] P. F. Lamas, "Enabling technologies and cyber-physical systems for mission-critical scenarios".

[8] M. A. Jabraeil Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, Towards the Internet of Things: Architectures, Security, and Applications. in EAI/Springer Innovations in Communication and Computing. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-18468-1.

[9] M. Bagga, P. Thakral, and T. Bagga, "A Study on IoT: Model, Communication Protocols, Security Hazards & Countermeasures," in 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), Solan Himachal Pradesh, India: IEEE, Dec. 2018, pp. 591–598. doi: 10.1109/PDGC.2018.8745984.

[10] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," IEEE Commun. Surv. Tutorials, vol. 21, no. 3, pp. 2671–2701, 2019, doi: 10.1109/COMST.2019.2896380.

[11] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, "Internet of Things Botnet Detection Approaches: Analysis and Recommendations for Future Research," Applied Sciences, vol. 11, no. 12, p. 5713, Jun. 2021, doi: 10.3390/app11125713.

[12] C.-L. Zhong, Z. Zhu, and R.-G. Huang, "Study on the IOT Architecture and Gateway Technology," in 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Guiyang, China: IEEE, Aug. 2015, pp. 196–199. doi: 10.1109/DCABES.2015.56.

[13] C. Musonda, "Security, Privacy and Integrity in Internet of Things – A Review," 2018.

[14] Ming-Chin Chuang and Jeng-Farn Lee, "TEAM: Trust-Extended Authentication Mechanism for Vehicular Ad Hoc Networks," IEEE Systems Journal, vol. 8, no. 3, pp. 749–758, Sep. 2014, doi: 10.1109/JSYST.2012.2231792.

[15] X. Yang, L. Shu, Y. Liu, G. P. Hancke, M. A. Ferrag, and K. Huang, "Physical Security and Safety of IoT Equipment: A Survey of Recent Advances and Opportunities," IEEE Trans. Ind. Inf., vol. 18, no. 7, pp. 4319–4330, Jul. 2022, doi: 10.1109/TII.2022.3141408.

[16] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," IEEE Access, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[17] H. Hamid et al., "IoT-based botnet attacks systematic mapping study of literature," Scientometrics, vol. 126, no. 4, pp. 2759–2800, Apr. 2021, doi: 10.1007/s11192-020-03819-5.

[18] J. Vania, A. Meniya, and H. B. Jethva, "A Review on Botnet and Detection Technique," International Journal of Computer Trends and Technology, 2013.

[19]"Multilayer_Framework_for_Botnet_Detection_Using_Machine_Learning_Algorithms.pdf."

[20] V. A. Memos and K. E. Psannis, "AI-Powered Honeypots for Enhanced IoT Botnet Detection," in 2020 3rd World Symposium on Communication Engineering (WSCE), Thessaloniki, Greece: IEEE, Oct. 2020, pp. 64–68. doi: 10.1109/WSCE51339.2020.9275581.

[21] H. Zhao, H. Shu, and Y. Xing, "A Review on IoT Botnet," in The 2nd International Conference on Computing and Data Science, Stanford CA USA: ACM, Jan. 2021, pp. 1–7. doi: 10.1145/3448734.3450911.

[22] [En ligne]. Available: https://github.com/topics/hydra-botnet

[23] [En ligne]. Available: https://github.com/v0idp/PsyBot

[24] [En ligne]. Available: https://github.com/ifding/iot-malware/blob/master/BASHLITE/client.c

[25] [En ligne]. Available: https://github.com/jgamblin/Mirai-Source-Code

[26] [En ligne]. Available: https://github.com/rawbypa/BrickerBot

[27] [En ligne]. Available: https://malpedia.caad.fkie.fraunhofer.de/details/elf.hajime

[28] [En ligne]. Available: https://www.opensourceforu.com/tag/wirex-botnet/

[29] [En ligne]. Available: https://github.com/0x1CA3/Reaper

[30] [En ligne]. Available: https://github.com/tori-bot

[31] [En ligne]. Available: https://github.com/topics/mozi

[32] A. Spognardi, M. D. Donno, N. Dragoni, and A. Giaretta, "Analysis of DDoS-Capable IoT Malwares," presented at the 2017 Federated Conference on Computer Science and Information Systems, Sep. 2017, pp. 807–816. doi: 10.15439/2017F288.

