

# Sécurisation de réseaux en environnement industriel : prototype de switch intelligent

Maxence Lannuzel

▶ **To cite this version:**

Maxence Lannuzel. Sécurisation de réseaux en environnement industriel : prototype de switch intelligent. Systèmes et contrôle [cs.SY]. Université de Brest, 2025. Français. ⟨NNT : 2025BRUN0032⟩. ⟨tel-05505278⟩

**HAL Id: tel-05505278**

**https://theses.hal.science/tel-05505278v1**

Submitted on 11 Feb 2026

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

ÉCOLE DOCTORALE N° 644
*Mathématiques et Sciences et Technologies de l'Information et de la Communication en Bretagne Océane*
Spécialité : Informatique et architecture numérique

Par
**Maxence LANNUZEL**

# Sécurisation de réseaux en environnement industriel : prototype de switch intelligent

**Thèse présentée et soutenue à l'École Navale, Lanvéoc, le 25/06/2025**
**Unité de recherche : Ecole Navale Institut de recherche (IRENav) EA3634**
**Thèse financée dans le cadre d'une convention CIFRE par Interface Concept**

**Rapporteurs avant soutenance :**

| | |
|---|---|
| Gilles GRIMAUD | Professeur, Université de Lille |
| Françoise SAILHAN | Professeure, IMT-Atlantique |

**Composition du Jury :**

| | | |
|---|---|---|
| Président : | Roland GAUTIER | Professeur, Université de Bretagne Occidentale |
| Examinateurs : | Gilles BUREL | Professeur, Université de Bretagne Occidentale |
| | Jean-Christophe CEXUS | Enseignant-chercheur, ENSTA Bretagne |
| | Gilles GRIMAUD | Professeur, Université de Lille |
| | Olivier JACQ | Conseiller en cybersécurité maritime, France Cyber Maritime |
| | Françoise SAILHAN | Professeure, IMT-Atlantique |
| Dir. de thèse : | David BROSSET | Maître de Conférences HDR, IRENAV |

**Invité(s) :**

| | |
|---|---|
| Dominique LETTY | Ingénieur R&D, Interface Concept |

# TABLE DES MATIÈRES

**Liste des acronymes** **7**

**Liste des figures** **12**

**Liste des tableaux** **14**

**Remerciements** **15**

**Introduction** **17**

Contexte général et problématique . . . . . 17
Objectifs de la recherche . . . . . 21
Questions de recherche . . . . . 22
Organisation du manuscrit . . . . . 23

**1 États de l'art** **27**

1.1 Architectures réseaux IT/OT . . . . . 29
1.1.1 Modèles de référence . . . . . 29
1.1.2 Sécurité et défense en profondeur . . . . . 31
1.1.3 Acteurs et protocoles des systèmes industriels . . . . . 33
1.2 Le trafic réseau : concepts et manipulation . . . . . 35
1.2.1 Acheminement du trafic . . . . . 35
1.2.2 Génération du trafic . . . . . 38
1.2.3 Capture et enregistrement du trafic . . . . . 39
1.2.4 Intérêt et objectifs . . . . . 40
1.3 Méthodes classiques d'analyse du trafic . . . . . 41
1.3.1 Analyse par couches (modèle OSI ou TCP/IP) . . . . . 41
1.3.2 Utilisation de bases de données protocolaires . . . . . 42
1.3.3 Segmentation des paquets . . . . . 42
1.3.4 Analyse par inspection des Flow . . . . . 42
1.3.5 Analyse par Deep Packet Inspection (DPI) . . . . . 44
1.3.6 Heuristiques et analyse avancée . . . . . 44
1.4 Classification réseau . . . . . 44
1.4.1 Détection et supervision : IDS, QoS, surveillance . . . . . 45

1.4.2 Machine Learning et IA pour l'analyse du trafic . . . . . 46
1.4.3 Classification du trafic : attaques, services, activités . . . . . 50
1.5 Analyse des Jeux de données existants . . . . . 55
1.5.1 Jeux de données orientés détection d'intrusion . . . . . 55
1.5.2 Synthèse des jeux de données réseau pour IDS . . . . . 60
1.5.3 Jeux de données orienté analyse de trafic réseau . . . . . 62
1.6 Activité humaine dans les réseaux . . . . . 64
1.6.1 Théorie de l'activité . . . . . 64
1.6.2 Différences entre l'homme et la machine . . . . . 66
1.6.3 Transposition de la théorie de l'activité à l'activité réseau . . . . . 67
1.7 Conclusion . . . . . 67

**2 Modélisation de l'activité humaine dans les réseaux 69**
2.1 Modélisation de la classification réseau . . . . . 71
2.1.1 Modèles théoriques de communications réseau . . . . . 72
2.1.2 Modèle d'analyseur réseau . . . . . 74
2.1.3 Modélisation de la granularité de l'activité . . . . . 78
2.2 Modélisation de l'activité humaine dans les réseaux informatiques . . . . . 84
2.2.1 Rappels théoriques . . . . . 84
2.2.2 Portage de la théorie de l'activité . . . . . 85
2.2.3 Application de la modélisation . . . . . 89
2.3 Conclusion . . . . . 92

**3 Génération de données pour l'analyse d'activités réseaux 93**
3.1 Définitions et scénarios . . . . . 95
3.2 Étiquetage des données . . . . . 97
3.2.1 TOMATO : Tools for Optimized Manual Assistance on Tagging Operations 98
3.2.2 SALAD : Script-based Automatic Labelling on Activity Dataset . . . . . 103
3.2.3 GAR7IC : Generalized Automatic Review and Labelling for S7COMM Industrial Communication . . . . . 106
3.2.4 Automatisation de l'étiquetage des données avec LLM . . . . . 109
3.3 Génération de données réalistes . . . . . 114
3.3.1 Enregistrement et rejeu des scénarios . . . . . 114
3.3.2 BASIL : Behavioral Action Scenario with Image Labelling . . . . . 122
3.4 Conclusion . . . . . 127

**4 Analyses et applications 129**
4.1 Proposition d'une solution intégrée au Switch . . . . . 131

4.1.1 Contexte et motivation . . . 131
4.1.2 Objectifs de la solution proposée . . . 132
4.1.3 Architecture fonctionnelle du switch "intelligent" . . . 133
4.1.4 Administration et supervision au sein du switch . . . 135
4.2 Analyse du trafic classique . . . 137
4.2.1 Comparaison des stratégies d'étiquetage : manuel vs automatique . . . 138
4.2.2 Extraction des caractéristiques . . . 140
4.2.3 Premières analyses et résultats . . . 143
4.2.4 La particularité du trafic WEB . . . 145
4.3 Analyse du trafic chiffré par VPN . . . 147
4.3.1 Principes d'un flow VPN . . . 148
4.3.2 Analyse de trafic Wireguard . . . 152
4.4 Analyse du trafic industriel . . . 155
4.4.1 Caractéristiques d'un flow industriel . . . 156
4.4.2 Analyse de trafic industriel S7COMM . . . 158
4.5 Conclusion . . . 160

**Conclusion et perspectives** **161**
Rappel de la problématique . . . 161
Contributions . . . 161
Perspectives . . . 163

**Liste des publications** **167**

**Annexes** **169**

**A Téléchargement des Datasets** **169**
A.1 IDS . . . 169
A.2 Analyse du trafic réseau . . . 170

**B Liens entre les différents auteurs des jeux de données** **172**

**C Captures d'écran** **173**

**Bibliographie** **177**

# Liste des acronymes

| | |
|---|---|
| ABB | ASEA Brown Boveri |
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| API | Application Programming Interface |
| ARP | Address Resolution Protocol |
| ARPANET | Advanced Research Projects Agency network |
| BASIL | Behavioral Action Scenario with Image Labelling |
| CCNA | Cisco Certified Network Associate |
| CIFRE | Conventions industrielles de formation par la recherche |
| CNN | Convolutional Neural Network |
| CPU | Central Proccessing Unit |
| CSV | Comma Separated Values |
| DCS | Distributed Control System |
| DDoS | Distributed Denial of Service |
| DFI | Deep Flow Inspection |
| DL | Deep Learning |
| DMZ | Demilitarized Zone |
| DNS | Domain Name System |
| DPI | Deep Packet Inspection |
| DoS | Denial of Service |
| FTP | File Transfer Protocol |
| GAR7IC | Generalized Automatic Review and Labelling for S7COMM Industrial Communication |
| HIDS | Host Intrusion Detection System |
| HMI | Human Machine Interface |
| HMS | Hardware Meets Software |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hyper Text Transfer Protocol Secure |
| IA | Intelligence Artificielle |
| ICMP | Internet Control Message Protocol |
| ICN | Information Centric Networking |
| ICS | Industrial Control System |
| IEC | International Electrotechnical Commission |

| | |
|---|---|
| IDS | Intrusion Detection System |
| IEEE | Institute of Electrical and Electronics Engineers |
| IGMP | Internet Group Management Protocol |
| IHM | Interface Homme Machine |
| IIoT | Industrial Internet of Things |
| IMAP | Internet Message Access Protocol |
| INES | International Nuclear Event Scale |
| iOS | iPhone Operating System |
| IoT | Internet of Things |
| IP | Internet Protocol |
| IPFIX | IP Flow Information Export |
| IPS | Intrusion Prevention System |
| ISO | International Organization for Standardization |
| IT | Information Technology |
| LAN | Local Area Network |
| LLC | Logical Link Control |
| LLM | Large Language Model |
| MAC | Media Access Control |
| MAN | Metropolitan Area Network |
| MDNS | Multicast Domain Name System |
| ML | Machine Learning |
| MQTT | Message Queuing Telemetry Transport |
| MTU | Maximum Transmission Unit |
| NB | Naïve Bayes |
| NAT | Network Address Translation |
| NIDS | Network Intrusion Detection System |
| NIS | Network and Information Security |
| OS | Operating System |
| OSI | Open Systems Interconnection |
| OT | Operational Technology |
| P2P | Peer-to-Peer |
| PDU | Protocol Data Unit |
| PCAP | Packet CAPture |
| PCAPNG | Packet CAPture New Generation |
| PLC | Programmable Logic Controller |
| QD | Question de développement |
| QoS | Quality of Service |

| | |
|---|---|
| QR | Question de recherche |
| R&D | Recherche et Développement |
| RF | Random Forest |
| RFC | Requests for comments |
| RINA | Recursive InterNetwork Architecture |
| RNN | Recurrent Neural Network |
| S7COMM | S7 Communication |
| SALAD | Script-based Automatic Labelling on Activity Dataset |
| SANS | SysAdmin, Audit, Network, Security |
| SCADA | Supervisory Control and Data Acquisition |
| SCTP | Stream Control Transmission Protocol |
| SIEM | Security Information and Event Management |
| SDN | Software Defined Network |
| SMTP | Simple Mail Transfer Protocol |
| SNAP | Sub-Network Access Protocol |
| SOC | Security Operations center |
| SVM | Support Vector Machine |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| TOMATO | Tools for Optimized Manual Assistance on Tagging Operations |
| TOR | The Onion Router |
| UDP | User Datagram Protocol |
| USB | Universal Serial Bus |
| VLAN | Virtual Local Area Network |
| VM | Virtual Machine |
| VoIP | Voice over IP |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| WWW | World Wide Web |
| YAML | Yet Another Markup Language |

# TABLE DES FIGURES


1 PLC Siemens S7-300 . . . . . 18
2 Relations entre l'IT et l'OT . . . . . 18
3 Évolution des publications sur Google Scholar sur la cybersécurité (2015-2024) par recherche des mots dans les articles (mots clés, résumé, texte, titre) . . . . . 20
4 Évolution des publications sur Scopus sur la cybersécurité (2015-2024) par recherche des mots dans le titre, résumé ou les mots-clés . . . . . 20
5 Carte ComEth4000e d'Interface Concept . . . . . 21
6 Liens entre les domaines et les sujets abordés dans la thèse . . . . . 24

1.1 Niveaux d'une architecture de Purdue[1] . . . . . 30
1.2 Modèle de référence ICS410 [2] . . . . . 31
1.3 Échelle INES . . . . . 32
1.4 Démarche de mise en évidence des lignes de défense . . . . . 32
1.5 Répartition des revenus générés par l'automatisation industrielle (2024) . . . . . 34
1.6 Représentation et étendue d'un LAN, MAN et WAN . . . . . 36
1.7 Carte ComEth4000e d'Interface Concept . . . . . 37
1.8 Architecture de la ComEth4000e . . . . . 38
1.9 Positionnement des IDS en fonction de leur type . . . . . 46
1.10 Relations entre IA, ML et DL . . . . . 48
1.11 Séparation de deux classes avec SVM . . . . . 49
1.12 Réseau de neurones à 3 couches . . . . . 50
1.13 Matrice ATT&CK de MITRE . . . . . 51
1.14 Suivi d'activités sur iOS . . . . . 53
1.15 Modèle hiérarchique de Leontiev . . . . . 65
1.16 Modèle général de l'activité de Engeström [3] . . . . . 66

2.1 Modèle OSI (à gauche) et modèle TCP/IP moderne (à droite) . . . . . 73
2.2 Détails d'une trame Ethernet II encapsulant du Modbus . . . . . 74
2.3 Champs de l'en-tête Ethernet II et IEEE 802.3 . . . . . 74
2.4 Modélisation du traitement des trames . . . . . 75
2.5 Modélisation d'un analyseur réseau traitant une trame Ethernet . . . . . 76
2.6 Modélisation d'un analyseur réseau intégré pour flows . . . . . 77
2.7 Pyramide des granularités de classification d'activités . . . . . 80

2.8 Modèle de l'outil réseau . . . 86
2.9 Modèle de l'activité humaine dans les réseaux . . . 87
2.10 Niveaux de chaque empreinte . . . 89
2.11 Modélisation de l'activité mail . . . 91

3.1 Piliers de la donnée d'apprentissage . . . 96
3.2 Méthodologie d'enregistrement et de traitement . . . 100
3.3 Distribution des activités et applications dans le jeu de données NAT-U1-2024 . . . 101
3.4 Architecture de l'enregistreur . . . 104
3.5 Exemple de fichier YAML décrivant la configuration d'un automate Siemens . . . 108
3.6 Principe des 4R (Hernández et Colom - 2023[4]) . . . 115
3.7 Architecture de l'hyperviseur type 1, l'hyperviseur type 2 et la conteneurisation . 117
3.8 Enregistrement et étiquetage des applications sous QubesOS . . . 120
3.9 Architecture du logiciel BASIL . . . 124
3.10 Positionnement des outils développés vis-à-vis de la donnée . . . 127

4.1 Architecture fonctionnelle du switch intelligent . . . 136
4.2 Interface web d'administration intégrée au switch intelligent . . . 137
4.3 Supervision temps réel du trafic au sein du switch . . . 138
4.4 Répartition et nombre d'occurences des étiquetages manuel et automatique. . . . 139
4.5 Top10 des caractéristiques de RandomForest (sur 1) . . . 143
4.6 Évolution de l'accuracy en fonction du nombre de caractéristiques utilisées . . . 144
4.7 Matrice de confusion des caractéristiques heuristiques . . . 145
4.8 Matrice de confusion des caractéristiques obtenues par filtrage . . . 146
4.9 Matrice de confusion des caractéristiques calculée par RandomForest . . . 147
4.10 Format et encapsulation du protocole Wireguard . . . 149
4.11 Quelques formats et mode d'encapsulation des protocoles OpenVPN et IPsec . . . 150
4.12 Taille de chaque paquet pour le trafic en clair et chiffré par Wireguard . . . 151
4.13 Quantité de données par seconde pour le trafic en clair et chiffré par Wireguard . 151
4.14 Matrice de confusion des données chiffrées . . . 154
4.15 Top 10 des caractéristiques pour le trafic VPN . . . 155
4.16 Détails du protocole S7COMM de Siemens . . . 157
4.17 Évolution du trafic en fonction des automates . . . 159
4.18 Résumé des contributions . . . 163
4.19 Résumé des perspectives . . . 165

B.1 Liens entre les différents auteurs des jeux de données . . . 172

C.1 Menu du logiciel SALAD . . . 173

C.2 Menu du logiciel SALAD après déploiement du conteneur . . . . . 173
C.3 Interface WEB du logiciel BASIL . . . . . 174
C.4 Capture d'écran de la documentation Docker . . . . . 174
C.5 Capture d'écran de PyCharm . . . . . 175
C.6 Capture d'écran de Thunderbird et OnlyOffice . . . . . 175

# Liste des tableaux


1 Correspondance entre chapitres et questions de recherche . . . . . 23

1.1 Répartition des revenus des grandes entreprises dans l'automatisation industrielle (2024) . . . . . 33
1.2 Comparaison entre Switch L2 et Routeur L3 . . . . . 37
1.3 Comparaison des logiciels de génération de trafic réseau . . . . . 39
1.4 Tableau de comparaison entre le NetFlow et le sFlow . . . . . 43
1.5 Jeux de données contenant des attaques dans les réseaux *classiques* . . . . . 58
1.6 Jeux de données contenant des attaques dans les réseaux mobiles . . . . . 59
1.7 Jeux de données contenant des attaques dans les réseaux industriels . . . . . 59
1.8 Jeux de données contenant des attaques dans les réseaux IoT . . . . . 60
1.9 Jeux de données de trafic réseau . . . . . 62
1.10 Jeux de données de trafic réseau . . . . . 63

2.1 Jeux de données triés par granularité . . . . . 83
2.2 Comparaison des deux objectifs de la modélisation . . . . . 88
2.3 Résumé des flows lors d'un envoie de mail . . . . . 90

3.1 Description des champs du format de métadonnée . . . . . 102
3.2 Comparaison des méthodes de génération de données . . . . . 106
3.3 Liste des champs extraits par GAR7IC . . . . . 109
3.4 Exemple d'extraction de données S7COMM par GAR7IC . . . . . 110
3.5 Classification des activités pour les trois images par ChatGPT . . . . . 112
3.6 Classification des activités pour les trois images par Mistral . . . . . 112
3.7 Comparaison des méthodes d'enregistrement et de rejeu des scénarios . . . . . 122
3.8 Résumé des commandes du langage utilisé dans le logiciel. . . . . 125
3.9 Comparaison entre BASIL, SikuliX et PyAutoGUI . . . . . 126

4.1 Scores des variables selon différentes méthodes de sélection de caractéristiques . . 142
4.2 Comparaison des performances selon la méthode de sélection de caractéristiques 145
4.3 Association entre les flows chiffrés (WireGuard) et les flows en clair encapsulés . 148

A.1 Statuts et liens de téléchargement des Datasets d'IDS . . . . . 170
A.2 Statuts et liens de téléchargement des Datasets d'analyse de trafic . . . . . 171

# Remerciements

Avant de parler de tout ce que j'ai appris pendant cette thèse, je dois commencer par dire merci à ceux qui ont rendu ce parcours possible.

Je remercie en tout premier lieu David Brosset, mon directeur de thèse... et surtout complice de longue date. Nous avions déjà travaillé ensemble avant cette aventure, et honnêtement, je ne me voyais pas me lancer là-dedans sans un encadrement en qui j'avais confiance. C'était simple : c'était avec lui ou c'était pas du tout. Honoré (et un peu désolé quand même) d'avoir été son premier doctorant. Il a même dû passer son HDR pour pouvoir m'encadrer et je ne sais pas lequel de nous deux a le plus souffert dans l'histoire. Merci David pour ton soutien, ta patience (ou la mienne peut-être ?), et ton humour même dans les moments difficiles.

Je remercie également Dominique Letty pour son encadrement chez Interface Concept, et toute l'équipe d'Interface Concept justement, pour m'avoir accompagné tout au long de ces années. Que ce soit par leur soutien matériel, personnel, ou moral. Toujours prêts à me demander si j'avais besoin de quelque chose.

Un grand merci aussi à mes collègues de la Chaire de Cyberdéfense des Systèmes Navals, avec qui on a (un peu) travaillé, (beaucoup) rigolé, et avec qui les liens se sont tissés bien au-delà du travail. Ce n'est pas parce que la thèse se termine qu'on va arrêter de se voir.

Merci à Clet, l'ami de toujours, ancien de la Chaire lui aussi, compagnon d'aventure depuis nos années fac. Merci pour ton soutien, tes conseils, et ton écoute pendant ces années, et même avant, quand c'était moi qui t'épaulais dans tes travaux. Le "binôme" se reformera, j'en suis sûr, ce n'est juste qu'une question de temps.

Je remercie ma famille, et en particulier ma sœur et ma mère, pour leur soutien constant. Mention spéciale pour ma mère qui a accepté de me loger, nourrir et blanchir pendant mes périodes à Quimper chez Interface Concept contre quelques tickets resto, évidemment ! Plus sérieusement, merci pour votre présence, surtout dans les périodes où ça n'allait pas fort. Parce qu'une thèse, c'est aussi un chemin solitaire : seul, car peu de gens peuvent vraiment comprendre ou aider... et seul, car au fond, c'est un travail que l'on fait pour soi.

Enfin, je termine avec ma petite famille : Minibus et Roger, mon chat et mon lapin, deux experts en sieste qui m'ont rappelé que parfois, il fallait lever le pied.

Mais surtout... Clara Berneuil, mon amour, mon amie, ma compagne dans cette aventure et dans toutes les autres. Tu as été là dans les moments difficiles, dans les plus beaux aussi, et je sais qu'après avoir traversé ça ensemble, il n'y a plus rien qui pourra nous arrêter. Cette thèse, ce n'est pas que la mienne, c'est aussi la tienne, et celle de toutes les personnes qui m'ont soutenu et accompagné. Merci, infiniment.

# INTRODUCTION

**Sommaire**


**Contexte général et problématique** . . . . . **17**
**Objectifs de la recherche** . . . . . **21**
**Questions de recherche** . . . . . **22**
**Organisation du manuscrit** . . . . . **23**


## Contexte général et problématique

Avec la montée en puissance des cyberattaques sur ICS (Industrial Control Systems) et la multiplication des réglementations, les industriels doivent faire face à un environnement de plus en plus contraignant en matière de cybersécurité. Des directives comme NIS2[5] ou le Cyber Resilience Act[6] imposent des exigences strictes aux entreprises, les obligeant à garantir un niveau de sécurité élevé pour leurs produits et services. Cette nécessité de conformité est essentielle pour protéger les infrastructures critiques et les systèmes industriels.

Dans la réalité, de nombreux industriels éprouvent des difficultés à répondre à ces nouvelles obligations. Les contraintes sont multiples : manque de ressources financières et humaines, compétences en cybersécurité insuffisantes, et incertitudes sur les futures évolutions réglementaires. À cela s'ajoute une pression accrue des clients, qui exigent des garanties de sécurité sous forme de certifications, de labels ou d'audits approfondis. Parfois, ces exigences relèvent d'une véritable démarche de sécurisation des infrastructures, mais elles peuvent aussi être motivées par des enjeux commerciaux et marketing, la cybersécurité devenant un facteur de réassurance et de différenciation sur le marché. Les réseaux de communication industriels sont particulièrement concernés par ces exigences de sécurité, car ils doivent protéger des infrastructures critiques tout en s'adaptant à des contraintes spécifiques liées à leur environnement et à leur fonctionnement.

Un réseau industriel est un réseau particulier composé de systèmes industriels appelés Industrial Control Systems (ICS) comme des automates (Programmable logic controllers (PLC), voir figure 1), des capteurs, des actionneurs, des systèmes de supervision (Supervisory control and data acquisition (SCADA)). Ils répondent à des critères et exigences propres à l'industrie, comme la résistance du matériel, la température de fonctionnement avec une longévité importante (15 ans en moyenne[7, 8] mais parfois bien plus en pratique[9]). Ils sont conçus pour des conditions et des environnements difficiles.




FIGURE 1 – PLC Siemens S7-300

Originellement, les systèmes industriels (OT) et les infrastructures informatiques (IT) évoluaient indépendamment. Les premiers fonctionnaient en circuit fermé, avec des technologies propriétaires, répondant à des exigences strictes en matière de temps réel et de fiabilité (cycles de traitement rapides, architectures redondantes, protocoles de communication industriels robustes, résistance aux environnements extrêmes). À l'inverse, les systèmes IT ont progressivement adopté des architectures flexibles, interconnectées et standardisées, favorisant l'échange d'informations et l'optimisation des processus de gestion.

Toutefois, avec l'essor de l'Industrie 4.0 depuis 2011, puis des objets connectés industriels (IIoT - Industrial Internet of Things) en 2015[10], la convergence entre l'IT et l'OT s'accélère. Si cette fusion ouvre la voie à de nouvelles fonctionnalités, elle pose également de nouveaux défis. En effet, la connexion des systèmes industriels les expose à des risques accrus, car ces infrastructures, historiquement peu sécurisées, deviennent exposées et vulnérables aux cyberattaques.

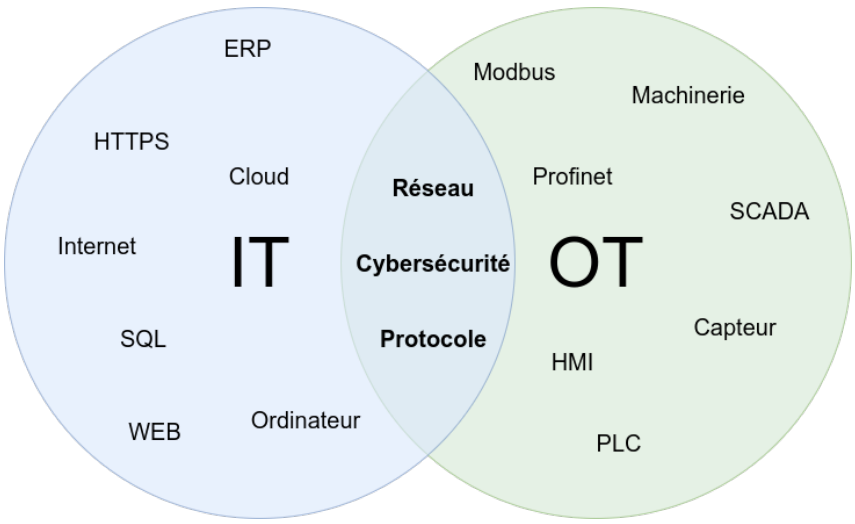



FIGURE 2 – Relations entre l'IT et l'OT

Les cyberattaques ciblant ces infrastructures critiques sont de plus en plus fréquentes et

sophistiquées, avec des conséquences potentiellement désastreuses. Parmi les exemples les plus marquants, l'attaque de 2015 contre le réseau électrique ukrainien reste une illustration emblématique des vulnérabilités des systèmes industriels face aux menaces cyber. En décembre 2015, des pirates informatiques ont pénétré le réseau de gestion de plusieurs sous-stations électriques en Ukraine, perturbant l'approvisionnement en électricité de centaines de milliers de personnes. Cette attaque a révélé des failles critiques dans la segmentation du réseau, l'absence de systèmes d'alerte précoce et le manque de visibilité en temps réel sur les flux réseau. Les systèmes de sécurité classiques n'ont pas pu réagir efficacement face à cette menace sophistiquée. Comme illustré par la figure 2, les environnements IT et OT sont désormais interconnectés via le réseau, facilitant leur communication. Cette évolution, due au passage de nombreux systèmes OT à la couche IP, les rend plus exposés aux menaces en cybersécurité. De plus, certains intègrent des serveurs web pour leur gestion, accentuant encore les risques potentiels.

Par ailleurs, la cybersécurité, bien qu'étant un sujet central, évolue désormais en synergie avec d'autres technologies. Les figures 3 et 4 représentent une analyse des publications académiques de Google Scholar [1] et Scopus [2] entre 2015 et 2024. Si la cybersécurité générale a connu une croissance exponentielle au cours des dix dernières années, elle semble aujourd'hui atteindre un plateau, voire une baisse, suggérant une maturité du domaine et une transition vers des approches plus spécialisées. En revanche, la cybersécurité intégrant l'intelligence artificielle (AI cybersecurity) continue de croître de manière exponentielle, illustrant l'importance croissante des techniques d'apprentissage automatique et d'automatisation dans la protection des systèmes informatiques. Parallèlement, la cybersécurité des environnements industriels (OT cybersecurity) commence à émerger comme un axe de recherche distinct, en réponse aux défis spécifiques posés par la convergence IT/OT et l'augmentation des cyberattaques ciblant les infrastructures critiques.

Ces tendances montrent que la cybersécurité ne se limite plus à une approche généraliste, mais s'intègre dans des domaines spécialisés pour répondre à des besoins spécifiques, notamment à travers l'intelligence artificielle et la protection des infrastructures industrielles. Ce phénomène se reflète également dans le monde industriel, où les utilisateurs recherchent non seulement des produits sécurisés, mais aussi des solutions dédiées à la cybersécurité, répondant aux nouvelles menaces pesant sur les infrastructures critiques.

Face à ces enjeux, l'entreprise Interface Concept [3] investit de manière significative dans la recherche et le développement (R&D) afin d'intégrer des fonctionnalités avancées de cybersécurité dans ses produits. L'objectif est double : proposer des équipements sécurisés, mais aussi développer des solutions spécifiques pour la cybersécurité. Parmi ces initiatives, l'un des axes majeurs concerne le switch Ethernet, un produit phare pour l'entreprise, qui évolue pour répondre aux

1. https ://scholar.google.com/
2. https ://www.scopus.com/
3. https ://www.interfaceconcept.com/

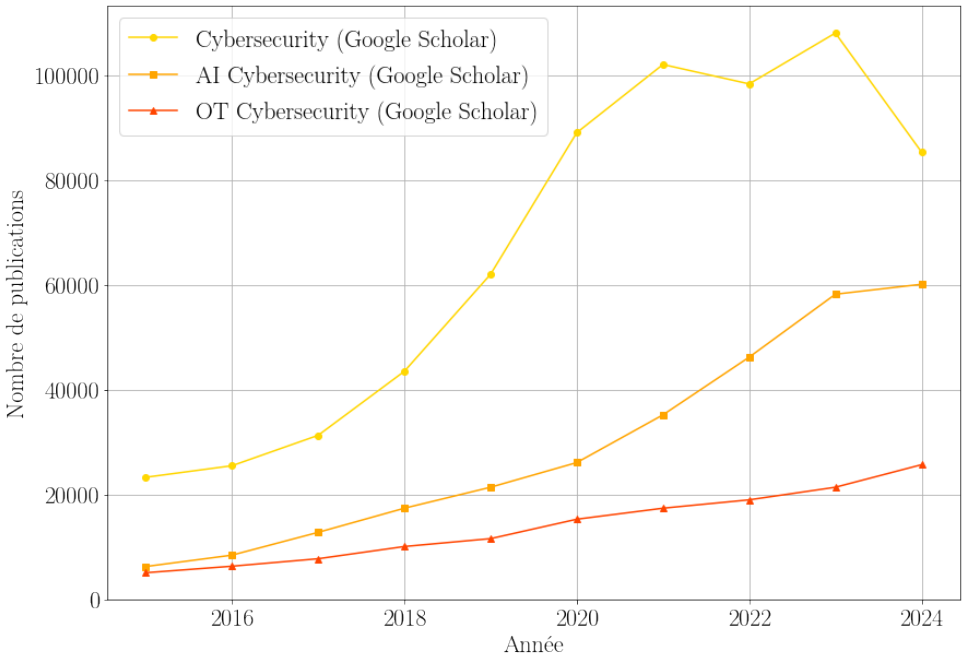



FIGURE 3 – Évolution des publications sur Google Scholar sur la cybersécurité (2015-2024) par recherche des mots dans les articles (mots clés, résumé, texte, titre)

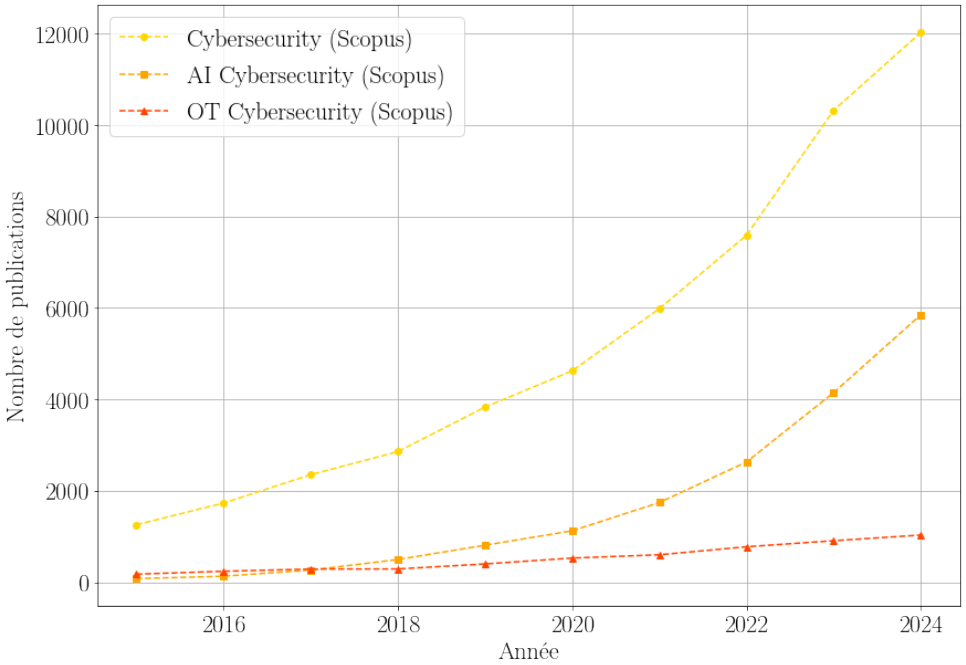



FIGURE 4 – Évolution des publications sur Scopus sur la cybersécurité (2015-2024) par recherche des mots dans le titre, résumé ou les mots-clés

exigences croissantes des clients et des réglementations.

C'est dans ce cadre que le travail de recherche présenté dans ce manuscrit s'inscrit. L'objectif est de concevoir une solution intégrée permettant d'améliorer la sécurité du trafic réseau au sein de ces équipements, en alliant classification, détection et résilience. Cette approche vise à répondre aux nouveaux besoins industriels, en offrant des capacités de cybersécurité embarquées directement dans l'infrastructure réseau, sans dépendre uniquement des solutions externes classiques comme les IDS (Intrusion Detection Systems).

## Objectifs de la recherche

Cette thèse CIFRE est réalisée en collaboration avec la société Interface Concept, une entreprise quimpéroise fondée en 1987, spécialisée dans la conception et la fabrication de cartes et de systèmes embarqués haute performance destinés à des applications industrielles. La problématique principale porte sur l'intégration de mécanismes de sécurité directement au sein des commutateurs réseau utilisés en environnements industriels, ces derniers constituant l'un des produits phares de l'entreprise, pour lesquels elle dispose d'une expertise technique reconnue. Les commutateurs d'Interface Concept, nommés ComEth, sont des équipements réseau avancés capables d'opérer à la fois sur les couches L2 (liaison de données) et L3 (réseau) du modèle OSI. Ils assurent la commutation rapide des trames Ethernet tout en intégrant des fonctions de routage IP pour gérer des communications interréseau. La figure 5 montre un commutateur répandu de chez Interface Concept, le ComEth4000e .




FIGURE 5 – Carte ComEth4000e d'Interface Concept

Une particularité de ces commutateurs réside dans la possibilité d'ajouter des cartes d'extension, une fonctionnalité qui, bien que rarement exploitée en pratique, permet d'étendre les capacités et les fonctionnalités du commutateur en fonction des besoins spécifiques des clients.

Ces extensions peuvent inclure plus de ports, des capacités de traitement supplémentaires, ou des fonctionnalités avancées comme du chiffrement matériel. Cette modularité offre une flexibilité accrue dans des environnements exigeants où les besoins évoluent en fonction des architectures réseau.

De ce constat, le commutateur apparaît comme en un carrefour stratégique pour l'analyse du trafic. En effet, les échanges réseau y transitent obligatoirement pour effectuer n'importe quelle communication. Cette opportunité permet, sans systèmes intermédiaires, d'effectuer une supervision du trafic ainsi qu'une première phase de détection et de classification des activités réseau. Cela permettrait de délester les systèmes existants qui analysent la totalité du trafic, souvent avec des algorithmes génériques. Dans une deuxième phase, il pourrait par exemple router intelligemment les différents flux vers des systèmes de détection d'intrusion (IDS) spécialisés.

L'objectif de ce travail de thèse est de concevoir un prototype de commutateur intelligent, capable d'améliorer la sécurité des réseaux en environnement industriel. Celui-ci intégrera des fonctions de supervision, de détection et de classification permettant d'assister, de redonder ou de de délester d'autres systèmes, tout en ayant une vue d'ensemble du trafic y transitant. En s'appuyant sur des statistiques ou des algorithmes d'apprentissage automatique, il sera en mesure d'analyser les flux réseau en temps réel, d'identifier les activités et de les classifier.

## Questions de recherche

Dans le cadre de cette thèse, plusieurs questions de recherche et de développements ont émergé autour de la conception et de l'implémentation d'un switch intelligent pour l'analyse du trafic pour la sécurisation des réseaux industriels.

1. **QR1 : Comment modéliser et représenter l'activité humaine dans un réseau informatique en tenant compte de ses spécificités ?** L'activité humaine dans un réseau est rarement étudiée en tant que génératrice de trafic. La théorie de l'activité, issue des sciences cognitives, peut aider à modéliser les comportements des utilisateurs. Il s'agit de proposer un cadre permettant une représentation exploitable des comportements réseau humains.
2. **QR2 : Comment détecter et classifier les activités observées dans un réseau informatique ?** La classification des flux réseau est complexe, car une même activité peut engendrer plusieurs types de traces et inversement. Plusieurs approches peuvent être combinées : heuristiques (ports et protocoles), analyse statistique (caractéristiques temporelles et volumétriques), apprentissage automatique et Deep Packet Inspection (DPI). L'objectif est de développer une méthode fiable minimisant les erreurs de classification, tout en tenant compte de la diversité des flux et des usages de chaque réseau.
3. **QR3 : Comment générer et étiqueter des ensembles de données représentatifs**

**des activités réseau pour entraîner des modèles de classification ?** L'étiquetage des données réseau est un vrai défi. Plusieurs approches existent : capturer des flux réels, générer du trafic synthétique et utiliser des méthodes d'étiquetage automatique. L'objectif est de créer des jeux de données équilibrés et représentatifs, essentiels pour améliorer la performance des modèles d'apprentissage automatique.

4. **QR4 : Comment intégrer un système d'analyse de trafic au sein d'un switch ou d'une architecture réseau ?** L'analyse du trafic en périphérie du réseau, directement dans un switch, permettrait une classification précoce et une répartition intelligente des flux vers des systèmes spécialisés (IDS, enregistreurs, filtres). Cela implique de relever des défis d'implantation (ressources matérielles limitées, latence, temps réel), de sécurité (résilience, cloisonnement), et d'architecture (interaction avec le reste du système). L'objectif est de proposer un modèle d'intégration efficace, modulaire, et interopérable.
5. **QD1 : Quels outils peuvent assister ou automatiser l'enregistrement et l'étiquetage des activités réseau ?** Pour répondre au manque d'outils disponibles, le développement d'outils pour l'enregistrement et l'étiquetage automatique des données réseau est effectué pour répondre aux problématiques.

Le tableau 1 met en évidence la correspondance entre les chapitres du manuscrit et les différentes questions de recherche (QR) et de développement (QD). Elle permet de structurer l'approche adoptée tout au long de cette étude et illustre comment chaque partie de la thèse contribue à la résolution des questions soulevées.

| Chapitre | Titre du chapitre | QR1 | QR2 | QR3 | QR4 | QD1 |
|---|---|---|---|---|---|---|
| 1 | État de l'art | | | | | |
| 2 | Modélisation de l'activité humaine dans les réseaux | ■ | | | | |
| 3 | Génération de données pour l'analyse d'activité réseau | | | ■ | | ■ |
| 4 | Analyses et applications | | ■ | | ■ | |
| 5 | Conclusion et perspectives | ■ | ■ | ■ | ■ | ■ |

TABLE 1 – Correspondance entre chapitres et questions de recherche

## Organisation du manuscrit

Pour répondre à la problématique décrite, le manuscrit de thèse se divise en quatre chapitres et une conclusion générale, ouvrant sur les perspectives et la suite des travaux. La figure 6 illustre les liens entre les principaux domaines abordés dans cette thèse et la manière dont ils interagissent pour répondre aux problématiques de modélisation, de génération de données et de classification des activités humaines numériques.

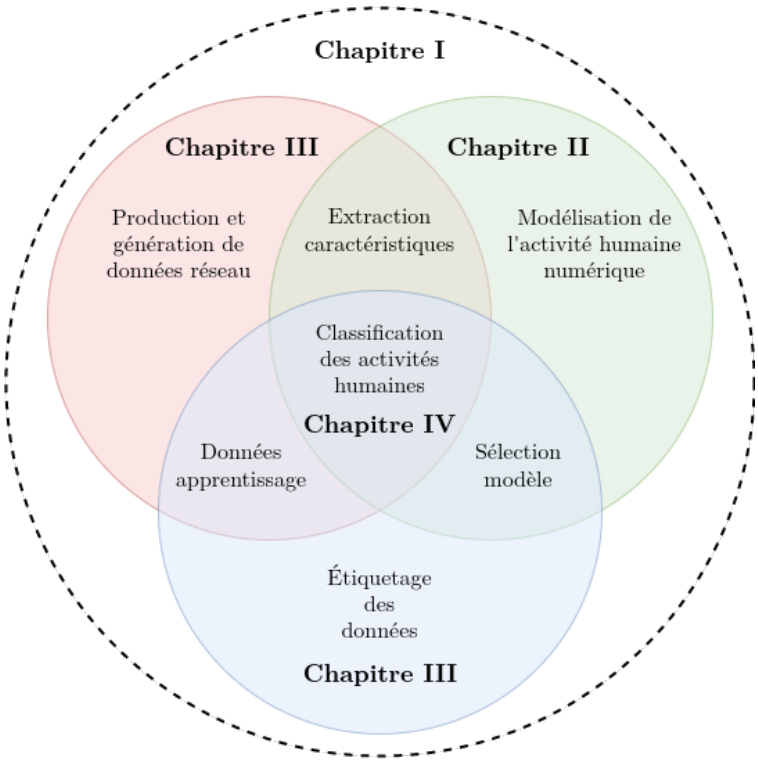



FIGURE 6 – Liens entre les domaines et les sujets abordés dans la thèse

Le **premier chapitre** introduit les bases théoriques et fait l'état de l'art des différents domaines abordés le long du manuscrit. Les concepts fondamentaux du réseau ainsi que les différents outils existants pour analyser le trafic sont abordés. De plus, les techniques d'apprentissage automatique ainsi que les jeux de données couramment utilisées sont détaillés et expliqués.

Le **deuxième chapitre** se concentre sur la modélisation de l'activité humaine dans les réseaux. Il s'appuie sur la théorie de l'activité, issue des sciences cognitives, pour contextualiser et décrire les activités humaines ainsi que la production de traces réseau associées.

Le **troisième chapitre** est axé sur la production de données pour l'entraînement d'algorithmes d'apprentissage automatique. La performance d'un algorithme dépend, bien souvent, de la qualité de ses données. Pour optimiser au mieux l'efficacité de nos détections et classifications, cette partie se concentre sur le développement de méthodes permettant de maximiser le réalisme, la précision et la qualité des données produites, notamment par la création de logiciels adaptés.

Le **quatrième chapitre** aborde l'application des méthodes et logiciels développés pour la production et l'exploitation des données générées pour l'apprentissage d'algorithme de détection et de classification d'activités. La seconde partie est consacrée à l'analyse des résultats obtenus, en particulier l'évaluation des jeux de données produits et la qualité de leur classification.

Enfin, un chapitre **conclusion** termine ce travail en récapitulant les contributions majeures

de cette recherche. Il propose également des perspectives pour de futurs travaux, notamment l'amélioration des méthodes de collecte et d'étiquetage de données, ainsi que l'intégration de nouvelles fonctionnalités aux commutateurs. Ce chapitre aborde également les défis restants à relever, tels que l'adaptation aux protocoles propriétaires et l'amélioration continue des algorithmes de détection et de classification.

Chapitre 1

# États de l'art

## Sommaire


**1.1 Architectures réseaux IT/OT . . . . . 29**
1.1.1 Modèles de référence . . . . . 29
1.1.2 Sécurité et défense en profondeur . . . . . 31
1.1.3 Acteurs et protocoles des systèmes industriels . . . . . 33
**1.2 Le trafic réseau : concepts et manipulation . . . . . 35**
1.2.1 Acheminement du trafic . . . . . 35
1.2.2 Génération du trafic . . . . . 38
1.2.3 Capture et enregistrement du trafic . . . . . 39
1.2.4 Intérêt et objectifs . . . . . 40
**1.3 Méthodes classiques d'analyse du trafic . . . . . 41**
1.3.1 Analyse par couches (modèle OSI ou TCP/IP) . . . . . 41
1.3.2 Utilisation de bases de données protocolaires . . . . . 42
1.3.3 Segmentation des paquets . . . . . 42
1.3.4 Analyse par inspection des Flow . . . . . 42
1.3.5 Analyse par Deep Packet Inspection (DPI) . . . . . 44
1.3.6 Heuristiques et analyse avancée . . . . . 44
**1.4 Classification réseau . . . . . 44**
1.4.1 Détection et supervision : IDS, QoS, surveillance . . . . . 45
1.4.2 Machine Learning et IA pour l'analyse du trafic . . . . . 46
1.4.3 Classification du trafic : attaques, services, activités . . . . . 50
**1.5 Analyse des Jeux de données existants . . . . . 55**
1.5.1 Jeux de données orientés détection d'intrusion . . . . . 55
1.5.2 Synthèse des jeux de données réseau pour IDS . . . . . 60
1.5.3 Jeux de données orienté analyse de trafic réseau . . . . . 62
**1.6 Activité humaine dans les réseaux . . . . . 64**
1.6.1 Théorie de l'activité . . . . . 64
1.6.2 Différences entre l'homme et la machine . . . . . 66
1.6.3 Transposition de la théorie de l'activité à l'activité réseau . . . . . 67
**1.7 Conclusion . . . . . 67**

En analysant l'intitulé du sujet de thèse, "Sécurisation de réseaux en environnement industriel : prototype de switch intelligent", on remarque quatre grands thèmes. Celui de la sécurité informatique, du monde industriel, des commutateurs réseau et enfin l'aspect "intelligence", caractéristique plutôt humaine, que l'on pourrait associer à l'intelligence artificielle et plus précisément à l'apprentissage automatique. Cette partie se concentre sur ces différents éléments nécessaires à la compréhension du sujet. Ils vont permettre de poser les bases de l'environnement dans lequel s'ancre la problématique du sujet et quelles sont ses particularités. Ensuite, les différentes méthodes d'analyse de trafic réseau et les techniques existantes pour y détecter des anomalies ont été recensées. Enfin, un état de l'art a été réalisé sur l'activité humaine dans les réseaux et les traces produites. Le but étant de caractériser des activités humaines à partir de traces réseau et ainsi essayer d'identifier une ou des activités différentes, voire même un profil utilisateur pouvant s'avérer anormal, dangereux ou non.

L'état de l'art actuellement disponible apporte déjà des éléments de réponse partiels aux questions de recherche, en posant les bases théoriques et techniques nécessaires à l'analyse de trafic réseau et à la compréhension des architectures industrielles. Toutefois, ces approches restent souvent générales, inadaptées aux environnements, contraints ou trop éloignées des problématiques réelles observées sur le terrain. C'est pourquoi les chapitres suivants viendront enrichir et répondre à ces questions par des contributions originales, concrètes et applicables, notamment en matière de modélisation de l'activité, de génération de données, et d'intégration embarquée dans des équipements réseau.

## 1.1 Architectures réseaux IT/OT

Dans le contexte industriel moderne, l'intégration des technologies de l'information (IT) et des technologies opérationnelles (OT) est devenue cruciale pour optimiser les processus de production et améliorer l'efficacité opérationnelle. Les architectures réseau OT/IT permettent de connecter les systèmes de contrôle industriel avec les systèmes d'information d'entreprise, facilitant ainsi la collecte, l'analyse et l'utilisation des données en temps réel.

Cette intégration, bien que bénéfique, présente également des défis significatifs, notamment en matière de sécurité et de gestion des risques. Les modèles de référence tels que le modèle de Purdue et le modèle ICS410 offrent des cadres structurés pour concevoir et gérer ces architectures complexes. De plus, la mise en œuvre de stratégies de défense en profondeur est essentielle pour protéger les réseaux OT/IT contre les cybermenaces croissantes.

Cette section explore les principaux modèles de référence, les stratégies de sécurité et les acteurs clés dans le domaine OT, tout en mettant en lumière l'importance des protocoles de communication dans ces environnements critiques.

### 1.1.1 Modèles de référence

Les réseaux industriels forment l'infrastructure essentielle des systèmes de contrôle industriel (ICS - Industrial Control Systems), assurant la communication entre capteurs, actionneurs, automates programmables (PLC), systèmes SCADA et autres équipements d'automatisation. Contrairement aux réseaux IT traditionnels, les réseaux industriels doivent garantir temps réel, fiabilité et disponibilité, souvent dans des environnements hostiles où toute interruption peut entraîner des pertes de production critiques.

Le modèle de Purdue est un modèle d'architecture d'entreprise datant des années 1990 et développé par Theodore J.Williams, ainsi que par d'autres membres de l'université Purdue. Ce modèle est devenu une référence dans le milieu de l'industrie et tout particulièrement ceux faisant usage d'Industrial Control Systems (ICS).

Cette architecture se divise en plusieurs couches permettant de segmenter chaque niveau du réseau et ainsi avoir un meilleur contrôle au niveau des échanges entre sous-réseaux (*Figure 1.1*).

Une des particularités de cette architecture est d'avoir une séparation entre le réseau d'entreprise et le réseau industriel. En effet, cela permet l'ajout d'une sécurité supplémentaire, car le réseau industriel ne sera pas directement exposé à l'extérieur. Chaque couche communique avec celles adjacentes et leurs rôles sont les suivants :

— **Level 5 : *Enterprise Network*** est la couche en charge du réseau de l'entreprise.

— **Level 4 : *Business Planning and Logistics*** est la couche qui contient les systèmes de logistique. On y retrouve les progiciels en charge de gérer l'ensemble des processus d'une entreprise (RH, gestion financière, distribution, approvisionnement, etc.).