[33] M. H. Syed, E. B. Fernandez, and J. Moreno, "A misuse Pattern for DDoS in the IoT," in Proceedings of the 23rd European Conference on Pattern Languages of Programs, Irsee Germany: ACM, Jul. 2018, pp. 1–5. doi: 10.1145/3282308.3282343.

[34] A. Arshad et al., "A novel ensemble method for enhancing Internet of Things device security against botnet attacks," Decision Analytics Journal, vol. 8, p. 100307, Sep. 2023, doi: 10.1016/j.dajour.2023.100307.

[35] C. A. Fadhilla, M. D. Alfikri, and R. Kaliski, "Lightweight Meta-Learning BotNet Attack Detection," IEEE Internet Things J., vol. 10, no. 10, pp. 8455–8466, May 2023, doi: 10.1109/JIOT.2022.3229463.

[36] S. Lee, A. Abdullah, N. Jhanjhi, and S. Kok, "Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning," PeerJ Computer Science, vol. 7, p. e350, Jan. 2021, doi: 10.7717/peerj-cs.350.

[37] F. Hussain et al., "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," IEEE Access, vol. 9, pp. 163412–163430, 2021, doi: 10.1109/ACCESS.2021.3131014.

[38] W. N. H. Ibrahim et al., "Multilayer Framework for Botnet Detection Using Machine Learning Algorithms," IEEE Access, vol. 9, pp. 48753–48768, 2021, doi: 10.1109/ACCESS.2021.3060778.

[39] "Using a Multi- Layered Framework for Botnet Detection Based on Machine Learning Algorithms," International Journal of Intelligent Systems and Applications in Engineering.

[40] S. Gupta and B. Singh, "An Intelligent Multi-Layer Framework with SHAP Integration for Botnet Detection and Classification," Computers & Security, p. 103783, Feb. 2024, doi: 10.1016/j.cose.2024.103783.

[41] N. S. Akash, S. Rouf, S. Jahan, A. Chowdhury, and J. Uddin, "Botnet Detection in IoT Devices Using Random Forest Classifier with Independent Component Analysis," Journal of Information and Communication Technology, vol. 21, 2022, doi: 10.32890/jict2022.21.2.3.

[42] N. Hasan, Z. Chen, C. Zhao, Y. Zhu, and C. Liu, "IoT Botnet Detection framework from Network Behavior based on Extreme Learning Machine," in IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA: IEEE, May 2022, pp. 1–6. doi: 10.1109/INFOCOMWKSHPS54753.2022.9798307.

[43] J. Velasco-Mata, V. González-Castro, E. Fidalgo, and E. Alegre, "Real-time botnet detection on large network bandwidths using machine learning," Sci Rep, vol. 13, no. 1, p. 4282, Mar. 2023, doi: 10.1038/s41598-023-31260-0.

[44] S. Srinivasan and D. P, "Enhancing the security in cyber-world by detecting the botnets using ensemble classification based machine learning," Measurement: Sensors, vol. 25, p. 100624, Feb. 2023, doi: 10.1016/j.measen.2022.100624.

[45] H. Alkahtani and T. H. H. Aldhyani, "Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications," Security and Communication Networks, vol. 2021, pp. 1–23, Sep. 2021, doi: 10.1155/2021/3806459.

[46] A. Agniel, D. Arnold, and J. Saniie, "Image Processing for Detecting Botnet Attacks: A Novel Approach for Flexibility and Scalability," in 2022 IEEE International Conference and Expo on Real Time Communications at IIT (RTC), Chicago, IL, USA: IEEE, Oct. 2022, pp. 8–12. doi: 10.1109/RTC56148.2022.9945055.

[47] B. Bojarajulu, S. Tanwar, and T. P. Singh, "Intelligent IoT-BOTNET attack detection model with optimized hybrid classification model," Computers & Security, vol. 126, p. 103064, Mar. 2023, doi: 10.1016/j.cose.2022.103064.

[48] S. Verma et al., "DNNBoT: Deep Neural Network-Based Botnet Detection and Classification," Computers, Materials & Continua, vol. 71, no. 1, pp. 1729–1750, 2022, doi: 10.32604/cmc.2022.020938.

[49] M. Al-Fawa'reh, J. Abu-Khalaf, P. Szewczyk, and J. J. Kang, "MalBoT-DRL: Malware Botnet Detection Using Deep Reinforcement Learning in IoT Networks," IEEE Internet Things J., pp. 1–1, 2023, doi: 10.1109/JIOT.2023.3324053.

[50]"GANIBOT_A_Network_Flow_Based_Semi_Supervised_Generative_Adversarial_Networks_Model_for_IoT_Botnets_Detection.pdf."

[51] N. Elsayed, Z. ElSayed, and M. Bayoumi, "IoT Botnet Detection Using an Economic Deep Learning Model." arXiv, May 28, 2023. Accessed: Feb. 09, 2024. [Online]. Available: http://arxiv.org/abs/2302.02013

[52] S. Rabhi, T. Abbes, and F. Zarai, "Transfer learning-based Mirai botnet detection in IoT networks," in 2023 International Conference on Innovations in Intelligent Systems and Applications (INISTA), Hammamet, Tunisia: IEEE, Sep. 2023, pp. 1–5. doi: 10.1109/INISTA59065.2023.10310379.

[53] Y. Yang and L. Wang, "LGANet: Local Graph Attention Network for Peer-to-Peer Botnet Detection," in 2021 3rd International Conference on Advances in Computer Technology, Information Science and Communication (CTISC), Shanghai, China: IEEE, Apr. 2021, pp. 31–36. doi: 10.1109/CTISC52352.2021.00013.

[54] M. Wazzan, D. Algazzawi, A. Albeshri, S. Hasan, O. Rabie, and M. Z. Asghar, "Cross Deep Learning Method for Effectively Detecting the Propagation of IoT Botnet," Sensors, vol. 22, no. 10, p. 3895, May 2022, doi: 10.3390/s22103895.

[55] [En ligne]. Available: https://neptune.ai/blog/autoencoders-case-study-guide

[56] [En ligne]. Available: https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2

[57] [En ligne]. Available : https://cs.taltech.ee/research/data/medbiot/

[58] [En ligne]. Available : https://www.virtualizationhowto.com/2022/11/vmware-workstation-17-pro-new-features-and-download-link/

[59] [En ligne]. Available : https://en.wikipedia.org/wiki/Ubuntu

[60] [En ligne]. Available : https://en.wikipedia.org/wiki/Python_%28programming-language%29

[61] [En ligne]. Available : https://colab.google/

[62] [En ligne]. Available : https://www.lucidchart.com/pages/examples/process management-software

[63] [En ligne]. Available : https://linux.die.net/man/1/tshark#:~ :text=TShark%20is%20a%20network%20protocol,the%20packets%20to%20a%20file.

[64] [En ligne]. Available : https://en.wikipedia.org/wiki/NumPy

[65] [En ligne]. Available : https://en.wikipedia.org/wiki/Pandas_(software)

[66] [En ligne]. Available : https://en.wikipedia.org/wiki/Scapy

[67] [En ligne]. Available : https://scikit-learn.org

[68] [En ligne]. Available : https://www.tensorflow.org/learn

[69] [En ligne]. Available : https://en.wikipedia.org/wiki/Keras

[70] [En ligne]. Available : https://en.wikipedia.org/wiki/Matplotlib

[71] [En ligne]. Available: https://github.com/ruCyberPoison/-Mirai-IotBotNet/blob/master/TUTORIAL.txt