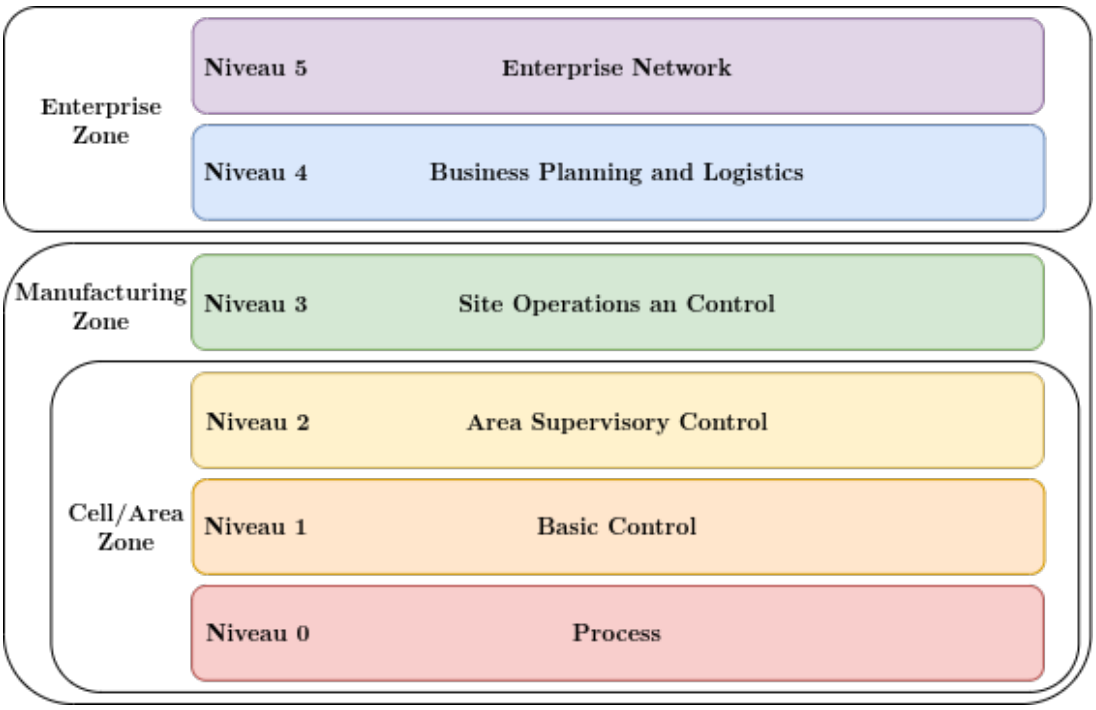



FIGURE 1.1 – Niveaux d'une architecture de Purdue[1]

— **Level 3 : *Site Operations and Control*** est la couche qui s'occupe du management de la production. On y retrouve par exemple les workflows pour la production des produits.

— **Level 2 : *Area Supervisory Control*** est la couche en charge de la supervision, du monitoring et du contrôle. Elle contient généralement des éléments comme des Interfaces Homme-Machine (IHM), des systèmes d'alarme ou encore des logiciels SCADA pour interagir avec les processus industriels.

— **Level 1 : *Basic Control*** est la couche de contrôle. C'est ici que les PLC se trouvent. Ils sont en charge de l'instrumentalisation des machines et des processus industriels.

— **Level 0 : *Process*** est la couche la plus basse. Elle est constituée de capteurs et actionneurs en charge des processus physiques.

Ce modèle a longtemps permis d'assurer une séparation claire entre l'OT (Operational Technology) et l'IT, minimisant ainsi les risques de cyberattaques en isolant les systèmes industriels.

Cependant, le modèle de Purdue n'a pas été pensé d'un point de vue sécurité initialement. Il est donc important de compléter cette architecture afin de sécuriser au mieux les ICS, qui, eux aussi, n'ont pas été pensés pour être sécurisés. Ceux-ci ont été conçus dans un but de fiabilité et robustesse dans des environnements souvent difficiles (conditions extérieures de température, d'humidité, de salinité ...), mais qui surtout ne doivent pas tomber en panne. Leur durée de vie se compte en dizaines d'années, laissant leurs vulnérabilités exposées autant de temps.

Le terme de Operational Technology (OT) est utilisé pour décrire tous les systèmes et technologies qui sont utilisés dans un environnement industriel. On utilise également des protocoles de communication différents des réseaux classiques, qui sont adaptés aux exigences spécifiques des environnements industriels. Le SANS Institute propose comme modèle de référence le modèle ISC410, que l'on peut voir sur la figure 1.2, qui vient corriger certains problèmes du modèle de Purdue.

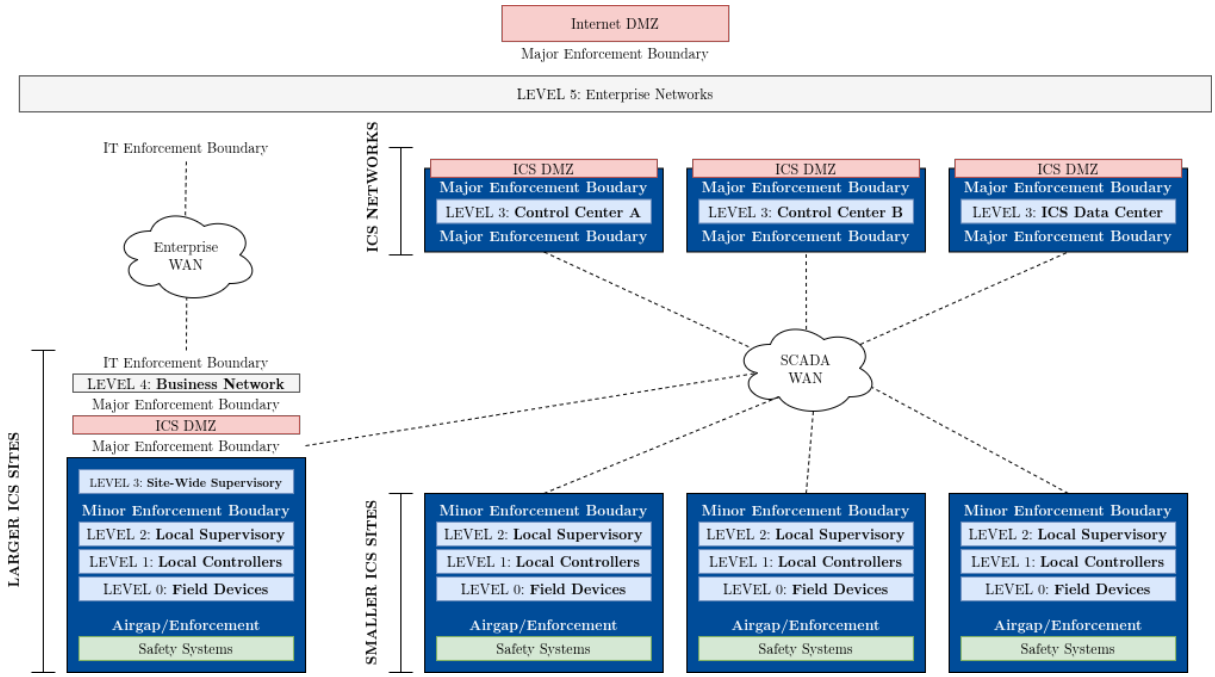



FIGURE 1.2 – Modèle de référence ICS410 [2]

Parmi ses avantages, on peut citer :

— La subdivision de certaines couches pour ne pas mélanger les systèmes

— L'ajout de DMZ (qui ne sont pas présentes dans le modèle de Purdue original, bien que souvent utilisées)

— Conseils pour sécuriser l'accès à distance

— Introduit le concept de plusieurs lignes de production/sites,

— Etc.

### 1.1.2 Sécurité et défense en profondeur

Face à la montée des cybermenaces, la Défense en profondeur (Defence in Depth) est une approche incontournable pour sécuriser les réseaux critiques[11]. Ce concept repose sur plusieurs couches de protection, de manière à rendre une attaque plus difficile à mener et à limiter son

impact en cas de compromission d'un élément du système. Le concept de défense en profondeur tient son origine du monde militaire. En effet, à l'image d'une ville assaillie par l'ennemi, la multiplication des lignes de défense permet de le ralentir voir le stopper. Cette notion est très présente dans le domaine de la sûreté nucléaire et la gravité d'un évènement est basée sur l'échelle INES (Figure 1.3).

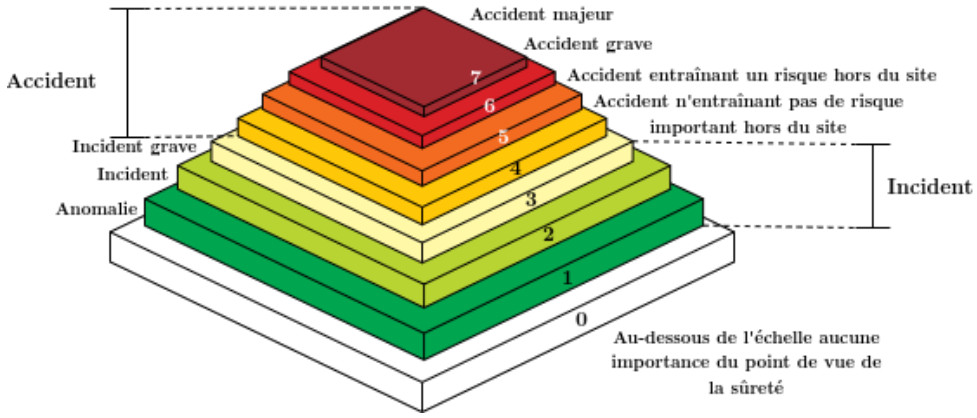



FIGURE 1.3 – Échelle INES

En effet, la défense en profondeur de celle-ci est divisée en 3 lignes de défense. Ce nombre n'a pas été choisi au hasard, car dans le cas où la première ligne serait franchie, même avec une anomalie rendant la deuxième inexploitable, il resterait toujours encore une dernière pour protéger la ressource critique avant l'incident grave. C'est pour cela que pour que le concept de défense en profondeur soit effectif il faut qu'il y ait au moins 3 lignes de défense, chaque ligne étant composée d'une ou plusieurs barrières. La figure 1.4 montre les différentes lignes de défense et leurs barrières associées.

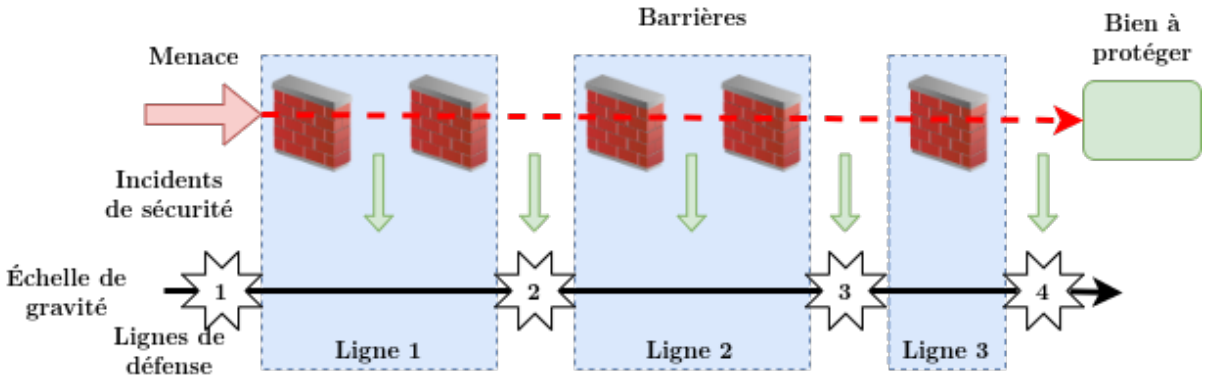



FIGURE 1.4 – Démarche de mise en évidence des lignes de défense

### 1.1.3 Acteurs et protocoles des systèmes industriels

Le secteur des systèmes industriels est dominé par de grands groupes internationaux, tels que Siemens, Schneider Electric, Rockwell Automation, ABB, Mitsubishi Electric ou encore Omron. Ces entreprises conçoivent et fournissent des automates programmables (PLC), des systèmes de contrôle (SCADA, DCS), ainsi que des solutions logicielles et matérielles pour l'automatisation des processus industriels. La table 1.1 détaille la part de revenu de ces entreprises dans l'automatisation industrielle.

| **Entreprise** | **Revenus 2024 (m€)** | **Revenus automatisation (m€)** | **Part** |
|---|---|---|---|
| Siemens AG[12] | 75 930 | 18 536 | 24,4% |
| Rockwell Automation, Inc.[13] | 7 950 (8 264m$) | 7 950 (8 264m$) | 100% |
| Mitsubishi Electric Corp.[14] | 33 335 (5 257,9M¥) | 4 859 (766,4M¥) | 14.58% |
| Schneider Electric SE[15] | 37 654 | 6 976 | 18,5% |
| Omron Corporation(2023)[16] | 5 188 (818,761M¥) | 2 494 (393,6M¥) | 48,1% |
| ABB Group[17] | 31 603(32 850m$) | 6 500(6 756m$) | 20,6% |

TABLE 1.1 – Répartition des revenus des grandes entreprises dans l'automatisation industrielle (2024)

Il convient de prendre ces chiffres avec précaution, car ils sont issus des rapports annuels 2024 des entreprises concernées. La classification des revenus varie d'un acteur à l'autre : certains regroupent leurs revenus sous des catégories larges comme Industrial Automation ou Digital Industries, sans détailler précisément la part attribuée aux automates programmables. De plus, ces entreprises ne se limitent pas aux PLC, mais développent également des moteurs industriels, des équipements électriques, des solutions logicielles, et d'autres systèmes d'automatisation. Certains de ces systèmes intègrent probablement des automates, mais pas dans les mêmes proportions ni avec les mêmes objectifs, rendant toute comparaison directe délicate. Ce tableau donne ainsi un ordre de grandeur plutôt qu'une répartition exacte du marché des automates. La figure 1.5 donne une estimation des parts de marché des différents acteurs importants de l'automatique.

Toutefois, la présence de ces grands acteurs implique souvent l'utilisation de protocoles propriétaires, développés spécifiquement pour leurs équipements. Cette approche, bien qu'optimisée pour leurs solutions, pose des problèmes d'interopérabilité, rendant parfois complexe l'intégration de matériels provenant de différents fabricants. Ce schéma issu du rapport HMS Networks 2024 donne un aperçu des tendances actuelles en matière d'utilisation des protocoles industriels. Il met en évidence la répartition des technologies adoptées, montrant la montée en puissance des solutions basées sur Ethernet industriel, tout en reflétant la persistance des bus de terrain et des technologies sans fil dans certains secteurs.

Toutefois, il est important de nuancer cette analyse. Tous les protocoles industriels ne sont pas propriétaires : certains, comme EtherNet/IP ou Modbus, sont des standards ouverts, permettant une interopérabilité plus large entre différents équipements. En revanche, d'autres, comme

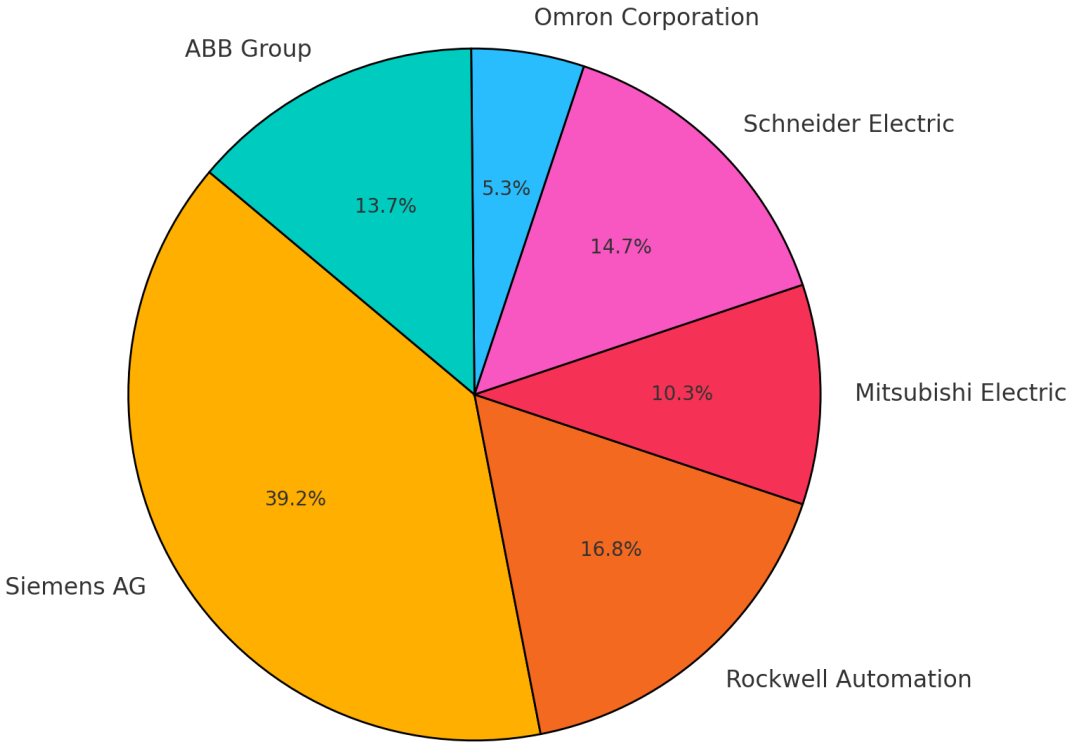



FIGURE 1.5 – Répartition des revenus générés par l'automatisation industrielle (2024)

S7COMM de Siemens ou certains protocoles spécifiques à Rockwell Automation, sont fermés et propriétaires, limitant ainsi leur compatibilité avec des équipements d'autres fabricants.

De plus, même parmi les standards ouverts, tous les automates ne sont pas compatibles entre eux. La mise en œuvre effective d'un protocole dépend des spécifications du constructeur, des versions logicielles, et des éventuelles extensions propres à chaque fabricant. Ainsi, bien que ces tendances montrent une évolution globale des réseaux industriels, la réalité sur le terrain reste plus fragmentée, avec des enjeux d'interopérabilité, de compatibilité et de migration des anciennes infrastructures vers des solutions plus modernes.

Avec la convergence croissante entre l'IT et l'OT, ces acteurs doivent désormais s'adapter aux standards ouverts, favoriser l'interopérabilité et répondre aux exigences croissantes en cybersécurité. Malgré ces défis, ils restent des piliers essentiels du monde industriel, jouant un rôle clé dans la transition vers l'Industrie 4.0 et l'automatisation intelligente.

## 1.2 Le trafic réseau : concepts et manipulation

L'univers du trafic réseau est un domaine relativement large. Que cela soit d'un point de vue physique, matériel, ou encore analytique, l'ensemble de ces champs se doivent d'être couverts pour une meilleure compréhension du sujet. Dans le cadre de ces recherches, la donnée a une place très importante. En effet, elle est nécessaire pour entraîner les algorithmes qui seront en charge de la détection et la classification des activités des utilisateurs. Il existe des jeux de données disponibles permettant de s'entraîner, mais ils ont leurs limites quant à leur réalisme quand le contexte se précise.

### 1.2.1 Acheminement du trafic

L'acheminement du trafic réseau repose sur un ensemble de composants physiques et logiques qui assurent le transport des données d'un point à un autre à travers un réseau. Ces infrastructures, comprenant des câbles, des routeurs, des switches et d'autres équipements, jouent un rôle crucial dans la transmission des paquets de données selon des protocoles bien établis. La qualité et la performance de l'acheminement du trafic dépendent directement de ces infrastructures, qui sont conçues pour garantir une communication rapide, sécurisée et fiable.

Dans les réseaux industriels, où les exigences en matière de temps réel et de sécurité sont plus strictes que dans les environnements classiques, l'efficacité de l'acheminement du trafic est primordiale pour maintenir la continuité des opérations. Cette section établit les fondamentaux des communications réseau, en expliquant le fonctionnement général des commutateurs et des routeurs. Elle se concentre ensuite sur le switch d'Interface Concept, un switch de niveau 2/3 (L2/L3) doté de nombreuses particularités

Les réseaux informatiques peuvent être classés en fonction de leur taille et de leur portée géographique comme le montre la figure 1.6. Un LAN (Local Area Network) est un réseau à petite échelle, généralement limité à un bureau, une maison ou un bâtiment. Il permet une communication rapide et sécurisée entre les appareils connectés. Un MAN (Metropolitan Area Network) couvre une zone urbaine ou un campus universitaire, reliant plusieurs LAN entre eux avec des infrastructures plus robustes. Enfin, un WAN (Wide Area Network) est un réseau à grande échelle, souvent intercontinental, qui relie plusieurs MAN ou LAN grâce à des technologies comme la fibre optique ou les satellites. Internet est l'exemple le plus connu de WAN. Cette hiérarchie de réseaux nécessite des équipements spécifiques pour gérer la transmission des données, notamment les switchs et les routeurs.

La structuration d'un réseau, qu'il soit local (LAN), métropolitain (MAN) ou étendu (WAN), repose sur des principes d'interconnexion et de communication entre les équipements. Pour assurer une transmission efficace et standardisée des données, des modèles de référence ont été établis, notamment le modèle OSI et le modèle TCP/IP, qui définissent les différentes couches

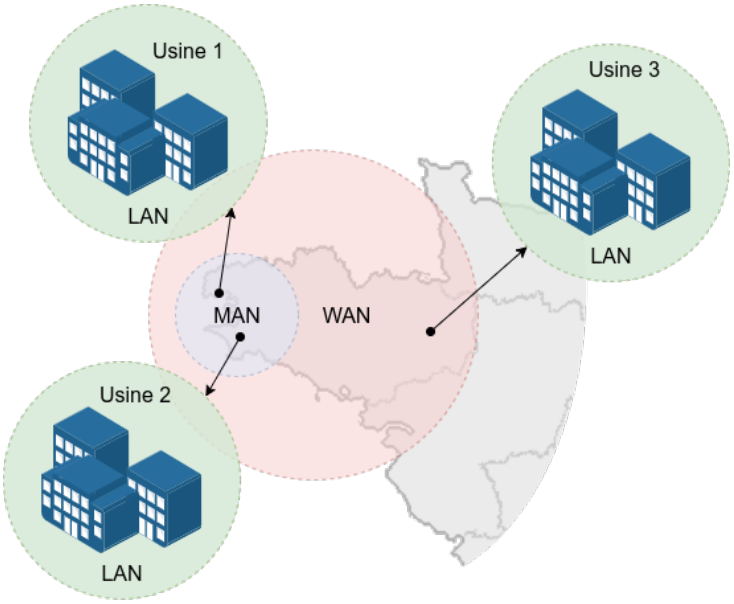



FIGURE 1.6 – Représentation et étendue d'un LAN, MAN et WAN

de communication. Ces modèles, détaillés dans le chapitre 2, permettent de comprendre comment les informations circulent d'un appareil à un autre et sont essentielles pour maîtriser le fonctionnement des équipements réseau tels que les switchs et routeurs, qui interagissent à différents niveaux du modèle OSI et TCP/IP.

Les équipements réseau jouent un rôle essentiel dans l'acheminement des données. Le switch (commutateur) fonctionne sur la couche 2 du modèle OSI (Liaison de données) et utilise les adresses MAC pour envoyer des trames uniquement aux destinataires concernés, optimisant ainsi les performances du réseau local. Il permet des communications en full-duplex, ce qui signifie qu'un appareil peut envoyer et recevoir des données simultanément. De son côté, le routeur opère sur la couche 3 (Réseau) et utilise les adresses IP pour acheminer les paquets entre plusieurs réseaux. Il joue un rôle clé dans la connexion à Internet et peut intégrer des fonctions avancées comme la traduction d'adresses (NAT), les pare-feux ou encore le routage dynamique.

Si le switch et le routeur assurent tous deux la transmission des données, leurs méthodes et usages sont bien distincts. Le switch (L2) est principalement utilisé pour relier des appareils au sein d'un même réseau local (LAN), en assurant un acheminement rapide des données entre eux. En revanche, le routeur (L3) permet d'interconnecter plusieurs réseaux, en utilisant des protocoles de routage pour trouver le meilleur chemin pour envoyer des paquets vers leur destination. Les switchs de niveau 3 (L3) existent également, combinant des fonctionnalités avancées des routeurs avec les performances des commutateurs classiques. Le tableau 1.2 compare les deux

matériels.

| **Caractéristiques** | **Switch (L2)** | **Routeur (L3)** |
|---|---|---|
| **Couche OSI** | Couche 2 (Liaison de données) | Couche 3 (Réseau) |
| **Adressage** | MAC (Media Access Control) | IP (Internet Protocol) |
| **Fonction principale** | Connexion des appareils en LAN | Connexion entre plusieurs réseaux |
| **Méthode de transmission** | Transmission selon l'adresse MAC | Routage selon l'adresse IP |
| **Table de correspondance** | Table MAC (commutation) | Table de routage (chemin optimal) |
| **Mode de communication** | Communication locale (LAN) | Communication inter-réseaux (WAN, MAN) |
| **Capacité de gestion** | VLAN, QoS sur modèles avancés | NAT, pare-feu, routage dynamique |
| **Utilisation typique** | Réseaux internes (bureaux, datacenters) | Connexion entre LAN, MAN et WAN |

TABLE 1.2 – Comparaison entre Switch L2 et Routeur L3

Les switchs d'interface concept, nommés ComEth, sont des équipements réseau avancés capables d'opérer à la fois sur les couches L2 (liaison de données) et L3 (réseau) du modèle OSI. Ils assurent la commutation rapide des trames Ethernet tout en intégrant des fonctions de routage IP pour gérer des communications interréseau. La figure 1.7 montre un switch ComEth4000e de chez Interface Concept.




FIGURE 1.7 – Carte ComEth4000e d'Interface Concept

Une particularité de ces switchs réside dans la possibilité d'ajouter des cartes d'extension, une fonctionnalité qui, bien que rarement exploitée en pratique, permet d'étendre les capacités et les fonctionnalités du switch en fonction des besoins spécifiques des clients. Ces extensions peuvent inclure plus de ports, des capacités de traitement supplémentaires, ou des fonctionnalités avancées comme du chiffrement matériel. Cette modularité offre une flexibilité accrue dans des environnements exigeants où les besoins évoluent en fonction des architectures réseau. La figure 1.8 permet de voir l'architecture de la carte ainsi que l'emplacement disponible.

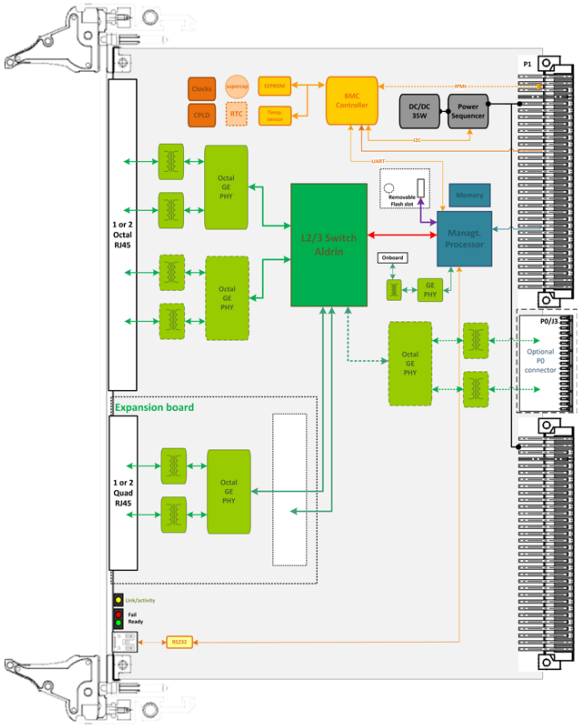



FIGURE 1.8 – Architecture de la ComEth4000e

### 1.2.2 Génération du trafic

Les logiciels de génération de trafic réseau sont des outils essentiels pour évaluer la performance, la sécurité et la résilience des infrastructures informatiques. Leur principal objectif est de produire du trafic synthétique ou réaliste afin de tester des équipements, d'entraîner des systèmes de détection d'intrusions (IDS), d'évaluer la capacité d'un réseau à supporter une charge donnée, ou encore de simuler des scénarios d'attaques pour améliorer la cybersécurité. On distingue plusieurs catégories de générateurs : ceux qui rejouent du trafic capturé, ceux qui synthétisent des paquets en suivant des modèles spécifiques, et ceux qui simulent des réseaux entiers.

L'utilisation de ces outils présente plusieurs avantages. Ils permettent d'anticiper les goulots d'étranglement et d'optimiser les infrastructures avant leur mise en production. Pour les IDS, ils offrent un cadre d'apprentissage et d'évaluation en soumettant ces systèmes à divers flux, du trafic normal aux attaques sophistiquées. Toutefois, ces logiciels présentent aussi des limites : certains ne reflètent pas fidèlement les comportements réels des utilisateurs, et leur mise en place peut être complexe, nécessitant des compétences avancées en réseau et en cybersécurité.

Il existe une multitude de solutions adaptées à des besoins spécifiques. Certains outils comme Tcpreplay permettent de rejouer des captures réelles, tandis que Iperf est spécialisé dans le test de performance réseau. Des solutions comme Metasploit ou Cuckoo Sandbox sont dédiées

à la simulation de trafic malveillant, et d'autres comme Conpot ciblent les environnements industriels. Le choix d'un générateur dépend donc de critères tels que le réalisme du trafic, l'orientation (sécurité, performance, simulation), et la compatibilité avec des infrastructures classiques ou industrielles. La table 1.3 liste quelques logiciels de génération de trafic utilisés pour divers objectifs.

| **Logiciel** | **Objectif** | | | **Réalisme** | **Industriel** | **Type de trafic** |
|---|---|---|---|---|---|---|
| | **Test Performance** | **Entraînement IDS** | **Simulation Réseau** | | | |
| **Iperf3** | ✓ | | | Synthétique | Non | TCP/UDP/SCTP |
| **Trex (Cisco)** | ✓ | | | Synthétique | Non | TCP/UDP, Stateful/Stateless |
| **Tcpreplay** | | ✓ | | Réaliste | Non | Capture PCAP |
| **Metasploit** | | ✓ | | Réaliste | Non | Exploits réseau |
| **Cuckoo Sandbox** | | ✓ | | Réaliste | Non | Exécution de malwares |
| **Slips** | | ✓ | | Synthétique | Non | Générateur de faux positifs IDS |
| **Scapy** | | | ✓ | Synthétique | Non | TCP/IP, DNS, HTTP, etc. |
| **Hping3** | ✓ | | | Synthétique | Non | TCP, UDP, ICMP |
| **Conpot** | | ✓ | ✓ | Réaliste | Oui | Protocoles industriels (SCADA) |
| **GridPot** | | ✓ | ✓ | Réaliste | Oui | Smart Grid / ICS |
| **Mininet** | | | ✓ | Réaliste | Non | OpenFlow / SDN |
| **GNS3** | | | ✓ | Réaliste | Oui | Multi-protocoles |
| **Core** | | | ✓ | Synthétique | Non | Multi-protocoles |
| **Locust** | ✓ | | | Réaliste | Non | HTTP/HTTPS |
| **Tsung** | ✓ | | | Réaliste | Non | HTTP/SIP/XMPP |

TABLE 1.3: Comparaison des logiciels de génération de trafic réseau

### 1.2.3 Capture et enregistrement du trafic

Le choix du format et de la manière de capturer les paquets ont leur importance. Il faut prendre en compte les fonctionnalités disponibles ainsi que la précision de chacun. Certains outils sont plus adaptés à l'enregistrement des données tandis que d'autres le sont plus pour leur traitement.

#### Wireshark

Wireshark est probablement l'un des analyseurs de paquet les plus populaires. Il possède une interface graphique permettant de rendre accessible son utilisation au plus grand nombre. Il prend en charge le format pcap et pcapng. Ce dernier est particulièrement intéressant, car il possède plus de fonctionnalité que son prédécesseur. Il permet par exemple d'ajouter de nouveaux champs à un protocole ce qui est assez intéressant dans le cadre de l'apprentissage automatique, car il est alors possible de directement labelliser les trames dans le même fichier durant l'enregistrement.

#### Tcpdump/Tshark

Tcpdump et Tshark sont deux outils incontournables pour la capture et l'analyse de trafic réseau. Tcpdump, réputé pour sa stabilité et sa légèreté, est idéal pour des captures rapides et des analyses de base, utilisant le format PCAP largement compatible. En revanche, Tshark, la version en ligne de commande de Wireshark, offre des fonctionnalités d'analyse plus avancées, telles que des filtres de visualisation puissants et des statistiques détaillées. Bien que Tshark puisse consommer plus de ressources, il est particulièrement adapté pour des analyses approfondies et peut être intégré dans des scripts pour des tâches automatisées.

#### Scapy

Contrairement aux logiciels cités précédemment, Scapy est une librairie python. Elle est très populaire pour la manipulation de données réseau. Elle est notamment pratique lorsqu'il est nécessaire de traiter un fichier de capture réseau déjà existant qui nécessite un traitement particulier comme l'extraction de certains champs par exemple.

#### NFStream

NFStream est un framework d'analyse de données réseau. Comme Scapy, il est développé en python et permet l'enregistrement et le traitement de fichiers réseau. Il a l'avantage d'avoir une structure de donnée claire, simple et intuitive particulièrement adaptée pour l'apprentissage automatique, ce qui en fait un candidat intéressant pour ce sujet.

### 1.2.4 Intérêt et objectifs

L'analyse du trafic réseau est une étape cruciale qui s'appuie sur les concepts et manipulations abordés précédemment, notamment l'acheminement, la génération, la capture et l'enregistrement du trafic. En intégrant des équipements réseau dédiés, tels que le switch d'Interface Concept, il devient possible d'obtenir une visibilité complète et en temps réel sur les communications réseau.

Ce switch, positionné stratégiquement, permet d'ajouter des cartes d'extension équipées de CPU ou d'autres composants dédiés à des fonctions spécifiques, comme l'exécution d'algorithmes d'analyse de trafic. Cette capacité d'extension offre une flexibilité et une puissance accrues pour surveiller et analyser les flux de données, détecter les anomalies et optimiser la qualité de service (QoS). De plus, l'intégration de l'intelligence artificielle dans ces processus ouvre de nouvelles perspectives pour automatiser et affiner l'analyse, permettant ainsi une réponse plus rapide et plus précise aux menaces potentielles. La section suivante explorera plus en détail ces avancées et les améliorations apportées par l'IA dans l'analyse du trafic réseau.

Dans la suite de l'étude, l'attention se porterait sur une analyse du trafic réseau personnalisable, capable de s'adapter à différents objectifs tels que la détection d'anomalies, la prévention des attaques, l'optimisation de la qualité de service (QoS) et la supervision continue. En tenant compte des divers types de trafic, qu'ils soient chiffrés, industriels, web ou VPN, l'approche envisagée visait à rediriger ces flux vers des sondes, des systèmes de détection d'intrusion (IDS) ou d'autres dispositifs spécialisés afin de permettre une analyse plus approfondie. Cette phase de préanalyse permettait non seulement d'identifier rapidement les comportements suspects, mais aussi de préparer le terrain pour des traitements locaux plus avancés. En intégrant ces capacités directement au niveau du switch développé par Interface Concept, il serait ainsi possible d'optimiser l'efficacité et la réactivité de l'infrastructure réseau face aux défis actuels en matière de cybersécurité.

## 1.3 Méthodes classiques d'analyse du trafic

Depuis quelques années, le domaine de l'analyse de trafic est en plein essor. En effet, l'amélioration des capacités de calcul permettent l'utilisation d'algorithmes laissés de côté, car trop consommateurs de ressources pour pouvoir être utilisés de manière opérationnelle. Ce domaine est tout particulièrement présent dans le milieu de la cybersécurité où les attaques sont de plus en plus nombreuses et empruntent cette voie. Dans cette section sont explorées les techniques et méthodes utilisées pour analyser le trafic réseau ainsi que les données qui y sont traitées.

### 1.3.1 Analyse par couches (modèle OSI ou TCP/IP)

Un analyseur de réseau comme Wireshark déduit le protocole utilisé et segmente les paquets en s'appuyant sur plusieurs aspects des en-têtes de paquets dans les différentes couches du modèle OSI (ou TCP/IP).

Wireshark analyse chaque paquet en fonction de la pile de protocoles du réseau, en partant de la couche physique jusqu'à la couche application. Lorsqu'un paquet arrive, il est interprété dans l'ordre des couches :

— Couche Liaison de données (Ethernet) : Le paquet commence par un en-tête Ethernet, qui

identifie le type de protocole de la couche supérieure. Par exemple, le champ EtherType dans l'en-tête Ethernet indique si le paquet est un paquet IPv4 (0x0800) ou IPv6 (0x86DD), ou un autre protocole de la couche réseau.

— Couche Réseau (IP) : Si le protocole de la couche réseau est IPv4 ou IPv6, Wireshark va examiner l'en-tête IP, qui contient un champ appelé Protocole. Ce champ spécifie le protocole de la couche transport (par exemple, TCP : 6, UDP : 17, ICMP : 1).

— Couche Transport (TCP/UDP) : Une fois le protocole de la couche transport identifié (TCP ou UDP), Wireshark va lire les informations spécifiques comme les numéros de ports. Par exemple, un port TCP 80 suggère que le protocole applicatif est probablement HTTP.

— Couche Application : Si Wireshark identifie un protocole applicatif (HTTP, DNS, etc.), il va alors analyser les données de cette couche en fonction de règles spécifiques au protocole. Certains protocoles (comme TLS) sont plus complexes à analyser directement en raison du chiffrement des données.

### 1.3.2 Utilisation de bases de données protocolaires

Wireshark s'appuie également sur une base de données de signatures de protocoles (ou heuristiques). Ces signatures sont des modèles définis qui correspondent aux en-têtes de certains protocoles. Chaque protocole a des schémas spécifiques dans ses en-têtes, et Wireshark les reconnaît pour pouvoir déterminer de quel protocole il s'agit.

### 1.3.3 Segmentation des paquets

La segmentation des paquets fait référence à l'identification des différentes couches d'encapsulation dans un paquet réseau. Une fois que Wireshark a identifié un protocole à chaque couche (Ethernet, IP, TCP/UDP), il peut extraire les données encapsulées et les interpréter en fonction des règles du protocole sous-jacent. Par exemple :

— Un paquet Ethernet encapsule un paquet IP.

— Un paquet IP encapsule un segment TCP ou UDP.

— Un segment TCP encapsule les données applicatives (par exemple, HTTP).

Chaque niveau d'encapsulation est analysé séparément, et Wireshark fournit une vue hiérarchique des différents en-têtes.

### 1.3.4 Analyse par inspection des Flow

Un *flow* (ou parfois *traffic flow*, *packet flow* ou *network flow*) est une séquence de paquets correspondant à un échange entre une machine source et un destinataire (un hôte, un groupe

| **Protocole** | NetFlow | sFlow |
|---|---|---|
| **Niveau OSI** | 3 | 2 |
| **Version** | 10 | 5 |
| **Échantillonnage** | Optionnel | Obligatoire |
| **Entête** | L2-L3 | L2-L7 |
| **Envoie** | UDP/SCTP | UDP |

TABLE 1.4 – Tableau de comparaison entre le NetFlow et le sFlow

multicast, etc.). Ce terme est couramment utilisé en français sous cette forme. Bien qu'il puisse être traduit par "flux réseaux", son anglicisme est préféré, car mieux défini.

Dans le cadre de nos travaux, nous retenons la définition d'un flow comme un ensemble de paquets partageant certaines caractéristiques communes — telles que les adresses IP source et destination, les ports et le protocole — et observés sur une période donnée. Cette définition, à la fois précise et opérationnelle, permet de regrouper efficacement les paquets en sessions logiques, facilitant ainsi l'analyse comportementale et statistique du trafic réseau. Elle est notamment utilisée par le protocole **NetFlow**, développé initialement par Cisco, et constitue aujourd'hui une référence dans le domaine.

Il existe cependant d'autres définitions, comme celle donnée par la **RFC 3917**, dans laquelle un flow est décrit comme « un ensemble de paquets IP passant par un point d'observation du réseau pendant un certain intervalle de temps ». Cette définition, bien que valide dans un contexte général, nous semble trop vague pour permettre une analyse fine et exploitable des comportements réseau.

Cisco est le premier à proposer une architecture de surveillance réseau avec **NetFlow**. En 2004, il publie la version 9 du protocole, qui servira de base au standard **IPFIX** (*IP Flow Information Export*), normalisé en 2008. IPFIX est aujourd'hui largement adopté pour l'exportation d'informations détaillées sur les flux IP. Un autre protocole de l'industrie est le **sFlow** (*sampled Flow*), initialement développé par InMon Corp. et désormais maintenu par le consortium sFlow.org. Contrairement à NetFlow/IPFIX, sFlow fonctionne en couche 2 et applique un échantillonnage aléatoire, ne sélectionnant qu'un paquet tous les $n$ paquets observés. Le tableau 1.4 compare le NetFlow avec le sFlow.

Enfin, le terme **Deep Flow Inspection (DFI)** est parfois employé dans la littérature[18, 19] pour désigner une analyse approfondie des flows réseau. Contrairement à la Deep Packet Inspection (DPI), qui se concentre sur le contenu des paquets (y compris les données applicatives), la DFI s'appuie sur des caractéristiques statistiques et temporelles extraites de l'ensemble d'un flux (durée, taille moyenne des paquets, intervalles entre les paquets, directionnalité, etc.). Cette approche permet notamment d'identifier des comportements ou des usages spécifiques, même lorsque le trafic est chiffré, en exploitant uniquement les métadonnées des échanges.

### 1.3.5 Analyse par Deep Packet Inspection (DPI)

Le Deep Packet Inspection (DPI) est une technique utilisée pour examiner en profondeur les données transitant sur le réseau. Cette méthode accède aux données utiles de chaque paquet tel que le contenu d'un e-mail, d'une image, d'une vidéo, etc. Il s'avère assez efficace pour détecter des attaques informatiques ou classifier du contenu (mail, image, etc.), mais il pose un principal problème, celui de la confidentialité des données des utilisateurs. En effet, bien que cette pratique se soit avérée efficace au niveau de la recherche, elle est beaucoup plus difficile à appliquer en réalité, ne serait-ce que d'un point de vue légal.

Il est important de souligner que si le trafic est chiffré, cette technique ne peut pas être exploitée. Il est possible de mettre en place, dans certaines entreprises, un système de proxy[20] permettant de déchiffrer le trafic des utilisateurs et ainsi pouvoir appliquer cette méthode.

En parallèle, il ne faut pas oublier que cette méthode requiert beaucoup plus de puissance de calcul et de temps pour appliquer son analyse. En effet, la quantité de données à traiter est beaucoup plus importante que pour une analyse uniquement basée sur l'en-tête des paquets. Notre commutateur, en particulier, est également un équipement embarqué. Il faut donc ajouter à la liste des contraintes celle de la consommation limitée, restreignant la puissance de calcul possible.

### 1.3.6 Heuristiques et analyse avancée

Dans certains cas où le protocole n'est pas immédiatement identifiable (par exemple, si les numéros de port ne sont pas standard ou si les en-têtes sont offusqués), Wireshark utilise des heuristiques, c'est-à-dire des méthodes plus complexes qui tentent d'identifier le protocole en se basant sur les motifs des données observées.

En résumé, Wireshark détecte le protocole principalement grâce à l'analyse des en-têtes des paquets, en partant des couches basses (Ethernet, IP, TCP/UDP) jusqu'aux couches applicatives. Il utilise des tables de correspondance et des signatures pour identifier chaque protocole, puis segmente les paquets en fonction des couches identifiées.

L'analyse des échanges contenus dans les jeux de données présentés dépend principalement de la forme des enregistrements. Deux formes peuvent être distinguées principalement : les captures de paquets sous la forme de fichiers PCAP et des résumés d'échanges appelés Flow qui peuvent avoir plusieurs formats différents.

## 1.4 Classification réseau

La classification du trafic réseau est essentielle pour la gestion et la sécurisation des infrastructures modernes. Elle permet d'identifier et de catégoriser les flux de données, facilitant ainsi la détection d'intrusions (IDS), la gestion de la qualité de service (QoS) et la surveillance

continue. L'intégration de l'intelligence artificielle et du machine learning offre de nouvelles possibilités pour automatiser et affiner cette classification, en identifiant des motifs complexes et des comportements anormaux. Selon le contexte, la classification peut se concentrer sur la détection d'attaques, l'optimisation des services ou l'analyse des activités des utilisateurs, nécessitant une approche adaptée pour garantir une protection optimale et une performance accrue.

### 1.4.1 Détection et supervision : IDS, QoS, surveillance

La détection et la supervision du trafic réseau sont essentielles pour garantir la sécurité, la qualité de service (QoS) et la disponibilité des infrastructures informatiques. Ces mécanismes permettent d'analyser en temps réel le comportement des flux réseau afin d'identifier des anomalies, détecter des intrusions ou encore optimiser les performances des services. On distingue trois grandes catégories d'outils et de techniques associées :

— **Les systèmes de détection d'intrusion (IDS)** , qui surveillent le trafic à la recherche de menaces potentielles.
— **Les outils de supervision QoS** , qui mesurent la latence, la bande passante et l'utilisation des ressources.
— **Les solutions de surveillance réseau** , qui collectent et analysent en continu les métriques des équipements et applications.

Chacune de ces approches répond à des objectifs spécifiques et repose sur des méthodologies adaptées aux enjeux des environnements modernes, qu'ils soient classiques ou industriels.

#### Intrusion Detection System (IDS)

Les IDS sont des systèmes de détection d'intrusion. Il en existe différents types, mais les plus connus sont les systèmes de détection d'intrusion réseau (NIDS) et d'hôte (HIDS). Le premier se présente comme un composant réseau (une sonde), souvent relié à un port miroir de tous les autres (*port mirroring* en anglais) d'un autre matériel réseau (switch ou routeur généralement) pour pouvoir analyser le trafic. Le second est plutôt sous la forme d'un logiciel, un peu à l'image d'un antivirus, mais celui-ci est beaucoup plus poussé. Il doit être capable de détecter des attaques au niveau des processus du système ou de dépassement de tampon (couramment appelé overflow en anglais). La figure 1.9 montre le positionnement de ces systèmes dans un réseau. Il existe de nombreux NIDS comme, parmi les plus connus : Suricata, Snort, Ossec, ou Wazuh. Beaucoup d'entre eux peuvent servir d'HIDS également.

Dans l'ensemble, les IDS fonctionnent sur des règles statiques basées sur les signatures que peut laisser une attaque. Le problème de cette méthode est qu'elle peut-être facilement contour-

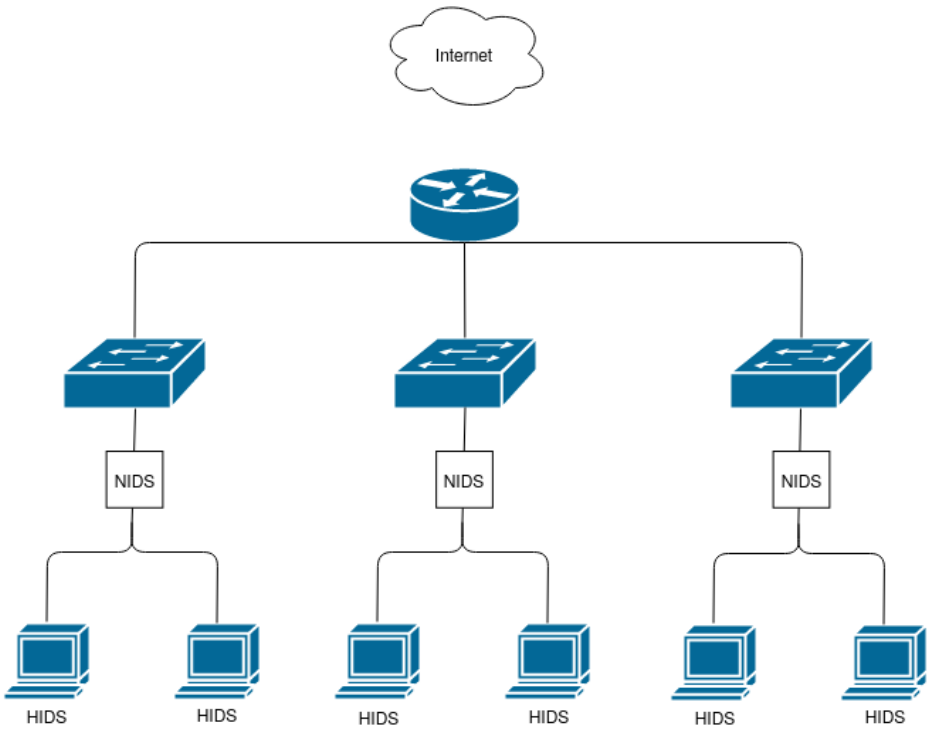



FIGURE 1.9 – Positionnement des IDS en fonction de leur type

nable pour un attaquant avancé, car il peut essayer de changer l'empreinte de son attaque. De plus, il est difficile, voire impossible, de détecter des attaques de type zéro-day, car elles ne possèdent pas (par définition) encore de signature identifiée.

Un autre type d'IDS est basé sur la détection d'anomalie. À l'aide d'algorithmes d'apprentissage automatique, l'IDS va "apprendre" le comportement normal au sein d'un réseau et tous les comportements inhabituels vont être remontés. Ce "comportement normal" doit être identifié au préalable. L'expression "Network Baseline" correspond aux échanges réseau habituels. Il varie en fonction des réseaux étudiés et même du contexte (Périodes de la journée, de la semaine ou spécifiques ...). Ils sont beaucoup plus récents et ne bénéficient pas encore d'une grande maturité, ce qui en fait un domaine de recherche très actif.

### 1.4.2 Machine Learning et IA pour l'analyse du trafic

L'analyse du trafic réseau est une composante essentielle pour la cybersécurité, la supervision des infrastructures et les tests de performance des réseaux. Elle permet de détecter des anomalies, d'identifier les tendances d'utilisation et d'assurer la qualité de service (QoS). Traditionnellement, cette analyse repose sur des techniques comme l'inspection approfondie des

paquets (Deep Packet Inspection - DPI) et l'analyse des flux réseau (Flow-based Analysis), qui permettent d'examiner le contenu ou les métadonnées des communications. Toutefois, ces approches sont limitées face à la complexité et au volume des données modernes, notamment avec la généralisation du trafic chiffré.

L'intelligence artificielle (IA) et plus spécifiquement le Machine Learning (ML) offrent des solutions avancées pour traiter et analyser ces flux de données de manière dynamique. Ces approches permettent non seulement de détecter les menaces et anomalies en temps réel, mais aussi d'optimiser les performances réseau en adaptant les configurations selon les usages. L'application du Machine Learning à l'analyse du trafic réseau couvre deux grands domaines :

— La détection d'intrusions (IDS - Intrusion Detection Systems), qui vise à identifier les activités malveillantes et les cyberattaques.

— L'analyse du trafic standard pour la supervision et l'optimisation du réseau, permettant de surveiller l'état des infrastructures, de tester des configurations et d'améliorer la gestion des ressources.

Dans cette section, nous explorons ces deux aspects, en détaillant les méthodes et algorithmes utilisés pour tirer parti du Machine Learning dans l'analyse du trafic.

L'apprentissage automatique, ou Machine Learning (ML) en anglais, est un domaine de l'Intelligence Artificielle (IA) très utilisé dans de nombreux domaines pour résoudre des problèmes de détection, de régression ou de classification. Son usage dans le domaine du réseau est également courant, notamment sur les travaux autour des systèmes de détection d'intrusion (IDS) qui sont des sondes analysant le trafic réseau et ayant pour but d'alerter le système lors d'une anomalie. Les règles sont établies et paramétrées manuellement en fonction du système dans lequel elles sont implémentées.

Pour éviter tout amalgame entre ces trois termes qui sont couramment utilisés et parfois confondus, voici les définitions proposées. La figure 1.10 représente le lien entre eux.

— L'**Intelligence Artificielle** est d'après le Larousse un "ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine."

— Le **Machine Learning** est un domaine de l'intelligence artificielle qui se base sur des techniques et des algorithmes permettant aux machines ou programmes d'apprendre automatiquement à partir des règles et données en entrées. Cela peut leur permettre de s'adapter en fonction de la situation.

— Le **Deep Learning** est un domaine du Machine Learning se basant sur des "réseaux de neurones" complexes (nombre de neurones important). À l'image du cerveau humain,

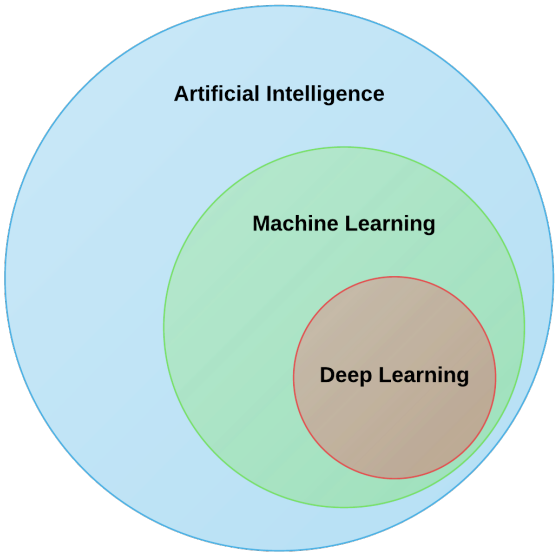



FIGURE 1.10 – Relations entre IA, ML et DL

ces algorithmes sont des couches de neurones reliées entre elles qui miment le processus d'apprentissage.

### Supervisé ou non supervisé

Au sein du Machine Learning nous avons deux types d'apprentissages, le supervisé et le non supervisé. Dans l'apprentissage supervisé, les données doivent être étiquetées, c'est-à-dire que l'algorithme sait à quelle catégorie chaque donnée appartient. Pour l'apprentissage non supervisé, les données d'apprentissage ne sont pas étiquetées, l'algorithme va donc catégoriser lui-même les données.

Par exemple, pour un algorithme d'apprentissage dont le but est de détecter des attaques dans des communications réseau, un jeu de données d'apprentissage contenant des catégories d'attaques identifiées ou bien seulement deux catégories : "normal" et "anormal". Il s'agit alors d'apprentissage supervisé. Par la suite, l'algorithme aura donc appris le "motif" de chaque attaque et sera en mesure de détecter toute autre nouvelle donnée ressemblante et donc de la classifier dans la bonne catégorie.

À l'inverse, nous aurions également pu, soit ne pas étiqueter les données d'apprentissage, soit ne pas fournir de données à l'algorithme. Nous aurions laissé l'algorithme trouver tout seul les motifs dans le trafic et il se serait chargé de les regrouper ensemble dans des catégories, on parle alors de clustering.

À mi-chemin entre ces deux types, nous avons l'apprentissage semi-supervisé. Cette mé-

thode est tout simplement une approche hybride des deux autres. Une partie des données seront étiquetés et ça sera à l'algorithme de déduire les étiquettes du reste du jeu de données.

### Algorithmes de classification courants

Il existe plusieurs algorithmes couramment utilisés dans la détection d'anomalies. Parmi les plus connus, nous avons :

— Les **Support Vector Machine (SVM)** sont des algorithmes d'apprentissage permettant de séparer les jeux de données de manière linéaire. Les fonctions noyau permettent de changer de dimensions pour régler les problèmes non séparables linéairement comme le montre la figure 1.11.

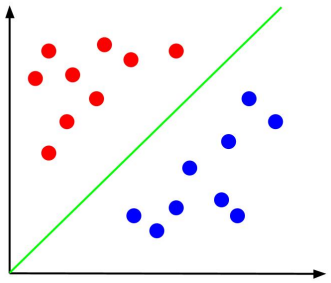


FIGURE 1.11 – Séparation de deux classes avec SVM

— Les **Random Forest (RF)** sont des algorithmes basés sur des arbres décisionnels. Une multitude d'arbres vont être générés de manière aléatoire, d'où le terme forêt, jusqu'à ce que chaque élément soit isolé. À la fin de cette étape, ils sont assemblés et permettent de prédire la classe d'une donnée inconnue à partir de tous les arbres.

— Les **Naïve Bayes (NB)** sont un classifieur basé sur le théorème de Bayes. À partir des données d'entraînement, l'espérance et la variance de chaque caractéristique sont calculées afin de déterminer la probabilité postérieure la plus grande.

— Le **C4.5** est un algorithme de classification supervisé qui génère un arbre de décision. Il se base sur l'entropie pour choisir l'attribut qui divise le mieux ses échantillons dans une classe ou une autre.

— Les **réseaux de neurones** sont des réseaux inspirés du fonctionnement des neurones biologiques. Chaque neurone possède des coefficients de pondération qui sont affinés à chaque passage. Il en existe différents types comme les réseaux de neurones artificiels

(ANN), convolutifs (CNN), récurrents (RNN) et bien d'autres encore. La figure 1.12 montre un réseau de neurones simple à trois couches.

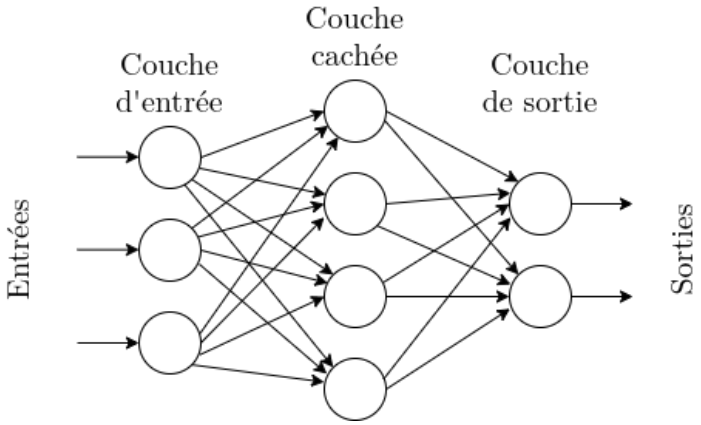



FIGURE 1.12 – Réseau de neurones à 3 couches

Évidemment, il en existe de nombreux autres, mais ceux-ci font partie des algorithmes incontournables et sont également ceux utilisés dans la littérature scientifique lors de comparaisons avec d'autres.

### 1.4.3 Classification du trafic : attaques, services, activités

La classification du trafic réseau couvre un spectre assez large de thématiques. En effet, elle peut se concentrer sur une partie seulement ou la totalité du trafic. Elle peut, par exemple, être utilisée dans un but de surveiller les activités des utilisateurs, détecter des anomalies, le type de trafic, déduire certains comportements, etc.

#### Classification par attaques

Un des objectifs principaux lors de la détection d'anomalie est aussi de l'identifier afin de la traiter en conséquence. Dans le cas d'une attaque, sa bonne classification va permettre de bien protéger son système.

MITRE propose la matrice ATT&CK (voir figure 1.13) qui répertorie l'ensemble des techniques et tactiques utilisées par les attaquants. C'est une base de données très complète qui permet également de s'informer sur les contre-mesures pour se protéger. De nombreuses applications utilisent cette matrice en référence.




FIGURE 1.13 – Matrice ATT&CK de MITRE

### Classification par type de trafic

De plus en plus de personnes utilisent couramment des moyens pour chiffrer leurs échanges ou anonymiser leur connexion. Les raisons de ces pratiques sont multiples et pas uniquement pour des questions de sécurité. Les Virtual Private Network (VPN) permettent de chiffrer ses échanges pour ne pas que les données, si elles étaient subtilisées, ne puissent être lisibles et exploitées par un attaquant. En effet, l'utilisateur va d'abord se connecter à un serveur, et donc emprunter l'adresse IP de celui-ci, avant d'accéder à Internet. Cela permet d'accéder à des sites bloqués par la géorestriction ou d'obtenir des contenus exclusifs à un endroit.

Une autre technique permettant l'anonymisation de la connexion est l'usage du navigateur Tor (The onion Router). La connexion va passer par plusieurs serveurs appelés "noeuds" qui permettent de "cacher" l'adresse IP d'origine de l'utilisateur et donc le rendre anonyme.

Le chiffrement ainsi que l'anonymisation du trafic permettent de renforcer sa protection contre l'analyse de trafic. Les chercheurs essayent donc de classifier ces types de trafic. Par exemple, Lashkari *et al.* montrent la possibilité d'identifier et caractériser le trafic transitant par un VPN en utilisant des caractéristiques basées sur le temps [21]. Les mêmes auteurs ont réalisé la même étude pour du trafic transitant par le réseau TOR [22].

### Classification par activités

Avant toute chose, il est important de définir ce qu'est une activité. Dans notre contexte, une activité est une série d'une ou plusieurs actions que l'on accomplit dans le but de réaliser quelque chose ou de produire un résultat. Au niveau du réseau, une activité va se traduire par un ensemble d'opérations qui permet de l'identifier. Il existe plusieurs manières de catégoriser les activités dont voici quelques-unes.

Une des plus simples est d'avoir chaque utilitaire étiqueté par sa catégorie d'activité. C'est la méthode utilisée par de nombreux logiciels pour estimer le temps passé par catégorie.

À titre d'exemple, les catégories d'activité sous iOS sont les suivantes :

— Réseaux sociaux
— Utilitaires
— Productivité
— Photo et vidéo
— Voyages
— Finance
— Divertissement
— Information et lecture
— Achats et nourriture

— Santé et forme
— Musique
— Jeux
— productivité et finances
— Éducation
— Autres
— Etc.




FIGURE 1.14 – Suivi d'activités sur iOS

Chaque application du téléphone est associée à une catégorie, il est donc facile par la suite de calculer le temps passé sur chaque type d'activité, comme on peut le voir sur la figure 1.14. Cependant, ces données ne sortent pas de la machine de l'utilisateur et ne transitent pas sur le réseau.

Dans la littérature scientifique, on retrouve la classification des activités lors de l'analyse du trafic. Les activités sont les suivantes :

— Browsing

— Email

— Chat

— Audio-Streaming

— Video-Streaming

— File Transfer

— VoIP

— P2P

Les catégories sont souvent celle-ci, car ce sont celles-ci qui sont labellisées dans les jeux de données utilisés.

En s'inspirant des catégories précédentes, voici un exemple de taxonomie des activités numériques, basée sur le domaine d'application :

— **Activités liées à la communication** : envoyer et recevoir des emails, utiliser des messageries instantanées, utiliser des réseaux sociaux, etc.

— **Activités liées au travail** : utiliser des logiciels de bureautique, partager des documents en ligne, travailler à distance en utilisant des outils de collaboration, etc.

— **Activités liées aux loisirs** : regarder des vidéos en ligne, écouter de la musique en ligne, jouer à des jeux vidéo en ligne, etc.

— **Activités liées à l'éducation** : suivre des cours en ligne, utiliser des ressources éducatives en ligne, participer à des forums ou des groupes de discussion pour échanger avec d'autres personnes sur des sujets d'intérêt, etc.

— **Activités liées à la gestion personnelle** : gérer son emploi du temps, suivre son alimentation, suivre ses finances, etc.

Cette taxonomie peut bien sûr être adaptée ou modifiée selon les besoins,les objectifs ou le contexte. Par exemple, on peut ajouter d'autres catégories (les activités liées à la santé, à la création de contenu, etc.) ou regrouper les activités de différentes manières (en fonction de leur niveau de complexité, de leur signature réseau, etc.).

En allant plus loin, l'objectif est de déduire des profils d'utilisateur à partir de ces données.

La taxonomie des profils d'utilisateurs numériques est un système de classification des utilisateurs des technologies numériques en fonction de leur comportement, de leurs intérêts, de leurs compétences ou de leur niveau d'expertise. Cette taxonomie peut être utilisée pour étudier les différents types d'utilisateurs et mieux comprendre leurs besoins, leurs préférences et leurs comportements en ligne.

#### Classification WEB

Un bon nombre de domaines s'intéressent à l'analyse du trafic Web. Mais avant tout, il est important de le définir. Le Web fait référence au World Wide Web (WWW) qui nous permet de consulter un ensemble de pages à l'aide d'un navigateur. C'est un des services accessibles sur internet, mais il en existe bon nombre d'autres, celui-ci étant le plus populaire. Aujourd'hui, pratiquement tous les services utilisant internet passent par le Web. Par exemple, beaucoup n'ont plus de client de messagerie dédié, mais bien une interface Web pour accéder à leurs mails. De même pour le transfert de fichier, beaucoup de personnes utilisent des services comme WeTransfert [1] ou encore Smash [2] mais plus rares sont ceux qui utilisent toujours FileZilla [3]. L'évolution de ces pratiques font du Web un domaine d'étude très important et intéressant. En effet, l'analyse de ce trafic peut permettre de détecter des comportements inhabituels.

## 1.5 Analyse des Jeux de données existants

Cette section regroupe l'ensemble des jeux de données populaires présents dans la littérature. On distingue deux principales finalités qui reviennent souvent, à savoir, la détection et classification des attaques au sein d'un réseau, et la catégorisation du trafic par l'analyse du réseau. Il arrive que certains soient utilisés pour les deux motifs, dans ce cas, ils n'apparaîtront, parfois, que dans une des deux tables pour plus de clarté.

Dans la dernière partie, nous introduisons la notion de granularité des classifications d'activités, puis nous proposons une vue d'ensemble de tous les jeux de données étiquetés en fonction de celle-ci, ainsi que le type de réseau dans lequel l'enregistrement a été produit.

### 1.5.1 Jeux de données orientés détection d'intrusion

Un des objectifs de ces jeux de données, est l'entraînement d'algorithmes d'intelligence artificielle, afin d'automatiser la détection et la classification des attaques. Les tableaux 1.5, 1.6, 1.7 et 1.8, tirés des travaux de Hindy etal.[23], Thakkar etal. [24] et Sharafaldin2017 etal. [25] pour les jeux de données datant de 2018 et antérieurs, puis enrichi par mes soins, offre une vue d'ensemble de plusieurs jeux de données populaires, ainsi que les différentes attaques présentes dont la plupart sont étiquetées.

— **KDD-Cup** : KDD-Cup est un jeu de données encore très largement utilisé dans le milieu de la recherche, notamment dans le domaine du machine learning et des systèmes de détection d'intrusion[26]. En effet, il a longtemps été l'un des rares jeux de données d'échanges réseau contenant des attaques, ce qui justifie sa présence dans de nombreux

1. https ://wetransfer.com/
2. https ://fromsmash.com/
3. https ://filezilla-project.org/

travaux. Malheureusement, il contient beaucoup de défauts ce qui n'en fait pas le jeu de données le plus adapté à la détection d'anomalies. De plus, il n'est pas représentatif du trafic réseau réaliste, car il a été réalisé lors d'un challenge. Aussi, datant de 1999 il ne contient pas de trafic proche du trafic réel (par exemple aucun trafic de type IOT ou OT).

— **NSL-KDD** : Le jeu de données NSL-KDD est une amélioration de KDD-Cup. Il résout certains des problèmes présents dans ce dernier, notamment la redondance des connexions. Malgré tout, il reste loin d'être représentatif du trafic réseau existant. Cependant, il peut parfois toujours être utilisé, notamment pour des évaluations de performance pour pouvoir facilement comparer les résultats avec d'autres algorithmes.

— **UNSW-NB15** : Le jeu de données UNSW-NB15 contient un set d'entraînement de 175 341 enregistrements ainsi qu'un set de test de 82 332 enregistrements. Il compte 9 attaques différentes parmi des DoS, des vers, backdoor et fuzzer. Il est paru en 2015 dans un article sur les systèmes de détection d'intrusion[27]. Il a été créé comme alternative aux précédents jeu de données KDD.

— **ISCXVPN2016** : En 2016, une publication sur la caractérisation de trafic chiffré et VPN est sortie[21]. C'est dans ce contexte qu'a été créé ce jeu de données. Il a permis de montrer comment détecter si un flux provenait d'un VPN ou non, mais également de détecter certaines activités (Email, Chat, streaming, etc.) bien que les échanges soient chiffrés.

— **ISCXTor2016** : Ce jeu de données a été produit dans le cadre de l'article Lashkari[22] sur la caractérisation du trafic Tor en utilisant des fonctions basées sur le temps. Il a permis de montrer comment détecter si un flux provenait de Tor, mais également de détecter certaines activités (Email, Chat, streaming, etc.).

— **CIC-IDS2017** : Ce jeu de données est probablement un des plus récents et représentatif du trafic réseau actuel. Il sert de référence dans beaucoup d'articles sur les IDS. Il contient un enregistrement sur 5 jours durant lesquels des attaques diverses ont lieu[28].

| General Purpose Networks [4] | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Year | Dataset | Normal | DoS | DDoS | Probe | U2R | R2L | Infiltrating/Scanning | Brute Force | | Heartbleed | Web | | | | | Botnet | Network and Host Events | Port Scan | Meterpreter |
| | | | | | | | | | SSH | FTP | | Brute Force | XSS | Sql Injection | Webshell | DCWA | | | | |
| 2019 | CIC DDoS dataset [29] | ✓ | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2018 | CICIDS2018 [28] | ✓ | ✓ | ✓ | - | - | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | - | ✓ | - |
| 2017 | CICIDS2017 [28] | ✓ | ✓ | ✓ | - | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | - | ✓ | - | ✓ | - |
| 2017 | CIC DoS dataset [30] | ✓ | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2017 & 2013 | AFDA-IDS [31] | ✓ | - | - | - | ✓ | ✓ | - | ✓ | ✓ | - | - | - | - | ✓ | - | - | - | - | ✓ |
| 2017 | Unified Network Dataset [32] | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | ✓ | - | - |
| 2016 | DDoSTB [33] | ✓ | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2015 | Boosters [34] | - | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2015 | TUIDS Coordinated Scan [35] | ✓ | - | - | ✓ | - | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - |
| 2015 | TUIDS DDoS [35] | ✓ | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2015 | TUIDS Intrusion [35] | ✓ | ✓ | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2015 | UNSW-NB15 [36] | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ | - | - | - | - | - | - | - | - | - | - | ✓ | - |
| 2014 | Botnet dataset [37] | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | ✓ | - | - | - |
| 2012 | STA2018 [38] | ✓ | ✓ | ✓ | - | - | - | ✓ | ✓ | - | - | - | - | - | - | - | - | - | - | - |
| 2011 | CTU-13 [39] | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | ✓ | - | - | - |
| 2010 | ISCXIDS2012 [40] | ✓ | ✓ | ✓ | - | - | - | ✓ | ✓ | - | - | - | - | - | - | - | - | - | - | - |
| 2009 | Twente [41] | ✓ | - | - | - | - | - | - | ✓ | ✓ | - | - | - | - | - | - | - | - | - | - |

4. Voir annexe A pour les liens de téléchargement

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2009 | Kyoto [42] | ✓ | not describe any details about the attack types | | | | | | | | | | | | | | | | |
| 2009 | Waikoto [43] | - | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2009 | ASNM-CDX [44] | ✓ | - | - | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2007 | CAIDA DDoS [45] | - | - | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2004-2005 | LBNL [46] | - | - | - | - | - | - | ✓ | - | - | - | - | - | - | - | - | - | ✓ | - |
| 2000 | DEFCON [47] | - | - | - | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | - | - | - | - | - | - | ✓ | - | - |
| 1999 | NSL-KDD [48] | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1999 | KDD'99 [49] | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 1998, 1999, 2000 | DARPA [50] | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | - | - | - | - | - | - | - | - | - | - | - | - |

TABLE 1.5: Jeux de données contenant des attaques dans les réseaux *classiques*

Il existe également d'autres jeux de données utilisés pour les systèmes de détection d'intrusion, mais sur des cibles plus spécifiques. On peut trouver parmi eux les réseaux mobiles, les systèmes industriels (ICS) ainsi que les réseaux d'internet des objets (IoT). Les tableaux suivant sont non exhaustifs et seront enrichis au fur et à mesure de l'évolution des problématiques. Le choix de leur classification est donc purement subjectif.

| General Purpose Mobile Networks | | | | | |
|---|---|---|---|---|---|
| Year | Dataset | Bening | Botnet | Adware | Malware |
| 2016 | Kiss2016 Malware Dataset [51] | - | - | - | ✓ |
| 2015-2017 | Android Adware and General Malware Dataset [22] | ✓ | - | ✓ | ✓ |
| 2010-201 | Android Botnet Dataset [52] | - | ✓ | - | - |
| 2010-2011 | Android Malware Genome [53] | - | - | - | ✓ |
| - | AndroMalShare | - | - | - | ✓ |

TABLE 1.6: Jeux de données contenant des attaques dans les réseaux mobiles

| General Purpose ICS Networks | | | | |
|---|---|---|---|---|
| Year | Dataset | Bening | Attack | |
| | | | Modbus | S7COMM |
| 2023 | CIC Modbus Dataset [54] | - | ✓ | - |
| 2020-2023 | HAI Security Dataset [55] | ✓ | - | ✓ |
| 2017 | Anomalies Water System [56] | ✓ | ✓ | - |
| 2015 | 4SICS ICS [57] | ✓ | - | ✓ |

TABLE 1.7: Jeux de données contenant des attaques dans les réseaux industriels

| General Purpose IoT Networks | | | | |
|---|---|---|---|---|
| Year | Dataset | Bening | Botnet | Malware |
| 2023 | CICIoT2023 [58] | - | ✓ | - |
| 2021 | Enriching IoT datasets [59] | - | ✓ | - |
| 2021 | TON-IoT [60] | ✓ | - | ✓ |
| 2018 | Bot-IoT [61] | - | ✓ | - |

TABLE 1.8: Jeux de données contenant des attaques dans les réseaux IoT

On remarque que tout le trafic considéré comme "normal", quand il existe, n'est représenté que par une seule et unique classe du même nom (normal ou bénin). Cela nous interroge sur la manière dont est catégorisé notre trafic. Comment caractériser une attaque dans le trafic bénin sachant que celui-ci peut avoir de multiples formes. La section suivante propose un résumé des contenus des jeux de données utilisé pour l'apprentissage des systèmes de détection d'intrusion dans les réseaux classiques.

### 1.5.2 Synthèse des jeux de données réseau pour IDS

La tableau 1.10 ci-dessous synthétise les principaux jeux de données utilisés pour les systèmes de détection d'intrusion en détaillant quelques-unes de leurs caractéristiques comme la durée d'enregistrement, les différentes classes présentes ou les formats disponibles.

| Summary of public datasets | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Year | Dataset | Period | # of users / machines | # of Classes | Traffic Types | # of available features | Labeled | Format |
| 2019 | CIC DDoS dataset | 1 day | 25 | 13 | 12 DDoS attacks and benign traffic | 80 | Yes | PCAP, CSV |
| 2018 | CICIDS2018 | 5 days | >420 | 13 | 12 attacks and benign traffic (vérifier labels) | 80 | Yes | PCAP, CSV |
| 2017 | CICIDS2017 | 5 days | 25 | 13 | 12 attacks and benign traffic (vérifier labels) | 80 | Yes | PCAP, CSV |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2017 | CIC DoS dataset | 1 day | - | - | 10 DoS attacks and normal traffic | - | - | PCAP |
| 2017 & 2013 | AFDA-IDS | 5 days | - | - | 10 attack vectors | System call traces | Yes | PCAP, flow |
| 2017 | Unified Network Dataset | 90 days | LAN | - | - | 11 | Yes | Flow, CSV |
| 2015 | Booters | 2 days | - | 9 | 14 booter attacks | - | Yes | PCAP |
| 2015 | TUIDS Coordinated Scan | 7 days | >250 | 2 | 6 attacks | 50 | Yes | PCAP, Flow |
| 2015 | TUIDS DDoS | | | | 6 attacks | | | |
| 2015 | TUIDS Intrusion | | | | 22 attacks | | | |
| 2015 | UNSW-NB15 | 31h | - | 10 | 9 kind of attacks and normal traffic | 42 | - | PCAP, CSV |
| 2014 | Botnet dataset | Dataset mix [5] | | 16 | 16 botnet types | 43 | Yes | PCAP |
| 2012 | STA2018 | 5 days | - | 2 | Attack and Normal | 550 | Yes | CSV |
| 2011 | CTU-13 | 1h-66h | 1-10 bot | 4 | 13 different malware captures with normal, botnet and background traffic. | - | Yes | PCAP, Flow |
| 2010 | ISCXIDS2012 | 7 days | - | 2 | Benign and attack traffic. | 41 | Yes | PCAP |
| 2009 | Twente | 6 days | 35-2000 | - | Honeypot with SSH, FTP and Apache servers. | - | Yes | PCAP, Flow |
| 2009 | Kyoto | 3 years | 22 | - | Attack and normal sessions on 22 different honeypots. | 24 | Yes | Flow |
| 2009 | Waikato | 56-620 days | - | - | Waikato University traffic. | - | - | PCAP |
| 2009 | CDX | 4 days | ∼50 | - | Cyber Defense Exercise which incorporated a professional NSA red team and automated white traffic generation. | - | No | PCAP |

5. Merge of ISOT, ISCX 2012 IDS and Botnet traffic generated by the Malware Capture Facility Project datasets.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2007 | CAIDA DDoS | 1h | - | - | Only attack traffic to the victim and responses to the attack from the victim are included in the traces. Non-attack traffic has as much as possible been removed. | - | No | PCAP |
| 2004-2005 | LBNL | 10m-1h | - | - | Malicious traffic. | - | No | PCAP |
| 2000 | DEFNET | 1-3 days | - | - | CTF traffic capture. | - | No | PCAP |
| 1999 | NSL-KDD | 7 weeks | 100 | 5 | Normal, probane, Dos, U2R and R2L. | 117 | Yes | ARFF, CSV |
| 1999 | KDD'99 | 7 weeks | - | 2 | 24 attack types | 41 | Yes | CSV |
| 1998 | DARPA | 7 weeks | 100 | 2 | 27 attack types and 10 anomalies | - | No | PCAP |

TABLE 1.9: Jeux de données de trafic réseau

### 1.5.3 Jeux de données orienté analyse de trafic réseau

Un autre usage, plus général, des jeux de données réseau, est la classification des applications, protocoles, services, etc., à des fins de supervision. C'est un champ d'intérêt notamment pour les fournisseurs d'accès internet ou entreprises afin d'assurer une qualité de service, contrôler l'usage de certaines applications ou encore détecter des activités malveillantes. Azab et al. [62] conclu que parmi les différentes méthodes, la plus ancienne basée sur l'identification des ports est la plus simple à mettre en place, mais souffre de plusieurs lacunes, et que le DPI, bien que beaucoup plus précis, mais également gourmand en ressources, devient vite inutile de par le chiffrement des échanges (HTTPS), devenu standard aujourd'hui.

La table 1.10 expose des jeux de données couramment utilisés pour la classification du trafic réseau. Certains d'entre eux possèdent plusieurs étiquetages comme l'application utilisée (Facebook, Skype, etc.), la catégorie d'application (Streaming, VoIP, etc.) ou encore le type de trafic (Tor, VPN, etc.). Dans ce cas-là, il est possible que certains datasets apparaissent plusieurs fois.

| Summary of public network datasets | | | | | |
|---|---|---|---|---|---|
| Year | Dataset | # of Classes | Traffic Types | Labeled | Format |
| 2022-2020 | LASS2023 [63][64] | Varies | Apple iCloud Private Relay | Yes | CSV |
| 2020 | CIC DoHBrw 2020 [65] | 2 | DoH, no-DoH | Yes | Full PCAP |
| 2000-2020 | MAWI [66] | Varies | HTTP, FTP, SSH, DNS, SSL, DoS, other | No | Full PCAP |
| 2016 | ISCX [6] | 8 | Browsing, Email, Chat, Streaming, File Transfert, VoIP, P2P | Yes | Full PCAP |
| 2016 | ISCXTor2016 [67] | 2 | Tor, no-Tor | Yes | Full PCAP |
| 2016 | ISCXVPN2016 [68] | 2 | VPN, no-VPN | Yes | Full PCAP |
| 2016 | USTC-TFC2016 [69] | 2 | Facetime, Gmail, Skype, Zeus, Cridex, Htbot | Yes | Full PCAP |
| 2014 | CTU-2013 [39] | 13 | Botnet sample for each scenario | Yes | Full PCAP & ARFF |
| 2013 | UJN traces [70] | 9 | Browsing, chat, cloud disk, live update, stream media, mail, ftp, p2p, other | Yes | Full PCAP |
| 2009 | UNIBS traces [71] | 10 | BitTorrent, Skype, HTTP, eDonkey, IMAP, POP3, MSN, SMTP, urd, SSH | Yes | Full PCAP |
| 2005 | Moore [72] | 10 | Web, SMTO, POP3, FTP, DNS, BitTorrent, MySQL, Virus, Windows media player, Telnet, WOW | Yes | ARFF |
| 2000 | Auckland II [73] | 10 | FTP, FTP-data, http, imap, pop3, smtp, nntp, ssh, dns, Telnet | No | Packet header (No payload) |
| Summary of public mobile network datasets | | | | | |
| 2017-2019 | Mirage [74] | 40 | 40 Android apps | Yes | JSON |
| 2016-2018 | MobileGT [75] | 12 | 12 Mobile apps | Yes | ARFF |
| 2016-2017 | UNSW [76] | 28 | 28 IoT devices | No | Full PCAP, CSV |
| Summary of public ICS network datasets | | | | | |
| Summary of public IoT network datasets | | | | | |
| 2017 | IoT devices captures [77] | 27 | 27 IoT devices | Yes | PCAP |

TABLE 1.10: Jeux de données de trafic réseau

6. Inclus les jeux de données ISCXTor2016 et ISCXVPN2016.

Une importante partie de la recherche a été effectuée au niveau de l'analyse des jeux de données disponibles. Il est important de garder du recul sur ces derniers. L'annexe B montre également les différents liens existant entre les auteurs, et donc, les possibles erreurs ou inexactitudes redondantes liées aux méthodes similaires qui sont utilisées.

## 1.6 Activité humaine dans les réseaux

L'activité humaine est un générateur majeur de trafic dans les réseaux informatiques, bien qu'elle soit rarement considérée comme telle dans les approches classiques d'analyse réseau. Comprendre et modéliser les comportements des utilisateurs constitue pourtant un levier essentiel pour améliorer la détection d'anomalies, la classification des usages ou encore l'optimisation de la qualité de service. Pour cela, il peut être pertinent de s'appuyer sur des cadres conceptuels issus des sciences sociales, tels que la théorie de l'activité, qui propose une grille de lecture structurée des actions humaines dans leur contexte.

### 1.6.1 Théorie de l'activité

La théorie de l'activité est un terme générique qui désigne un ensemble de recherches et théories en sciences sociales ayant pour origine commune les travaux de psychologues soviétiques dans les années 30.

En psychologie, la théorie de l'activité est un courant de pensée qui s'intéresse à l'influence de l'environnement sur le comportement et le développement des individus. Elle met l'accent sur le rôle des activités et des interactions sociales dans l'apprentissage et la croissance personnelle.

La théorie de l'activité a eu une influence importante sur de nombreux domaines de la psychologie, notamment l'éducation, la cognition et le développement de l'enfant. Elle a également été utilisée pour étudier les dynamiques de groupe et la motivation.[78]

#### Théorie de l'activité selon Vygotski

Lev Semionovitch Vygotski est un psychologue soviétique qui est un des précurseurs de la théorie de l'activité. Il est considéré comme un des plus grands psychologues du XX$^{\text{ème}}$ siècle [79]. En effet, il est à l'origine de la théorie historico-culturelle du psychisme sur laquelle se base la théorie de l'activité. Cette théorie soutient que le psychisme humain n'est pas inné et héréditaire, mais est le résultat des influences sociales et culturelles dans une société.

### Théorie de l'activité selon Leontiev

Alexei Leontiev est un psychologue et philosophe soviétique. Il est le fondateur de la théorie de l'activité et était le disciple de Vygotski. Il s'est inspiré des travaux de ce dernier pour concevoir sa théorie. Comme le montre la figure 1.15, il introduit 3 niveaux d'analyse de la notion d'activité [80] :

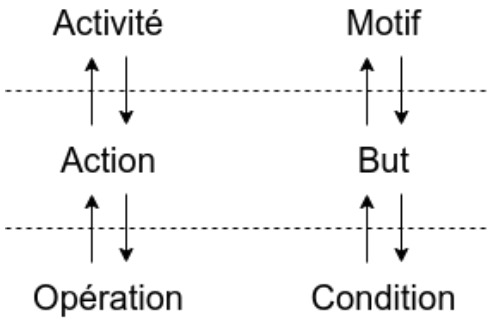



FIGURE 1.15 – Modèle hiérarchique de Leontiev

— L'**activité** elle-même, comme organisation fonctionnelle du comportement dans laquelle s'inscrivent les échanges entre le sujet et son milieu.
— Les **actions** ou conduites particulières imputables à un agent qui peuvent être considérées comme faisant partie de l'activité.
— Les **opérations** et les tâches qui sont impliquées dans la réalisation d'une action.

### Théorie de l'activité selon Engeström

La théorie de l'activité a été popularisée par le chercheur finlandais Yrjö Engeström. Il considère l'individu et la communauté comme un système à part entière, et permet d'analyser comment les objectifs collectifs sont atteints. Il a développé un modèle sous la forme d'un triangle avec une composante de l'activité à chacun de ses pôles, comme le montre la figure 1.16.

— Le **Sujet** est l'individu ou le sous-groupe que l'observateur a choisi d'analyser.
— L'**Objet** est la transformation de l'environnement qui est visé par l'activité (tâche à réaliser, objectif à atteindre).
— Les **Outils** sont les outils matériels ou symboliques qui médiatisent l'activité.
— La **Communauté** est l'ensemble des sujets (ou des sous-groupes) qui partagent le même objet et se distinguent par là même d'autres communautés.
— La **Division du travail** reprend à la fois la répartition horizontale des actions entre les sujets / les membres de la communauté, et la hiérarchie verticale des pouvoirs et des statuts.

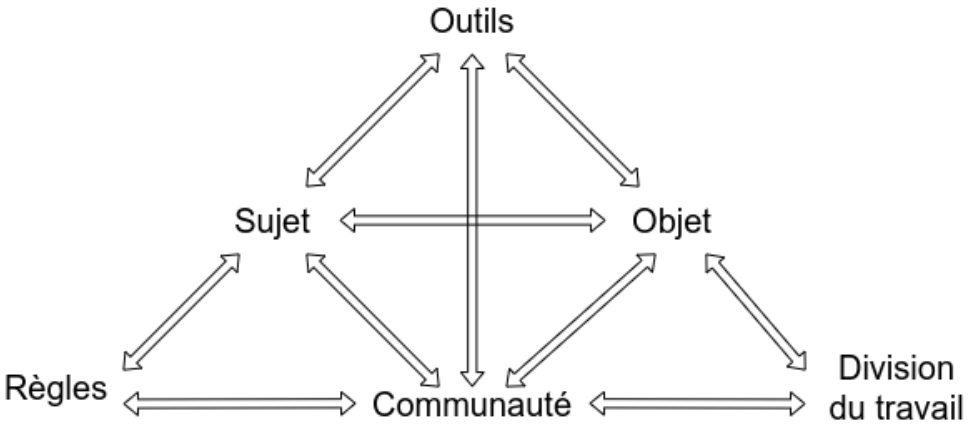



FIGURE 1.16 – Modèle général de l'activité de Engeström [3]

— Les **Règles** font référence aux normes, conventions, habitudes, etc., implicites et explicites qui maintiennent et régulent les actions et les interactions à l'intérieur du système.

### 1.6.2 Différences entre l'homme et la machine

Il est important de noter qu'une activité sur un réseau a forcément pour origine soit un humain soit une machine (programmes informatiques inclus). Le comportement d'une machine peut souvent s'avérer plus facile à déterminer du fait de la répétition cyclique de ses tâches. Le motif de son activité sera alors facilement mis en évidence parmi le trafic réseau. Il est possible que certains dispositifs s'activent de manière aléatoire et pas toujours avec le même comportement. Cependant, la plupart du temps, ils répondent à un ou plusieurs des critères suivants :

— Séquence prédéfinie

— Réponse à un évènement

— Temporalité dans la fréquence des échanges

— Exécution cyclique

— Protocole machine

Il est possible qu'il en existe d'autres, mais en prenant en compte l'ensemble de ces derniers, il est déjà possible de différencier la machine de l'humain. En prenant en compte le contexte et l'ensemble de ces informations, on peut également parfois déduire quelle est la machine utilisée et ce qu'elle exécute.

### 1.6.3 Transposition de la théorie de l'activité à l'activité réseau

Le but est de transposer cette théorie, initialement appliquée aux domaines des sciences sociales, de la psychologie ou de l'apprentissage, au domaine de l'informatique. L'objectif est de décomposer chaque composante d'une activité utilisateur au sein d'un réseau. Pour cela, nous allons baser notre approche sur le modèle général d'Engeström [3].

## 1.7 Conclusion

L'architecture des réseaux industriels évolue avec l'Industrie 4.0, poussant à une intégration de plus en plus forte avec l'IT. Si cette évolution offre de nouvelles opportunités d'optimisation et d'automatisation, elle impose aussi une refonte des stratégies de cybersécurité pour garantir la sûreté des systèmes industriels.

L'adoption de modèles éprouvés comme le modèle Purdue, les recommandations du SANS ICS410 et la stratégie de Défense en profondeur est essentielle pour assurer la résilience des ICS face aux cybermenaces croissantes. L'analyse du trafic réseau joue un rôle clé dans cette sécurisation, en permettant la supervision, la détection d'anomalies et l'amélioration des performances des infrastructures critiques. L'étude des méthodes classiques et avancées, appuyée par l'intelligence artificielle et l'apprentissage automatique, montre l'importance de la classification et de l'exploitation des jeux de données pour affiner les capacités de détection.

Cependant, il est crucial de comprendre les techniques permettant de contourner ces classifications. Le padding, qui consiste à ajouter des données de bourrage aux paquets, et la manipulation de la temporalité des échanges sont des méthodes couramment utilisées pour brouiller les signatures réseau. La technique du morphing, combinant ces deux approches, permet de faire correspondre les échanges à la signature d'un autre type de trafic, dans le but de mal classifier la communication. Bien que ces techniques puissent être efficaces, les méthodes actuelles permettent souvent de les contrer, les rendant obsolètes.

Dans cette perspective, le switch d'Interface Concept occupe une place stratégique, étant positionné au cœur du trafic. Son rôle central lui confère un potentiel d'extension particulièrement intéressant. L'ajout d'une carte d'extension lui permettrait d'élargir ses capacités d'analyse et d'intégrer des fonctionnalités avancées dédiées à l'observation et à la classification des activités humaines sur le réseau. Une telle approche offrirait un moyen efficace d'identifier les comportements suspects, d'optimiser la qualité de service (QoS) et d'améliorer la supervision globale du réseau.

L'intégration de la dimension humaine dans l'analyse du trafic souligne l'impact des comportements des utilisateurs sur la dynamique des flux réseau. L'application de la théorie de l'activité à l'étude des communications permet une meilleure compréhension des interactions et une adaptation plus fine des mécanismes de classification et de sécurité.

L'évolution des réseaux industriels exige ainsi une approche combinant expertise technique et analyse comportementale pour garantir un équilibre entre performance, automatisation et protection contre les menaces émergentes. En s'appuyant sur des solutions comme l'extension des capacités du switch d'Interface Concept, il devient possible de mieux comprendre ce qui se passe sur le réseau et d'agir en conséquence, que ce soit pour des fins de détection, de QoS ou de supervision.

Chapitre 2

# Modélisation de l'activité humaine dans les réseaux

## Sommaire


**2.1 Modélisation de la classification réseau . . . . . 71**
2.1.1 Modèles théoriques de communications réseau . . . . . 72
2.1.2 Modèle d'analyseur réseau . . . . . 74
2.1.3 Modélisation de la granularité de l'activité . . . . . 78
**2.2 Modélisation de l'activité humaine dans les réseaux informatiques . 84**
2.2.1 Rappels théoriques . . . . . 84
2.2.2 Portage de la théorie de l'activité . . . . . 85
2.2.3 Application de la modélisation . . . . . 89
**2.3 Conclusion . . . . . 92**


Ce chapitre ce concentre sur la modélisation de l'identification en classe de l'activité humaine dans les réseaux. Une première partie abordera la modélisation des méthodes utilisées dans les réseaux informatiques pour classifier le trafic, ainsi que des propositions d'amélioration dans le cadre du trafic industriel.

En effet, ce dernier correspond aux échanges de données au sein des systèmes de contrôle industriels (ICS), intégrant les automates programmables industriels (PLC), les systèmes SCADA, ainsi que divers capteurs et actionneurs [81]. Malgré la convergence croissante entre IT et OT, la sécurité des réseaux industriels demeure nettement moins étudiée que celle des réseaux informatiques. À titre d'exemple, sur la période 2020-2024, une recherche sur Google Scholar retourne environ 9 000 publications pour *OT security*, contre près de 200 000 pour *IT security*. Cette disparité s'explique par plusieurs facteurs. Historiquement, les réseaux industriels ont été conçus pour fonctionner de manière isolée, limitant ainsi les préoccupations sécuritaires. De plus, l'utilisation de protocoles propriétaires et la longévité du matériel rendent les mises à jour et l'adoption de nouvelles solutions plus complexes. Enfin, un manque de compétences spécialisées freine l'évolution du domaine, en partie en raison de la différence marquée entre les métiers d'informaticien et d'automaticien, et parfois d'une collaboration insuffisante entre ces deux domaines [82, 83].

Ce chapitre contribue à répondre à la question de recherche **QR1 :** Comment modéliser et représenter l'activité humaine dans un réseau informatique en tenant compte de ses spécificités ?, en proposant un cadre structuré basé sur l'analyse des interactions observées à travers le trafic.

Après un rappel sur la modélisation de la théorie de l'activité, une seconde partie proposera son adaptation au domaine informatique. La théorie de l'activité est un cadre conceptuel permettant d'expliquer, de comprendre et de modéliser les actions humaines dans un contexte donné. Dans cette étude, elle est appliquée à l'informatique pour analyser et représenter l'activité d'un utilisateur sur une machine, en tenant compte de ses interactions avec le système.

Enfin, une dernière partie abordera le concept de granularité des activités, qui introduit une échelle de précision permettant d'affiner la classification des activités. Cette granularité définit jusqu'à quel niveau de détail l'activité peut être observée et analysée, offrant ainsi une flexibilité dans la modélisation : elle permet d'adapter le niveau d'abstraction selon les besoins.

## 2.1 Modélisation de la classification réseau

La classification du trafic réseau repose sur différentes approches, adaptées aux objectifs spécifiques de l'analyse. Selon le contexte, elle peut viser à identifier des activités utilisateur (navigation web, streaming, messagerie, etc.), à détecter des anomalies ou des attaques, ou encore à assurer la supervision et l'optimisation du réseau. Chaque objectif influence les méthodes employées : la classification peut se baser sur les ports, les signatures de protocole, l'inspection approfondie des paquets (DPI), ou encore l'analyse statistique et l'apprentissage automatique[84]. Ainsi, la pertinence d'une approche dépend à la fois du niveau de précision recherché et des contraintes opérationnelles du réseau étudié. Malheureusement, avec la complexification croissante des réseaux due à la convergence IT/OT, à l'essor du cloud, à la virtualisation des infrastructures et à l'augmentation du chiffrement, les approches les plus simples de classification montrent leurs limites. En effet, se référer à un simple numéro de port ne suffit plus à identifier avec précision les applications et les activités réseau, car de nombreux services utilisent des ports dynamiques. Chaque environnement réseau possède ses propres particularités, avec des schémas de communication qui varient en fonction des usages, des équipements et des contraintes métier. Il est donc essentiel d'adopter une méthode flexible et adaptable, capable d'apprendre et de s'ajuster aux spécificités de chaque réseau.

Dans le cadre de la classification des activités réseau, une approche hybride combinant plusieurs méthodes est privilégiée en fonction des besoins spécifiques, qu'il s'agisse de supervision ou d'une analyse plus approfondie des activités. Certaines approches, comme l'inspection approfondie des paquets (DPI), bien qu'efficaces pour identifier les applications, sont trop gourmandes en ressources et difficilement applicables en environnement temps réel. De plus, elles posent des problèmes de confidentialité et deviennent inexploitables lorsque le trafic est chiffré.

Pour concilier performance et précision, une première classification basée sur les en-têtes des flows est mise en place. Cette approche permet un tri rapide et peu coûteux en ressources en exploitant des attributs comme la durée des communications, le nombre de paquets échangés, les intervalles entre les paquets ou encore les protocoles utilisés. Ces caractéristiques sont indépendantes du contenu du trafic et donc robustes face au chiffrement.

Cependant, pour affiner l'analyse dans des cas plus complexes, des algorithmes de machine learning sont intégrés afin d'extraire des patterns plus subtils et d'améliorer la détection des différentes activités. L'avantage de cette approche progressive est qu'elle s'adapte dynamiquement aux besoins : une classification rapide pour la supervision globale du trafic et une analyse plus poussée lorsque la détection d'activités spécifiques ou d'anomalies est nécessaire. Ainsi, la granularité et la précision de la classification peuvent être ajustées en fonction des objectifs opérationnels et des contraintes techniques du réseau.

Dans cette section, nous présentons tout d'abord le fonctionnement classique des analyseurs réseau, en expliquant comment les informations issues des en-têtes permettent une classifica-

tion basique. Ensuite, nous introduirons notre propre approche de modélisation, qui adapte la technique de classification en fonction des spécificités du trafic analysé. Notre modèle propose un traitement plus fin et contextuel des flux, intégrant des paramètres supplémentaires pour améliorer la précision et l'efficacité de la classification, notamment dans les réseaux industriels.

### 2.1.1 Modèles théoriques de communications réseau

Dans les réseaux, plusieurs modèles théoriques permettent de structurer les communications. L'un des plus courants est le modèle OSI, qui définit des couches spécifiques pour l'échange d'informations. Un concept fondamental associé à ces modèles est l'encapsulation des données. Ce mécanisme permet d'organiser les informations en trames, en ajoutant progressivement des en-têtes et des métadonnées nécessaires au bon acheminement des paquets. L'encapsulation garantit ainsi une transmission efficace et fiable des données, en respectant les règles définies par chaque couche du modèle de communication.

Le modèle OSI est le modèle de référence théorique utilisé comme standard pour décrire le fonctionnement des communications réseaux[85, 86]. Il est divisé en 7 couches distinctes, chacune ayant une fonction spécifique, allant de la transmission physique des données (couche physique) jusqu'aux applications utilisateur (couche application). Il s'agit d'un modèle théorique conçu pour standardiser et faciliter l'interopérabilité des réseaux, offrant une vue claire et détaillée des interactions entre les différentes couches. C'est en 1978 que ce modèle est présenté pour la première fois, mais il sera supplanté par le modèle TCP/IP, publié en 1974[87] et adopté en 1983 par ARPANET, pour sa meilleure résilience[88, 89].

Le modèle TCP/IP est, quant à lui, l'implémentation pratique du réseau, directement issue du développement d'Internet. Il était composé initialement de 4 couches comme décrites dans la RFC 1122 [90] : Application, Transport, Internet et Accès réseau. Bien que parfois toujours présenté ainsi, il tend à se standardiser sur 5 couches, reflétant mieux la réalité des réseaux actuels (voir figure 2.1). En effet, ce modèle est utilisé comme référence dans des certifications bien établies dans le domaine des réseaux, telles que le CCNA.[91].

Étant donné que le modèle TCP/IP est l'implémentation pratique du réseau, c'est ce dernier qui servira de référence dans la suite du manuscrit, ainsi que les couleurs données pour chacune de ses couches. Des alternatives comme le modèle à trois couches de John Day (Recursive Internetwork Architecture - RINA[92]) ou encore les architectures orientées vers l'information-centric networking (ICN[93]) sont également envisagées pour répondre aux défis des réseaux modernes et palier aux lacunes de TCP/IP. Par ailleurs, au sein de chaque couche de ces modèles, d'autres modèles de communication plus spécialisés existent. Par exemple, le paradigme client-serveur structure les échanges entre un client demandant un service et un serveur qui le fournit, tandis que le modèle peer-to-peer (P2P) repose sur une architecture décentralisée où chaque nœud peut agir comme client et serveur. De même, des protocoles spécifiques comme HTTP,

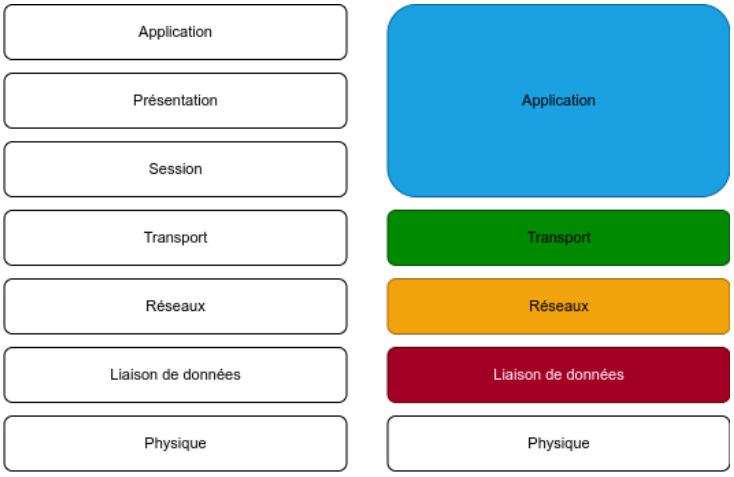



FIGURE 2.1 – Modèle OSI (à gauche) et modèle TCP/IP moderne (à droite)

FTP ou MQTT définissent des schémas de communication particuliers, mais ils restent internes à certaines couches (ex. application ou transport) et ne remplacent pas les modèles généraux comme OSI ou TCP/IP. Ainsi, si ces derniers servent de cadre conceptuel pour comprendre la communication réseau dans son ensemble, d'autres modèles viennent préciser et optimiser les interactions à différents niveaux du réseau.

L'encapsulation est un processus fondamental dans les réseaux de communication, consistant à ajouter des informations de contrôle à une donnée pour permettre son transfert efficace et fiable à travers un réseau. À chaque couche du modèle de communication, des en-têtes (headers) et parfois des en queues (trailers) sont ajoutés aux données transmises. Ces informations supplémentaires fournissent des métadonnées nécessaires, telles que les adresses source et destination, les numéros de port, ou encore les mécanismes de contrôle d'erreurs. Par exemple, dans une communication utilisant Modbus sur TCP/IP, un protocole industriel ouvert de la couche applicative permettant aux composants d'échanger des données, les données d'application (couche 5) sont encapsulées dans des segments au niveau transport (couche 4), encapsulées dans des paquets au niveau réseau (couche 3), puis dans des trames au niveau liaison de données (couche 2), avant d'être transformées en signaux au niveau physique.

Une trame est donc une unité de données utilisée dans les communications réseau. La figure 2.2 montre les détails d'une trame Ethernet II encapsulant du Modbus.

De plus, chaque protocole est pensé pour être compatible, en théorie, avec les autres. Si l'on prend Ethernet II et IEEE 802.3, le commutateur va tout d'abord vérifier la taille des trames pour pouvoir correctement les traiter. La figure 2.3 détaille les différents champs des deux protocoles. Il est à noter qu'Ethernet II est aujourd'hui le format de trame majoritairement utilisé. Par ailleurs, le champ LLC est une sous-couche du protocole IEEE 802.3, ajouté en 1998 et commun à tous les autres protocoles MAC 802.x, permettant leur fiabilisation.

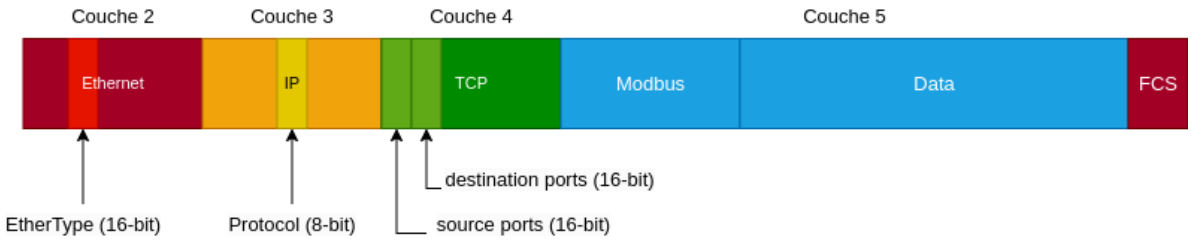



FIGURE 2.2 – Détails d'une trame Ethernet II encapsulant du Modbus




FIGURE 2.3 – Champs de l'en-tête Ethernet II et IEEE 802.3

La figure 2.4 permet de suivre le cheminement et les différents traitements par lesquels passe une trame. Le premier champ qui diffère entre les deux protocoles est celui de l'Ethertype pour l'Ethernet II et de la longueur pour l'IEEE 802.3. Ceux-ci ayant la même taille (2 octets), leur valeur permet de les différencier. En effet, si celle-ci est supérieure à 1500, le commutateur l'interprétera donc comme un Ethertype et non une taille, la longueur des données de la trame étant limitée à 1500 octets (0x0800 en hexadécimale). Afin de permettre une parfaite compatibilité, l'Ethertype possède une valeur supérieure à 1500, permettant d'identifier directement le protocole encapsulé.

De l'autre côté, si le protocole IEEE 802.3 est identifié, le code du protocole encapsulé se trouve dans l'en-tête LLC situé à la suite du champ longueur. Celui-ci possède les champs *dsap* et *ssap*, permettant l'identification des protocoles d'accès aux sous-réseaux source et destination. Il existe un cas spécifique, lorsque les valeurs sont 0xAA ou 0xAB, qui indique la présence d'un en-tête SNAP à la suite, une extension de l'en-tête LLC. Ce dernier permet d'apporter des précisions sur le protocole encapsuler comme l'organisation ou le fabricant en plus du protocole encapsulé lui-même.

### 2.1.2 Modèle d'analyseur réseau

Un analyseur réseau est un outil logiciel ou matériel permettant de capturer, inspecter et analyser le trafic circulant sur un réseau informatique. Il est utilisé pour diagnostiquer des problèmes de communication, surveiller la performance du réseau, détecter des anomalies ou encore assurer la sécurité en identifiant d'éventuelles menaces.

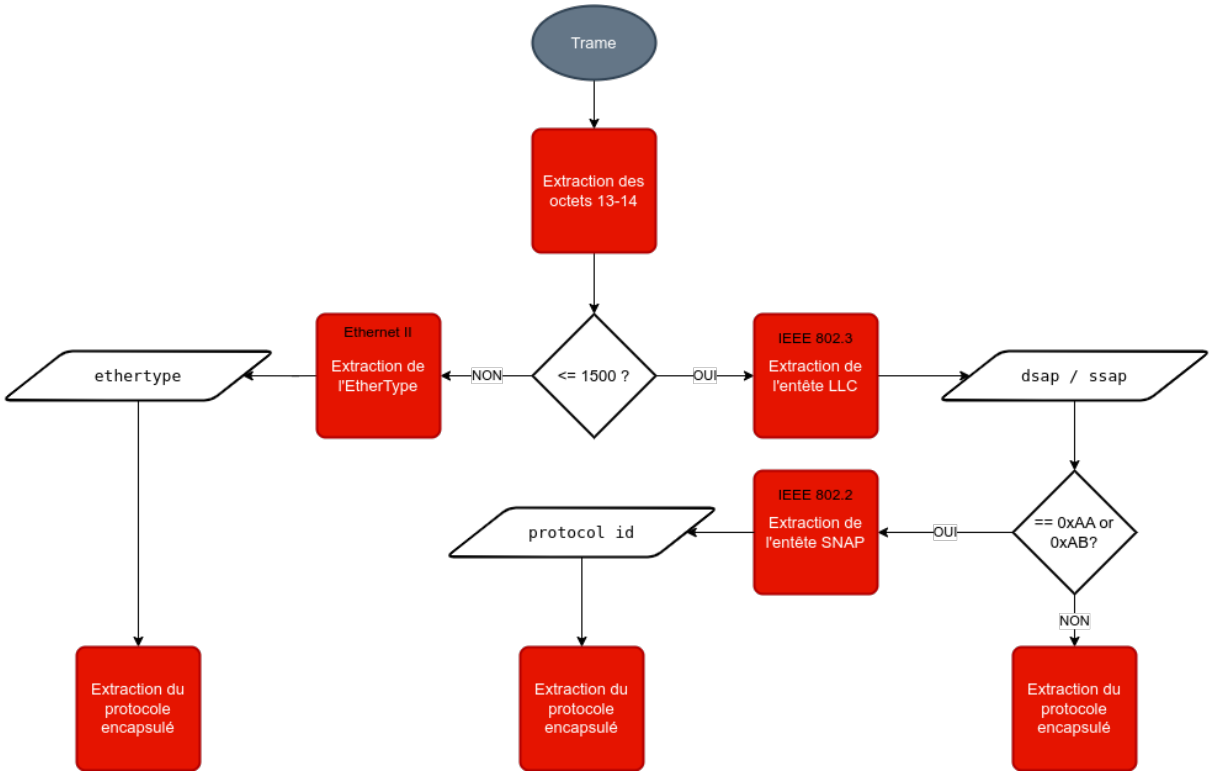



FIGURE 2.4 – Modélisation du traitement des trames

Dans un premier temps, le fonctionnement d'un analyseur réseau classique sera décrit. Ce type d'outil capture et inspecte les paquets circulant sur un réseau afin d'extraire des informations détaillées sur les communications. Bien que performants, ces analyseurs peuvent être coûteux en ressources et peu adaptés à une analyse à grande échelle en temps réel.

Ensuite, l'attention sera portée sur le fonctionnement d'un analyseur de flow, une approche optimisée pour la classification d'activité. Contrairement aux analyseurs traditionnels, un analyseur de flow ne traite pas chaque paquet individuellement, mais agrège les informations essentielles sur les communications en cours. Cette approche permet une analyse plus efficace, particulièrement adaptée aux besoins d'identification des types d'activités réalisées sur le réseau sans surcharger l'infrastructure de collecte et de traitement des données.

Un analyseur classique suit plusieurs étapes méthodiques pour classifier un protocole et identifier les communications qui circulent sur un réseau. La méthode utilisée, la désencapsulation, est similaire à celle qu'utilisent les équipements réseaux comme les commutateurs et les routeurs, à l'exception que ces derniers sont respectivement limités aux traitements de la couche 2 et 3. Tout d'abord, il capture les paquets au niveau de la couche physique ou liaison de données. Ensuite, il effectue une analyse séquentielle des en-têtes en appliquant les règles d'encapsulation. Par exemple, il examine d'abord l'en-tête de la couche liaison, ici Ethernet II, pour déterminer

le type de protocole de la couche supérieure (IPv4, IPv6, ARP, etc.) grâce au champ Ethertype. Une fois le protocole identifié, l'analyseur passe à la couche suivante, comme l'en-tête IP, pour extraire des informations clés, telles que les adresses source et destination, ou encore les numéros de protocole pour la couche transport (TCP, UDP). Ensuite, il approfondit l'examen, au niveau de la couche transport, en identifiant des ports spécifiques, qui révèlent souvent le type d'application (HTTP, FTP, DNS, etc.).

Le diagramme de la figure 2.5 montre les étapes utilisées par un analyseur réseau pour extraire les informations de la trame de la figure 2.2. La suite des étapes concernant le traitement des données utiles n'est pas représentée sur le modèle, étant donné que nous n'y aurons pas accès au niveau des flows.

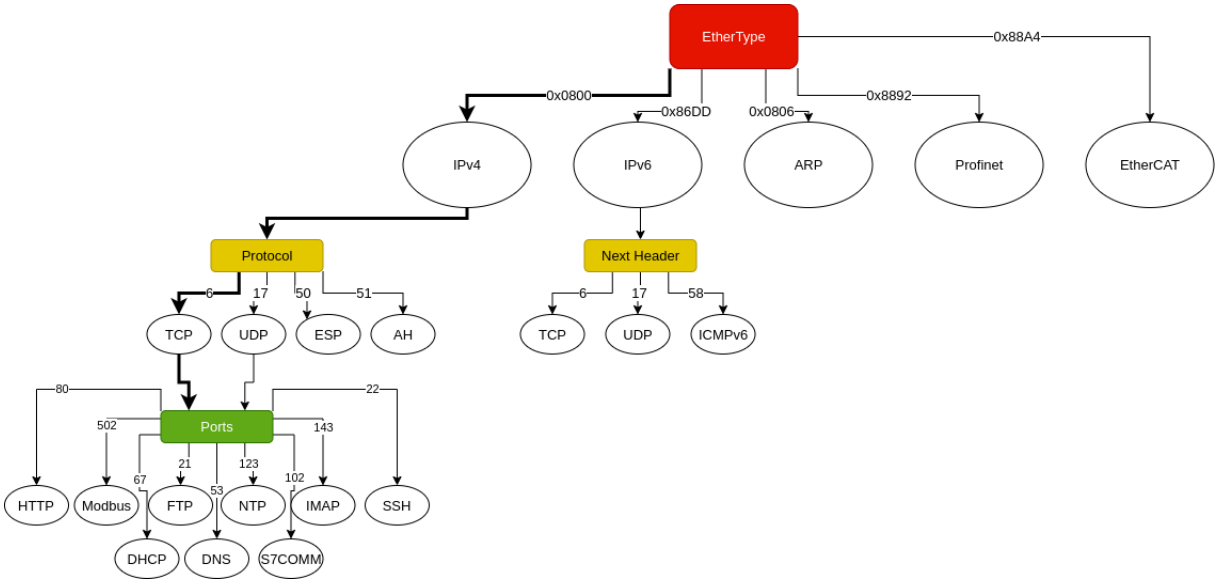



FIGURE 2.5 – Modélisation d'un analyseur réseau traitant une trame Ethernet

Les analyseurs intègrent souvent des fonctionnalités plus poussées permettant parfois d'identifier des anomalies, des problèmes de performance, etc. Mais ils n'agissent en aucun cas sur le trafic.

Dans le cadre de cette thèse, un modèle d'analyseur intégré spécifique aux flows a été développé. Ce modèle repose sur l'hypothèse que les flows ont déjà été extraits du trafic réseau circulant à travers le commutateur. Contrairement à un analyseur classique, ce modèle est conçu pour être directement intégré au commutateur et adapté aux contraintes spécifiques des environnements industriels. Son objectif principal est de caractériser les comportements des utilisateurs en analysant les flows de manière structurée et ciblée. C'est cette finalité qui justifie son approche différente des analyseurs traditionnels, en mettant l'accent sur l'identification et la classification des activités plutôt que sur une analyse exhaustive du trafic. Le diagramme de flux proposé se

trouve sur la figure 2.6.

L'objectif étant de caractériser le plus rapidement possible le flow, le modèle traite les couches dans l'ordre inverse de l'analyseur classique en commençant par la quatrième, la couche transport, puisque la désencapsulation a déjà été effectuée. Plus on se rapproche de cette couche, plus le protocole extrait peut nous donner des informations sur l'application, voir l'activité qui est effectuée. En fonction des informations récupérées et au fur et à mesure de l'avancée dans le diagramme, les différents flows sont orientés vers un traitement spécifique, dépassant le rôle classique de l'équipement réseau, voire de l'analyseur. Les ports et/ou protocoles connus permettent de faciliter la classification de certains flows.

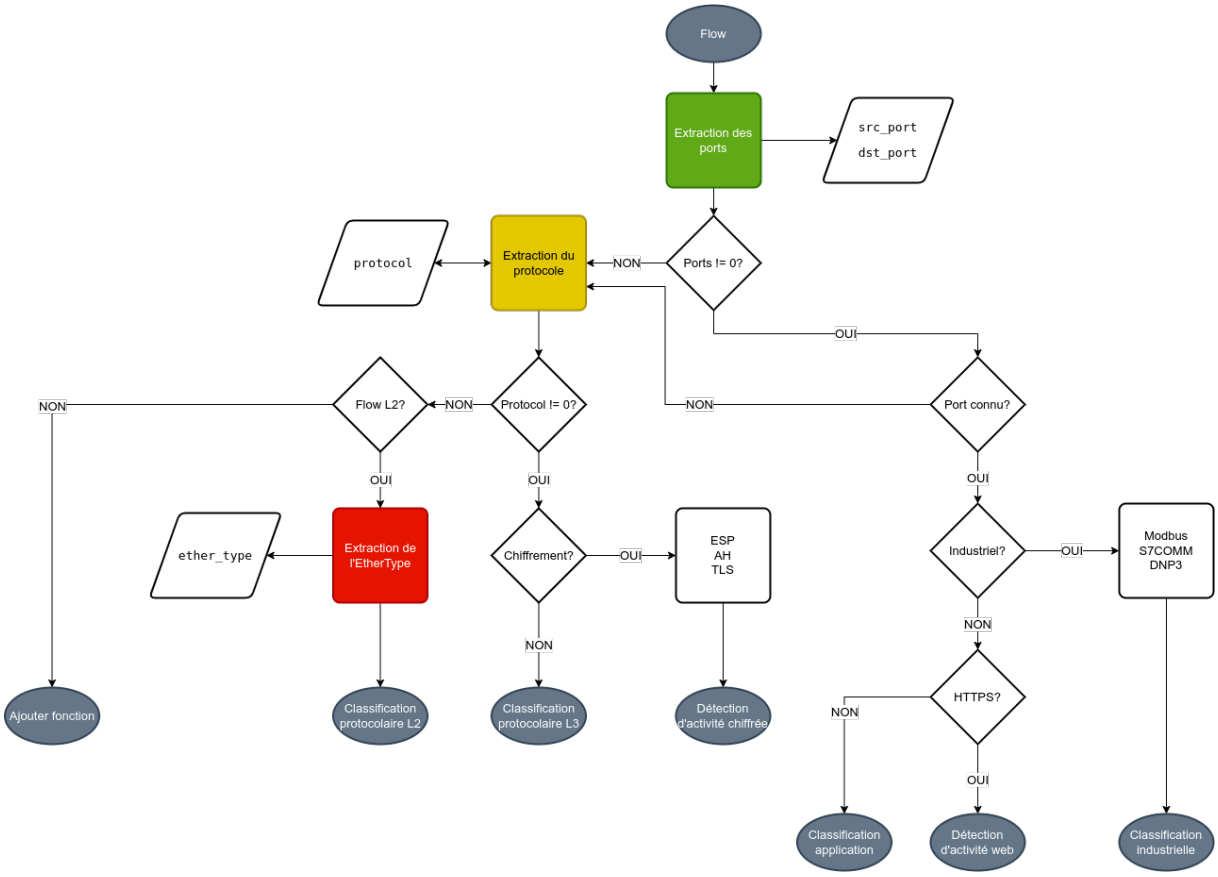



FIGURE 2.6 – Modélisation d'un analyseur réseau intégré pour flows

Ce modèle met en avant plusieurs catégories types, souvent abordées dans la littérature. En effet, il est largement reconnu que certaines situations nécessitent un traitement spécifique, soit en raison de leurs différences marquées, soit en raison de leur forte similarité. Parmi ces catégories récurrentes figurent le trafic chiffré ou non chiffré [94, 95], le trafic web ou non web [96, 97], ainsi que la distinction entre les environnements IT et OT [98, 99]. Les caractéristiques spécifiques de ces catégories sont explicitées ci-dessous, afin de mettre en lumière leurs implica-

tions pour l'analyse et la classification.

— IT ou OT : Pour des raisons de temps réel, l'informatique industrielle (OT) est rarement chiffrée et relève couramment d'un port connu. Ses échanges sont donc facilement discriminables et leurs traitements peuvent donc être optimisés en fonction des spécificités de l'environnement. Par contre, par leur nature fortement orientée propriétaire, il faut que l'analyseur soit en mesure de manipuler leurs protocoles. Cela demande le développement d'un dissecteur qui permet d'extraire tous les champs précisément grâce à une cartographie complète du protocole.

— Chiffré ou non-chiffré : Avec l'avènement des VPN et la sécurité croissante des échanges, le trafic se retrouve de plus en plus chiffré à bas niveau (couche 2, 3 ou 4 du modèle TCP/IP). Cela rend plus difficile la classification des échanges et demande une analyse plus poussée et spécifique en fonction du protocole et donc de la couche où il a lieu.

— WEB ou non-WEB : La distinction entre ces 2 types de trafic est importante. En effet, une même activité ou application peut passer par ces différents moyens, mais ne suivra pas les mêmes schémas de communication. Il est donc nécessaire d'avoir 2 modèles différents pour traiter les différents cas possibles. Un email peut très bien être envoyé par une application native comme Thunderbird ou passer par une application Webmail, apparaissant respectivement dans l'analyse comme utilisant SMNP et HTTPS.

Ces comparaisons entre OT et IT, trafic chiffré et non chiffré, ainsi que trafic web et non-web, illustrent des différences et similitudes qui se manifestent à différents niveaux du modèle de communication réseau. Ces distinctions reposent sur des protocoles situés à des couches variées, influençant ainsi la manière dont ces caractéristiques doivent être analysées. Certains de ces critères sont visibles dès les couches basses du modèle réseau (Chiffrement MACSec), tandis que d'autres nécessitent une analyse à des niveaux plus élevés (HTTPS).

Ainsi, ces différences de granularité impliquent qu'il n'est pas pertinent de traiter toutes ces dimensions de la même manière ou au même niveau d'analyse. Pour mieux structurer l'approche de classification, il est donc nécessaire de modéliser la granularité de l'activité en prenant en compte la hiérarchie des protocoles et la pertinence des informations disponibles à chaque couche.

### 2.1.3 Modélisation de la granularité de l'activité

La notion de granularité de l'activité désigne le niveau de détail avec lequel une activité humaine est observée et analysée dans un environnement donné. Dans le contexte des réseaux

informatiques, cette granularité permet de classifier et d'interpréter les actions des utilisateurs en fonction de différents niveaux. Les niveaux commencent des distinctions générales, comme le trafic normal ou anormal, à des descriptions précises des comportements individuels, tels que l'envoi d'un email ou la consultation d'une application métier.

Cette approche progressive offre une meilleure compréhension des interactions dans un réseau, en reliant les données brutes aux contextes organisationnels et techniques.

### Définition des différentes granularités

La notion de granularité est abordée par Wang etal. [100], qui en identifient huit différentes au total, dont six sont retenues pour cette classification. Les deux derniers niveaux, bien que pertinents dans des contextes spécifiques, ne sont pas adaptés à notre approche. Premièrement, le cas des appareils intelligents, tel qu'examiné dans l'article de Wang dans un contexte "domicile" où ces dispositifs interagissent souvent entre eux, n'est pas traité dans notre modèle. En effet, notre application cible principalement des environnements industriels, où ces dispositifs sont rarement présents. Deuxièmement, l'identité des applications, bien que précise, dépasse le cadre de notre projet. L'accès à ces informations, comme un numéro de téléphone ou un nom d'utilisateur, pourrait contrevenir à des exigences légales, notamment en matière de respect du RGPD, qui interdit l'exploitation de données personnelles sensibles.

Par ailleurs, comme expliqué dans la section 1.3.5 du chapitre précédent, le traitement détaillé des données issues des paquets nécessite une puissance de calcul significative, particulièrement lorsqu'il s'agit d'exploiter des techniques avancées comme le deep learning. Étant donné les contraintes de ressources associées à notre projet, ce type de traitement dépasse le cadre de notre étude. Notre objectif demeure de concevoir un modèle généraliste, conforme aux réglementations en vigueur et adapté aux environnements industriels. La figure 2.7 présente l'échelle des granularités de chaque activité, associée à un exemple concret.

— **Binaire** : il n'y a que deux classes, comme séparer le trafic normal/anormal, chiffré/non-chiffré ou encore VPN/non-VPN.

— **Protocoles** : la classification se base sur les protocoles, leur numéro de port, souvent utilisé à des fins de supervision réseau.

— **Groupes de service** : les différents services sont regroupés en fonction de leurs usages comme les services de streaming, de téléchargement ou la navigation internet. On arrive sur un premier niveau de classification d'activités exploitable.

— **Applications** : la classification gagne en précision en permettant de catégoriser le trafic

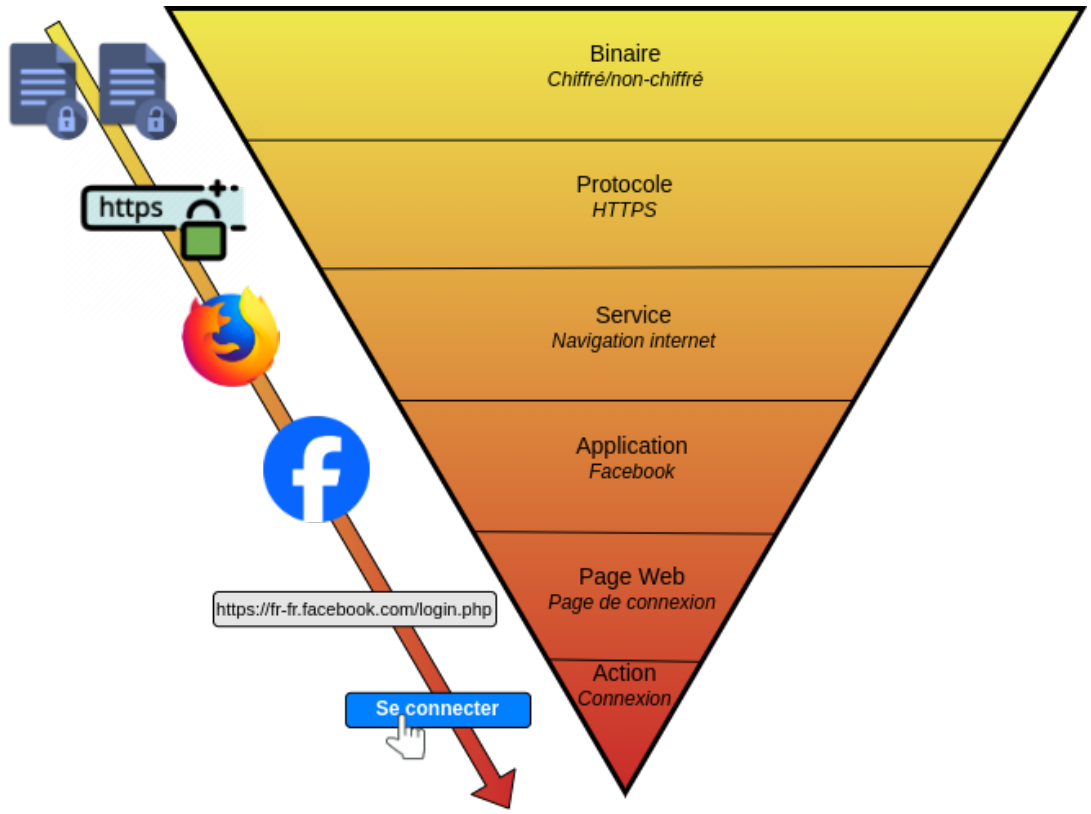



FIGURE 2.7 – Pyramide des granularités de classification d'activités

en fonction de l'application utilisée.

— **Site WEB** : à cette échelle, on essaye de distinguer les pages WEB visitées.

— **Comportements utilisateurs** : on arrive a un niveau de précision élevé où chaque action de l'utilisateur peut être classifiée comme par exemple, poster une image sur Twitter, se connecter à sa boîte mail ou rédiger un document sur Office 360.

### Classification des jeux de données par granularité de l'activité

Pour illustrer la sous-section dernière, la table 2.1 classifie le ou les niveaux de granularité des activité étiquetées dans quelques jeux de données populaires. Les jeux de données retenus et leur présentation sont les suivants :

— **MAWI (Measurement and Analysis on the WIDE Internet)"** : Le dataset MAWI capture du trafic réseau réel provenant de l'Internet public grâce au WIDE Project au Japon. Il est utilisé pour analyser le trafic général, détecter des anomalies, et évaluer les

performances des systèmes réseau. Les données, anonymisées pour des raisons de confidentialité, incluent à la fois du trafic normal et anormal, ce qui en fait une ressource précieuse pour la recherche.

— **ISCX (Canadian Institute for Cybersecurity Dataset)** : Le dataset ISCX fournit une simulation réaliste de trafic réseau, intégrant à la fois des scénarios de trafic normal et d'attaques. Il est couramment utilisé pour évaluer les systèmes IDS/IPS et les approches de détection d'anomalies. Ce dataset, richement étiqueté, contient diverses attaques telles que les attaques par déni de service (DoS) et les Botnets.

— **USTC-TFC2016** : USTC-TFC2016 est un dataset chinois conçu pour l'analyse de trafic, avec des échantillons de trafic normal et malveillant. Il cible principalement la classification de trafic et la détection d'anomalies, incluant des types variés de trafic malveillant tels que les Botnets, Ransomwares, et autres.

— **CTU-2013** : CTU-2013 se concentre sur le trafic Botnet, capturé dans un environnement semi-contrôlé. Ce dataset est conçu pour détecter les communications malveillantes et les Botnets. Il est bien étiqueté et utilisé pour entraîner des modèles de détection spécialisés.

— **UJN** : Les UJN Traces sont issues de captures de trafic réseau effectuées dans une université chinoise. Ce dataset se concentre sur l'analyse des comportements réseau spécifiques au milieu universitaire, mettant en lumière des interactions propres à cet environnement.

— **UNIBS** : Le dataset UNIBS provient de captures réalisées à l'Université de Brescia en Italie. Il est utilisé pour l'analyse du trafic applicatif et comportemental, avec des données issues d'utilisateurs réels exploitant diverses applications réseau.

— **Moore** : Créé spécifiquement pour la classification de trafic par application, le Moore Dataset est une ressource clé pour les méthodes supervisées de classification. Il offre des données étiquetées selon les applications (par exemple, HTTP, FTP), facilitant ainsi les études ciblées sur le trafic applicatif.

— **Auckland** : Capturé à l'Université d'Auckland, ce dataset propose des traces historiques de trafic réseau. Il est utilisé pour l'analyse de trafic général et les études sur la qualité de service dans un environnement académique.

— **Mirage** : Mirage est un dataset simulant des environnements réseau critiques tels que les réseaux industriels et IoT. Il cible la détection d'anomalies et d'attaques, en se basant sur des scénarios réalistes typiques de ces environnements sensibles.

— **MobileGT** : MobileGT est conçu pour capturer et analyser le trafic réseau généré par des appareils mobiles dans un environnement contrôlé. Il inclut des interactions utilisateurs avec des applications mobiles et met l'accent sur la distinction entre trafic chiffré et non chiffré.

— **UNSW** : UNSW-NB15 combine des captures de trafic réseau normal et des scénarios simulés d'attaques, offrant un étiquetage détaillé pour neuf types d'attaques, tels que les Fuzzers, Backdoors, et Worms. Il est particulièrement adapté à l'entraînement et à l'évaluation de systèmes de détection d'intrusions et à la classification des activités réseau. Ce dataset se distingue par sa diversité, permettant une analyse approfondie des interactions entre trafic légitime et malveillant.

Il est possible que certains jeux de données aient plusieurs granularités, en effet, certains possèdent plusieurs étiquettes, permettant d'effectuer différents types de classifications. Il ne suffit pas de considérer uniquement la granularité des activités dans les jeux de données. Hindy etal. [23] répertorie des jeux de données populaires dans différentes tables en fonction de spécificités, réseau (IT, mobile, SCADA, IoT, etc.). Il apparaît donc important de préciser quels types de machines sont représentés dans chaque jeu de données, car le support peut avoir un potentiel impact sur la nature et les caractéristiques du trafic réseau capturé. Par exemple, les appareils IT, OT, mobiles ou IoT génèrent des trafics très différents en raison de leurs usages spécifiques, des protocoles qu'ils utilisent et des environnements dans lesquels ils évoluent. Ces distinctions, incluses dans le tableau, permettent de mieux comprendre les limites et les applications potentielles de chaque jeu de données en fonction du type de machine étudié. Les catégories principales incluent :

— **IT** : Ces machines englobent les ordinateurs, serveurs et autres équipements traditionnels utilisés pour les activités bureautiques et les services réseau standards. Elles génèrent principalement du trafic web, des transferts de fichiers et des communications internes.

— **OT** : Les dispositifs OT sont spécifiquement conçus pour les environnements industriels, comme les automates programmables, les systèmes SCADA et les capteurs. Leur trafic est souvent prévisible et critique, orienté vers des protocoles spécifiques au contrôle industriel.

— **Mobile** : Cette catégorie inclut les smartphones, tablettes et autres appareils mobiles connectés au réseau. Ces machines produisent un trafic réseau varié, allant des communications web classiques aux interactions spécifiques d'applications mobiles.

— **IoT** : Les appareils IoT, comme les caméras de sécurité, thermostats intelligents ou autres objets connectés, génèrent un trafic souvent caractérisé par des communications machine-to-machine. Ces dispositifs utilisent fréquemment des protocoles propriétaires ou légers adaptés à leurs besoins.

| | | Summary of public datasets | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Granularity | | | | | | Network type | | | |
| Year | Dataset | Binary | Protocols | Service groups | Applications | Websites | User behaviors | IT | Mobile | OT | IoT |
| 2000-2020 | MAWI | - | ✓ | - | - | - | - | ✓ | - | - | - |
| 2016 | ISCX | - | ✓ | ✓ | - | - | - | ✓ | - | - | - |
| 2016 | USTC-TFC2016 | - | - | ✓ | - | - | - | ✓ | - | - | - |
| 2014 | CTU-2013 | - | - | - | - | - | - | ✓ | - | - | - |
| 2013 | UJN traces | - | - | ✓ | - | - | - | ✓ | - | - | - |
| 2009 | UNIBS traces | - | ✓ | ✓ | - | - | - | ✓ | - | - | - |
| 2005 | Moore | - | ✓ | ✓ | - | - | - | ✓ | - | - | - |
| 2000 | Auckland II | - | ✓ | - | - | - | - | ✓ | - | - | - |
| 2017-2019 | Mirage | - | - | - | ✓ | - | - | - | ✓ | - | - |
| 2016-2018 | MobileGT | - | - | - | ✓ | - | - | - | ✓ | - | - |
| 2016-2017 | UNSW | - | - | - | ✓ | - | - | - | - | - | ✓ |

TABLE 2.1: Jeux de données triés par granularité

On remarque que la majorité des jeux de données actuels ont été enregistrés sur des réseaux *classiques*. Cela s'explique par le fait que les réseaux mobiles soient plus récents, ce qui est d'autant plus vrai pour les réseaux IoT. Le faible nombre de jeux de données industriels s'explique par deux principales raisons. La première, pour des raisons de confidentialités, les industriels ne veulent pas partager leurs données, car elles peuvent s'avérer possiblement sensibles ou donner des informations sur leur architecture. La deuxième, pour des raisons d'expositions. En effet, pendant longtemps, les systèmes industriels n'étaient pas autant exposés qu'aujourd'hui puisque protégés de l'extérieur, et par conséquent, ne nécessitant pas la détection d'attaques.

L'analyse des échanges réseau à travers différentes métriques permet d'extraire des caractéristiques précises sur le trafic, telles que son volume, sa fréquence ou encore les protocoles utilisés. Toutefois, ces informations seules ne suffisent pas à répondre à une question essentielle : quelle est l'origine réelle de cette activité ? Derrière un flux réseau, il peut y avoir un utilisateur humain, mais également des processus automatisés tels que des scripts, des tâches planifiées ou encore des communications entre machines.

L'enjeu est donc de modéliser l'activité humaine dans un réseau informatique, afin de distinguer les interactions initiées par un utilisateur réel de celles générées par des mécanismes automatiques. Cette modélisation nécessite d'identifier des signatures comportementales propres aux actions humaines, en prenant en compte des éléments comme la temporalité des échanges, les habitudes d'utilisation, ou encore les types d'applications sollicitées. L'objectif est ainsi de mieux comprendre et classifier les comportements observés sur le réseau en fonction de leur origine et de leur nature.

## 2.2 Modélisation de l'activité humaine dans les réseaux informatiques

Cette section explore comment la théorie de l'activité, issue des sciences sociales, peut être adaptée pour analyser les interactions humaines dans les réseaux informatiques. Dans un premier temps, nous rappelons les principes fondamentaux de cette théorie, développée à l'origine pour comprendre les dynamiques sociales et éducatives. Ensuite, nous examinons comment ces concepts peuvent être portés au domaine informatique en intégrant les spécificités des environnements informatiques. Enfin, un exemple concret illustre l'application de ce cadre théorique revisité, mettant en lumière les interactions complexes entre les utilisateurs et les systèmes interconnectés.

### 2.2.1 Rappels théoriques

La théorie de l'activité, issue des travaux de psychologues soviétiques des années 1930, est une approche des sciences sociales qui examine l'influence de l'environnement social et culturel sur le comportement et le développement humain. En psychologie, elle met en lumière le rôle central des interactions sociales et des activités dans l'apprentissage et la croissance individuelle, ayant des implications majeures dans des domaines tels que l'éducation, la cognition et la motivation. Parmi ses figures majeures, Vygotski a établi une base théorique en affirmant que le psychisme humain est façonné par des influences culturelles et sociales, tandis que Leontiev a introduit une hiérarchie d'analyse comprenant les activités, les actions et les opérations. Plus tard, Engeström a enrichi cette théorie en concevant un modèle systémique, représenté sous forme de triangle, qui explore les interactions complexes entre individus, objectifs, outils, communautés, règles et division du travail. Ces contributions ont permis à la théorie de l'activité de devenir un cadre puissant pour analyser les dynamiques humaines dans des contextes variés.

### 2.2.2 Portage de la théorie de l'activité

L'application de la théorie de l'activité au domaine informatique nécessite une révision et une adaptation du modèle original d'Engeström, initialement développé pour analyser les processus humains dans des contextes sociaux et éducatifs. En effet, la spécificité des réseaux informatiques et des interactions humaines avec des systèmes numériques impose de repenser certaines composantes de l'activité, notamment en tenant compte de la médiation par des outils technologiques tels que les ordinateurs, les logiciels et les protocoles réseau. Ce modèle revisité intègre des éléments essentiels du monde numérique, notamment l'Environnement informatique, les Séquences ou programmes d'opérations, ainsi que la Division des tâches et des Droits au sein du réseau. Ce cadre permet de mieux comprendre comment les utilisateurs interagissent avec des machines et des systèmes interconnectés, en tenant compte de la manière dont ces outils influencent l'exécution des tâches et la production de traces dans le réseau. Ainsi, en enrichissant le modèle d'Engeström avec ces nouvelles composantes, il devient possible de saisir les dynamiques complexes entre les individus, les technologies et les objectifs au sein des réseaux informatiques modernes. Pour cela, il est nécessaire de décomposer et parfois redéfinir certaines composantes d'une activité utilisateur au sein d'un réseau. On reprend donc pour base, le modèle général d'Engeström, qui généralise assez bien l'activité utilisateur au sein d'un environnement et d'une communauté.

La particularité de l'activité humaine au sein d'un réseau informatique est la dépendance de celle-ci vis-à-vis des **outils** utilisés (ordinateurs, logiciels, machines, etc.). Il est donc important de développer la modélisation de cette partie qui n'est pas autant pris en compte dans le modèle général d'Engeström. En effet, une même action sur un ordinateur ne va pas forcément générer les mêmes opérations et donc produire les mêmes traces sur le réseau par rapport aux caractéristiques du système, du logiciel ou encore des protocoles utilisés.

La figure 2.8 propose un modèle de l'outil utilisé lors de l'activité.

Les nouvelles composantes de ce modèle sont les suivantes :

— L'**Environnement** qui se place d'un point de vue informatique et qui regroupe l'ensemble des matériels et logiciels qui compose le système (système d'exploitation, architecture physique, etc.).

— Les **Séquences** ou programmes qui sont des suites d'opérations utilisées par l'outil pour effectuer les actions.

— Les **Outils** ou machines sont les outils matériels ou symboliques qui médiatisent l'activité.

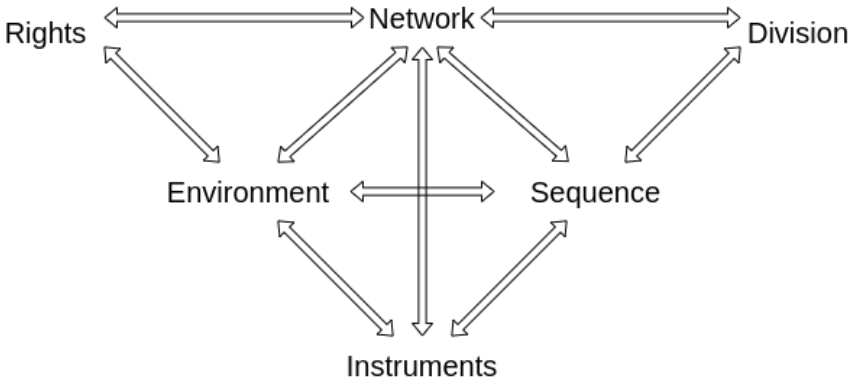



FIGURE 2.8 – Modèle de l'outil réseau

— Le **Réseau** est l'ensemble des outils qui partagent le même réseau.

— La **Division** reprend à la fois la répartition horizontale des séquences entre les outils du réseau, et la hiérarchie verticale des pouvoirs et des statuts.

— Les **Droits** sont des règles explicites qui maintiennent et régulent les actions et les interactions à l'intérieur de l'environnement et du réseau.

Ce modèle de l'outil réseau peut ainsi être ajouté à la modélisation générale de l'activité pour en créer un nouveau sur l'activité humaine dans les réseaux informatiques comme le montre la figure 2.9.

Maintenant que la structure du modèle d'activité humaine dans les réseaux a été définie, il est important de compléter ou revoir la notion même de l'activité, à savoir ses 3 niveaux d'analyse, l'activité, l'action et l'opération, comme introduite par Vigotski.

— L'**activité** est le composant central de l'analyse. Ce sont les raisons et les motifs qui ont incité à agir, à réaliser cette activité.

— Les **actions** sont les éléments unitaires composant l'activité. Elles permettent l'exécution de celle-ci et sont d'origine humaine (interaction humaine avec la machine ou lancement d'un programme).

— Les **opérations** sont les tâches ou séquences qui sont produites lors de la réalisation d'une

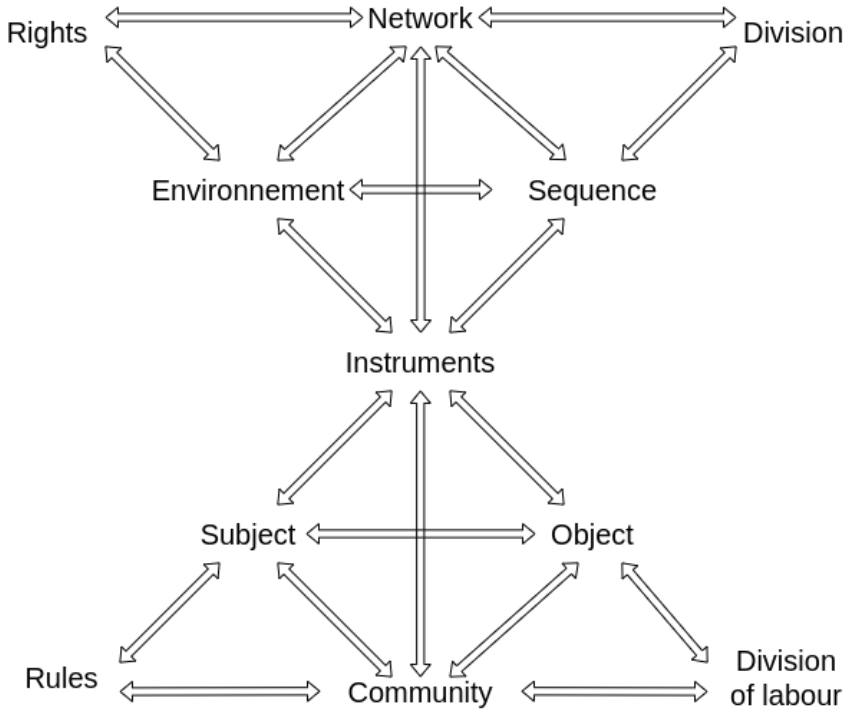



FIGURE 2.9 – Modèle de l'activité humaine dans les réseaux

action et qui permettent d'interagir avec le ou les systèmes.

L'objectif est de construire cette modélisation à partir d'une capture réseau, en analysant les interactions et le trafic produit par une ou plusieurs machines. L'idée est de modéliser l'environnement, en prenant en compte les réseaux, les machines, les droits d'accès, etc., puis de remonter jusqu'au profil des utilisateurs. À partir de ces données, il s'agira de déduire leurs activités les plus probables en fonction des patterns observés dans les traces réseau. Cette approche permettrait de mieux comprendre les comportements des utilisateurs au sein d'un réseau et de rendre possibles des inférences précises sur leurs actions et leur contexte environnemental.

Par ailleurs, la quantité et la qualité des données réseau jouent un rôle déterminant dans la précision de cette modélisation. Plus elles sont qualitatives et diversifiées, plus elles contiennent d'informations exploitables, permettant ainsi une analyse plus fine et une modélisation plus détaillée des interactions et des comportements.

Cependant, cette modélisation ne se limite pas à fournir une vue d'ensemble des activités en cours ou à déduire ce que fait un utilisateur. Elle permet aussi, dans l'autre sens, de prendre conscience de la richesse et de la nature révélatrice de nos propres activités réseau. Cela peut révéler des failles ou des manques dans les pratiques visant à préserver la confidentialité, incitant ainsi à une réflexion pour réduire les traces laissées et améliorer la protection de la vie privée.

En ce sens, cette approche ne se limite pas à une analyse descriptive, mais ouvre également des perspectives pour renforcer la sécurité et la confidentialité des utilisateurs. La table 2.2 résume les usages possibles de la modélisation.

| | **Objectif 1 : Comprendre les activités** | **Objectif 2 : Améliorer la confidentialité** |
|---|---|---|
| **But principal** | Identifier les activités probables, les postes ou les entreprises en fonction des patterns observés dans les traces réseau. | Détecter les éléments révélateurs dans les activités réseau pour améliorer la protection des données personnelles. |
| **Perspective** | Fournir une vue d'ensemble des comportements au sein du réseau. | Identifier les failles dans les pratiques de confidentialité et proposer des améliorations. |
| **Utilisation des données** | Exploiter la richesse des données pour modéliser précisément les interactions et comportements. | Mettre en lumière la sensibilité des données pour inciter à réduire les traces laissées sur le réseau. |
| **Impact attendu** | Meilleure compréhension des dynamiques réseau et des comportements utilisateurs. | Sensibilisation à la protection des données et mise en place de mesures pour limiter la traçabilité des activités. |

TABLE 2.2 – Comparaison des deux objectifs de la modélisation

Cette modélisation de l'activité humaine dans les réseaux vise à caractériser le profil d'un utilisateur à partir de ses traces réseau. Une activité $A$ est guidée par un motif et se compose d'un ensemble d'actions $a$ qui interagissent avec un ou plusieurs systèmes à l'aide d'opérations $o$.

$$A = (a_0, ..., a_n) \iff a \in Action$$

$$a = (o_0, ..., o_n) \iff o \in Operation$$

Dans son ensemble, une activité laisse nécessairement un ensemble de traces sur les systèmes avec lesquels elle interagit. La portée des actions est limitée à la machine sur laquelle elles sont effectuées, car elles sont directement instanciées par l'utilisateur. Cependant, la portée des opérations s'établit sur trois niveaux principaux : la machine locale, le réseau local et le réseau Internet.

Chaque passage d'un plan à un autre laisse des traces réseau ($N$) et, sur chacun de ces niveaux, des mémoires vives ($M_1$) et mortes ($M_2$) peuvent stocker des données liées aux opérations effectuées. Les communications sans fil ($S$) couvrent également l'ensemble des niveaux, ajoutant un type spécifique d'empreinte qui peut se manifester à travers différents plans.

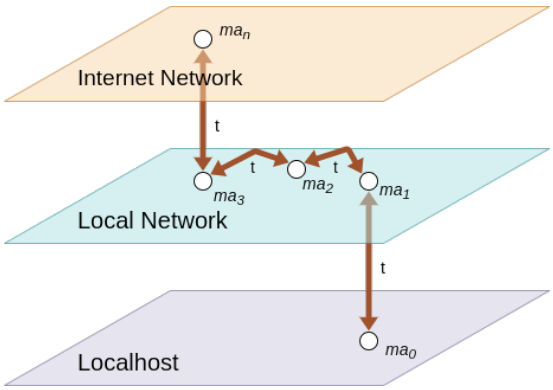



FIGURE 2.10 – Niveaux de chaque empreinte

$$P = \{L, LN, I\}$$

$$F = \{M_1, M_2, N, S\}$$

Ainsi, les empreintes laissées par une opération sont caractérisées par sa portée ($P$) et les types d'empreintes ($F$) présentes sur les plans traversés (figure 2.10).

$$o = \{P, F(P)\}$$

### 2.2.3 Application de la modélisation

L'objectif de cette modélisation est de prendre en compte tous les paramètres possibles de l'environnement, susceptible d'influer sur les traces réseau. Avoir une vue d'ensemble de tous les liens et échanges qui peuvent avoir lieu, dans un environnement, en particulier avec l'ensemble des acteurs présents. Prenons comme exemple, un employé d'une entreprise qui envoie un email professionnel via un client de messagerie (par exemple, Outlook ou Thunderbird) configuré pour utiliser les protocoles SMTP et IMAP via TLS. Nous obtenons les flows suivants résumés dans la table 2.3.

| n° | Source IP | Destination IP | Source Port | Destination Port | Protocol | Action | SMTP Server code | Sended Bytes | Received Bytes |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 192.168.1.20 | 10.0.0.10 | 55000 | 993 | IMAP (TLS) | Récupération des métadonnéees de la boîte de réception | - | 500 | 800 |
| 2 | 192.168.1.20 | 10.0.0.10 | 55000 | 993 | IMAP (TLS) | Vérification des messages non lus | - | 300 | 400 |
| 3 | 192.168.1.20 | 10.0.0.10 | 55000 | 993 | IMAP (TLS) | Synchronisation des drapeaux de lecture | - | 250 | 200 |
| 4 | 192.168.1.20 | 10.0.0.10 | 55000 | 465 | SMTP (TLS) | Authentification SMTP pour l'envoi d'email | 235 | 350 | 220 |
| 5 | 192.168.1.20 | 10.0.0.10 | 55000 | 465 | SMTP (TLS) | Déclaration de l'expéditeur | 250 | 200 | 100 |
| 6 | 192.168.1.20 | 10.0.0.10 | 55000 | 465 | SMTP (TLS) | Déclaration du destinataire | 250 | 200 | 100 |
| 7 | 192.168.1.20 | 10.0.0.10 | 55000 | 465 | SMTP (TLS) | Envoi du contenu du mail | 250 | 1200 | 200 |
| 8 | 192.168.1.20 | 10.0.0.10 | 55000 | 993 | IMAP (TLS) | Enregistrement du mail dans "Messages envoyés" | - | 600 | 300 |

TABLE 2.3: Résumé des flows lors d'un envoie de mail

À partir de ces flows, si nous faisons une synthèse des informations qui peuvent être extraites, il est possible d'obtenir la synthèse suivante des différentes composantes :

— **Environnement** : Serveur SMTP, ordinateur utilisateur et le réseau local.

— **Outils** : Ordinateur de l'utilisateur et Thunderbird.

— **Séquences** : Authentification, déclaration des adresses, envoi contenu

— **Réseau** : Connexion réseau et protocoles utilisés (TLS, SMTP)

— **Division** : Interaction client-serveur, rôles utilisateurs/admin

— **Droits** : Authentification, droits d'accès au SMTP

Si nous complétons la pyramide utilisateur, nous pouvons déduire grâce à la pyramide matériel les composantes suivantes :

— **Sujet** : Utilisateur (employé).

— **Objet** : Transmettre une information ou un document à un collègue ou client.

— **Communauté** : Destinataire de l'email (collègues, clients) et équipe IT.

— **Règle** : Politiques d'entreprise, bonnes pratiques.

— **Division du travail** : L'utilisateur rédige le mail seul.

Si nous rassemblons l'ensemble de ces informations, nous obtenons au final la figure 2.11 suivante, résumant l'ensemble des indices laissés par l'enregistrement réseau.

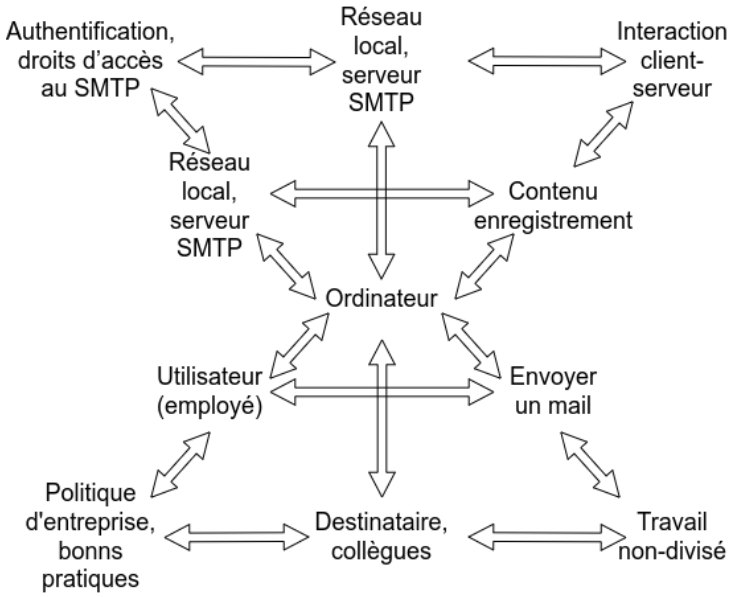



FIGURE 2.11 – Modélisation de l'activité mail

L'analyse peut être approfondie et l'accès au domaine de l'adresse ou aux résolutions DNS peut apporter des précisions sur ce cas d'étude. Une liste non exhaustive d'une analyse plus poussée :

**Profil utilisateur probable** :

— *Métier* : Par exemple, un gestionnaire de compte client (basé sur les échanges fréquents avec des destinataires externes).
— *Environnement de travail* : Utilisation d'un poste fixe ou portable avec une connexion sécurisée au réseau de l'entreprise.
— *Comportement* : Travail pendant les horaires standard avec des interactions fréquentes dans un contexte professionnel structuré.

**Contexte organisationnel probable** :

— *Entreprise* : Une organisation de taille moyenne ou grande avec une infrastructure technique bien établie.
— *Sécurité* : Une politique IT structurée (authentification, droits d'accès, chiffrage des échanges).
— *Culture* : L'ouverture à l'utilisation de logiciels libres (Thunderbird) ou une culture orientée vers la flexibilité technologique.

## 2.3 Conclusion

Après avoir exploré les bases théoriques et pratiques de la classification réseau, ce chapitre a proposé une vision globale et structurée de la modélisation de l'activité humaine dans les réseaux informatiques. La première section a permis d'examiner les principaux modèles de classification réseau, en abordant à la fois les fondements théoriques, les spécificités d'un modèle d'analyseur réseau et le concept clé de granularité des activités, qui offre une approche plus fine pour catégoriser le trafic.

Dans la seconde section, un focus particulier a été fait sur le portage de la théorie de l'activité vers le domaine des réseaux informatiques, mettant en lumière son potentiel pour mieux comprendre les comportements utilisateurs. Cette réflexion théorique a été enrichie par une application concrète de la modélisation, démontrant sa pertinence, notamment dans un cadre industriel.

En conclusion, ce chapitre établit les bases nécessaires pour une meilleure compréhension et une classification plus précise des activités humaines dans les réseaux informatiques, tout en répondant aux exigences spécifiques des environnements industriels. Ces travaux ouvrent la voie à des solutions plus adaptées pour l'analyse et la sécurisation du trafic réseau.

Chapitre 3

# Génération de données pour l'analyse d'activités réseaux

## Sommaire


**3.1 Définitions et scénarios . . . . . 95**
**3.2 Étiquetage des données . . . . . 97**
3.2.1 TOMATO : Tools for Optimized Manual Assistance on Tagging Operations 98
3.2.2 SALAD : Script-based Automatic Labelling on Activity Dataset . . . . 103
3.2.3 GAR7IC : Generalized Automatic Review and Labelling for S7COMM Industrial Communication . . . . . 106
3.2.4 Automatisation de l'étiquetage des données avec LLM . . . . . 109
**3.3 Génération de données réalistes . . . . . 114**
3.3.1 Enregistrement et rejeu des scénarios . . . . . 114
3.3.2 BASIL : Behavioral Action Scenario with Image Labelling . . . . . 122
**3.4 Conclusion . . . . . 127**


Les jeux de données disponibles sont principalement orientés vers les systèmes de détection d'intrusions (IDS) et mettent l'accent sur des scénarios d'attaques, plutôt que sur des activités réseau générales (voir Chapitre 1). La génération de données s'impose donc comme une nécessité face aux limites des jeux de données existants dans le domaine de l'analyse réseau. Ces jeux de données sont souvent anciens, partiellement réalistes, et majoritairement synthétiques, produits par des scripts ou des générateurs automatiques sans réelle implication humaine. Cette nature artificielle limite leur pertinence dans des contextes spécifiques et rend leur utilisation complexe, car les conditions exactes dans lesquelles ils ont été produits ne sont pas connues.

L'approche présentée dans ce chapitre vise à combler ces lacunes en développant des méthodes et outils de génération et d'étiquetage de données, adaptés à l'apprentissage d'algorithmes, tout en tenant compte des spécificités d'un réseau particulier. L'objectif est de proposer des solutions capables de produire des jeux de données réalistes de manière automatisée, flexibles et ajustables aux cas d'utilisation spécifiques. Ces travaux permettent ainsi de répondre à la question de recherche **QR3 :** Comment générer et étiqueter des ensembles de données représentatifs des activités réseau pour entraîner des modèles de classification ? Bien que les jeux de données

existants puissent être utiles pour des expérimentations ponctuelles, ils restent difficilement exploitables comme ressource d'apprentissage universelle. En générant des données sur mesure, nous ouvrons la voie à une meilleure adéquation entre les besoins des systèmes d'analyse et la réalité des environnements réseau étudiés.

Dans ce chapitre, plusieurs outils développés pour répondre à ces besoins seront présentés. Ils permettent de générer, enregistrer et étiqueter automatiquement des données adaptées à l'analyse des activités réseau. Ils intègrent des mécanismes de simulation et de capture, ainsi que des algorithmes pour associer des étiquettes précises aux données générées, facilitant ainsi leur exploitation dans des tâches variées, en particulier l'analyse des activités réseau. Ces développements s'inscrivent pleinement dans la question de développement **QD1 :** Quels outils peuvent assister ou automatiser l'enregistrement et l'étiquetage des activités réseau ? Chaque jeu de données est qualifié grâce à la modélisation développée (voir Chapitre 2).

## 3.1 Définitions et scénarios

Avant de présenter les différents outils développés, il est essentiel de définir certains termes et de préciser le contexte dans lequel ces outils évolueront. Il est à noter que ces définitions ne sont pas nécessairement des standards universels, mais plutôt des distinctions adoptées dans ce contexte afin de clarifier les concepts et d'assurer une compréhension cohérente pour la suite.

Lorsqu'on parle de génération de données, on fait référence à des données synthétiques, c'est-à-dire non directement issues d'un environnement opérationnel en temps réel. La distinction entre génération et production est fondamentale : une donnée générée peut provenir d'une donnée initialement produite, par exemple lorsqu'un enregistrement de données réelles est rejoué. Cependant, ce rejeu introduit inévitablement des biais ou artefacts, même infimes, notamment liés au décalage temporel, aux conditions d'exécution ou à d'autres facteurs imprévisibles. À l'inverse, une donnée produite est capturée en direct dans son contexte original, avec les machines émettrices et les conditions d'exploitation réelles.

— La **génération** concerne des données synthétiques, pouvant être inspirées de données réelles, mais reproduites artificiellement ou rejouée, donc non produite par les émetteurs originels.

— La **production** correspond aux données réellement capturées dans un environnement opérationnel par les émetteurs originaux.

— La **capture** correspond à l'enregistrement des données, qu'elles soient générées ou produites.

— L'**étiquetage** est l'attribution d'une étiquette à une donnée ou un ensemble de données afin de les classifier selon une catégorie spécifique. Il est indispensable pour l'apprentissage automatique supervisé.

En résumé, la production représente la réalité terrain, tandis que la génération vise à imiter ou à se rapprocher au maximum des conditions réelles. Par défaut, sauf indication contraire, les données réseau utilisées sont au format PCAP ou PCAPNG. Le format PCAPNG[101] offre des fonctionnalités avancées comme l'ajout de métadonnées, la prise en charge de plusieurs interfaces et l'annotation des trames, mais il n'est pas devenu la norme principalement pour des raisons de compatibilité et de simplicité. Le format PCAP, plus ancien, est largement adopté et supporté par de nombreux outils, ce qui freine la transition. De plus, le format PCAPNG est plus complexe à implémenter et peut introduire des surcoûts en performance. Enfin, de nombreux systèmes et équipements réseau utilisent encore seulement le format PCAP, garantissant une meilleure interopérabilité.

Dans notre cas, la génération de données vise à alimenter des algorithmes d'apprentissage automatique afin de détecter et classifier des activités. Pour rappel, une activité est constituée

d'actions, elles-mêmes composées d'opérations (Voir Chapitre 2 pour plus de détails). Dans notre contexte :

— Une **activité** correspond à une tâche globale réalisée sur un système numérique.

— Les **actions** regroupent toutes les interactions physiques avec l'outil numérique (clavier, souris, voix, etc.), indispensables à la réalisation de l'activité.

— Les **opérations** représentent les traces laissées par ces actions sur la machine ou le réseau.

Notre travail se focalise exclusivement sur les traces réseau, c'est donc à ce niveau que les données sont captées. À partir de ses dernières, on relève trois piliers fondamentaux comme dans la figure 3.1, dont l'association donne les objectifs suivants :

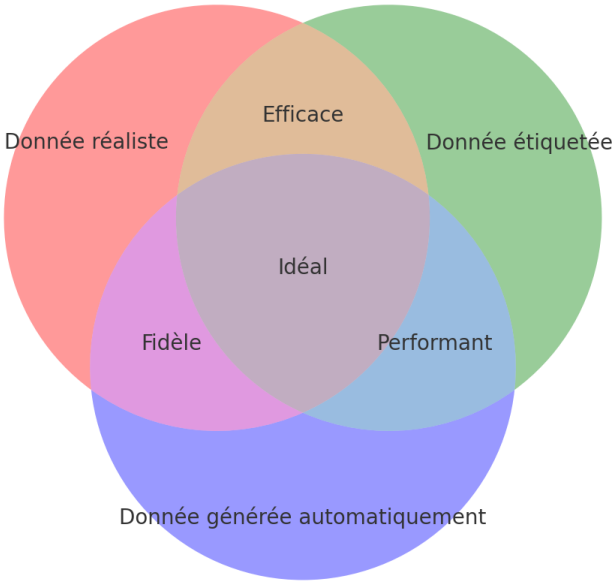



FIGURE 3.1 – Piliers de la donnée d'apprentissage

— Une **donnée générée automatiquement**, souvent à l'origine du développement d'outils, permet de gagner du temps et d'obtenir des données quasi instantanément. Elle évite également le recours à des infrastructures physiques parfois coûteuses, comme des systèmes industriels spécifiques.

— Une **donnée réaliste**, essentielle pour garantir leur pertinence dans le contexte d'application final, afin qu'elles reflètent fidèlement les conditions réelles d'utilisation.

— Une **donnée étiquetée**, dont la qualité et la précision sont cruciales. Un étiquetage rigoureux influence directement les performances des algorithmes en fournissant des données bien annotées et exploitables.

— Une **donnée efficace**, obtenue lorsque la donnée est à la fois réaliste et étiquetée. Elle permet un apprentissage pertinent et une exploitation immédiate par les algorithmes.

— Une **donnée fidèle**, qui combine des données réalistes et générées automatiquement. Elle permet de simuler automatiquement des données réalistes tout en conservant une haute qualité d'imitation des conditions terrain.

— Une **donnée performante**, issue de données générées automatiquement et étiquetées. Elle permet d'accélérer et d'optimiser l'apprentissage grâce à un volume important de données annotées.

— Une **donnée idéale**, qui regroupe les trois piliers : réaliste, étiquetée et générée automatiquement. Elle représente l'équilibre parfait entre pertinence, scalabilité et annotation de qualité, assurant ainsi des performances optimales des modèles d'apprentissage.

Le développement d'outils est principalement axé sur la partie IT, car c'est un domaine où l'environnement est moins maîtrisé et plus dynamique. À l'inverse, dans un contexte OT, l'architecture est censée être bien définie, stable et sujette à peu de changements, ce qui facilite son contrôle. Toutefois, les réseaux industriels, qui constituent notre objet d'étude, présentent une particularité : ils sont constitués d'un mélange de trafic IT et OT. Cette cohabitation complexifie l'analyse, car elle combine la variabilité et les enjeux de sécurité propres au monde IT avec les contraintes de fiabilité et de disponibilité caractéristiques des environnements OT.

La méthodologie d'enregistrement des scénarios repose sur la capture des données réseau générées par différentes activités, qu'elles soient humaines ou automatisées. Un scénario désigne un ensemble d'activités réalisées dans un périmètre défini. Il se compose d'un contexte, incluant par exemple le nombre de machines impliquées, ainsi que d'une liste d'activités ou de profils utilisateurs types. Le niveau de précision d'un scénario peut varier en fonction de l'objectif visé. Lorsqu'il s'agit d'étiquetage, un degré de détail élevé est privilégié afin d'assurer une annotation rigoureuse et exploitable.

## 3.2 Étiquetage des données

L'étiquetage des données est une étape obligatoire pour l'apprentissage automatique supervisé. Il apparaît que toutes les méthodes actuelles d'étiquetage présentent des inconvénients fondamentaux concernant la qualité, le volume et la rapidité de traitement des données obtenues[102, 103]. Les méthodes actuelles pour l'étiquetage de trafic classique reposent encore largement sur des données synthétiques générées par des machines, où chaque activité est simulée de manière isolée, ce qui limite la représentativité des scénarios réels (ISCXTor2016[104]

et ISCXVPN2016[105]). Cette section présente une méthode pour étiqueter précisément et manuellement des données d'activité réseau ainsi que de nouveaux outils pour automatiser cet étiquetage.

### 3.2.1 TOMATO : Tools for Optimized Manual Assistance on Tagging Operations



TOMATO est, avant d'être un outil, une méthode destinée à optimiser et assister l'étiquetage de jeux de données. Elle est conçue pour faciliter l'enregistrement manuel et précis des activités d'un utilisateur, en capturant à la fois son trafic réseau, son écran, ainsi que ses interactions clavier et souris. Cette approche permet de produire une trace riche et synchronisée, mettant en évidence les actions utilisateur à partir de ses interactions visuelles et comportementales.

TOMATO s'inscrit pleinement dans le modèle de granularité présenté au Chapitre 2, en offrant un cadre permettant à l'utilisateur d'annoter manuellement les données à différents niveaux. Grâce aux informations visuelles et contextuelles, l'utilisateur peut associer les flows capturés à des opérations (représentées par les paquets), mais également les regrouper en actions (clics, soumissions de formulaire, requêtes...), et enfin en activités (navigation, streaming, envoi d'un e-mail...). De plus, TOMATO permet de distinguer des granularités variées, allant d'un usage binaire d'un service à une navigation sur un site web particulier, selon le niveau de détail choisi par l'annotateur.

Les outils proposés par TOMATO viennent ensuite assister l'utilisateur dans l'étiquetage, en synchronisant automatiquement les données réseau avec les actions observées, en fournissant des suggestions contextuelles, tout en laissant l'expert finaliser manuellement les annotations. L'objectif est d'allier rapidité et précision, tout en garantissant la cohérence et la qualité des étiquettes produites, même sur des cas complexes difficilement automatisables.

Concrètement, TOMATO s'appuie sur une collection de scripts spécialisés qui automatisent les étapes critiques du processus. Un ensemble de scripts permet ainsi d'enregistrer toutes les entrées utilisateur (clics, frappes clavier, mouvements de souris) avec un horodatage précis. D'autres scripts convertissent automatiquement les traces réseau en flows, puis réalisent une synchronisation temporelle entre ces flows et les actions utilisateur enregistrées. Enfin, un script capture une capture d'écran à chaque action utilisateur, offrant ainsi une vue contextuelle immédiate de l'environnement à ce moment précis. Grâce à cette image, l'utilisateur peut facilement comprendre l'intention de l'action et étiqueter manuellement les données avec un niveau de précision élevé.

### Protocole d'enregistrement

La méthode débute par la capture d'écran, accompagnée de l'enregistrement des entrées clavier et souris, suivie du lancement de la capture réseau (PCAP). Les données collectées sont ensuite traitées : le fichier PCAP est converti en flows grâce à NFStream et est fusionné avec les autres informations pour produire un fichier CSV. Ce dernier est conçu pour être complété avec les actions correspondant aux différentes entrées. Parallèlement, une analyse vidéo peut être exploitée de deux façons : soit l'opérateur visionne la vidéo pour replacer les actions dans leur contexte visuel et ainsi les étiqueter précisément, soit il utilise un script automatisé qui, à partir des horodatages des actions, extrait automatiquement des captures d'écran aux moments clés. Cela permet d'obtenir directement une vue contextuelle de chaque action sans avoir à parcourir l'intégralité de la vidéo. L'opérateur peut donc enrichir le CSV de manière optimale et précise. La figure 3.2 illustre les étapes de la méthodologie.

Les premières étapes d'enregistrement utilisent des outils communs comme l'enregistreur d'écran par défaut sous Windows ou Ubuntu. De même pour les captures réseau qui peuvent être obtenues avec Tcpdump ou Tshark.

### Outils

TOMATO s'appuie sur plusieurs scripts pour capturer, traiter et annoter les interactions utilisateur ainsi que le trafic réseau associé.

— **Recorder.py** : Ce script utilise la bibliothèque *python-evdev*[1], permettant d'interagir avec le sous-système evdev (event device) de Linux, qui gère les périphériques d'entrée tels que les claviers, souris, manettes, écrans tactiles, etc. Cela permet d'enregistrer les entrées clavier et souris dans notre cas avec un format plus clair et structuré. Avant son exécution, il est nécessaire d'identifier les périphériques afin de configurer correctement les paramètres dans le code.

— **Converter.py** : Une fois le trafic réseau capturé au format PCAP, ce script est utilisé pour le convertir en flows exploitables. Il s'exécute via la commande suivante : La conversion est assurée par la bibliothèque NFStream[2].

— **Processing.py** : Ce script fusionne les fichiers d'événements capturés (entrées utilisateur) avec les données réseau extraites du PCAP, afin de labelliser automatiquement chaque flow en fonction des actions effectuées sur la machine. Il permet ainsi de créer un ensemble de données enrichi pour l'analyse des activités.

— **VideoExtractor.py** : Pour affiner l'annotation des actions utilisateur, ce script extrait des captures d'écran à partir d'une vidéo enregistrée lors de la session. Ces images servent

1. https ://pypi.org/project/evdev/
2. https ://www.nfstream.org/

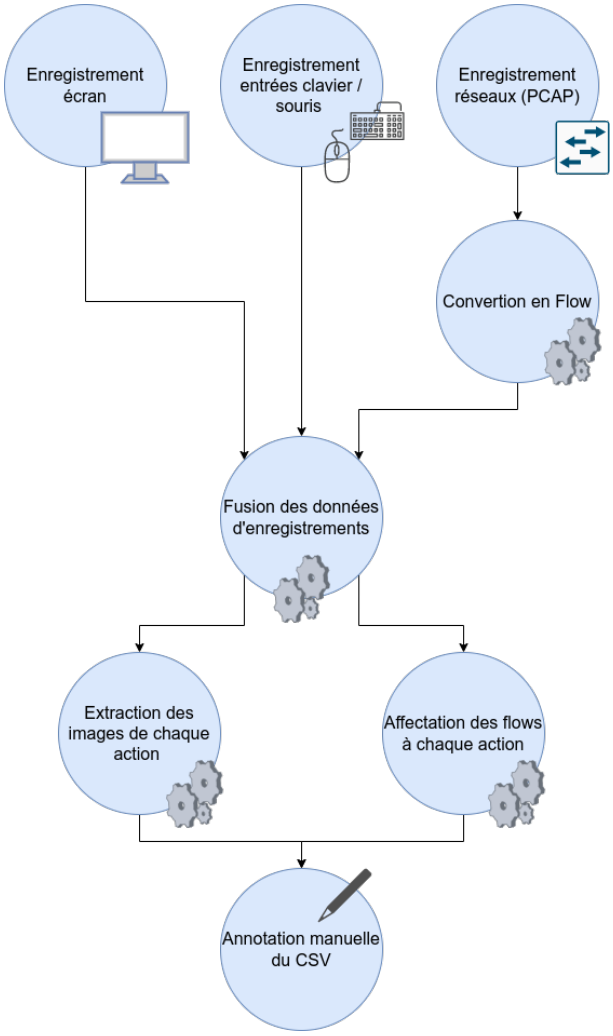



FIGURE 3.2 – Méthodologie d'enregistrement et de traitement

ensuite de référence pour compléter manuellement les annotations des interactions, garantissant une correspondance précise entre l'activité utilisateur et le trafic réseau généré.

En combinant ces scripts, TOMATO permet de capturer fidèlement un scénario d'utilisation, de l'associer avec le trafic réseau généré, et d'obtenir un jeu de données structuré et annoté pour l'analyse d'activité.

### Jeu de donnée

Nous avons produit un jeu de données original à l'aide de la méthode et de l'outil TOMATO, conçu pour capturer automatiquement les interactions utilisateur et générer un trafic réseau représentatif d'une session d'utilisation réelle. Le jeu de données NAT-U1-2024, d'une

durée d'enregistrement d'une heure, est disponible publiquement sur le dépôt GitHub [3]. Ce jeu de données offre une bonne répartition des activités réalisées (visioconférence, navigation, développement, messagerie, etc.) ainsi que des applications utilisées (Zoom, Thunderbird, Google Meet, Firefox, etc.), comme illustré dans la figure 3.3.

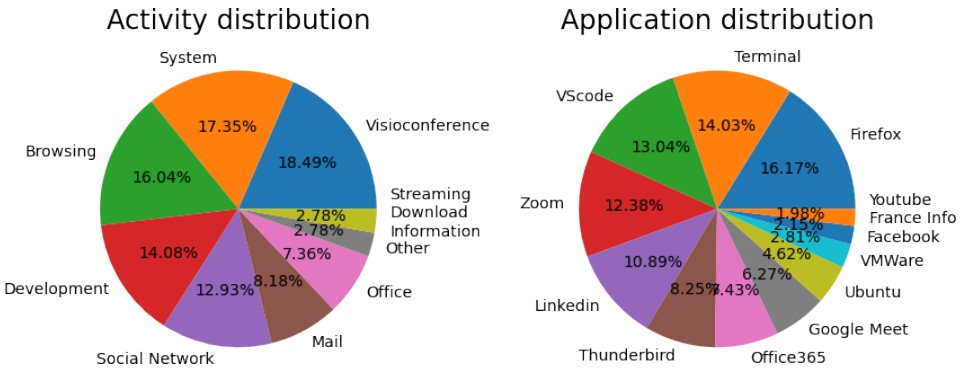



FIGURE 3.3 – Distribution des activités et applications dans le jeu de données NAT-U1-2024

Toutes les granularités de la pyramide d'activité sont étiquetées, allant du niveau binaire jusqu'aux comportements utilisateurs, ce qui permet une grande flexibilité pour l'évaluation de méthodes de classification ou de détection de comportements réseau.

Afin de proposer une représentation synthétique, nous avons défini un format de métadonnée compact qui permet, en une seule ligne, de décrire les caractéristiques essentielles d'un jeu de données réseau. Ce format facilite la comparaison rapide entre jeux de données en résumant leur origine, environnement, niveau de granularité, types d'étiquetage, ainsi que les dimensions observables du modèle d'Engeström étendu au domaine informatique.

Cette structuration repose sur l'application de la méthode présentée au chapitre 2, fondée sur la théorie de l'activité, adaptée aux environnements numériques. Elle permet de caractériser chaque jeu de données selon deux dimensions complémentaires : d'une part les composantes sociotechniques (sujet, outils, règles, etc.), et d'autre part les paramètres techniques (séquences d'opérations, environnement machine, droits, etc.) qui influencent les traces réseau générées. L'objectif est de rendre visible la richesse contextuelle contenue dans les captures réseau, et de situer chaque jeu de données par rapport à son potentiel d'analyse comportementale.

Le format retenu est le suivant :

```
[NAME] | [SOURCE] | [ENV] | [TRAFIC TYPE] | [ACTIVITY] | [LABELS] |
| [USERS] | [USER DIVERSITY] | [DURATION] | [LOWER ENGESTRÖM: S/O/I/C/Ru/Dl] |
[UPPER ENGESTRÖM: E/I/Sq/N/D/Ri] | [YEAR]
```

3. https://github.com/MaxouPicsou/NAT-U1-2024

Le tableau 3.1 décrit les champs constitutifs de ce format de métadonnée.

| **Champ** | **Description** |
|---|---|
| `NAME` | Nom du jeu de données. |
| `SOURCE` | Origine des données : `real`, `simulated` ou `semi-simulated`. |
| `ENV` | Type d'environnement observé : `IT`, `OT`, `mobile`, `IoT`, ou combinaison de ces éléments. |
| `TRAFIC TYPE` | Nature du trafic capturé : `normal`, `attack`, `mix`, etc. |
| `ACTIVITY` | Niveau maximal d'analyse atteint selon la pyramide d'activité : `B→P→S→A→W→U` (pour `B`inairy, `P`rotocol, `S`ervices, `A`pplication, `W`ebpage, `U`ser action). |
| `LABELS` | Type d'étiquettes disponibles : `mono-label`, `multi-labels`, `no-labeled`, etc. |
| `USERS` | Nombre d'utilisateurs représentés dans le jeu de données. |
| `USER DIVERSITY` | Diversité comportementale : `single` (un seul utilisateur), `multi-generic` (plusieurs utilisateurs, mais avec un seul profil généraliste) ou `multi-profiled` (un profil par utilisateur). |
| `DURATION` | Durée de la capture : format libre, mais standardisé, par exemple `1h`, `5min`, `2d`, etc. |
| `LOWER ENGESTRÖM` | Composantes observables du modèle classique : `S` (Suject), `O` (Object), `I` (Instruments), `C` (Community), `Ru` (Rules), `Dl` (Division of labor). |
| `UPPER ENGESTRÖM` | Composantes observables du modèle numérique : `E` (Environnement), `I` (Instruments), `Sq` (Sequences), `N` (Network), `D` (Division), `Ri` (Rights). |
| `YEAR` | Année/date de création ou d'enregistrement du jeu de données. |

TABLE 3.1 – Description des champs du format de métadonnée

L'application de ce format au jeu de données NAT-U1-2024 donne la ligne suivante :

```
NAT-U1-2024 | real | IT | office activities | B→P→S→A→W→U | multi-labels | 1 | single |
1h | developer / office work / HP Elitebook i7 / developer team / company rule / - |
ubuntu 22.04 / HP elitebook i7 / all / 192.168.3.10:24 / - / administrator | 2024
```

Ce résumé met en évidence que le jeu de données permet d'observer des comportements utilisateurs complets, depuis les opérations réseau jusqu'aux intentions sous-jacentes, dans un environnement contrôlé et étiqueté à tous les niveaux de la pyramide d'activité.

### 3.2.2 SALAD : Script-based Automatic Labelling on Activity Dataset



Nous avons précisé les étapes de génération et de capture dans la section 3.1. Cette section présente un outil de génération avec étiquetage automatique. Pour rappel, nous appelons génération des données synthétiques, pouvant être inspirées de données réelles, mais reproduites artificiellement ou rejouées.

SALAD (Script-based Automatic Labelling on Activity Dataset) est un outil conçu pour simplifier la création et l'étiquetage de jeux de données d'activités à partir d'environnements virtualisés. Face à la complexité de la génération de données d'activités précises et étiquetées, le conteneur semble faire partie des solutions sortant du lot comme vu dans la table 3.7. Nous proposons une approche automatisée, où des scripts d'actions peuvent être déployés facilement sur des conteneurs. L'objectif est de permettre une maîtrise totale des scénarios d'exécution et d'assurer une reproductibilité des enregistrements, tout en intégrant automatiquement l'étiquetage des flux d'activités associés. Par exemple, pour capturer le trafic généré par l'envoi d'un email, il suffit d'écrire un script Python automatisant l'envoi d'un message, puis de l'exécuter dans SALAD. Ce dernier déploiera un conteneur isolé, à partir duquel le script sera lancé, ainsi que l'enregistrement du trafic réseau. De cette manière, seules les données correspondant à l'envoi de l'email seront capturées, ce qui simplifie grandement l'étiquetage. Chaque script est exécuté individuellement, et la capture est arrêtée une fois son exécution terminée, avant de relancer une nouvelle session pour le script suivant. Cela garantit que seules les données générées par le script en cours sont enregistrées, évitant toute interférence avec d'autres processus

#### Architecture

Au niveau de l'architecture, le logiciel se décompose en 3 parties. La première est l'interface utilisateur qui permet d'interagir avec les différentes commandes comme le déploiement du conteneur, le lancement des scripts, l'enregistrement et le traitement des captures réseau. La seconde partie concerne la construction du conteneur ainsi que la gestion des évènements. En effet, elle doit gérer la génération du *Dockerfile*[4] et le déploiement de celui-ci en fonction des personnalisations ajoutées par l'utilisateur. Une fois déployé, un programme est en charge de gérer les évènements qui sont envoyés par l'utilisateur à l'aide de l'interface graphique décrite précédemment. Enfin, la dernière partie porte sur le stockage des enregistrements. Un volume docker est créé lors du déploiement du conteneur et permet de facilement stocker et récupérer les enregistrements.

4. Fichier contenant la configuration d'un conteneur Docker

### Fonctionnement

Au lancement du programme, l'utilisateur peut ajouter ou effacer des scripts de la base de données du logiciel, puis sélectionner ceux qui seront lancés depuis le conteneur. La figure C.1, en annexe, montre un aperçu de l'interface graphique actuelle du logiciel au moment de l'écriture de ce rapport. On peut y voir deux scripts disponibles qui peuvent être sélectionnés pour être lancé depuis le conteneur qui sera déployé.

Lorsque l'utilisateur a sélectionné les scripts nécessaires pour son expérience, il suffit de cliquer sur *Deploy* pour lancer la génération du *Dockerfile* ainsi que le lancement de celui-ci qui est en charge du déploiement du conteneur. Un autre script appelé *Receiver.py* est aussi généré et lancé au démarrage du conteneur. Il permet de recevoir les ordres et commandes émises par l'utilisateur via le programme, afin de lancer les scripts précédemment sélectionnés. Il est aussi en charge d'envoyer l'état d'exécution de ceux-ci afin d'attendre qu'ils se terminent.

Une fois déployé, l'utilisateur a accès à de nouvelles commandes, comme le montre la figure C.2 en annexe, lui permettant de lancer/arrêter un enregistrement, mais également d'exécuter manuellement chaque script qu'il a précédemment sélectionné. Les enregistrements sont stockés dans un volume Docker, permettant ainsi à l'utilisateur de les récupérer sur son système. La figure 3.4 résume les actions entre les différentes parties au sein de l'architecture.

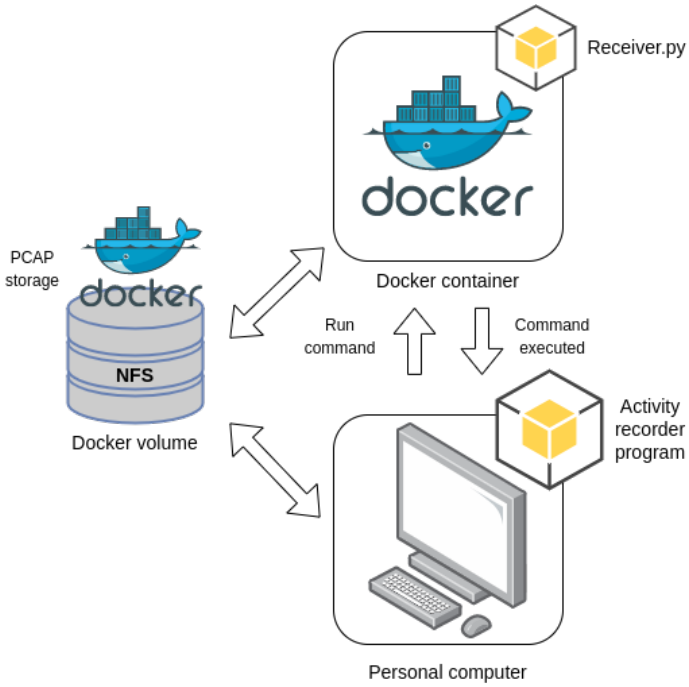



FIGURE 3.4 – Architecture de l'enregistreur

Le logiciel permet également le traitement des captures réseau. Celui-ci est en charge d'éti-

queter les enregistrements. Au lancement et à la fin de chaque script, le temps ainsi que les différentes connexions utilisées sont relevés. Ceux-ci sont ensuite utilisés pour étiqueter les portions de la capture qui concerne les actions effectuées lors de l'exécution du script.

### Comparaison

La solution d'automatisation des enregistrements permet un gain de temps important et apporte une meilleure précision temporelle par rapport à une méthode entièrement manuelle. Cependant, il est probable que cette méthode donne des enregistrements assez différents qu'avec des actions directement initiées par une vraie personne. En effet, un utilisateur réel adopte un comportement plus naturel et imprévisible, avec des temps de latence variables, des erreurs, des hésitations, ou encore des réactions contextuelles à l'interface. De plus, certaines sources de bruit ou de trafic d'arrière-plan, présentes lors d'un usage normal, sont souvent absentes dans un environnement automatisé contrôlé. Il est donc utile de s'intéresser à ces différences et dans quelles mesures celles-ci impacteraient les résultats lors d'un apprentissage automatique par exemple. Pour cela, nous allons comparer 3 méthodes, les actions lancées manuellement, via à conteneur ou via une machine virtuelle.

— **Actions manuelles** : La première méthode est la plus réaliste et la plus simple à mettre en œuvre. Cependant, c'est également la plus chronophage puisque tout est manuel. Les actions seront effectuées le plus naturellement possible et donneront des enregistrements avec un rythme d'action plausible sans artefacts liés à l'exécution automatique. L'étiquetage sera peut-être un peu plus approximatif, car les temps seront également relevés manuellement, et donc plus sujet à des erreurs.

— **Actions automatiques via conteneur** : La deuxième méthode vise à exploiter le logiciel SALAD qui a été développé et qui est décrit plus tôt. L'avantage principal est l'enregistrement et l'étiquetage automatique des captures. Cela permet un gain de temps considérable. Il faut cependant étudier les différences et biais que cette méthode peut engendrer. En effet, est-il nécessaire de gérer une temporisation entre les actions pour se rapprocher d'un comportement réel, ou cette différence est-elle négligeable une fois l'apprentissage effectué par l'algorithme de classification.

— **Actions automatiques via machine virtuelle** : Cette dernière méthode est assez similaire à la précédente, mais n'est pas intégrée dans SALAD. Une méthode comparable a été proposée[106], dans le contexte de la génération de jeux de données pour la détection d'anomalies et l'apprentissage d'IDS. La principale différence réside dans l'environnement d'exécution qui est ici une machine virtuelle. Celle-ci est en effet plus proche d'une machine utilisateur qu'un conteneur dépourvu d'interface graphique. Encore une fois, il est ici question de vérifier quels possibles biais peuvent ou non impacter la qualité des captures

lors de l'étape d'apprentissage de nos algorithmes d'apprentissage automatique qui seront détaillés dans la section suivante.

| **Critère** | **Actions manuelles** | **Automatiques via conteneur (SALAD)** | **Automatiques via machine virtuelle** |
|---|---|---|---|
| Réalisme du comportement utilisateur | Très élevé | Moyen | Moyen |
| Complexité de mise en œuvre | Faible | Moyenne | Élevée |
| Précision de l'étiquetage | Faible | Élevée | Élevée |
| Soumis à des biais temporels | Faible | Moyen | Moyen |
| Diversité comportementale | Élevée | Moyenne | Moyenne |
| Adapté à l'apprentissage automatique | Oui, mais lent | Oui | Oui |

TABLE 3.2 – Comparaison des méthodes de génération de données

### 3.2.3 GAR7IC : Generalized Automatic Review and Labelling for S7COMM Industrial Communication



Après avoir présenté les outils d'étiquetage TOMATO et SALAD, principalement conçus pour le trafic réseau classique (IT), nous introduisons à présent un outil dédié aux systèmes industriels.

Jusqu'à présent, l'approche adoptée consistait à utiliser et étiqueter des flows adaptés au trafic en clair, typique des environnements IT classiques tels que le web ou la messagerie. Il apparaît cependant que pour d'autres types de trafic, un problème se pose : les flows générés deviennent trop volumineux et difficilement exploitables, notamment parce que la connexion est maintenue en continu selon le protocole utilisé.

Concernant les systèmes industriels, et plus particulièrement le protocole S7COMM, il est nécessaire de comprendre que celui-ci permet des opérations de lecture et d'écriture dans des registres spécifiques. L'idée est alors d'en extraire des flows spécialisés afin de les traiter de manière plus adaptée. De la même façon, les échanges VPN doivent être segmentés pour permettre une meilleure interprétation des données. Un découpage temporel semble être une piste pertinente pour extraire des informations exploitables, bien que cette question mérite une analyse plus approfondie par la suite.

Un dernier outil est présenté dans cette section, il s'agit de GAR7IC. Il vient compléter les précédents en permettant, à partir d'une capture réseau et d'un fichier de configuration représentant l'adressage des automates du réseau, d'annoter automatiquement le fichier au format PCAPNG grâce au champ *commentaire* ou alors de générer un CSV comprenant l'ensemble des échanges S7COMM et leurs caractéristiques associées. Dans certains cas le traitement par flows peut s'avérer plus compliqué avec les automates, car une connexion peut contenir de multiples communications, lectures, écritures ou autres et peut faire perdre en précision sur la nature des activités produites, c'est pour cela que cet outil a été développé.

### Architecture

GAR7IC est un script Python en ligne de commande conçu pour analyser les communications industrielles utilisant le protocole S7COMM (compatibles avec les automates Siemens des séries S7-200, S7-300, S7-400, S7-1200 et S7-1500). Il repose sur plusieurs bibliothèques spécialisées pour traiter les captures réseau et extraire les informations pertinentes.

L'outil fonctionne exclusivement en mode post-traitement, c'est-à-dire qu'il analyse une capture réseau existante au format PCAP plutôt qu'un flux en temps réel. Pour cela, il utilise Pyshark afin d'extraire les attributs des trames capturées et de structurer les échanges S7COMM. L'interprétation des échanges avec les automates Siemens est facilitée par la bibliothèque Snap7 [5], qui permet d'analyser et d'interagir avec les données des équipements industriels.

L'organisation des équipements du réseau est définie par un fichier de configuration au format YAML, qui suit une structure spécifique permettant de représenter l'architecture des automates ainsi que leur adressage comme sur la figure 3.5. Ce fichier est utilisé pour contextualiser les échanges détectés dans la capture réseau et enrichir l'analyse.

L'export des résultats peut se faire selon deux modes :

— **Annotation directe de la capture réseau** : Les informations extraites sont intégrées dans le champ commentaire des paquets, ce qui nécessite de convertir le fichier d'entrée PCAP en PCAPNG pour supporter ces annotations.

— **Génération d'un fichier CSV** : L'ensemble des échanges S7COMM est structuré dans un DataFrame avec Pandas, permettant une analyse ultérieure des données sous un format standardisé.

L'architecture modulaire de GAR7IC permet une extensibilité future, avec la possibilité d'ajouter de nouvelles fonctionnalités comme l'intégration d'autres protocoles industriels ou une interface interactive pour faciliter son utilisation.

5. https ://snap7.sourceforge.net/

```yaml
plc:
  - name: Water pump
    model: CPU 315-2 PN/DP
    ip: 127.0.0.1
    rack: 0
    slot: 0
    port: 102
    io_mapping:
      data_block:
        - name: DB1
          number: 1
          variables:
            - name: Machine state
              address: 0.0
              type: BOOL
            - name: Target water level
              address: 2.0
              type: INT
            - name: Current water level
              address: 4.0
              type: REAL

      inputs:
        - name: Water Sensor
          address: 0.0
          type: BOOL

      outputs:
        - name: Pump State
          address: 0.0
          type: BOOL
```

FIGURE 3.5 – Exemple de fichier YAML décrivant la configuration d'un automate Siemens

### Fonctionnement et données

L'outil fonctionne en ligne de commande avec les arguments suivants :

`-f` : Spécifie le fichier de capture réseau en entrée (`.pcap` ou `.pcapng`).

`-c` : Indique le fichier de configuration YAML contenant l'adressage des automates.

`-t` : Active l'exportation des données sous forme d'un fichier CSV.

`-p` : Active l'annotation du fichier PCAPNG avec les noms des variables extraites.

Lorsque l'option `-t` est activée, GAR7IC extrait les informations données dans la table 3.3 pour chaque requête S7COMM :

TABLE 3.3: Liste des champs extraits par GAR7IC

| Champ | Description |
|---|---|
| `Frame_Number` | Numéro de la trame dans le fichier PCAP |
| `Timestamp` | Horodatage de la trame (format lisible) |
| `Timestamp_Epoch` | Horodatage en format Unix (Epoch) |
| `Timestamp_Shift` | Décalage temporel relatif depuis la première trame |
| `Source_IP` | Adresse IP de l'équipement émetteur |
| `Destination_IP` | Adresse IP de l'équipement récepteur |
| `Length` | Taille du paquet en octets |
| `Header_Rosctr` | Type de requête (JOB, ACK, etc.) |
| `Header_PduRef` | Numéro de référence du PDU (Protocol Data Unit) |
| `Param_Function` | Fonction S7COMM (READ, WRITE, PLC_STOP, etc.) |
| `Param_Item_Count` | Nombre d'éléments contenus dans la requête |
| `Param_Item_Transport_Size` | Taille du type de données transporté |
| `Param_Item_Length` | Longueur du champ de données |
| `Param_Item_DB` | Numéro du Data Block concerné (si applicable) |
| `Param_Item_Area` | Zone mémoire (INPUT, OUTPUT, DB, etc.) |
| `Param_Address_Byte` | Offset mémoire en octets |
| `Param_Address_Bit` | Offset mémoire en bits (pour les BOOL) |
| `Variable_Name` | Nom de la variable associée (issu du fichier YAML) |
| `Data_Type` | Type de donnée (`BOOL`, `INT`, `REAL`, etc.) |
| `Data_Value` | Valeur extraite (si applicable) |
| `Data_Return_Code` | Code de retour de la requête (SUCCESS, FAILURE, etc.) |

Les données extraites sont enregistrées sous forme d'un fichier CSV avec la structure donnée par la table 3.4 contenant quelques-uns des champs précédemment cités :

Cet exemple illustre une extraction où une requête `READ` est effectuée sur un Data Block pour la variable `Machine State`. La réponse retourne une valeur booléenne `TRUE` confirmant l'état de marche de la machine.

Grâce à ces fonctionnalités, GAR7IC facilite l'analyse et l'interprétation des échanges S7COMM, permettant d'identifier rapidement les interactions entre automates et d'améliorer la visibilité sur les processus industriels.

### 3.2.4 Automatisation de l'étiquetage des données avec LLM

Pour automatiser l'étiquetage du trafic réseau, une approche assistée par intelligence artificielle a été explorée à l'aide de plusieurs LLM, à savoir, ChatGPT et Mistral. Dans un premier

| Frame | Timestamp_Epoch | Header_Rosctr | Param_Function | Param_Item_Area | Variable_Name | Data_Value |
|---|---|---|---|---|---|---|
| 1 | 1740841672 | JOB | READ | DATA_BLOCK | Machine State | UNKNOWN |
| 2 | 1740841672 | ACK_DATA | READ | Unknown | Machine State | TRUE |
| 3 | 1740841673 | JOB | WRITE | DATA_BLOCK | Temperature | 45.3 |
| 4 | 1740841673 | ACK_DATA | WRITE | Unknown | Temperature | SUCCESS |
| 5 | 1740841675 | JOB | READ | INPUTS | Sensor A | 23.1 |
| 6 | 1740841676 | ACK_DATA | READ | Unknown | Sensor A | UNKNOWN |
| 7 | 1740841677 | JOB | WRITE | OUTPUTS | Motor Speed | 1500 |
| 8 | 1740841677 | ACK_DATA | WRITE | Unknown | Motor Speed | SUCCESS |

Table 3.4 – Exemple d'extraction de données S7COMM par GAR7IC

temps, des tests manuels ont été réalisés en prenant des captures d'écran de la machine afin d'identifier la ou les activités en cours. Pour affiner la précision de l'étiquetage, la notion de granularité d'activité leur a été introduite, permettant de moduler le niveau de détail de la classification.

### Traitement d'une image

Dans un premier temps, avant d'aborder des documents plus complexes et d'introduire la notion de granularité, un premier test a été réalisé sur une capture d'écran, illustrée à la figure C.4. Avant d'envoyer l'image à ChatGPT, la question suivante lui a été posée.

**Question**

si je te donne une image de mon écran d'ordinateur, peux-tu me décrire l'action que j'effectue

À cela, on obtient la réponse suivante :

**Réponse**

L'image que tu as partagée montre une capture d'écran d'une documentation Docker, plus précisément une page concernant la configuration du réseau en mode "bridge" et l'utilisation d'IPv6. Voici une description des actions que tu sembles effectuer :

- **Consultation de la Documentation Docker** : Tu es en train de lire ou de consulter la documentation officielle de Docker sur la création et la gestion des réseaux, notamment l'utilisation de l'IPv6 dans un réseau défini par l'utilisateur. Le texte visible inclut des instructions sur la création d'un réseau Docker avec prise en charge de l'IPv6 en utilisant la commande Docker suivante : docker network create –ipv6 –subnet 2001 :db8 :1234 : :/64 my-net.
- **Navigation dans Plusieurs Onglets** : Plusieurs onglets sont ouverts dans ton navigateur, ce qui suggère que tu fais probablement des recherches ou que tu travailles sur plusieurs tâches en même temps, liées à Docker, à l'administration de serveurs (peut-être via NoVNC), et possiblement d'autres sujets techniques. Parmi les onglets visibles, on peut distinguer des termes liés à Docker, aux conteneurs, à l'authentification CAS, et au réseau, ce qui indique un travail potentiellement lié à l'administration ou à la configuration d'une infrastructure.
- **Environnement de Travail** : Tu utilises un navigateur web dans un environnement de bureau sous Linux (probablement Ubuntu, étant donné la barre de titre). Plusieurs outils et services sont ouverts en arrière-plan, ce qui montre un environnement de travail technique et focalisé sur la gestion des réseaux ou des systèmes.

En résumé, tu sembles être en pleine configuration ou en train d'apprendre à configurer des réseaux Docker, avec une attention particulière à l'IPv6, tout en jonglant entre différentes ressources et outils liés à ton travail.

À partir de cet exemple, ChatGPT est capable d'analyser l'écran et de déduire l'activité en cours en s'appuyant sur l'ensemble des informations présentes dans la capture, telles que la fenêtre active, les onglets ouverts et l'environnement de bureau. Étant donné que cette capture ne concerne qu'un écran secondaire, encore plus d'informations auraient pu être déduites avec une vue complète de l'affichage, notamment grâce aux autres fenêtres ouvertes, à la barre des

tâches affichant les applications en arrière-plan ou favorites, ainsi qu'au fond d'écran. Cela en fait un outil puissant pour identifier automatiquement les applications utilisées, sans nécessiter d'intervention humaine.

| **Granularité** | **Image 1** (Docker Docs) | **Image 2** (PyCharm) | **Image 3** (Bureautique) |
|---|---|---|---|
| Binaire | Normal | Normal | Normal |
| Protocoles | HTTPS | Python | IMAP/SMTP, HTTPS |
| Groupes de services | Documentation | Développement | Bureautique, Communication |
| Applications | Navigateur | PyCharm | ONLYOFFICE, Mail |
| Site Web | Docker Docs | N/A | N/A |
| Comportements | Consultation | Prog. | Rédaction, Mails |

TABLE 3.5 – Classification des activités pour les trois images par ChatGPT

Suite aux tests réalisés avec une image, il lui a été demandé de fournir une réponse sous forme d'un seul mot pour chaque niveau de granularité, en fonction des éléments visibles sur une capture d'écran. Pour évaluer cette approche, trois tests ont été effectués, chacun basé sur une capture d'écran distincte que l'on retrouve sur les figures C.4, C.5 et C.6. Les résultats de ces expérimentations sont illustrés dans la table 3.5.

ChatGPT démontre une excellente capacité à classifier les activités de l'utilisateur en fonction de différents niveaux de granularité, en prenant en compte les applications ouvertes et leur contexte d'utilisation. Il est capable d'identifier des tâches variées et de contextualiser la simultanéité des applications : l'ouverture d'un éditeur de texte et d'une boîte mail.

À titre de comparaison, la même expérience a été réalisée avec Mistral avec un modèle Pixtral[6]. Les résultats se trouvent à la table 3.6.

| **Granularité** | **Image 1** (Docker Docs) | **Image 2** (PyCharm) | **Image 3** (Bureautique) |
|---|---|---|---|
| Binaire | Normal | Normal | Normal |
| Protocoles | HTTPS | N/A | N/A |
| Groupes de services | Navigation | Développement | N/A |
| Applications | Navigateur WEB | PyCharm | Traitement de texte, Messagerie |
| Site Web | docs.docker.com | N/A | N/A |
| Comportements | Consultation documentation | Développement d'une application WEB | Rédaction, Consultation emails |

TABLE 3.6 – Classification des activités pour les trois images par Mistral

6. Modèles dotés de capacité de vision, leur permettant d'analyser des images et de fournir des informations basées sur le contenu visuel, en plus du texte.

Il est à noter que l'apprentissage du modèle a été bien plus rapide avec Mistral, en une seule définition de la granularité et le format de réponse attendu il a été capable de répondre très précisément sans erreur. Il aura fallu plusieurs tentatives et d'apprentissage à ChatGPT pour obtenir ces réponses. Cela peut aussi s'expliquer par la sélection du modèle plus spécialisé chez Mistral alors que GPT-4 est très généraliste.

**Traitement de plusieurs images**

Afin d'accélérer le traitement, plusieurs techniques d'automatisation ont été testées. Grâce à la capacité de ChatGPT à traiter une vidéo, l'analyse des activités utilisateur peut être optimisée en concaténant une série d'images capturées au fil du temps pour générer un fichier vidéo unique. Cette approche permet de suivre l'évolution des actions effectuées sur un ordinateur de manière fluide et continue, facilitant ainsi l'interprétation du contexte global d'utilisation. Plutôt que d'examiner chaque image individuellement, la vidéo offre une vue d'ensemble des interactions successives entre différentes applications, mettant en évidence les transitions, les chevauchements d'activités et les enchaînements logiques.

Malheureusement, les tests sur une vidéo contenant 20 images soit moins d'une seconde n'ont pas été concluants sur le modèle précédemment utilisé pour les images seules. En effet, malgré le fait de préciser la manière dont est construite la vidéo et d'analyser chaque image, le modèle ne nous donne pas de granularités satisfaisantes voire de fausses analyses (logiciel non utilisé, mauvaise activité).

Une autre méthode est l'utilisation de l'API de Mistral qui est gratuite, contrairement à celle de ChatGPT. Un agent a été déployé avec pour seules données d'apprentissage, les définitions des différentes granularités ainsi que le format de réponse attendu qui est le suivant :

**Instructions**

Ton rôle est d'analyser des images de captures d'écran d'ordinateur et de fournir des descriptions basées sur différents niveaux de granularité de l'activité. Les niveaux de granularité sont les suivants : 1. Binaire : Classification simple comme trafic normal/anormal, chiffré/non-chiffré. 2. Protocoles : Classification basée sur les protocoles réseau et leurs ports. 3. Groupes de service : Classification des services en groupes comme streaming, téléchargement, navigation internet. 4. Applications : Identification de l'application utilisée. 5. Site WEB : Identification des pages web visitées. 6. Comportements utilisateurs : Description détaillée des actions de l'utilisateur, comme poster une image sur les réseaux sociaux ou rédiger un document. Pour chaque image reçue, fournis une description en un mot pour chaque niveau de granularité, en laissant des cases vides si nécessaire. Retourne les descriptions au format CSV.

Il suffit ensuite de lui envoyer chaque image en base64 et obtenir la réponse automatiquement dans le format souhaité. Les données renvoyées correspondent parfaitement aux granularités de la table 3.6.

## 3.3 Génération de données réalistes

Après avoir présenté les méthodes et outils dédiés à l'étiquetage des données réseau, nous nous intéressons dans cette section à la génération de données, une étape tout aussi essentielle dans le cadre de l'apprentissage automatique. La génération de données est une étape essentielle pour l'entraînement et l'évaluation des algorithmes d'apprentissage automatique, permettant d'explorer différents paramètres afin d'optimiser leur performance ou de tester leur robustesse face à diverses conditions. Toutefois, le réalisme des données constitue un critère fondamental, car des jeux de données trop artificiels risquent d'induire des biais et de compromettre la fiabilité des modèles entraînés [102, 103]. En effet, pour que les résultats obtenus soient exploitables et transposables à des environnements réels, il est essentiel que les données générées reproduisent fidèlement la complexité et la diversité du trafic observé dans des conditions réelles. De ce manque de réalisme sont nés différents outils présentés dans cette section.

### 3.3.1 Enregistrement et rejeu des scénarios

L'enregistrement et le rejeu de scénarios apparaissent souvent comme la méthode la plus simple à mettre en place pour générer des flux de données réalistes. Ils font partie des étapes

fondamentales pour la création de jeux de données destinés à l'analyse réseau ou à l'entraînement de modèles d'apprentissage automatique. Ces techniques consistent à capturer les interactions réseau ou système lors de l'exécution d'un scénario donné, puis à les reproduire dans des environnements contrôlés.

Dans ce contexte, les principes des 4 R – Repeatability, Reproducibility, Replicability, et Reusability – jouent un rôle clé pour garantir la validité et l'utilité des scénarios enregistrés et rejoués. La figure 3.6 est tirée et traduite des travaux de Hernández et Colom (2023) [4].

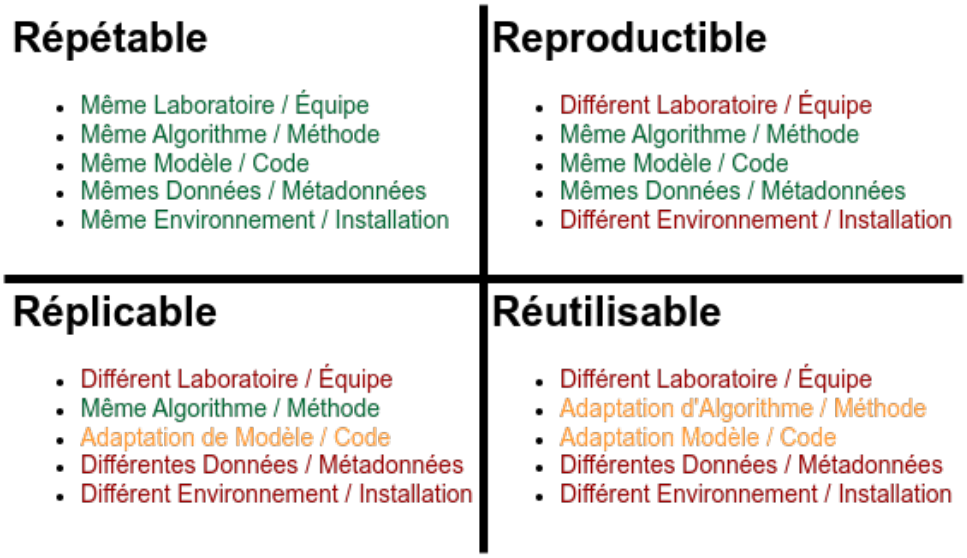



FIGURE 3.6 – Principe des 4R (Hernández et Colom - 2023[4])

— **Repeatability (Répétabilité)** : Il s'agit de la capacité à obtenir des résultats identiques en exécutant plusieurs fois le même scénario dans un environnement inchangé. Cela est essentiel pour les tests en cybersécurité ou l'entraînement de modèles d'apprentissage automatique, où la stabilité des jeux de données est cruciale.

— **Reproducibility (Reproductibilité)** : Ce concept va au-delà de la répétabilité en s'assurant que d'autres chercheurs ou ingénieurs peuvent obtenir des résultats similaires en suivant les mêmes instructions, mais dans des environnements légèrement différents. Par exemple, un scénario enregistré sur une machine peut être reproduit sur une autre avec des variations mineures.

— **Replicability (Réplicabilité)** : Ici, il est question de reproduire un scénario ou une expérience avec un autre ensemble de données ou un contexte différent, tout en conservant une cohérence dans les observations obtenues. Cela permet de tester la généralisation des modèles ou la robustesse des systèmes d'analyse.

— **Reusability (Réutilisabilité)** : Enfin, la réutilisabilité concerne la capacité à adapter et à utiliser les scénarios enregistrés dans de nouvelles expériences, potentiellement avec des objectifs différents. Cela est particulièrement important pour l'apprentissage automatique, où un jeu de données peut être exploité pour différents types d'analyses ou d'applications.

En intégrant ces principes, l'enregistrement et le rejeu de scénarios deviennent non seulement des outils d'analyse puissants, mais aussi des mécanismes fondamentaux pour garantir la fiabilité et l'évolution des modèles et des infrastructures d'analyse réseau.

L'enregistrement global sur une machine unique, les approches basées sur la virtualisation avec machines virtuelles ou conteneurs, ainsi que l'utilisation de systèmes d'exploitation basés sur des hyperviseurs comme Qubes OS, représentent différentes méthodes permettant de répondre à ces besoins. Chacune de ces solutions présente des avantages et des inconvénients en termes de réalisme, de reproductibilité, de contrôle, et de facilité d'étiquetage.

Dans cette section, nous explorerons ces approches en détail, en mettant en lumière leurs particularités et leurs cas d'utilisation respectifs. Nous conclurons par une comparaison globale, permettant d'éclairer le choix de la méthode la plus adaptée selon les contraintes et les objectifs des travaux.

#### Enregistrement Global sur une Machine Unique

L'enregistrement global sur une machine unique est une méthode simple et rapide pour capturer le trafic réseau. Elle nécessite uniquement un outil de capture et permet de collecter une vue complète de toutes les interactions réseau d'un système. Cependant, cette approche manque de précision et de contrôle : les flux capturés incluent non seulement les activités d'intérêt, mais également les processus en arrière-plan, les logiciels qui tournent de manière imprévisible, et parfois même des informations personnelles ou non désirées. Cette absence d'isolation rend difficile l'étiquetage précis des données, ce qui la rend inappropriée pour des jeux de données destinés à l'apprentissage supervisé, où chaque échantillon doit être rigoureusement associé à une activité spécifique. En revanche, pour des données de test, cette méthode est particulièrement intéressante. En effet, elle reproduit une situation réaliste où les données sont hétérogènes et bruitées, ce qui correspond mieux à des scénarios réels. Dans ce cas, l'objectif est de développer des modèles capables de classifier efficacement les activités à partir de ce type de données globales, telles qu'elles apparaissent dans un environnement réel et non contrôlé.

#### Approche par Machines Virtuelles et Conteneurs

La virtualisation, qu'elle soit réalisée à l'aide de machines virtuelles (VMs) ou de conteneurs, offre une solution efficace pour l'enregistrement et le rejeu de scénarios réseau tout en permettant une meilleure isolation des processus.

Pour rappel, il existe différents types de virtualisation, parmi eux l'hyperviseur de type 1, de type 2 et les conteneurs (Figure 3.7).

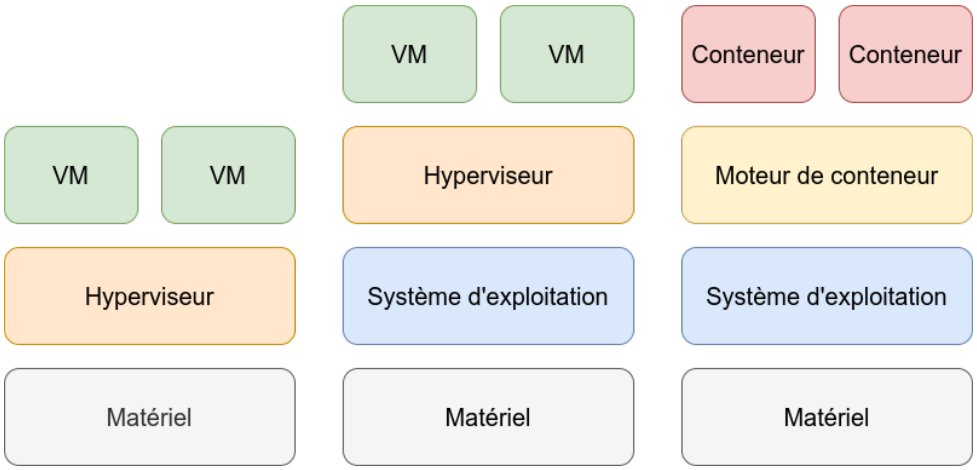



FIGURE 3.7 – Architecture de l'hyperviseur type 1, l'hyperviseur type 2 et la conteneurisation

L'hyperviseur de type 1, aussi appelé "bare-metal", s'exécute directement sur le matériel physique de la machine hôte et gère les machines virtuelles sans nécessiter de système d'exploitation intermédiaire. L'hyperviseur de type 2, en revanche, fonctionne au-dessus d'un système d'exploitation hôte et repose sur ce dernier pour gérer les ressources matérielles, ce qui le rend plus simple à mettre en place, mais avec une légère perte de performance. Les conteneurs, quant à eux, ne virtualisent pas le matériel, mais plutôt le système d'exploitation, permettant l'exécution d'applications isolées partageant le même noyau. Cette approche offre une légèreté et une rapidité de déploiement accrues par rapport aux autres systèmes de virtualisation.

En revenant sur nos objectifs d'enregistrement de données réseau, les machines virtuelles, par exemple, créent des environnements entièrement séparés avec leurs propres systèmes d'exploitation, ce qui permet de tester des scénarios dans des configurations réalistes et contrôlées. Elles offrent un excellent niveau de contrôle sur les applications et les ressources, garantissant ainsi que seules les interactions pertinentes sont capturées. Cependant, cette approche peut être gourmande en ressources et nécessite des configurations spécifiques pour être mise en œuvre de manière optimale.

Les conteneurs, en revanche, sont plus légers et permettent une isolation partielle des applications tout en partageant le noyau du système hôte. Cette solution est plus rapide à déployer et plus économique en termes de ressources, ce qui en fait une option intéressante pour les environnements où la flexibilité et la rapidité sont cruciales. Cependant, l'isolation fournie par les conteneurs n'est pas aussi forte que celle des machines virtuelles, ce qui peut rendre plus difficile le contrôle précis des interactions entre les différents processus.

Ainsi, ces deux technologies de virtualisation présentent des avantages significatifs en matière de flexibilité, d'isolation et de réalisme des scénarios, mais elles nécessitent un compromis entre la consommation de ressources et la granularité du contrôle. Elles sont particulièrement adaptées pour des tests où plusieurs environnements ou services doivent être simulés simultanément, offrant une reproduction fidèle des interactions dans des conditions réalistes tout en permettant un certain degré de personnalisation.

L'un des grands avantages de la virtualisation, que ce soit avec des machines virtuelles ou des conteneurs, réside dans la possibilité de maîtriser précisément quelles applications ou services sont actifs au sein de chaque environnement. En isolant chaque application dans un conteneur ou une machine virtuelle dédiée, il devient possible de contrôler rigoureusement les processus en cours d'exécution, d'empêcher l'exécution de logiciels non désirés et de réduire ainsi le bruit réseau et les interférences dans les données enregistrées. Cela permet de garantir que seules les activités pertinentes sont capturées et enregistrées, facilitant ainsi l'étiquetage précis des flux réseau.

De plus, ces technologies permettent une reproduction fidèle d'un environnement d'enregistrement particulier, offrant ainsi la possibilité de rejouer un scénario avec les mêmes configurations, applications et conditions de réseau qu'au moment de l'enregistrement initial. Cette reproductibilité est essentielle pour les tests et le calibrage des modèles d'apprentissage automatique, permettant une évaluation plus précise et consistante des performances du modèle sur des données simulées. La capacité de rejouer des scénarios réseau impacte fortement les 4R, améliorant la répétabilité et la reproductibilité, et facilitant la réplicabilité sous certaines conditions. La réutilisabilité est également accrue, bien que les biais potentiels du rejeu puissent nécessiter une attention particulière pour éviter des interprétations erronées des performances des modèles. La table 3.7 en fin de section résume les avantages et inconvénients de chaque solution.

Cependant, la gestion de l'enregistrement dans ces environnements virtualisés présente des défis. Lorsque plusieurs machines virtuelles ou conteneurs sont utilisés simultanément, la synchronisation des enregistrements devient un point critique. Chaque instance doit être configurée pour capturer les données réseau de manière cohérente et synchronisée, ce qui peut rapidement devenir complexe. En particulier, il peut être difficile de garantir que les estampilles temporelles des enregistrements provenant de différentes machines ou conteneurs soient alignées, ce qui complique l'analyse ultérieure des flux. Cette complexité ajoute une couche de gestion supplémentaire, nécessitant des outils et des stratégies bien définis pour assurer que tous les environnements fonctionnent de manière coordonnée et efficace.

#### Enregistrement avec un hyperviseur de type 1

Tous les systèmes d'exploitation ne reposent pas sur le même modèle de sécurité, et cette différence impacte directement la fiabilité de l'enregistrement et l'analyse des activités. Certains

systèmes d'exploitation utilisent un hyperviseur intégré au système hôte, les rendant dépendants de sa robustesse. En cas de compromission, toutes les machines virtuelles exécutées dessus sont exposées, compromettant ainsi l'intégrité des données capturées.

C'est dans cette optique que des systèmes comme Qubes OS [7] ont été développés. Qubes OS est un système d'exploitation axé sur la sécurité, développé par Invisible Things Lab [8], une entreprise spécialisée en cybersécurité fondée par Joanna Rutkowska. La première version a été publiée en 2012, avec pour objectif de proposer une approche innovante de la sécurité informatique basée sur la compartimentalisation des tâches via la virtualisation. Il repose sur l'hyperviseur Xen, permettant d'exécuter différentes applications et activités dans des machines virtuelles isolées (qubes).

Au fil des années, Qubes OS s'est imposé comme une référence en matière de sécurité, attirant un public composé de journalistes, chercheurs en cybersécurité et utilisateurs soucieux de la protection de leurs données. Il est notamment recommandé par Edward Snowden[107] et la Freedom of the Press Foundation[108] pour les communications sécurisées.

En raison de son approche rigoureuse en matière d'isolation des processus, Qubes OS a une adoption plus limitée que des systèmes d'exploitation classiques comme Windows ou Linux, mais sa communauté d'utilisateurs reste active. Le nombre exact d'utilisateurs n'était pas officiellement connu, mais les estimations basées sur les téléchargements et l'activité de la communauté indiquent plusieurs dizaines de milliers d'utilisateurs à travers le monde[109].

Dans Qubes OS, l'approche de compartimentalisation ne se limite pas aux applications et aux processus utilisateurs. Tout élément matériel sensible est également isolé dans une machine virtuelle dédiée afin de prévenir les attaques visant les périphériques. Par exemple :

— Les périphériques USB sont gérés par une VM spécifique (USB qube), empêchant tout accès direct au système principal, limitant ainsi les risques d'attaques via des périphériques malveillants (BadUSB[110]).

— Les interfaces réseau sont centralisées dans une VM dédiée appelée NetVM, qui contrôle les connexions réseau avant de les transmettre aux autres qubes.

— Les pare-feux et VPN peuvent être configurés dans des qubes spécialisés (FirewallVM, VPN qube), permettant un contrôle fin des flux réseau tout en garantissant un cloisonnement strict des communications.

Cette architecture offre un cadre optimal pour l'enregistrement et l'analyse des interactions, tout en limitant les risques d'altération ou de fuite de données. Son principe de compartimentalisation permet d'exploiter pleinement l'étiquetage des données, chaque qube étant dédié à une activité unique. Cela garantit une association précise des flux réseau et des interactions à une tâche spécifique, comme illustré sur la figure 3.8.

7. https ://www.qubes-os.org/
8. https ://invisiblethingslab.com/

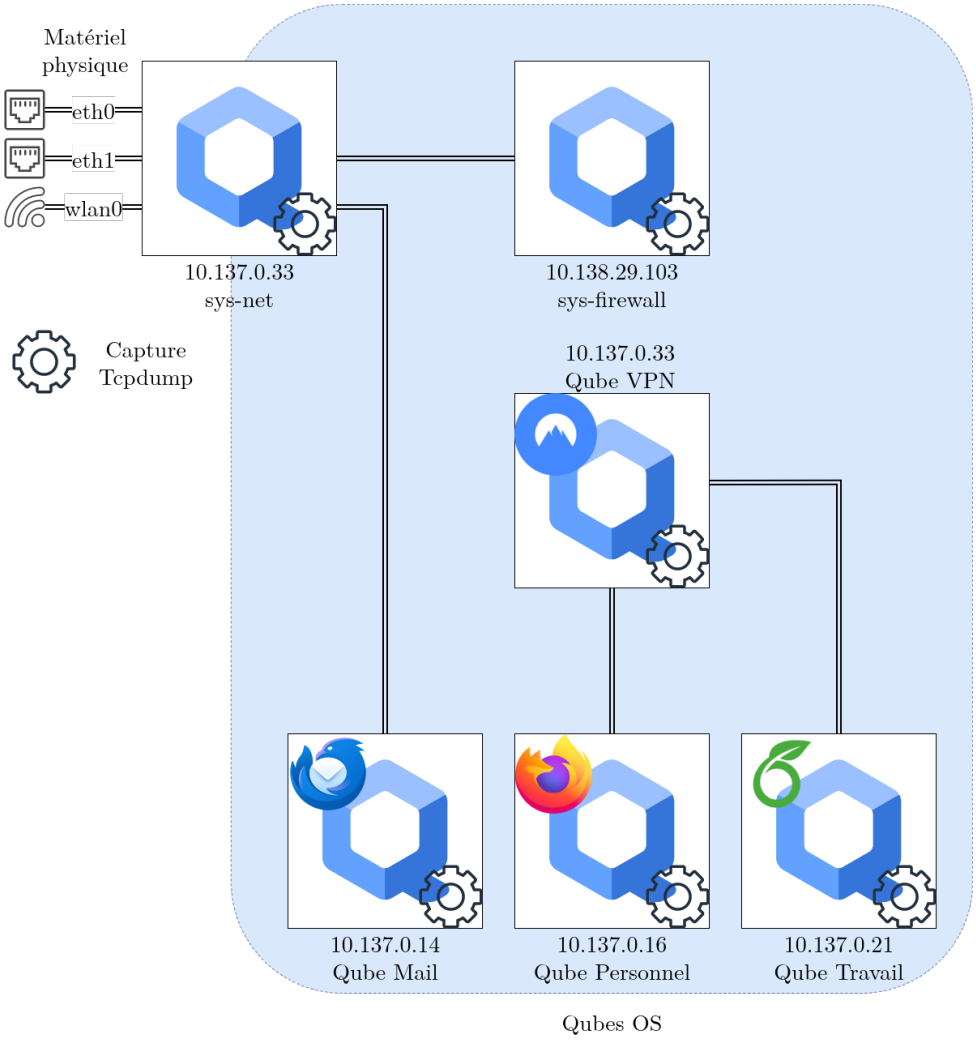



FIGURE 3.8 – Enregistrement et étiquetage des applications sous QubesOS

L'un des avantages majeurs de Qubes OS pour l'enregistrement de scénarios est sa capacité à gérer plusieurs applications simultanément, chacune fonctionnant dans son propre qube, tout en maintenant une séparation claire et sécurisée des flux de données. Cette isolation des applications permet non seulement d'assurer une sécurité accrue, mais aussi de faciliter l'enregistrement des échanges réseau associés à chaque application de manière indépendante.

Grâce aux NetVM, qui centralisent le trafic réseau avant de le distribuer aux autres qubes, il devient particulièrement simple d'intercepter et d'analyser les échanges réseau sans interférence entre les différents qubes. En capturant les flux directement dans NetVM, il est possible de reconstituer précisément les interactions réseau associées à chaque application, garantissant ainsi un suivi précis et une meilleure évaluation des communications.

Cependant, bien que cette isolation soit idéale pour l'enregistrement des activités réseau de manière précise, elle ajoute également une complexité en termes de gestion de l'enregistrement. Chaque qube génère son propre flux réseau, nécessitant des outils spécifiques pour capturer et analyser les données de manière cohérente et synchronisée, tout en maintenant l'intégrité et la confidentialité des informations.

**Comparaison des Approches**

La table 3.7 présentée compare différentes approches pour l'enregistrement et le rejeu de scénarios réseau en fonction de plusieurs critères clés. La méthode d'enregistrement sur une machine unique est la plus simple à mettre en œuvre, nécessitant uniquement des outils de capture. Cependant, elle offre un contrôle limité sur les applications enregistrées et complique l'étiquetage des activités en raison du bruit réseau et des processus en arrière-plan.

Les machines virtuelles (VMs) offrent une isolation complète en embarquant un système d'exploitation complet distinct de celui de l'hôte. Cela permet un réalisme élevé des scénarios et un contrôle total sur l'environnement d'exécution, ce qui est particulièrement utile pour simuler des postes utilisateurs ou des systèmes industriels complexes. Cependant, cette approche est plus gourmande en ressources et plus complexe à configurer, car elle nécessite le choix d'un système d'exploitation, son installation complète, ainsi que la configuration et l'installation manuelle de tous les logiciels nécessaires.

À l'inverse, les conteneurs sont plus légers et plus rapides à déployer, car ils partagent le noyau du système hôte. Cette architecture rend leur démarrage quasi instantané et simplifie la gestion des dépendances via des fichiers de configuration comme les Dockerfiles. Toutefois, cette légèreté se fait au prix d'une isolation moindre : contrairement aux VMs, les conteneurs n'isolent pas totalement les processus du système hôte, ce qui peut poser des limites en matière de sécurité ou d'émulation fidèle de comportements utilisateurs réels.

Enfin, Qubes OS propose une isolation unique en virtualisant chaque application ou groupe d'applications associé à une activité, ce qui facilite l'enregistrement simultané de plusieurs flux tout en offrant une excellente capacité d'étiquetage. Toutefois, son utilisation demande une configuration initiale plus complexe qu'un OS conventionnel en raison de sa compatibilité et gestion du matériel. Le tableau 3.7 met en évidence les avantages et les inconvénients de chaque approche, aidant à choisir la méthode la plus adaptée selon les besoins et les contraintes spécifiques.

| **Critères** | **Machine Unique** | **Machines Virtuelles (VMs)** | **Conteneurs** | **Qubes OS** |
|---|---|---|---|---|
| Facilité de mise en œuvre | Très simple | Moyenne | Moyenne | Complexe |
| Consommation de ressources | Faible | Élevée | Faible à modérée | Élevée |
| Contrôle des applications | Faible | Élevé | Moyenne | Très élevé |
| Réalisme des scénarios | Moyen | Élevé | Moyen | Élevé |
| Facilité d'étiquetage | Faible | Moyenne | Moyenne | Très élevée |
| Isolation des activités | Aucune | Complète | Partielle | Complète |
| Adaptabilité à des scénarios complexes | Faible | Élevée | Élevée | Moyenne |
| Coût de mise en œuvre | Faible | Élevé | Faible | Modéré |
| Repeatability | Faible | Moyenne | Élevée | Très élevée |
| Reproducibility | Faible | Moyenne | Élevée | Très élevée |
| Replicability | Très faible | Faible | Moyenne | Élevée |
| Reusability | Faible | Élevée | Très élevée | Élevée |

TABLE 3.7 – Comparaison des méthodes d'enregistrement et de rejeu des scénarios

### 3.3.2 BASIL : Behavioral Action Scenario with Image Labelling



BASIL (Behavioral Action Scenario with Image Labelling) est un logiciel conçu pour enregistrer et rejouer fidèlement les interactions utilisateur sur un ordinateur dans le but de générer du trafic réseau réaliste. Il a été développé pour répondre au besoin d'un outil capable de simuler des comportements utilisateurs complexes tout en capturant leur contexte d'exécution, afin de générer des données annotées précises et exploitables pour l'analyse réseau. Contrairement aux solutions existantes, souvent limitées à des scripts d'automatisation simples ou à des tests d'interface graphique, BASIL offre une approche complète centrée sur la reconstitution fidèle d'un scénario humain. Il capture les entrées d'une machine (actions clavier et souris) en associant chaque clique à une capture d'écran de la zone concernée, créant ainsi un fichier de scénario détaillant les actions, leur chronologie et les zones interactives. En s'appuyant sur la reconnaissance d'image, BASIL identifie dynamiquement les zones de l'écran lors de la relecture

et exécute les actions correspondantes via le script du scénario précédemment généré. Cet outil offre une solution puissante pour rejouer un comportement utilisateur réel et faciliter l'analyse liée au trafic réseau, le tout à partir d'une application WEB, le rendant facilement accessible à partir de n'importe quel navigateur (voir figure C.3 en annexe). Une comparaison avec des outils similaires tels que SikuliX et PyAutoGUI est présentée à la fin de cette section afin de situer BASIL dans son écosystème fonctionnel et de mettre en évidence ses spécificités.

### Architecture

Le logiciel BASIL permet d'enregistrer, modifier et rejouer des scénarios d'automatisation, comme illustré sur la figure 3.9. Il dispose d'une interface simple et accessible depuis un navigateur, facilitant son utilisation sur différents supports.

En arrière-plan, un programme central gère l'organisation des fichiers et l'exécution des tâches. L'échange d'informations entre l'interface et ce programme est fluide, permettant une mise à jour en temps réel lors de l'enregistrement et de la relecture des scénarios.

Pour capturer les actions de l'utilisateur et le trafic réseau, BASIL utilise des outils spécialisés, notamment un logiciel d'analyse réseau [9], ainsi que des outils permettant d'enregistrer les mouvements de la souris et du clavier. Lors du rejeu des scénarios, seul l'enregistrement des actions est nécessaire pour reproduire fidèlement ce qui a été fait précédemment.

Le module création/édition facilite la modification des scripts .basil grâce à une interface intuitive proposant des commandes préconfigurées. Enfin, tous les fichiers sont stockés localement sur la machine, assurant une gestion autonome des scénarios au sein de l'application.

### Langage d'actions

Le logiciel utilise un langage simple et intuitif conçu pour définir des scénarios d'interaction utilisateur. Ce langage repose sur des commandes clés associées à des paramètres qui spécifient les actions à réaliser. Chaque commande est un mot-clé qui décrit une action élémentaire, comme cliquer, écrire, ou presser une touche. Les paramètres permettent de détailler l'action, qu'il s'agisse d'un chemin d'image pour identifier une zone cliquable, d'une chaîne de texte à saisir, ou d'une touche ou combinaison de touches à presser. Cette structure claire et lisible facilite la création, la modification et la lecture des scénarios, tout en permettant un contrôle précis des actions. La table 3.8 répertorie les commandes existantes.

Par exemple, le programme ci-dessous ouvre un nouvel onglet en cliquant sur le bouton "+" du navigateur, clique sur la case recherche et y écrit "Vol Brest-Paris". Une fois la touche entrée actionnée, le programme attend deux secondes, avant de cliquer sur un lien de résultat et faire défiler vers le bas.

9. Comme Wireshark, TCPDump ou Tshark selon les besoins.

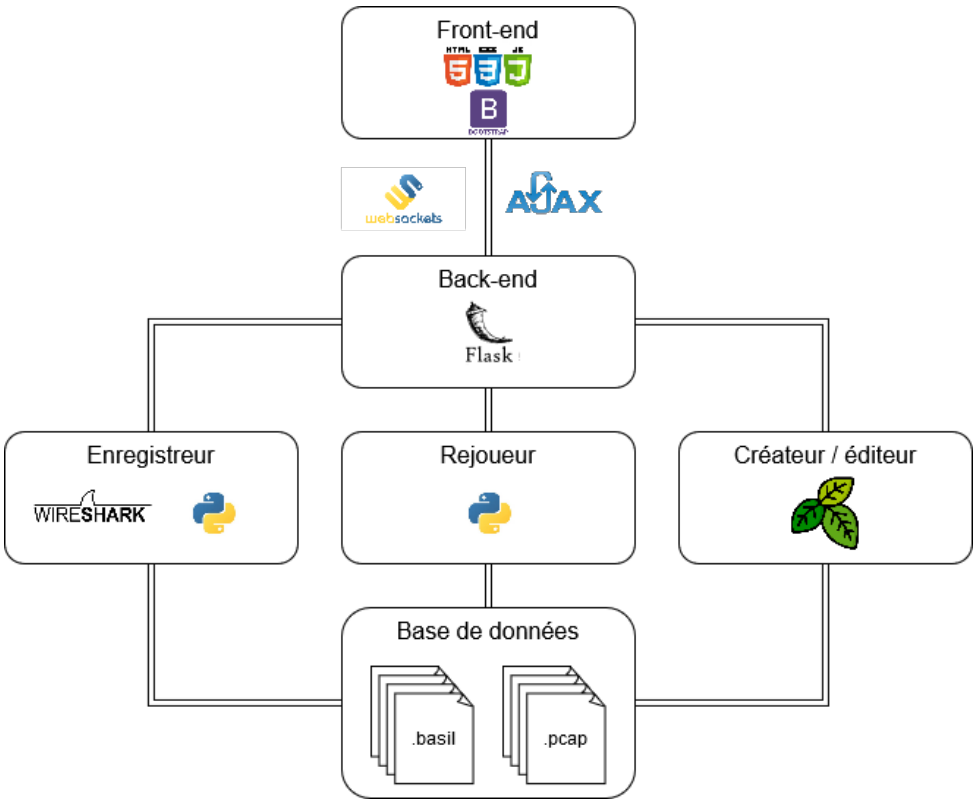



FIGURE 3.9 – Architecture du logiciel BASIL

```
CLICK new_tab.png
CLICK search.png
WRITE Vol Brest-Paris
PRESS enter
WAIT 2
CLICK result_link.png
SCROLL down 2
```

Grâce à cette structure simple, les utilisateurs peuvent rapidement concevoir des scénarios complexes tout en conservant une lisibilité optimale. De plus, la syntaxe se veut proche d'autres langages connus comme par exemple pour le Rubber Ducky [10], une clé USB permettant de lancer une charge automatiquement en se faisant passer pour un clavier (Vulnérabilité BadUSB([110]).

10. https ://payloadstudio.com/community/

| Commande | Paramètre attendu | Description |
|---|---|---|
| **CLICK** | `image` | Simule un clic au centre de la zone correspondant à l'image spécifiée (possibilité de simuler le déplacement jusqu'au clique). |
| **WRITE** | `texte` | Simule la saisie d'une chaîne de caractères (possibilité de simuler la durée entre chaque touche). |
| **PRESS** | `touche` ou `combinaison de touches` | Simule l'appui sur une touche ou une combinaison de touches (e.g., `ctrl+c`). |
| **WAIT** | `durée` (en secondes) | Mets en pause le scénario pendant la durée spécifiée. |
| **MOVE** | `image` | Déplace le curseur vers la zone correspondant à l'image spécifiée sans cliquer. |
| **SCROLL** | `up/down` et `intensité` | Effectue un défilement vers le haut (`up`) ou le bas (`down`) avec une intensité définie. |

Table 3.8 – Résumé des commandes du langage utilisé dans le logiciel.

#### Fonctionnement

Le fonctionnement du logiciel est conçu pour être simple et intuitif, permettant à l'utilisateur de se concentrer sur ses activités tout en enregistrant chaque interaction. Il faut tout d'abord lancer l'enregistrement avant de commencer à effectuer les actions souhaitées. Que ce soit en cliquant sur des éléments à l'écran, en saisissant du texte ou en pressant des touches, chaque interaction est enregistrée automatiquement. Une fois l'enregistrement terminé, l'appui sur le bouton "stop" génère en sortie l'ensemble des captures d'écran des actions effectuées, l'enregistrement du trafic réseau (PCAP), ainsi que le script du scénario (.basil).

Pour aller plus loin, le script généré peut être modifié à l'aide de l'éditeur intégré, permettant d'ajouter, modifier ou supprimer des actions selon les besoins. Cela offre une grande flexibilité pour ajuster les scénarios avant de les rejouer. En outre, il est possible de convertir le fichier PCAP en flux (flow) et d'associer les actions aux événements enregistrés, ce qui permet de mieux comprendre le lien entre les actions utilisateur et les échanges réseau, facilitant ainsi l'analyse et le traitement des données collectées.

#### Comparaison

Dans le cadre du développement de BASIL, un logiciel permettant d'enregistrer et de rejouer des scénarios utilisateur via la reconnaissance d'image et l'automatisation des entrées clavier/-

souris, il est essentiel de le comparer aux solutions existantes afin d'évaluer ses forces et ses éventuelles limites. Plusieurs outils, tels que SikuliX et PyAutoGUI, offrent des fonctionnalités similaires en matière d'automatisation et d'interaction avec l'interface graphique. Cette comparaison vise à analyser leurs capacités respectives en termes d'enregistrement automatique des actions utilisateur, de rejeu des scénarios, de reconnaissance d'image, et de flexibilité d'usage. L'objectif est de déterminer dans quelle mesure BASIL se distingue de ces alternatives et s'il apporte une valeur ajoutée unique, notamment dans le cadre de la simulation d'activités utilisateur pour la génération de trafic réseau. La comparaison est disponible dans la table 3.9.

| **Critère** | **BASIL** | **SikuliX** | **PyAutoGUI** |
|---|---|---|---|
| **Enregistrement** | Automatique | Script | Script |
| **Reconnaissance** | Image | Image | Image |
| **OCR** | Non | Oui | Non |
| **Clavier/Souris** | Oui | Oui | Oui |
| **Multi-OS** | Oui | Oui | Oui |
| **Facilité** | Facile | Facile | Moyenne |
| **Personnalisation** | Élevée | Bonne | Simple |
| **Usage** | Simulation | Tests UI | Automatisation |

TABLE 3.9 – Comparaison entre BASIL, SikuliX et PyAutoGUI

## 3.4 Conclusion

La génération et l'étiquetage de données réseau représentent un défi fondamental pour la recherche et l'analyse du trafic réseau. Les jeux de données disponibles souffrent de plusieurs limitations, notamment un manque de diversité, une focalisation excessive sur les attaques et une production souvent artificielle ne reflétant pas fidèlement les activités réelles des utilisateurs. Ces contraintes freinent l'évolution des modèles d'apprentissage et compromettent la pertinence des analyses réalisées à partir de ces sources.

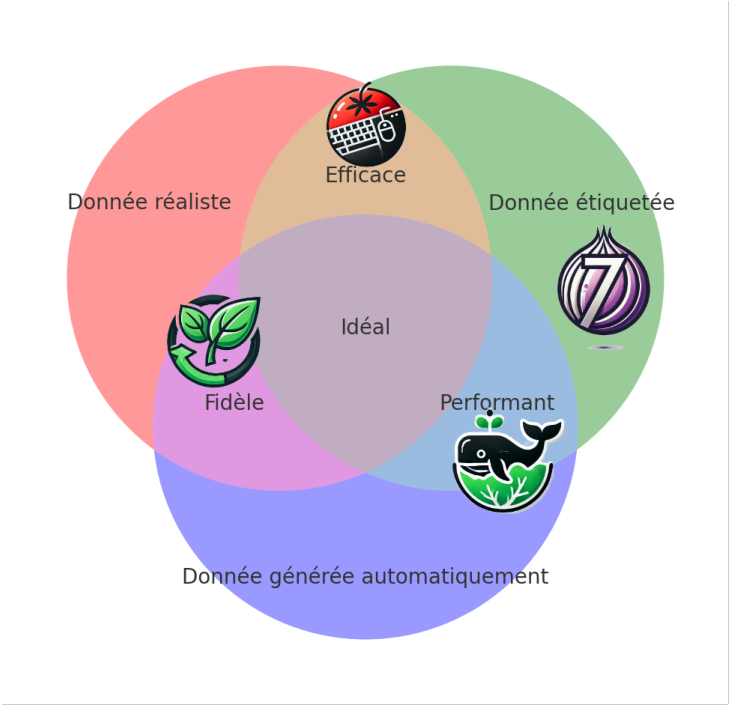



FIGURE 3.10 – Positionnement des outils développés vis-à-vis de la donnée

Face à ces enjeux, les outils développés dans ce chapitre offrent des solutions complémentaires pour surmonter ces limites. L'enregistrement et le rejeu de scénarios apparaissent comme des approches essentielles pour capturer des comportements authentiques et reproductibles. Différentes stratégies, comme l'enregistrement global, la virtualisation ou encore l'usage de systèmes cloisonnés tels que Qubes OS, permettent d'adapter la collecte de données aux contraintes spécifiques des études menées.

La figure 3.10 place les différents logiciels dans le diagramme de Venn exploré en début de section vis-à-vis des exigences de la donnée.

SALAD, en exploitant des environnements virtualisés et une approche scriptée, simplifie considérablement la génération et l'étiquetage des jeux de données. Son architecture permet

de structurer des scénarios d'exécution contrôlés tout en garantissant la reproductibilité et la précision des annotations. TOMATO, quant à lui, s'inscrit dans une démarche d'assistance active à l'étiquetage, en synchronisant différentes sources d'information (flux réseau, actions utilisateur, captures d'écran) afin de faciliter l'identification et la classification des données collectées. En offrant des suggestions contextuelles, il accélère le processus tout en limitant les erreurs inhérentes à une automatisation brute. BASIL se distingue en proposant une approche centrée sur le comportement utilisateur. En associant les entrées clavier et souris à des captures d'écran et en permettant la relecture des scénarios grâce à la reconnaissance d'image, cet outil garantit une restitution fidèle des interactions. Cette approche est particulièrement précieuse pour la génération de trafic réseau réaliste, où la dimension comportementale joue un rôle clé dans l'analyse des usages et la détection d'anomalies. Enfin, GAR7IC simplifie l'extraction des communications S7COMM à partir d'une capture réseau en les structurant sous forme de flows spécifiques. Cette mise en forme optimisée améliore leur traitement et facilite leur classification, permettant une analyse plus efficace des échanges dans un environnement industriel.

Chaque outil présenté dans ce chapitre répond à un besoin spécifique au sein de la chaîne de traitement menant à l'intégration d'un modèle de classification dans un switch intelligent. Par exemple, GAR7IC est directement exploitable dans un contexte industriel : il permet d'extraire et de structurer les communications S7COMM sous forme de flows normalisés, à la fois pour entraîner les modèles de classification, mais aussi pour prétraiter en temps réel les données entrantes au niveau du switch, assurant ainsi une compatibilité directe avec les algorithmes embarqués.

De leur côté, TOMATO et SALAD sont plus adaptés à la phase de génération de jeux de données étiquetées, nécessaires pour l'apprentissage supervisé. Ils permettent de créer des datasets réalistes, reproductibles et annotés manuellement ou automatiquement, utilisables pour entraîner des modèles sur mesure, adaptés à un réseau ou une politique de sécurité spécifique. Ces outils peuvent être utilisés ponctuellement lors du déploiement initial d'un système, mais également pour alimenter une base de données communautaire, afin de mutualiser l'effort d'étiquetage sur des types d'activités variées (web, messagerie, etc.).

Enfin, BASIL peut intervenir en amont, dans un scénario collaboratif de création de modèles personnalisés. Il pourrait être déployé temporairement sur les postes de travail de certains utilisateurs d'une entreprise, afin d'enregistrer leurs journées types. Ces scripts scénarisés, une fois récupérés avec les captures réseau et les spécificités de leur environnement, peuvent être rejoués en interne pour reproduire fidèlement leur comportement. Cela permet d'entraîner des modèles par profil ou par machine, augmentant la précision du filtrage comportemental directement intégré dans le switch.

Chapitre 4

# Analyses et applications

## Sommaire


**4.1 Proposition d'une solution intégrée au Switch . . . . . 131**
4.1.1 Contexte et motivation . . . . . 131
4.1.2 Objectifs de la solution proposée . . . . . 132
4.1.3 Architecture fonctionnelle du switch "intelligent" . . . . . 133
4.1.4 Administration et supervision au sein du switch . . . . . 135
**4.2 Analyse du trafic classique . . . . . 137**
4.2.1 Comparaison des stratégies d'étiquetage : manuel vs automatique . . . . 138
4.2.2 Extraction des caractéristiques . . . . . 140
4.2.3 Premières analyses et résultats . . . . . 143
4.2.4 La particularité du trafic WEB . . . . . 145
**4.3 Analyse du trafic chiffré par VPN . . . . . 147**
4.3.1 Principes d'un flow VPN . . . . . 148
4.3.2 Analyse de trafic Wireguard . . . . . 152
**4.4 Analyse du trafic industriel . . . . . 155**
4.4.1 Caractéristiques d'un flow industriel . . . . . 156
4.4.2 Analyse de trafic industriel S7COMM . . . . . 158
**4.5 Conclusion . . . . . 160**


Comme expliqué dans le chapitre 1 section 1.5, les jeux de données communément utilisés dans les travaux de recherche en cybersécurité présentent plusieurs limites : ils sont souvent datés, artificiellement construits, et centrés sur des scénarios d'attaque. Très peu d'entre eux représentent fidèlement des usages normaux, quotidiens ou industriels, ce qui limite leur pertinence pour entraîner des modèles capables de fonctionner dans des environnements réels. Or, pour garantir des performances fiables en détection ou classification, il est essentiel de disposer de données représentatives des comportements légitimes.

Ce chapitre s'inscrit dans une logique d'application concrète des outils développés dans les chapitres précédents. Dans cette démarche, deux jeux de données ont été constitués, représentant des activités courantes et permettant d'observer des comportements réseau typiques tout en testant différentes approches de classification.

La première section (4.1) introduit la solution technique développée pour embarquer des capacités de capture, d'analyse et de classification directement dans un commutateur réseau, en s'appuyant sur l'infrastructure d'Interface Concept. Les sections suivantes déclinent cette approche selon trois types de trafic : le trafic classique (section 4.2), issu d'usages bureautiques ou personnels comme la navigation web ou le streaming ; le trafic chiffré via VPN (section 4.3), pour lequel des méthodes de simulation et d'extraction adaptées sont proposées ; et enfin le trafic industriel (section 4.4), avec un format de flow spécifique centré sur les opérations de lecture/écriture échangées entre automates.

L'ensemble de ces analyses permet de valider la pertinence de l'approche de classification par flow dans différents contextes (IT, OT, chiffré), et d'alimenter les modèles embarqués dans le switch. Les travaux abordés répondent ainsi à plusieurs des questions de recherche de la thèse (cf. Questions de recherche) :

— **QR2 :** Comment détecter et classifier les activités observées dans un réseau informatique ?

— **QR4 :** Comment intégrer un système d'analyse de trafic au sein d'un switch ou d'une architecture réseau ?

Enfin, les résultats obtenus permettent d'estimer la faisabilité d'un traitement en temps réel embarqué, tout en posant les bases d'une supervision intelligente distribuée. Ces conclusions seront approfondies dans la dernière partie du chapitre, avant d'ouvrir sur les perspectives futures du projet.

## 4.1 Proposition d'une solution intégrée au Switch

L'objectif de cette section est de présenter la solution envisagée : un switch dit "intelligent", capable de réaliser en local une analyse du trafic transitant, d'en extraire des caractéristiques pertinentes, et de classifier les flux en fonction de leur nature (ex. : technique, professionnel, personnel, web, etc.). Ce prétraitement permettrait de rediriger les flux vers des modules spécialisés (systèmes de détection, enregistreurs, analyseurs), optimisant ainsi les ressources globales du système de supervision tout en améliorant la pertinence des alertes.

Nous présentons dans un premier temps le contexte et les motivations ayant conduit à cette proposition, avant de détailler les objectifs spécifiques visés par cette solution. Nous décrivons ensuite l'architecture fonctionnelle du switch intelligent ainsi que ses principaux composants.

### 4.1.1 Contexte et motivation

Dans le contexte industriel, les équipements réseau tels que les commutateurs jouent un rôle central, tout le trafic transitant nécessairement par eux. Cette position stratégique en fait un point d'observation idéal pour analyser et superviser les échanges en temps réel.

Interface Concept, entreprise spécialisée dans les cartes et systèmes embarqués industriels, développe notamment la gamme de commutateurs ComEth. Ces équipements, modulaires et performants, offrent la possibilité d'ajouter des fonctions avancées via des extensions matérielles. Cette flexibilité ouvre la voie à l'intégration de capacités de traitement local dédiées à la détection et à la classification du trafic.

L'objectif de cette thèse est de tirer parti de ces possibilités pour proposer une solution embarquée directement dans le commutateur : un prototype capable de surveiller le trafic, de l'analyser localement à l'aide de méthodes statistiques ou d'apprentissage, et de rediriger intelligemment les flux vers des systèmes de sécurité spécialisés. Cette approche vise à alléger les dispositifs centraux tout en améliorant la réactivité et la précision de la détection dans les réseaux industriels.

Un commutateur « intelligent » présente plusieurs avantages dans un réseau industriel. C'est un point de contrôle idéal pour intégrer des capacités de supervision et de préanalyse. En réalisant localement des tâches de capture, de classification ou de redirection, le switch peut soulager les systèmes centraux (comme les IDS ou les SIEM), qui sont souvent saturés et peu spécialisés.

De plus, ce traitement en amont permet d'orienter les flux de manière plus fine vers les outils appropriés, en fonction de leur nature. Par exemple, un trafic critique ou inconnu peut être dupliqué et envoyé vers un IDS spécialisé, tandis qu'un trafic reconnu et bénin est simplement enregistré ou ignoré.

Ce rôle de filtrage intelligent peut être comparé au fonctionnement hiérarchisé d'un SOC (Security Operations Center). À l'image des niveaux N1/N2/N3, le switch intelligent agit comme

un premier niveau d'analyse (N1) capable de trier les alertes et les flux pertinents, en ne transmettant aux niveaux supérieurs que les événements jugés critiques ou ambigus. Cela permet de réduire significativement la charge de travail des analystes de niveaux N2 et N3, qui peuvent alors se concentrer sur l'investigation approfondie et la réponse aux incidents les plus sensibles. Le switch joue ainsi un rôle de sentinelle embarquée, apportant une couche d'intelligence et de filtrage dès les premières couches du réseau.

Enfin, cette approche permet une adaptation par port, en exploitant la régularité des usages machine par machine. Chaque port peut embarquer un modèle personnalisé, entraîné spécifiquement pour les comportements observés sur la machine qui y est connectée. Cela augmente fortement la précision et réduit les faux positifs.

### 4.1.2 Objectifs de la solution proposée

La solution envisagée vise à intégrer des capacités de supervision, de détection et de classification directement au sein du commutateur réseau, afin d'exploiter sa position centrale dans l'architecture industrielle.

Les objectifs principaux sont les suivants :

— **Détection et classification locale du trafic** : analyser les flux en temps réel pour identifier leur nature (trafic technique, professionnel, personnel...) à l'aide de méthodes statistiques ou d'apprentissage automatique embarquées.

— **Allègement des systèmes centraux** : effectuer un premier niveau d'analyse directement dans le switch afin de réduire la charge des systèmes de détection classiques (IDS, SIEM...).

— **Redirection intelligente des flux** : router les flux en fonction de leur classification vers les outils les plus adaptés pour une analyse approfondie ou un enregistrement.

— **Adaptabilité aux contraintes industrielles** : respecter les contraintes spécifiques des environnements industriels, notamment en matière de latence, de disponibilité, et de faible taux de faux positifs.

— **Intégration transparente dans les architectures existantes** : permettre un déploiement progressif sans modifier profondément l'infrastructure réseau en place.

Cette approche vise à renforcer la sécurité du réseau en apportant une couche d'intelligence dès les premières couches du modèle OSI, tout en conservant une compatibilité avec les systèmes de supervision existants.

Le switch intelligent s'intègre naturellement dans une architecture de supervision en tant que point de préanalyse et de filtrage. Il ne remplace pas les outils existants (IDS, SIEM, enregistreurs...), mais agit en amont pour améliorer leur efficacité.

Il peut notamment :

— Filtrer et préclassifier les flux avant qu'ils n'atteignent les IDS, réduisant ainsi leur charge.

— Générer des alertes simples localement, transmises ensuite au SIEM.

— Produire des logs condensés exploitables par des outils de corrélation ou de forensique.

— Transmettre des flux spécifiques vers des outils spécialisés en fonction de leur type (ex. : envoyer les flux industriels vers un IDS OT, et les flux VPN vers un IDS réseau classique).

— Être utilisé comme point d'apprentissage distribué, grâce aux clients installés sur les machines, qui permettent de générer des modèles adaptés à l'usage réel.

Ce fonctionnement modulaire et distribué s'intègre parfaitement dans une démarche de défense en profondeur, en ajoutant une couche d'intelligence dès les premières couches réseau.

### 4.1.3 Architecture fonctionnelle du switch "intelligent"

La solution proposée repose sur une architecture modulaire embarquée dans un commutateur réseau, enrichie par une carte d'extension dédiée. Cette architecture permet d'analyser le trafic en temps réel, de le classifier selon sa nature, puis de le rediriger intelligemment vers des interfaces ou des systèmes d'analyse adaptés. La figure 4.1 montre les différents composants identifiés par des chiffres dont voici le rôle et leur description :

**1. Capture et d'extraction de caractéristiques** Le trafic réseau transitant par le commutateur est intercepté via une opération de mirroring (sans impact sur le service), puis traité par un premier module chargé d'extraire des caractéristiques simples et rapides à calculer. Cette approche passive, similaire à celle utilisée par les IDS classiques, permet de réaliser une analyse et une classification en parallèle, sans perturber le fonctionnement normal du réseau. Ces caractéristiques incluent, par exemple, le protocole réseau utilisé, les ports source et destination, ou encore des statistiques élémentaires telles que la durée ou le débit du flux. Cette extraction rapide permet une première orientation du trafic sans alourdir le traitement.

**2. Orientation vers un modèle spécialisé** Sur la base des premières caractéristiques extraites, le système oriente chaque flux vers l'un des modèles de classification spécialisés. Trois grandes familles sont actuellement envisagées : un modèle pour le trafic VPN, un autre pour le trafic industriel, et un modèle dit « classique » pour les usages bureautiques ou génériques. Cette orientation initiale permet de traiter chaque flux avec un modèle adapté à son contexte, afin d'en améliorer la pertinence et la performance de classification. En fonction du modèle sélectionné, une agrégation spécifique des flows est ensuite réalisée. Celle-ci permet d'adapter la granularité de l'analyse au type de trafic considéré — qu'il s'agisse, par exemple, d'un découpage temporel pour le trafic VPN ou d'une reconstruction d'opérations unitaires dans le cas du trafic industriel. Ces mécanismes d'agrégation seront détaillés plus loin dans ce chapitre. Il est prévu de pouvoir ajouter d'autres modèles au besoin, selon les cas d'usage. Parmi les modèles

de classification envisagés, l'algorithme Random Forest présente un compromis particulièrement adapté au contexte réseau. Robuste face au bruit, efficace sur des données tabulaires comme les statistiques de flux, il offre de bonnes performances tout en restant simple à entraîner et léger à exécuter. Cette légèreté permet de l'embarquer facilement dans un environnement contraint comme un switch, où les ressources sont limitées. De plus, il conserve une certaine interprétabilité, ce qui facilite l'analyse des décisions prises et l'ajustement des modèles en fonction des usages réels observés sur chaque port du commutateur.

Plusieurs travaux académiques viennent conforter ce choix. L'étude pForest, est une implémentation matérielle de Random Forest dans des commutateurs programmables, notamment sur des architectures P4 (un langage de programmation pour le data plane réseau), démontre la possibilité d'exécuter des modèles Random Forest directement dans des commutateurs programmables, avec des performances compatibles avec une inférence en ligne et en environnement embarqué [111]. Dans un contexte de détection d'intrusions, Random Forest a également montré d'excellents résultats, atteignant jusqu'à 98% de précision sur des jeux de données comme CICIDS2017 [112]. Ce modèle se distingue en particulier par sa robustesse face aux données bruitées et aux distributions déséquilibrées. L'intégration d'approches comme ADASYN pour le rééquilibrage des classes a permis de renforcer encore ses performances dans des contextes réalistes de classification réseau [113].

Ces qualités — robustesse, légèreté et capacité d'interprétation — font de Random Forest un excellent candidat pour une intégration embarquée dans un switch intelligent. En fonction du type de modèle activé (trafic VPN, industriel ou classique), une agrégation spécifique des flows est par ailleurs appliquée, comme cela sera détaillé dans la suite du chapitre.

**3. Classification spécialisées** Chaque modèle reçoit les flux qui lui sont attribués et procède à une classification plus fine. Ces modules peuvent reposer sur des règles simples ou des algorithmes d'apprentissage automatique adaptés aux contraintes embarquées. Ils sont chargés d'identifier le type d'activité (téléchargement, streaming, contrôle industriel, etc.) ou le niveau de confiance du trafic. L'apprentissage de ces modèles, particulièrement coûteux en ressources, est déporté vers un serveur extérieur. Celui-ci reçoit des données d'entraînement labellisées, procède à l'entraînement, puis renvoie au commutateur une version optimisée et compacte du modèle, adaptée à une exécution embarquée rapide.

**4. Redirection intelligente** Une fois classifiés, les flux sont redirigés selon leur nature. Les flux jugés normaux ou bénins suivent leur chemin habituel, tandis que ceux considérés comme suspects ou inconnus peuvent être envoyés vers des systèmes spécialisés, tels que des IDS (Intrusion Detection Systems) ou des enregistreurs. Ce filtrage permet de soulager ces systèmes en ne leur transmettant que les données pertinentes.

**5. Interface de supervision et de gestion** Le système inclut une interface de communication avec un serveur de supervision. Celle-ci permet de remonter des statistiques, des décisions de classification, ou des alertes, mais également de recevoir des mises à jour de configuration ou de modèles. Cette interface assure ainsi la supervision et l'évolutivité de la solution.

**6. Intégration dans le réseau industriel** Le commutateur intelligent s'intègre sans rupture dans une architecture existante, notamment grâce à son emplacement permettant l'ajout de carte d'extension. Son déploiement peut se faire de manière progressive, sans modification de la topologie, facilitant ainsi les phases de test et l'acceptation par les exploitants.

**7. Client d'apprentissage local pour l'étiquetage personnalisé** Afin d'améliorer la qualité des données d'apprentissage, un client léger peut être installé sur certaines machines du réseau. Ce client analyse et étiquette localement le trafic réseau selon les activités effectuées (ex. : navigation web, mail, communication industrielle). Il transmet ensuite ces données labellisées au serveur d'apprentissage externe. Cette approche permet de créer des modèles adaptés aux usages réels, et surtout personnalisés à chaque port du commutateur. En effet, dans un environnement industriel, chaque machine — automate, poste de supervision, terminal utilisateur — présente un comportement réseau distinct. En associant un modèle de classification spécifique à chaque port, le système améliore sensiblement la précision tout en réduisant les faux positifs.

**8. Journalisation des flux classifiés** Enfin, les flux réseau classifiés sont archivés dans un format condensé, à un niveau plus abstrait que les paquets individuels ou les flows. Cette journalisation permet de conserver une trace des activités observées, tout en limitant l'espace de stockage nécessaire. Ces logs pourront ensuite être utilisés pour de futures analyses, requalifications ou comme base d'entraînement pour de nouveaux modèles.

### 4.1.4 Administration et supervision au sein du switch

Afin de faciliter le déploiement, la configuration et la supervision de la solution embarquée, une interface d'administration a été développée dans le but d'être intégrée directement au sein du commutateur. Cette interface permet aux administrateurs réseau de configurer facilement les ports, de sélectionner les modèles de classification à utiliser, de consulter les journaux d'activité, et d'assurer un suivi en temps réel du trafic.

**Vue d'ensemble de l'interface** La figure 4.2 montre un exemple de l'une des sections de l'interface d'administration intégrée au switch. Cette section permet de configurer individuellement chaque port, en lui assignant un rôle (machine industrielle, poste utilisateur, serveur, etc.) ainsi qu'un modèle de classification spécifique adapté au type de trafic attendu.

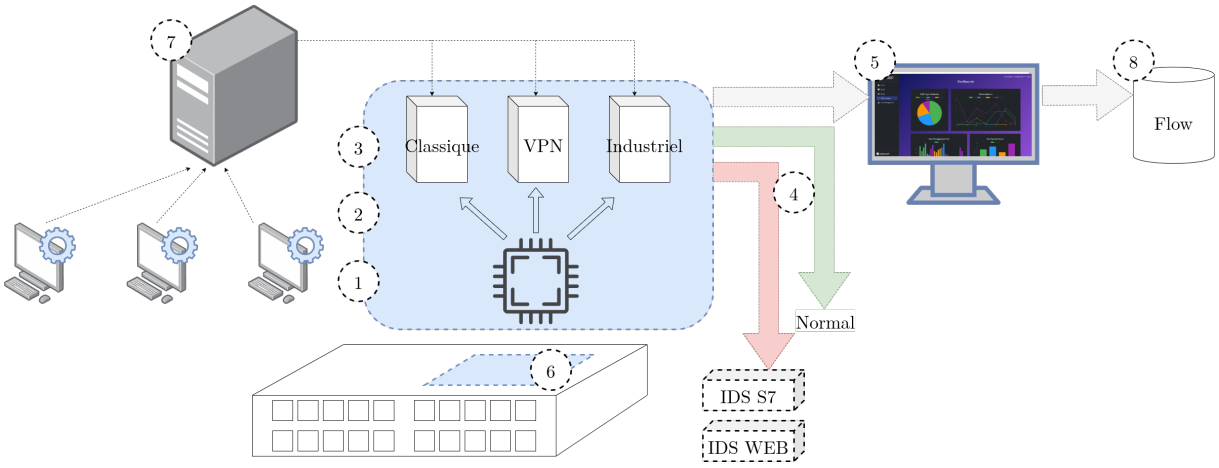



FIGURE 4.1 – Architecture fonctionnelle du switch intelligent

Au-delà de cette fonctionnalité, d'autres modules sont envisagés ou en cours de développement :

— **Configuration des ports** : permet d'assigner à chaque port un rôle particulier (trafic industriel, utilisateur, serveur, etc.) et d'associer un modèle de classification spécifique en fonction du type de trafic attendu.

— **Sélection des modèles** : chaque modèle pré-entraîné peut être activé ou désactivé dynamiquement selon les besoins. Il est également possible d'uploader un modèle personnalisé généré à partir de données locales.

— **Tableau de supervision** : affiche en temps réel les flux capturés, leur classification, le modèle utilisé, et les éventuelles redirections déclenchées. Des alertes simples peuvent également être remontées.

— **Journalisation et export** : propose l'accès à l'historique des classifications effectuées, avec possibilité d'export en CSV ou JSON pour un traitement ultérieur.

— **Mise à jour et gestion** : permet d'envoyer de nouveaux modèles, de modifier les paramètres de classification ou de redémarrer les services internes sans redémarrage complet du switch.

— **Déploiement d'agents** : un module dédié permettra de gérer le déploiement d'agents légers sur les machines du réseau. Ces agents auront pour rôle de capturer localement le trafic réseau, de l'étiqueter selon les activités observées, et de remonter les données vers le switch ou un serveur d'apprentissage. Cela permet de générer des modèles adaptés aux usages spécifiques de chaque poste ou équipement, tout en gardant une approche décentralisée et personnalisée de la supervision.

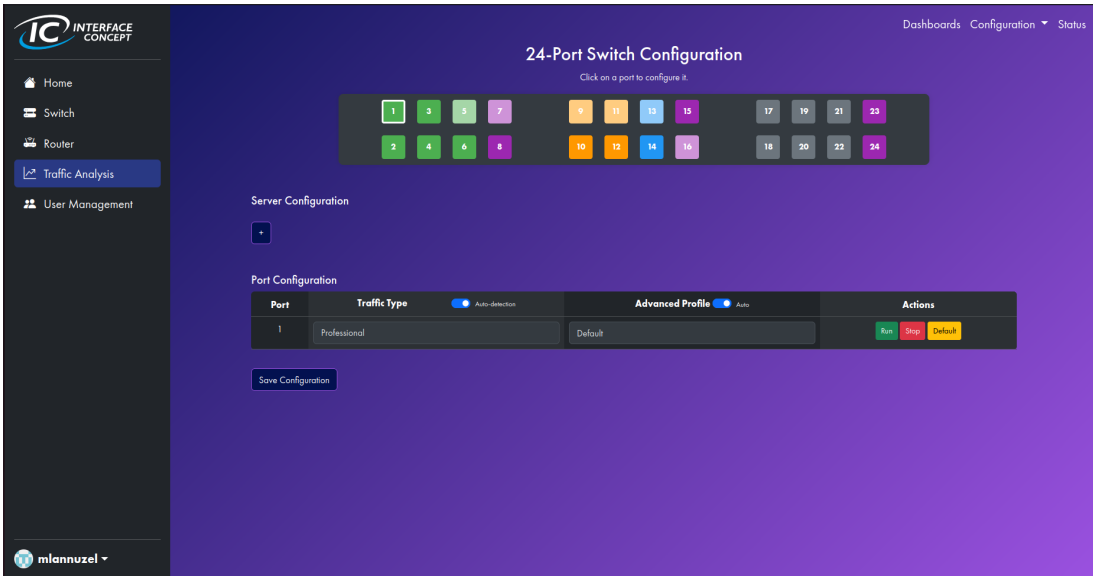



FIGURE 4.2 – Interface web d'administration intégrée au switch intelligent

**Supervision en temps réel** La figure 4.3 présente un exemple de tableau de bord intégré à l'interface d'administration. Ce tableau permet de suivre en temps réel les flux réseau, leur classification, le modèle utilisé ainsi que les décisions prises (redirection, journalisation, alerte). L'objectif est de fournir une visualisation centralisée de l'activité du switch, facilitant ainsi le travail des opérateurs.

Ce tableau n'est qu'un exemple : la solution a été pensée pour être extensible et personnalisable, à la manière d'un outil tel que Kibana [1]. Plusieurs tableaux de bord peuvent être configurés pour répondre à des besoins spécifiques (vue par port, par type de trafic, par modèle actif, etc.), offrant ainsi une souplesse d'usage et une visibilité complète depuis un seul et même point d'accès.

**Flexibilité et évolutivité** L'architecture retenue permet une grande modularité. Les modèles peuvent être mis à jour individuellement, et chaque port peut être géré indépendamment, ce qui permet une adaptation fine aux usages de chaque machine connectée. Ce fonctionnement distribué participe pleinement à la logique de défense en profondeur en apportant une supervision intelligente au plus près des équipements.

## 4.2 Analyse du trafic classique

Dans cette section, nous nous intéressons au trafic dit "classique", c'est-à-dire à l'ensemble des communications réseau issues d'activités humaines courantes sur un ordinateur. Il ne s'agit donc

1. https ://www.elastic.co/fr/kibana




FIGURE 4.3 – Supervision temps réel du trafic au sein du switch

pas de trafic industriel, automatisé ou propre à des équipements spécifiques, mais bien de trafic généré par des usages quotidiens : navigation web, envoi et réception de courriels, visionnage de vidéos en streaming, téléchargement de fichiers, utilisation de services cloud, etc. Ce trafic, que l'on pourrait qualifier de "grand public", est souvent produit par des utilisateurs dans le cadre d'un usage personnel ou professionnel standard. L'objectif ici n'est pas de s'appuyer uniquement sur des indicateurs techniques classiques comme les ports ou les protocoles, mais bien d'aller vers une classification plus fine et plus pertinente des activités réseau, en identifiant le type d'usage sous-jacent au trafic observé.

### 4.2.1 Comparaison des stratégies d'étiquetage : manuel vs automatique

Dans cette étude, il a été choisi de s'appuyer sur l'analyse de flows réseau. Il est rappelé qu'un flow (voir section 1.3.4) correspond à un ensemble de paquets partageant certaines caractéristiques communes — telles que les adresses IP source et destination, les ports et le protocole —, observés sur une période donnée (30s par défaut pour NFstream, voir section 1.2.3). Cette représentation permet de résumer efficacement l'activité réseau en évitant de conserver l'intégralité des paquets, trop volumineux et en grande partie inutiles pour l'analyse visée. Il est ainsi considéré que les flows offrent une abstraction pertinente, à la fois légère et synthétique, pour étudier le trafic. Plutôt que de traiter des données brutes complètes, dont une part importante ne serait pas exploitée, l'approche retenue consiste à extraire un sous-ensemble restreint, mais significatif de caractéristiques (comme la durée, le volume, le nombre de paquets ou les temps d'interarrivée) permettant une classification efficace des usages réseau.

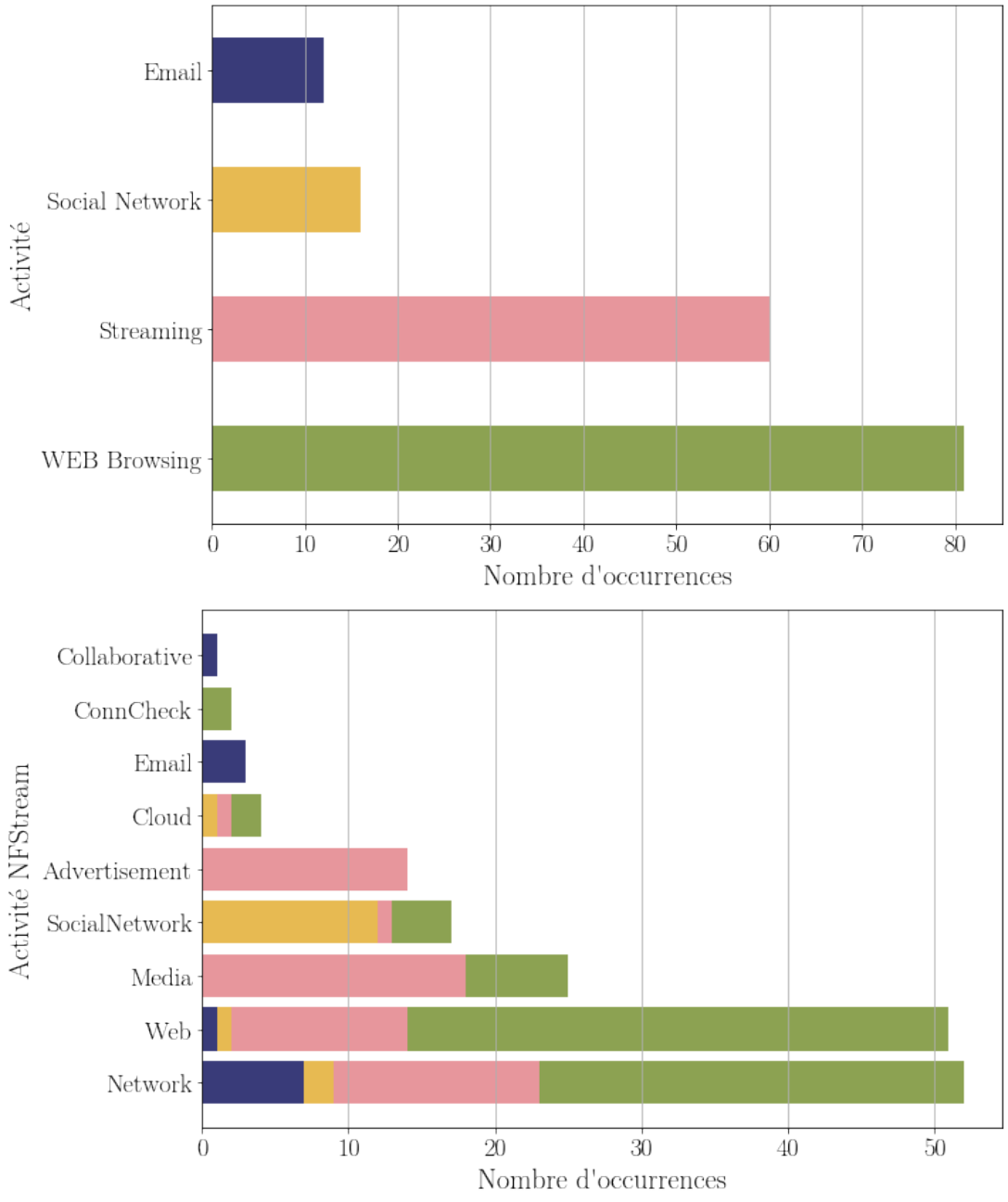



FIGURE 4.4 – Répartition et nombre d'occurences des étiquetages manuel et automatique.

Un enregistrement de courte durée, d'environ cinq minutes, a été réalisé afin de tester les capacités d'étiquetage offertes par la méthode et les outils développés dans le cadre de TOMATO. Durant cette capture, plusieurs activités courantes ont été effectuées : consultation de courriels, navigation web incluant l'usage d'un réseau social, et visionnage d'une vidéo sur YouTube. À l'issue de cette session, les flux ont été étiquetés manuellement selon cinq catégories générales : Email, Web Browsing, Social Network et Streaming. Ces étiquettes ont ensuite été comparées à celles attribuées automatiquement par l'outil NFStream [2]. Il a été constaté que ce dernier proposait non seulement un plus grand nombre de catégories, mais également des distinctions plus fines : Network, Email, Web, Cloud, ConnCheck, SocialNetwork, Media, Advertisement et Collaborative. La figure 4.4 montre les résultats.

Cette comparaison a permis de mettre en évidence les apports potentiels d'un étiquetage automatisé plus granulaire pour une classification fine des usages. En analysant le fonctionnement de la librairie, elle se base uniquement sur le protocole pour associer son étiquette. Toutefois, des niveaux de granularité plus poussés ne sont pas encore possibles avec NFStream à l'heure actuelle. Cela justifie le maintien de l'usage de l'outil TOMATO, qui permet une annotation plus ciblée et adaptée aux besoins spécifiques de classification fine.

Par exemple, lors de l'utilisation d'une plateforme comme Facebook, un utilisateur effectue en réalité plusieurs types d'actions : consulter du contenu, regarder des vidéos, discuter en messagerie instantanée, ou encore charger des éléments depuis le cloud. Un étiquetage manuel regroupera l'ensemble de ces activités sous une seule étiquette fonctionnelle cohérente comme "Réseaux sociaux", car il prend en compte l'intention globale de l'utilisateur.

En revanche, un outil comme NFStream segmente cette activité en plusieurs catégories techniques. Ainsi, une vidéo hébergée sur Facebook pourra être catégorisée comme "Streaming", une conversation Messenger comme "Chat", et un transfert de fichier comme "Cloud". NFStream attribue donc une catégorie à chaque flux réseau indépendamment du contexte global, ce qui peut donner une vision fragmentée d'une seule et même action utilisateur.

De plus, certains protocoles comme DNS, DHCP, ICMP, ARP, ou d'autres protocoles système essentiels sont automatiquement classés dans la catégorie "Network". Ces protocoles sont présents dans quasiment toutes les activités, car ils assurent la résolution de noms, l'attribution d'adresses IP, ou encore la gestion de la connectivité. Ainsi, la catégorie "Réseau" est omniprésente dans les traces, mais elle ne correspond pas à une activité fonctionnelle spécifique de l'utilisateur.

### 4.2.2 Extraction des caractéristiques

La sélection des caractéristiques constitue une étape déterminante en apprentissage automatique, dans la mesure où elle permet de réduire la dimensionnalité des données, de limiter le

2. Il est à noter que l'association de la catégorie est effectuée par la bibliothèque nDPI (https ://github.com/ntop/nDPI/tree/dev) dont fait appel NFstream.

surapprentissage et d'améliorer les performances des modèles. On distingue principalement trois grandes familles de méthodes : les méthodes de filtrage, les méthodes d'enveloppe (wrapper) et les méthodes intégrées (embedded). Les méthodes de filtrage reposent sur des critères statistiques indépendants du modèle, comme le test du Chi², l'information mutuelle ou l'ANOVA F-value. Les méthodes d'enveloppe testent différentes combinaisons de variables en fonction des performances obtenues avec un algorithme donné, comme dans la sélection récursive de variables (RFE). Enfin, les méthodes intégrées réalisent la sélection directement lors de l'entraînement du modèle, en exploitant les propriétés de certaines techniques comme la régularisation (Lasso) ou les forêts d'arbres. L'objectif de ces approches est d'identifier un sous-ensemble optimal de caractéristiques permettant de maximiser la qualité de la prédiction tout en conservant un modèle interprétable et efficace.

Dans le cadre de cette étude, les données utilisées sont majoritairement quantitatives. Elles se composent principalement de mesures statistiques extraites des flux réseau, telles que la durée du flux, le nombre de paquets échangés ou encore le volume de données transféré. Ces métriques permettent de décrire le comportement d'un flux de manière générique, sans dépendre du contenu ni du protocole spécifique. Ce choix de variables permet de travailler sur des données neutres vis-à-vis de l'application, tout en conservant une capacité de discrimination pertinente pour la classification du trafic réseau.

Une première approche heuristique a été explorée, qui repose sur une analyse exploratoire et une connaissance préalable du domaine afin d'identifier les attributs les plus pertinents. Cette méthode a ensuite été complétée par une sélection plus automatisée, reposant sur des techniques telles que l'analyse d'importance des variables ou la sélection récursive. L'objectif est de comparer les résultats obtenus par ces différentes approches afin de déterminer la stratégie la plus adaptée aux données étudiées.

À la suite d'analyses exploratoires, il a été décidé d'utiliser les caractéristiques *bidirectional_duration_ms*, *bidirectional_packets* et *bidirectional_bytes* pour mener une première phase de tests de classification du trafic réseau. Elles représentent respectivement, le temps total en milliseconde d'un flow, le nombre total de paquets échangés ainsi que sa taille en octet. Ce choix, effectué de manière heuristique, repose sur l'hypothèse que ces attributs offrent une représentation simple, mais pertinente de l'activité d'un flux. En effet, la durée d'un échange, le nombre de paquets transmis et le volume de données échangées sont apparus comme des indicateurs globaux capables de distinguer différentes typologies de trafic. Ces caractéristiques ont également été retenues pour leur disponibilité systématique, quel que soit le protocole ou le chiffrement, ce qui les rend intéressantes pour une première approche de caractérisation. Elles présentent en outre l'avantage de ne pas dépendre directement des ports, des adresses IP ou du protocole utilisés. Contrairement à certaines méthodes classiques de classification qui reposent sur ces éléments, ces caractéristiques restent stables même lorsque ces paramètres changent, ce

qui les rend plus robustes et plus fiables pour analyser du trafic dans des environnements variés ou peu contrôlés.

L'analyse des résultats issus des différentes méthodes de sélection de caractéristiques met en évidence un ensemble cohérent de variables particulièrement discriminantes, comme on peut le voir à la table 4.1. Pour cela, plusieurs tests statistiques ont été utilisés afin d'évaluer l'importance de chaque variable par rapport à la cible. Le test du Chi² mesure l'indépendance entre une variable et la cible : plus le score est élevé, plus la variable est considérée comme dépendante de la classe à prédire, ce qui en fait un bon candidat pour la classification. L'ANOVA F-value (Analyse de la Variance) évalue la capacité d'une variable continue à distinguer les groupes définis par la cible ; une valeur élevée indique que la moyenne de la variable diffère significativement selon les classes. Enfin, l'information mutuelle mesure la quantité d'information qu'une variable apporte sur la cible : plus cette valeur est élevée, plus la variable réduit l'incertitude sur la classe. Dans les résultats obtenus, les variables bidirectional_max_ps, src2dst_max_ps et dst2src_max_ps obtiennent des scores très élevés dans les trois méthodes, indiquant une forte capacité de discrimination. Ces variables représentent les pics de débit dans les deux sens de communication, souvent révélateurs du type d'activité (streaming, téléchargement, etc.). De même, les volumes de données (bidirectional_bytes, src2dst_bytes, dst2src_bytes) ainsi que certaines variables temporelles (src2dst_duration_ms, bidirectional_stddev_piat_ms) apportent des informations précieuses sur les usages. Enfin, le champ dst_port, bien qu'un peu plus générique, reste pertinent dans la mesure où certains ports sont fortement associés à des services particuliers. Ces observations permettent d'identifier un noyau robuste de caractéristiques à retenir pour l'entraînement de modèles de classification efficaces.

| **Feature** | **Chi² Score** | **ANOVA F** | **Mutual Info** |
|---|---|---|---|
| bidirectional_max_ps | 417.25 | 121.08 | 0.8305 |
| src2dst_max_ps | 106.01 | 58.55 | 0.7787 |
| dst2src_max_ps | 468.67 | 106.08 | 0.7633 |
| bidirectional_bytes | 29.16 | 4.93 | 0.7518 |
| dst_port | 118.91 | 20.33 | 0.7406 |
| src2dst_bytes | 29.66 | 6.51 | 0.7227 |
| dst2src_bytes | 28.39 | 4.53 | 0.6910 |
| bidirectional_min_ps | 148.15 | 21.14 | 0.6830 |
| bidirectional_stddev_piat_ms | 149.43 | 88.26 | 0.6754 |
| src2dst_duration_ms | 130.51 | 93.41 | 0.6701 |

TABLE 4.1 – Scores des variables selon différentes méthodes de sélection de caractéristiques

Pour le troisième cas, bien que l'ensemble des caractéristiques disponibles ait été utilisé pour entraîner le modèle Random Forest lors de l'analyse principale, la figure 4.5 nous montre le top 10 des caractéristiques ayant le plus fort impact sur la classification. Cette analyse offre un éclairage complémentaire sur les facteurs déterminants du trafic, bien que toutes les variables

aient été conservées dans l'évaluation finale pour maximiser la performance du modèle.

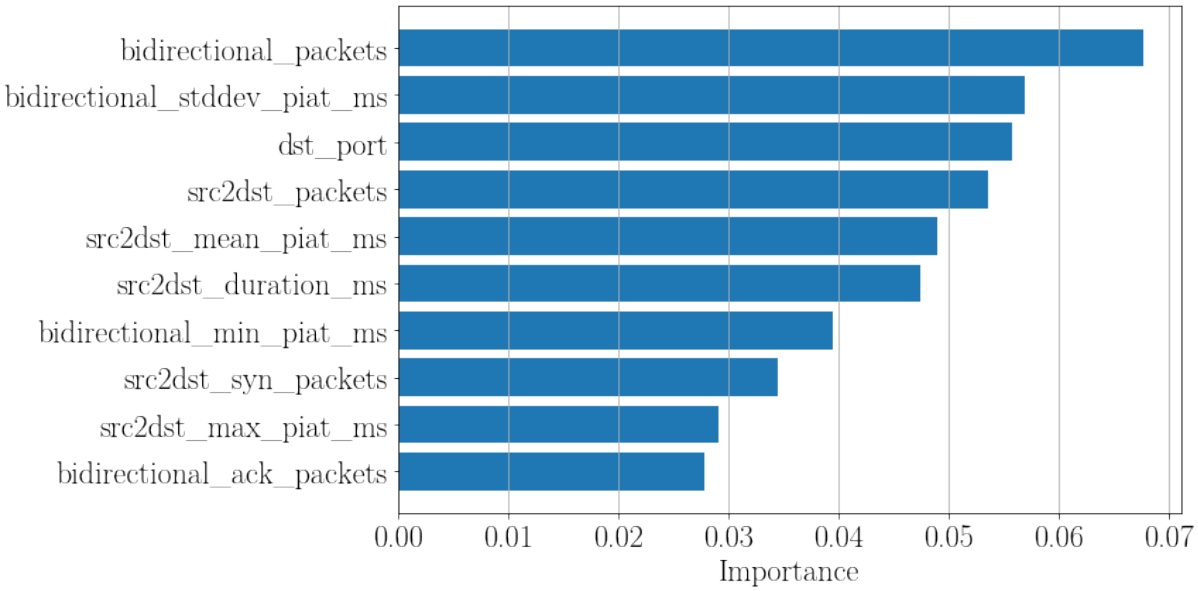



FIGURE 4.5 – Top10 des caractéristiques de RandomForest (sur 1)

Il a été choisi d'évaluer l'effet du nombre de caractéristiques utilisées sur la performance du modèle Random Forest. Les résultats ont montré une amélioration progressive des performances avec l'ajout de nouvelles caractéristiques, jusqu'à atteindre une forme de saturation aux alentours de 20 à 30 variables, avec une accuracy dépassant les 91%. Un pic de performance a été observé autour de 92,5% lorsque 60 caractéristiques étaient utilisées. On observe cependant un palier de performance au-delà d'un certain nombre de variables, ce qui suggère que les dernières caractéristiques apportent peu d'information supplémentaire. Cette tendance confirme l'intérêt de l'analyse d'importance des variables pour réduire la dimensionnalité sans perte significative de performance. La figure 4.6 met en évidence cette saturation, et peut servir de guide pour une future réduction de dimension ou optimisation de modèle.

### 4.2.3 Premières analyses et résultats

Afin d'évaluer l'impact de la sélection de caractéristiques sur les performances du modèle, trois expériences ont été menées en utilisant un classifieur Random Forest classique avec des paramètres constants (n_estimators=50 et random_state=42) afin de garantir la reproductibilité des résultats. Le choix de ces paramètres résulte d'un compromis entre la stabilité des prédictions et le temps de traitement, une contrainte essentielle dans le cadre d'une implémentation embarquée sur un switch réseau, où les ressources computationnelles sont limitées.

Trois stratégies de sélection de caractéristiques ont été explorées. La première repose sur

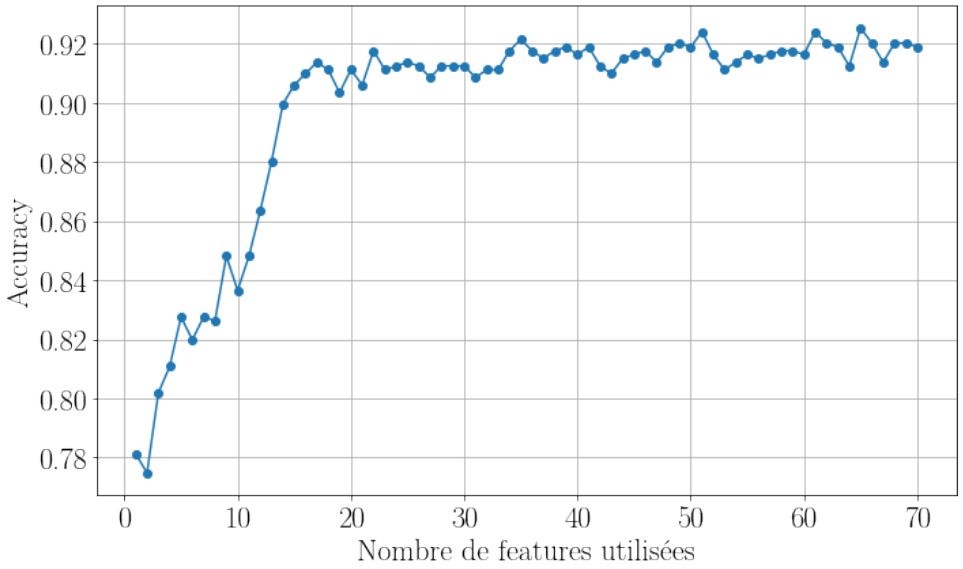



FIGURE 4.6 – Évolution de l'accuracy en fonction du nombre de caractéristiques utilisées

un choix heuristique de trois caractéristiques simples, mais informatives : la durée du flux, le nombre total de paquets, et le volume de données échangées (bidirectional_duration_ms, bidirectional_packets, bidirectional_bytes). Ces variables présentent l'avantage d'être peu coûteuses à calculer, tout en fournissant un aperçu global de l'intensité et de la nature de l'activité réseau. Cette approche minimaliste est particulièrement adaptée aux environnements contraints où un traitement rapide et léger est requis. La seconde approche utilise un filtrage statistique pour sélectionner les 10 caractéristiques les plus discriminantes, tandis que la troisième exploite les importances de variables calculées par le modèle Random Forest lui-même après un premier entraînement global. Il est important de noter que l'ajout de caractéristiques améliore généralement les performances de classification, mais augmente aussi les coûts de calcul. Or, dans un contexte embarqué tel qu'un switch, la rapidité de traitement et la faible consommation mémoire priment. Ces expériences visent ainsi à identifier le meilleur compromis entre complexité, précision, et efficacité temporelle.

Pour chacune de ces approches, une matrice de confusion normalisée est générée, respectivement aux figure 4.7, 4.8 et 4.9, permettant d'analyser visuellement les performances par classe. Ces matrices sont accompagnées des scores globaux (accuracy, précision, rappel, F1-score), que l'on peut retrouver à la table 4.2, afin de comparer les résultats de manière complète.

la dernière approche montre des performances nettement supérieures, avec une accuracy de 92% et des scores de précision, rappel et F1-score voisins, soulignant l'efficacité de cette stratégie de sélection automatique des caractéristiques.

Dans le cas d'une classification multi-classes, ces métriques sont généralement moyennées

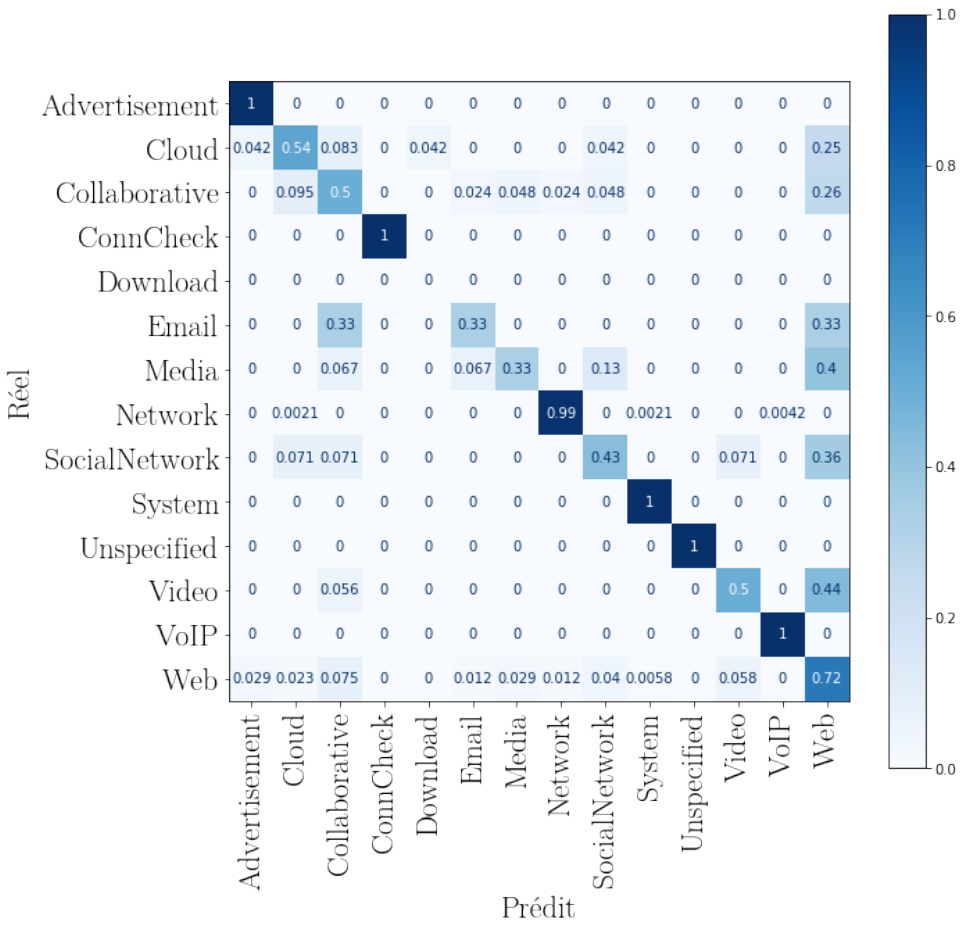



FIGURE 4.7 – Matrice de confusion des caractéristiques heuristiques

| Méthode | Accuracy | Précision | Rappel | F1-score |
|---|---|---|---|---|
| Heuristique manuelle | 0.853 | 0.855 | 0.853 | 0.850 |
| Filtrage (top 10) | 0.871 | 0.870 | 0.871 | 0.870 |
| Forêt aléatoire (all) | 0.920 | 0.912 | 0.918 | 0.913 |

TABLE 4.2 – Comparaison des performances selon la méthode de sélection de caractéristiques

(macro, micro ou pondérée) pour refléter les performances globales du modèle.

### 4.2.4 La particularité du trafic WEB

Une grande partie des erreurs de classification concerne le trafic associé à l'activité "Web". Cela s'explique notamment par la complexité inhérente à cette catégorie, qui regroupe non seulement la navigation classique sur des sites web, mais aussi de nombreuses autres activités – telles que le streaming, la consultation de mails via des webmails, ou encore l'usage d'applications web

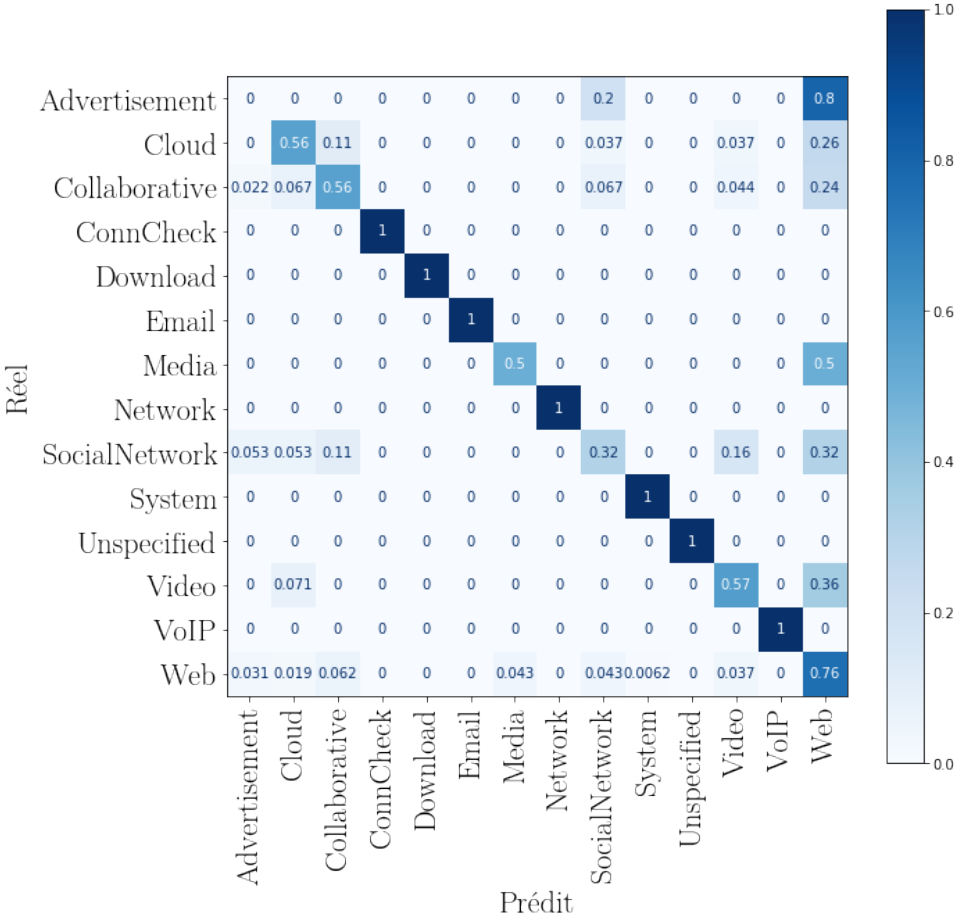



FIGURE 4.8 – Matrice de confusion des caractéristiques obtenues par filtrage

– qui transitent elles aussi par un navigateur. Cette superposition d'une couche applicative sur un même support de transport rend la distinction plus difficile, car des usages fonctionnellement très différents peuvent présenter des caractéristiques réseau similaires. La situation est d'autant plus complexe que la quasi-totalité de ce trafic est désormais chiffrée via HTTPS, transitant par un même protocole (TCP) et un même port (443), ce qui neutralise les indices habituellement utilisés pour distinguer les applications (comme les ports ou signatures spécifiques). Ainsi, seule une analyse fine des métriques statistiques permet de différencier les usages. Le trafic "Web" constitue donc un regroupement hétérogène d'activités, ce qui limite la capacité du modèle à le différencier précisément des autres classes.

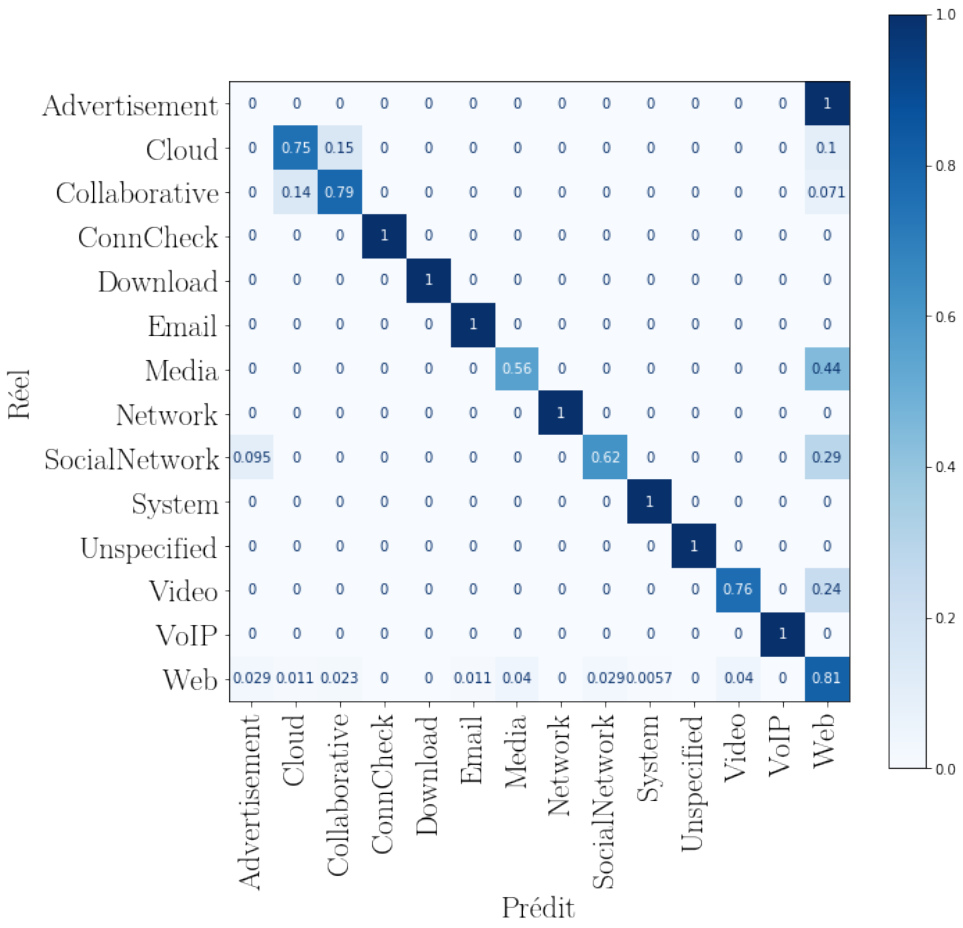



FIGURE 4.9 – Matrice de confusion des caractéristiques calculée par RandomForest

## 4.3 Analyse du trafic chiffré par VPN

Avec la montée en puissance des préoccupations liées à la vie privée et à la sécurité en ligne, l'utilisation des VPN (Virtual Private Network) s'est largement démocratisée ces dernières années. Des services grand public comme NordVPN, CyberGhost ou ExpressVPN sont désormais largement médiatisés, vantant leur capacité à sécuriser les connexions, à contourner les restrictions géographiques ou encore à obtenir des tarifs plus avantageux en ligne en passant par d'autres pays. Cette popularisation s'est accentuée avec la crise du COVID-19, qui a entraîné une généralisation du télétravail et, par conséquent, une augmentation significative de l'usage des VPN, tant chez les particuliers que dans les environnements professionnels. Du côté des entreprises, les VPN sont également utilisés pour garantir un accès sécurisé aux ressources internes depuis l'extérieur. Dans le contexte d'une adoption croissante des VPN, le trafic chiffré via VPN représente aujourd'hui une part non négligeable du trafic réseau global, ce qui en fait un sujet d'étude incontournable pour toute analyse moderne du trafic réseau.

### 4.3.1 Principes d'un flow VPN

L'utilisation d'un VPN modifie en profondeur le comportement d'un flux réseau classique. En effet, les protocoles VPN fragmentent et encapsulent le trafic afin de le chiffrer, ce qui masque certaines de ses caractéristiques initiales, comme le protocole d'origine ou les ports utilisés. Il devient donc difficile, voire impossible, de se baser sur ces éléments pour classifier le trafic. À titre d'illustration, un enregistrement simultané du même trafic en version claire et chiffrée via le protocole WireGuard a permis d'observer respectivement 621 flows en clair contre 17 flows dans la version chiffrée. Parmi ces derniers, un seul flow encapsulait l'ensemble du trafic initial. La table 4.3 présente en détail cette correspondance. Il convient de noter que les autres flows observés dans le trafic chiffré proviennent de communications réseau usuelles (ICMP, IGMP, MDNS, etc.) adressées à l'interface utilisée comme passerelle. Ces échanges ne figurent pas dans la capture en clair, car ils ne sont pas directement générés par l'activité utilisateur, mais résultent du rôle particulier de l'interface dans l'environnement réseau.

| ID flow chiffré | Durée (ms) | Nb paquets | Taille (o) | Protocole | Nb flows clairs | Durée totale clairs (ms) | Taille totale clairs (o) |
|---|---|---|---|---|---|---|---|
| 0 | 64203 | 4 | 432 | ICMPv6 | 0 | - | - |
| 1 | 80092 | 3 | 180 | IGMP | 0 | - | - |
| 2 | 0 | 1 | 46 | IGMP | 0 | - | - |
| 3 | 41984 | 2 | 92 | IGMP | 0 | - | - |
| 4 | 0 | 1 | 689 | WSD | 0 | - | - |
| 5 | 3211 | 3 | 396 | MDNS | 0 | - | - |
| 6 | 803 | 2 | 300 | ICMPv6 | 0 | - | - |
| 7 | 72455 | 3 | 138 | IGMP | 0 | - | - |
| 8 | 60076 | 12 | 1795 | TLS | 0 | - | - |
| 9 | 24057 | 13 | 3437 | MDNS | 0 | - | - |
| 10 | 88826 | 16047 | 10018446 | Wireguard | 621 | 88871 | 8607230 |
| 11 | 74707 | 8 | 664 | ICMPv6 | 0 | - | - |
| 12 | 75264 | 6 | 540 | ICPMv6 | 0 | - | - |
| 13 | 36864 | 2 | 92 | IGMP | 0 | - | - |
| 14 | 60502 | 16 | 2219 | TLS | 0 | - | - |
| 15 | 0 | 1 | 46 | IGMP | 0 | - | - |
| 16 | 24057 | 14 | 4346 | MDNS | 0 | - | - |

TABLE 4.3 – Association entre les flows chiffrés (WireGuard) et les flows en clair encapsulés

Par ailleurs, la structure même des flux est perturbée. Comme les flux sont formés à partir des échanges entre deux points, ils ont tendance à rester ouverts plus longtemps, souvent jusqu'à un timeout ou une fermeture explicite de la connexion. Cela génère des flows plus longs et volumineux, qui effacent davantage les traces caractéristiques du trafic en clair.

Il n'est donc pas pertinent de conserver ce format de flow tel quel pour analyser du trafic

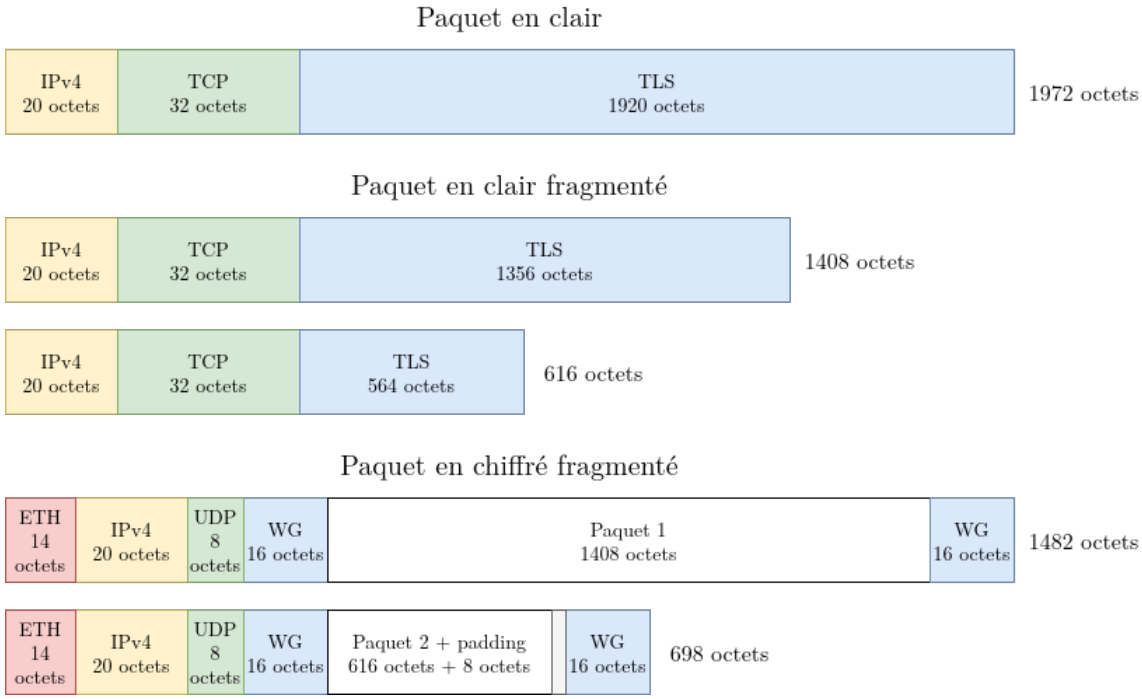



FIGURE 4.10 – Format et encapsulation du protocole Wireguard

chiffré. D'autres approches sont envisageables. Par exemple, sur le plan temporel, on observe que la somme de la taille des paquets en fonction du temps, ainsi que leur fréquence, restent généralement cohérentes. À partir du trafic en clair, il est souvent possible de prédire la forme que prendra le trafic chiffré en fonction du protocole utilisé, notamment en ce qui concerne la taille des paquets, le sens des échanges et les timestamps. On peut voir concrètement sur la figure 4.10 le format et le processus d'encapsulation utilisé par le protocole Wireguard.

Le premier segment illustre un paquet classique, composé d'un en-tête IPv4 (20 octets), TCP (32 octets), et d'une charge utile TLS de 1920 octets. Ce paquet dépasse la taille maximale de transmission (MTU) une fois encapsulé, et doit donc être fragmenté comme on peut le voir au milieu. La taille de la fragmentation dépend des protocoles utilisés pour la suite comme IP ou IPv6.

Dans la partie inférieure du schéma, on observe la transformation de ce paquet en deux paquets WireGuard. Le premier paquet encapsulé (Paquet 1) contient une première partie des données TLS, en respectant la contrainte de MTU de 1500 octets. Après ajout des en-têtes Ethernet (14 octets), IPv4 (20 octets), UDP (8 octets), et WireGuard (16 octets), la payload total est ajusté à 1408 octets. Le second paquet (Paquet 2) transporte la suite des données TLS, complétée par un padding de 8 octets pour aligner la structure, et est encapsulé de la même manière.

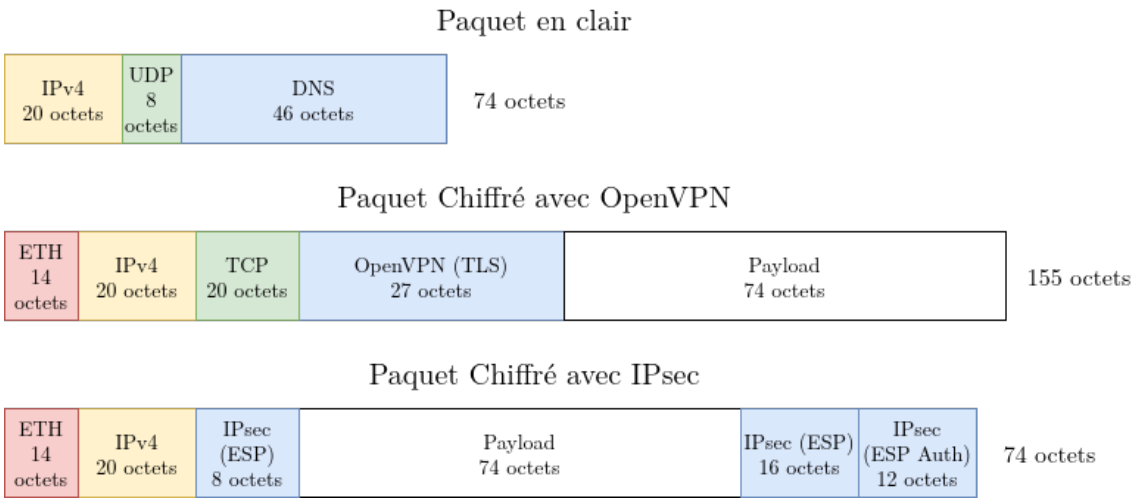



FIGURE 4.11 – Quelques formats et mode d'encapsulation des protocoles OpenVPN et IPsec

WireGuard repose sur UDP comme protocole de transport. Ainsi, une fois chiffrés, les paquets d'origine ne conservent aucune trace explicite de leurs couches applicatives (TCP, TLS), ni de leurs ports d'origine. Ce mode de fonctionnement assure la confidentialité, mais rend plus complexe toute tentative d'analyse ou de classification du trafic, celui-ci apparaissant comme un flux uniforme de paquets UDP.

Il existe une multitude de protocoles de chiffrement, parmi lesquels les plus couramment utilisés sont OpenVPN, IPsec et WireGuard. Chacun possède ses propres spécificités, notamment en termes d'algorithmes de chiffrement, de protocoles de transport (UDP ou TCP), ou encore de mécanismes d'authentification. En revanche, tous reposent sur un principe commun : l'encapsulation du trafic dans leur propre format, ce qui implique l'ajout d'un en-tête plus ou moins conséquent comme le montre la figure 4.11.

Par exemple, la figure 4.12 met en évidence la fragmentation établie par le protocole wireguard sur le trafic d'origine et efface les variations de taille qui pouvait exister. La figure 4.13 montre que le volume de données échangées au cours du temps reste relativement stable, même après chiffrement. Dans ce cas précis, cela s'explique par l'utilisation de WireGuard, qui encapsule les données dans un datagramme UDP (8 octets), puis y ajoute son propre en-tête (16 octets) et pied (16 octets), soit un total de 40 octets supplémentaires par paquet (sans compter l'encapsulation IP et Ethernet). Il est également important de noter que les paquets sont fragmentés pour respecter une taille maximale de 1482 octets en IPv4 dans la configuration actuelle.

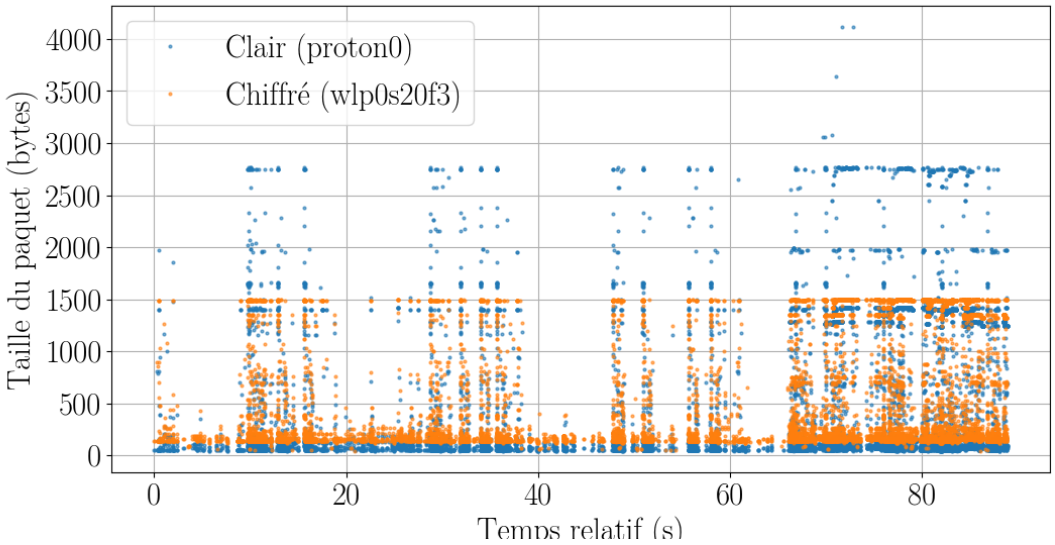



FIGURE 4.12 – Taille de chaque paquet pour le trafic en clair et chiffré par Wireguard

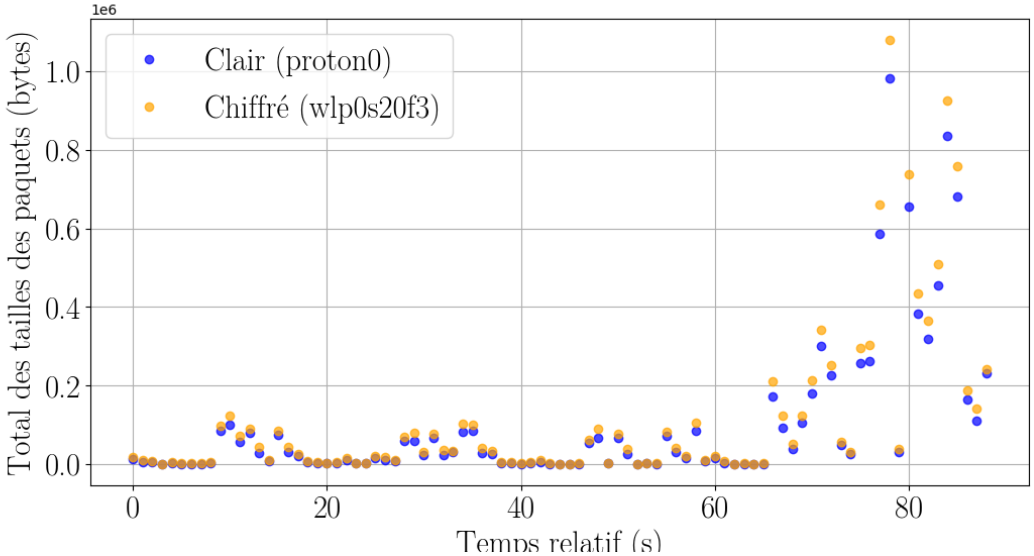



FIGURE 4.13 – Quantité de données par seconde pour le trafic en clair et chiffré par Wireguard

### 4.3.2 Analyse de trafic Wireguard

Mes travaux se sont orientés autour de la question de l'apprentissage supervisé à partir de trafic chiffré, et plus particulièrement sur les méthodes permettant d'étiqueter ce type de données. Pour cela, un même échange réseau a été enregistré en clair et en version chiffrée via VPN, afin de permettre une comparaison fine entre les deux captures.

L'analyse des fichiers PCAP montre qu'il est possible d'associer les paquets chiffrés et non chiffrés en s'appuyant sur des éléments tels que leur taille et leur timestamp. Les écarts temporels étant très faibles, la correspondance entre les paquets reste précise.

Ce constat permet de poser l'hypothèse qu'il est envisageable de prédire la taille d'un échange chiffré à partir de son équivalent en clair, ainsi que le schéma de fragmentation des paquets en fonction de leur taille. Un algorithme a donc été conçu pour estimer le ou les paquets chiffrés générés à partir d'un paquet en clair comme le montre l'algorithme 1.

À partir de cette estimation de la taille des nouveaux paquets chiffrés, il devient possible de recalculer l'ensemble des caractéristiques qui en découlent, tel que le nombre de paquets échangés, leur direction, ou encore les statistiques de débit. Pour cela, la bibliothèque NFStream est à nouveau exploitée, notamment pour sa capacité d'extension permettant de créer facilement de nouvelles caractéristiques sur mesure.

Un plugin a ainsi été développé, intégrant l'algorithme 1, et permettant de créer et d'actualiser dynamiquement l'ensemble des données notamment liées à la taille et la quantité des paquets qui ne sont plus les mêmes que les flows originels.

Toutes les autres caractéristiques existantes restent inchangées. Cette approche permet ainsi de conserver le label initialement associé au trafic en clair, tout en réutilisant les échanges simulés comme s'ils avaient transité par un protocole chiffrant tel que WireGuard.

Dans la continuité de l'expérience précédente, le jeu de données a été converti dans sa version chiffrée, en appliquant l'algorithme de simulation basé sur le protocole WireGuard. Le même modèle de classification a ensuite été testé sur ces nouvelles données, sans modification des hyperparamètres ni du pipeline d'entraînement. Les résultats obtenus, présentés dans la figure 4.14, confirment la robustesse du modèle, avec une matrice de confusion très proche de celle observée sur les données en clair.

Cela s'explique par le fait que seules certaines caractéristiques ont été modifiées : la taille des paquets et, par conséquent, le nombre de paquets échangés. Les autres attributs, tels que les statistiques temporelles ou les patterns d'échange directionnels, sont restés globalement inchangés, comme le montre la figure 4.15, qui met en évidence les dix caractéristiques les plus discriminantes sur ce nouveau jeu de données.

Cette phase constitue une étape intermédiaire importante avant d'aborder un cas plus réaliste, dans lequel il ne s'agira plus de simuler du trafic chiffré à partir d'un trafic clair, mais bien de partir directement d'un flux chiffré réel capturé dans un tunnel VPN. L'enjeu consistera alors

**Algorithm 1:** Estimation des tailles de paquets WireGuard à partir d'un paquet clair

**Data:** Taille du paquet clair : `packet_length`, protocole IP (IPv4 ou IPv6), protocole de transport (UDP ou TCP)

- `ETH_HDR_LEN` = 14 octets (en-tête Ethernet)
- `IP_HDR_LEN` = 20 octets (IPv4), `IP6_HDR_LEN` = 40 octets (IPv6)
- `UDP_HDR_LEN` = 8 octets, `TCP_HDR_LEN` = 32 octets
- `WG_HDR_LEN` = 16 octets, `WG_TLR_LEN` = 16 octets
- `TOTAL_WG_HDR = ETH + IP + UDP + WG_HDR + WG_TLR`
- `IP_MAX_SEGMENT_SIZE` = 1408 octets, `IPV6_MAX_SEGMENT_SIZE` = 1420 octets
- `PADDING_MULTIPLE` = 16 octets

**Result:** Liste des tailles des paquets chiffrés WireGuard correspondant au paquet clair

```
Fonction PrédireTailleWireGuard(packet_length, isIPv6, isUDP) :
    if isIPv6 then
        header_length ← IP6_HDR_LEN
        max_segment_size ← IPV6_MAX_SEGMENT_SIZE
    else
        header_length ← IP_HDR_LEN
        max_segment_size ← IP_MAX_SEGMENT_SIZE
    if isUDP then
        header_length ← header_length + UDP_HDR_LEN
    else
        header_length ← header_length + TCP_HDR_LEN
    data_length ← packet_length − header_length
    encrypted_sizes ← liste vide
    nb_packet ← 1
    if data_length > max_segment_size then
        data_length ← data_length − (max_segment_size − header_length)
        Ajouter max_segment_size + TOTAL_WG_HDR à encrypted_sizes
        while data_length > 0 do
            nb_packet ← nb_packet + 1
            if data_length > max_segment_size then
                data_length ← data_length − (max_segment_size + header_length)
                Ajouter max_segment_size + header_length + TOTAL_WG_HDR à
                  encrypted_sizes
            else
                padding ← PADDING_MULTIPLE − ((data_length +
                  header_length) mod PADDING_MULTIPLE)
                data_length ← 0
                Ajouter data_length + header_length + TOTAL_WG_HDR + padding
                  à encrypted_sizes
    else
        padding ← PADDING_MULTIPLE − (packet_length mod PADDING_MULTIPLE)
        Ajouter packet_length + TOTAL_WG_HDR + padding à encrypted_sizes
```

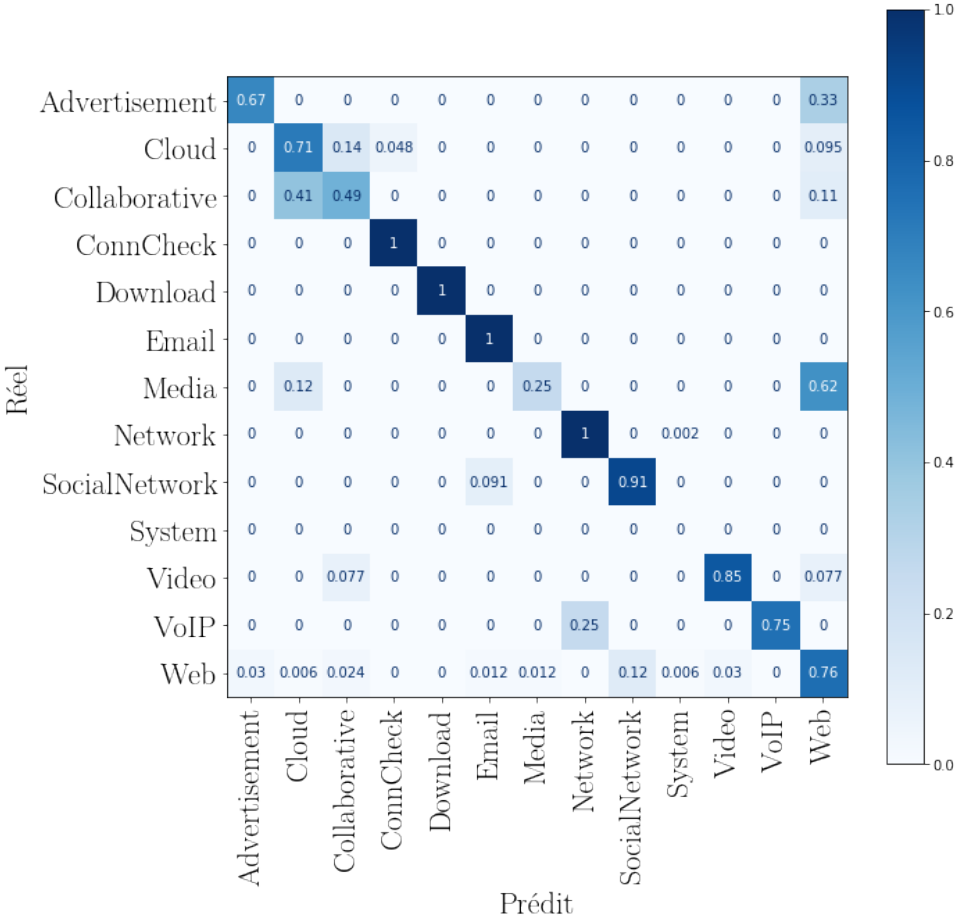



FIGURE 4.14 – Matrice de confusion des données chiffrées

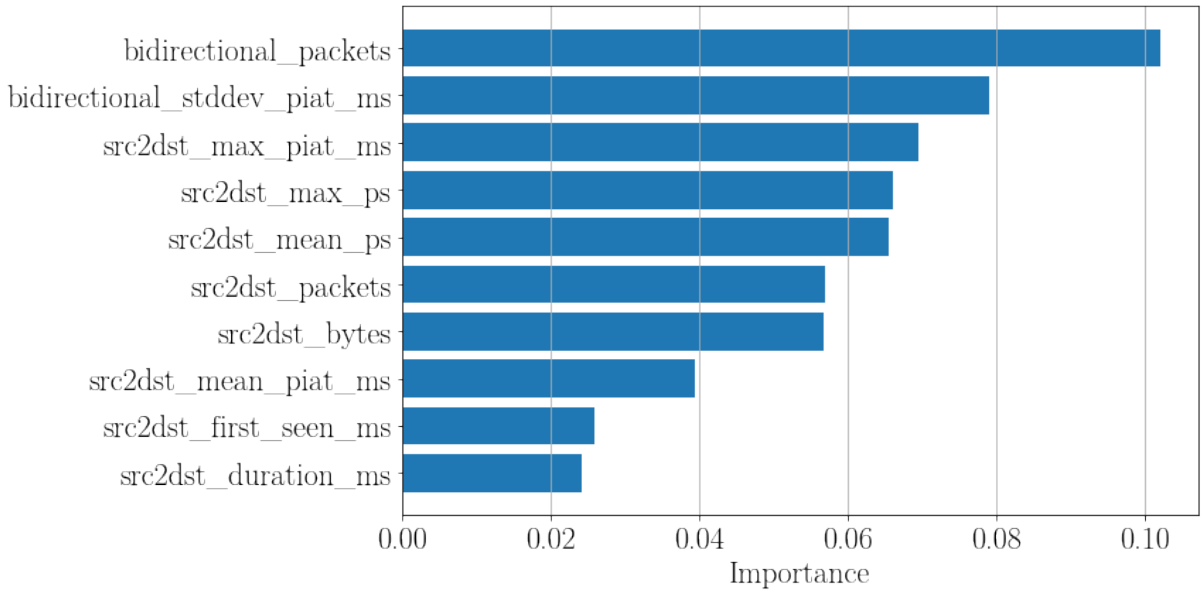



FIGURE 4.15 – Top 10 des caractéristiques pour le trafic VPN

à déterminer le bon découpage temporel et la taille de fenêtre optimale pour générer des flows pertinents et exploitables par le modèle, ce qui reste un paramètre inconnu à ce stade.

Plusieurs approches sont envisageables pour déterminer la stratégie de découpage temporel la plus pertinente à appliquer sur des flux chiffrés réels. Une première solution consiste à utiliser une fenêtre de taille fixe, par exemple en se basant sur la moyenne des tailles de fenêtres observées lors des enregistrements précédents. Une seconde approche, plus flexible, proposerait d'utiliser une fenêtre dynamique, qui pourrait varier soit aléatoirement, soit selon un pattern défini à l'avance (croissant, décroissant, périodique, etc.).

Enfin, une troisième méthode, plus coûteuse en calcul, mais potentiellement plus fine, consisterait à analyser le nombre moyen de paquets échangés par classe (application ou activité), afin d'en déduire une taille de fenêtre optimale pour chaque type de trafic. Cette approche pourrait être ensuite généralisée dynamiquement en temps réel, en adaptant la fenêtre au fur et à mesure de la capture. Bien que cette dernière stratégie implique davantage de traitements, elle pourrait permettre une meilleure segmentation du trafic chiffré, en s'adaptant plus précisément aux comportements observés.

## 4.4 Analyse du trafic industriel

L'analyse du trafic réseau industriel, en particulier celui généré par les automates programmables, constitue un enjeu majeur pour la supervision et la sécurisation des systèmes de contrôle industriels (ICS). Contrairement aux réseaux informatiques classiques, les réseaux industriels

présentent des caractéristiques propres : les communications sont souvent très longues, avec des connexions maintenues sur de longues durées, générant des flows massifs et peu exploitables tels quels. Ce phénomène, également observé dans le cas du trafic chiffré, complique l'identification de séquences pertinentes pour la détection d'anomalies ou la classification d'activités.

Cependant, les systèmes industriels disposent de caractéristiques qui peuvent se révéler un avantage pour l'analyse comportementale. Le trafic y est généralement hautement cyclique, en lien direct avec les tâches récurrentes et déterministes des automates, ce qui facilite l'établissement d'un profil de comportement « normal ». De plus, l'architecture réseau est souvent figée dans le temps : on ne modifie pas quotidiennement une chaîne de production, et les échanges entre les composants sont relativement stables et bien documentés. Ces éléments offrent un cadre propice à l'analyse et à la modélisation.

Dans ce travail, nous nous concentrons sur le trafic généré par des automates Siemens S7-300, largement répandus dans l'industrie, en raison de leur robustesse, de leur modularité et de leur déploiement massif dans des systèmes critiques tels que les chaînes de production et les architectures SCADA. Siemens indique que ces automates ont « fait leurs preuves des millions de fois dans le monde entier »[114], ce qui témoigne de leur adoption généralisée. Bien que leur fin de commercialisation soit annoncée, la société prévoit un support et une disponibilité des pièces détachées jusqu'en 2033, ce qui montre qu'elle anticipe une transition étalée sur près d'une décennie pour permettre aux industries de migrer progressivement vers la gamme S7-1500. Cette nouvelle génération reste compatible avec le protocole S7COMM, confirmant la pertinence de l'étude de ce protocole dans une perspective durable. L'objectif est d'étudier en profondeur la structure de leurs communications afin d'identifier un format de flow à la fois compact et exploitable pour des traitements en cybersécurité : détection d'anomalies, segmentation de comportement, ou encore classification d'activités. Il s'agira notamment de surmonter la difficulté liée à la faible granularité des flows traditionnels en proposant des méthodes de découpage adaptées aux contraintes industrielles.

### 4.4.1 Caractéristiques d'un flow industriel

Le protocole S7COMM est un protocole propriétaire utilisé par la gamme d'automates Siemens S7, dont les S-300, au cœur de notre étude. Il s'agit d'un protocole applicatif encapsulé dans ISO-on-TCP (port 102), permettant d'échanger des commandes entre une interface de supervision (SCADA ou HMI) et les automates. Le protocole S7COMM se structure en trois parties principales : un header, une section paramètre, et une section data. Le header contient notamment le type de message, essentiel pour interpréter le sens de la requête. Deux types courants de messages sont JOB, qui désigne une requête envoyée à l'automate, et ACK_DATA, qui correspond à la réponse de ce dernier. La figure 4.16 détaille les différentes parties du protocole.

La section paramètre décrit l'opération demandée : on y retrouve les instructions READ et

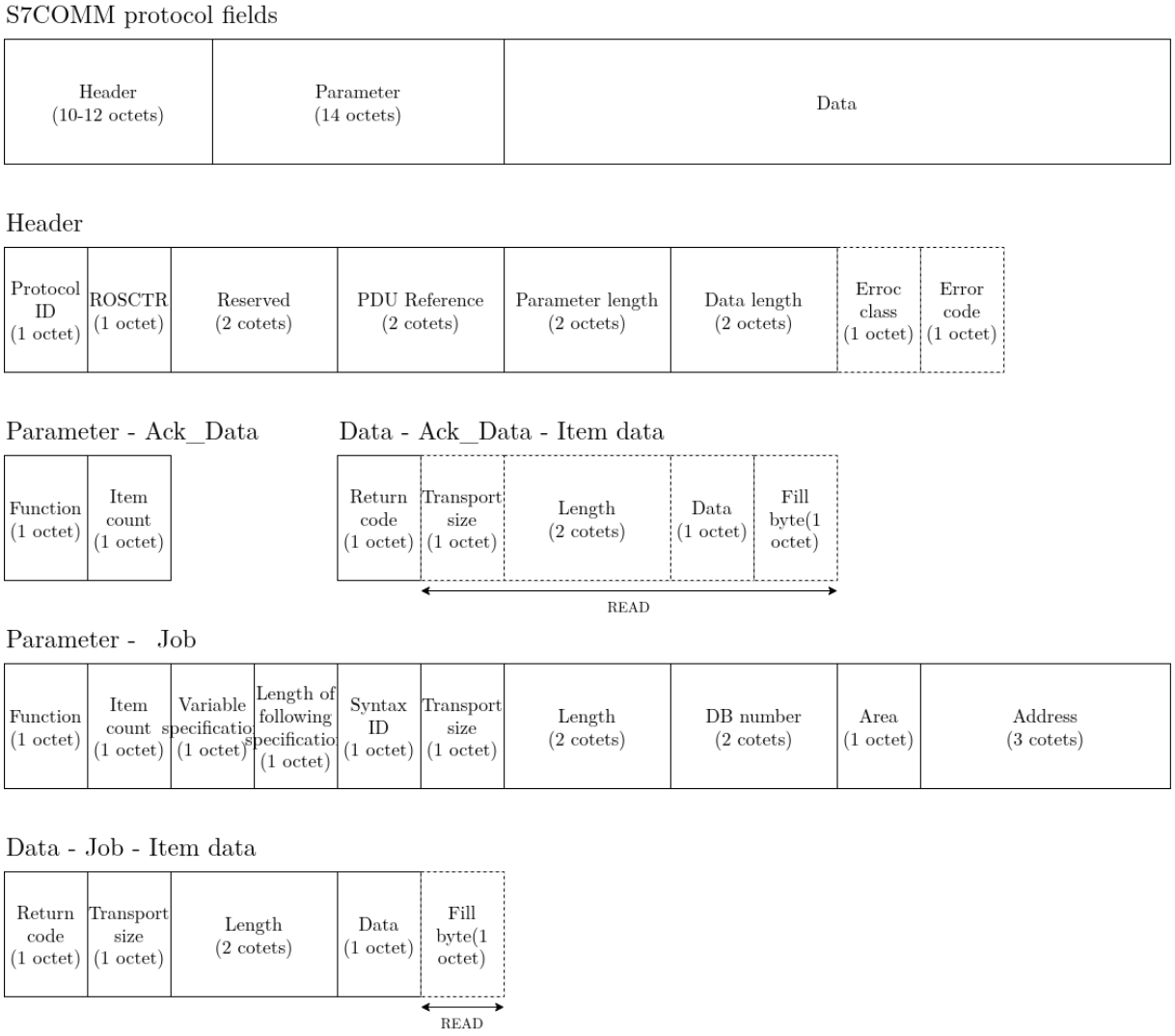



FIGURE 4.16 – Détails du protocole S7COMM de Siemens

WRITE, qui sont les plus représentatives des interactions avec un automate. En effet, un automate industriel réalise essentiellement des lectures (par exemple, lire une valeur de capteur ou un état mémoire) et des écritures (émettre une commande, changer une valeur dans la mémoire interne, etc.). Ces paramètres précisent aussi les zones mémoire visées (comme les Data Blocks, Inputs/Outputs, M area...), le nombre d'éléments et leur type.

Enfin, la section data contient, dans le cas des réponses, les valeurs lues ou les confirmations d'écriture. C'est cette structure riche et bien délimitée qui rend le protocole S7COMM particulièrement intéressant pour construire des flows industriels plus intelligibles et exploitables, en extrayant les opérations unitaires (lecture/écriture) comme éléments de base d'analyse, plutôt que de se limiter à des flux globaux peu informatifs.

Afin de rendre l'analyse du trafic industriel plus efficace et ciblée, nous avons défini un format de flow spécifique aux communications S7COMM, recentré sur les opérations à valeur métier, à savoir les lectures et écritures de variables réalisées par les automates. Contrairement aux pcap bruts, très volumineux et majoritairement constitués de paquets réseau sans valeur fonctionnelle directe (handshake TCP, keepalive, ARP, etc.), ce format extrait uniquement les transactions S7COMM pertinentes. Un enregistrement d'une heure de trafic peut atteindre plus de 500 Mo, alors que seulement 2% des paquets concernent réellement les interactions entre les équipements industriels. Ce filtrage est donc crucial pour rendre l'analyse exploitable.

Le format de flow que nous proposons repose sur une représentation structurée et synthétique de chaque opération individuelle (lecture ou écriture), intégrant les métadonnées essentielles : numéro de trame (Frame_Number), timestamp absolu (Timestamp et Timestamp_Epoch) et relatif (Timestamp_Shift), adresses IP source et destination, et longueur du paquet. Les champs protocolairement critiques sont conservés : type de commande (Header_Rosctr, ex. JOB ou ACK_DATA), identifiant de transaction (Header_PduRef), fonction (Param_Function, ex. READ/WRITE), type et nombre de données échangées, ainsi que les informations ciblant la zone mémoire concernée : le bloc de données (Param_Item_DB), la zone (Param_Item_Area), l'adresse en octet et bit. Pour une meilleure lisibilité, le nom de la variable manipulée (Variable_Name), son type logique (Data_Type), sa valeur (Data_Value) et son code de retour (Data_Return_Code) sont également extraits.

Ce format permet donc de réduire drastiquement la volumétrie des données tout en conservant une richesse sémantique élevée, centrée sur ce qui est réellement exécuté au niveau de l'automate. Il constitue une base solide pour des traitements en détection, classification ou reconstitution d'activité.

### 4.4.2 Analyse de trafic industriel S7COMM

L'étude du trafic réseau échangé entre automates permet d'identifier les grandes tendances comportementales propres aux environnements industriels. Dans cette section, nous analysons

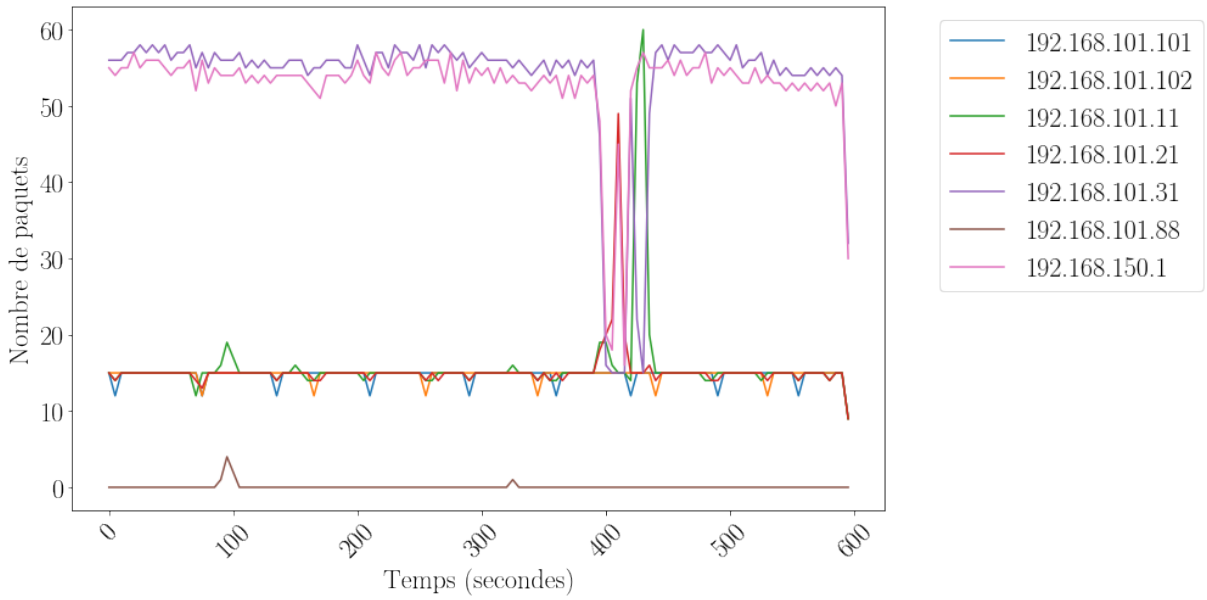



FIGURE 4.17 – Évolution du trafic en fonction des automates

une capture représentative de communications S7COMM afin de mettre en évidence les caractéristiques typiques de ce trafic, ainsi que sa sensibilité aux événements inhabituels.

La figure 4.17 présente une visualisation du trafic S7COMM observé sur une période de 10 minutes, réparti selon les adresses IP des différents équipements. On remarque immédiatement une grande stabilité du trafic, notamment pour certaines IP comme 192.168.101.21 ou 192.168.150.1, avec un nombre de paquets quasiment constant au fil du temps. Ce comportement reflète le fonctionnement cyclique des automates industriels, où les mêmes opérations de lecture et d'écriture sont effectuées de manière périodique, souvent à une fréquence fixe définie dans le programme automate.

Une anomalie marquée apparaît cependant au centre de la capture : une chute brutale du trafic, suivie d'un redémarrage progressif. Cette séquence correspond à un PLC_STOP, c'est-à-dire un arrêt temporaire des automates. Deux pics de trafic très nets sont également visibles avant et après cet arrêt, sur plusieurs IP simultanément. Ces pics sont liés à des opérations manuelles de lecture/écriture initiées par le logiciel TIA Portal, l'outil utilisé pour programmer et superviser les automates Siemens. Ces interactions humaines génèrent un volume anormalement élevé de requêtes sur une courte période, contrastant avec la régularité habituelle.

Il est important de noter que ces événements ne représentent pas des attaques à proprement parler, mais ils illustrent bien comment des changements dans le comportement du trafic S7COMM, même ponctuels, sont facilement détectables dans un contexte aussi régulier. Cette stabilité du trafic en conditions normales constitue un levier puissant pour la détection d'anomalies dans les réseaux industriels.

## 4.5 Conclusion

Ce chapitre a mis en évidence la richesse des données collectées grâce aux outils développés et leur pertinence dans le cadre d'une analyse fine du trafic réseau, tant sur des activités classiques que sur des échanges chiffrés ou industriels. À travers différentes expérimentations, il a été démontré que l'approche par flows constitue un compromis pertinent entre granularité, performance et faisabilité d'analyse. Les flows, en agrégeant les paquets selon des critères spatio-temporels et protocolaires, permettent de réduire la complexité des données tout en conservant l'essentiel des dynamiques d'échange. Ils offrent ainsi une base robuste pour l'entraînement de modèles de classification embarqués dans un environnement contraint, comme celui d'un switch réseau.

Cependant, cette approche n'est pas universelle. Dans certains cas, les flows générés s'avèrent trop larges ou trop imprécis, notamment lorsque les connexions sont longues ou persistantes, comme dans le cas des trafics chiffrés (VPN) ou industriels (SCADA/ICS). Pour y remédier, des formats de flows personnalisés ont été proposés. Dans le cas du VPN, une reconstruction synthétique du trafic chiffré à partir de données claires a permis de simuler fidèlement la structure des paquets après encapsulation, en préservant la pertinence des caractéristiques exploitées pour la classification. Cette approche permet de maintenir les étiquettes originales tout en rendant le jeu de données compatible avec l'environnement réel ciblé.

Concernant le trafic industriel, la structure classique des flows ne permet pas de capturer la dimension fonctionnelle des échanges entre automates, souvent enfouie dans des connexions longues et massives. Il a donc été nécessaire de descendre à un niveau plus fin, en extrayant des paquets métier (READ/WRITE) en tant qu'unités de base d'analyse. Ces « micro-flows » industriels permettent de réduire drastiquement la volumétrie, tout en augmentant la lisibilité et l'exploitation des données pour la détection ou la classification.

Ces choix méthodologiques s'inscrivent pleinement dans la logique d'un switch intelligent, dont l'objectif est d'adapter le traitement à la nature du trafic et aux contraintes locales (temps réel, ressources limitées, spécialisation par port). En combinant différentes représentations — flows standards, flows personnalisés, opérations unitaires — et en adaptant les modèles à chaque type de trafic, il devient possible d'intégrer des capacités de supervision, de classification et de détection efficaces, embarquées directement dans le cœur du réseau. Cette approche modulaire, personnalisable et optimisée, ouvre la voie à une surveillance plus fine, réactive et contextualisée des systèmes industriels et hybrides modernes.

# Conclusion et perspectives

**Sommaire**

**Rappel de la problématique** . . . . . **161**
**Contributions** . . . . . **161**
**Perspectives** . . . . . **163**

## Rappel de la problématique

Dans un contexte où les réseaux industriels deviennent de plus en plus interconnectés et exposés aux menaces extérieures, la sécurisation de ces environnements critiques représente un enjeu majeur. Cette thèse s'est attachée à explorer comment un commutateur réseau "intelligent" peut contribuer à une détection plus fine et plus ciblée des activités réseau. La problématique centrale repose sur la nécessité de modéliser, détecter et classifier les activités humaines dans un réseau informatique, afin de mieux comprendre les usages et renforcer les mécanismes de protection. Pour y répondre, plusieurs questions de recherche ont guidé ce travail : (QR1) comment modéliser et représenter l'activité humaine dans un réseau, en s'appuyant notamment sur la théorie de l'activité ; (QR2) comment classifier les activités observées malgré la diversité et l'ambiguïté des flux ; (QR3) comment générer ou étiqueter des ensembles de données représentatifs pour entraîner efficacement les modèles. En complément, une question de développement essentielle a été abordée : (QD1) quels outils peuvent permettre d'assister ou d'automatiser l'enregistrement et l'étiquetage des données réseau. Ces axes ont structuré les contributions de la thèse, tant sur le plan méthodologique que technique, et posent les bases pour une intégration de la classification du trafic au cœur des équipements réseau.

## Contributions

Les travaux menés dans cette thèse s'inscrivent dans une problématique pluridisciplinaire, à la croisée des domaines des réseaux, de la cybersécurité et de la modélisation des comportements humains. L'objectif global était de concevoir une solution intégrée de classification du trafic réseau au sein d'un commutateur, en intégrant des capacités de supervision, de détection, et de classification. En réponse à cette problématique, plusieurs contributions originales ont été apportées, structurées autour des questions de recherche et de développement définies en introduction.

La première contribution, en lien direct avec la question **QR1** *"Comment modéliser et représenter l'activité humaine dans un réseau informatique en tenant compte de ses spécificités ?"*, consiste en la transposition et à l'enrichissement de la théorie de l'activité au domaine de l'analyse réseau. Ce cadre conceptuel a permis de structurer les comportements numériques en les associant à des objectifs, des actions et des contextes techniques, ouvrant ainsi la voie à une lecture plus contextualisée et interprétable des traces réseau. Cette approche permet de mieux distinguer les activités humaines réelles des comportements automatisés ou malveillants, en ancrant chaque flux observé dans une activité cohérente.

En réponse à la question **QR2** *"Comment détecter et classifier les activités observées dans un réseau informatique ?"*, une méthodologie de classification des flux réseau a été développée. Celle-ci repose sur une approche hybride, combinant des heuristiques classiques (ports, protocoles), des indicateurs statistiques (volumes, temporalité), et des techniques d'apprentissage automatique supervisé. Cette stratégie permet de dépasser les limites des approches traditionnelles, en prenant en compte la variabilité des flux associés à une même activité, et en renforçant la robustesse des classifications face aux environnements hétérogènes. Ces méthodes ont été appliquées et évaluées dans le cadre d'expérimentations réalistes sur du trafic réseau humain.

La question **QR3** *"Comment générer ou étiqueter des ensembles de données représentatifs des activités réseau pour entraîner des modèles de classification ?"* a conduit à une contribution majeure autour de la production de données réalistes et annotées. Une attention particulière a été portée à la conception de scénarios représentatifs d'activités humaines, à la capture fine des interactions réseau, et à l'étiquetage automatique ou semi-automatique des flux produits. Cette démarche a permis de générer des jeux de données contextualisés, indispensables à l'entraînement et à l'évaluation de modèles de détection et de classification. Elle constitue également un socle méthodologique réutilisable pour des recherches futures.

En lien avec la question de développement **QD1** *"Quels outils peuvent assister ou automatiser l'enregistrement et l'étiquetage des activités réseau ?"*, plusieurs outils logiciels ont été conçus et développés spécifiquement au cours de cette thèse, afin de faciliter la génération, l'annotation et l'exploitation de données réseau contextualisées. Le premier de ces outils, *BASIL* (Behavioural Action Scenario With Image Labelling), permet l'enregistrement des interactions utilisateurs à travers une combinaison d'entrées clavier/souris et de captures d'écran. BASIL génère un scénario temporel détaillé des actions effectuées, permettant de rejouer fidèlement des comportements humains à des fins de génération de trafic. En complément, l'outil *TOMATO* (Tools for Optimised Manual Assistance on Tagging Operations) a été développé pour capturer, structurer et enrichir les traces réseau issues de ces scénarios, en leur associant des métadonnées descriptives — telles que l'application utilisée, le contexte d'exécution ou encore l'action de l' utilisateur — facilitant ainsi un étiquetage plus précis et automatisé.

Pour répondre aux besoins spécifiques de l'analyse de trafic industriel, l'outil *GAR7IC* (Ge-

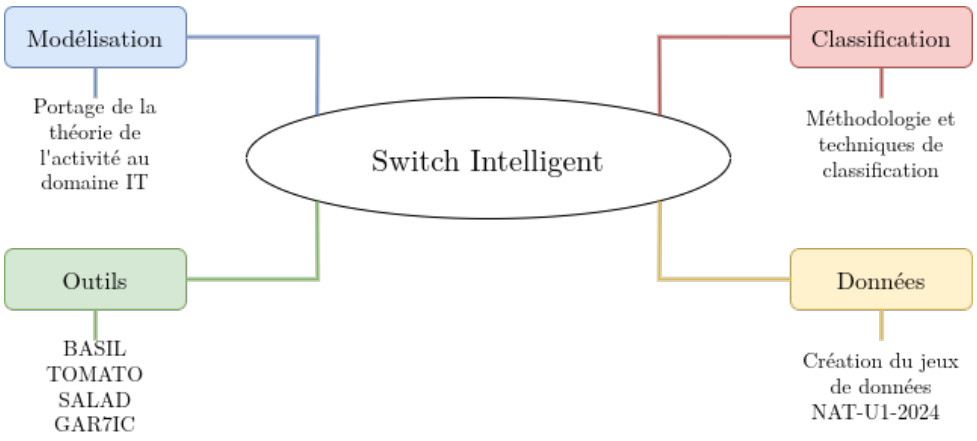



FIGURE 4.18 – Résumé des contributions

neralised Automatic Review and Labelling for S7COMM Industrial Communication) a été conçu pour extraire, annoter et convertir des communications industrielles (protocole S7COMM) en jeux de données exploitables. Configurable via des fichiers YAML, il permet d'associer des adresses de variables d'automates à des opérations réseau concrètes, assurant une annotation fine des paquets issus des environnements industriels. Enfin, *SALAD* (Script-based Automatic Labelling on Activity Dataset) vient compléter cet écosystème d'outils en offrant la possibilité de scripter et de rejouer automatiquement des scénarios utilisateurs au sein d'un environnement virtualisé maîtrisé. Cette approche permet d'associer précisément les actions exécutées aux traces réseau générées, grâce à une segmentation fine selon les fenêtres temporelles d'exécution, garantissant ainsi un étiquetage rigoureux et reproductible des activités observées.

Enfin, la conception d'une architecture de switch intelligent intégrant ces différentes briques de détection et de classification représente une contribution transversale à l'ensemble des questions abordées. Ce modèle architectural, fondé sur l'analyse embarquée en temps réel, offre un nouveau paradigme d'intégration de la cybersécurité directement dans les infrastructures réseau. Il constitue une base concrète pour la mise en œuvre de solutions embarquées capables d'assister les systèmes de détection existants, de préfiltrer les flux, ou d'interagir dynamiquement avec le réseau. L'ensemble de ces contributions, que l'on peut voir sur la figure 4.18, répond aux enjeux spécifiques de cybersécurité dans les environnements industriels en offrant une solution intégrée, embarquée directement au cœur de l'infrastructure réseau, et capable de fournir une analyse fine et contextualisée du trafic, sans dépendre exclusivement de dispositifs de supervision externes.

## Perspectives

Les travaux présentés dans ce manuscrit se sont inscrits dans un champ de recherche particulièrement dense, à l'intersection de plusieurs domaines tels que la cybersécurité, l'analyse

de trafic, la modélisation de l'activité humaine, ou encore l'apprentissage automatique. Cette richesse thématique a permis d'aborder de nombreux types de trafic — qu'il soit chiffré, non chiffré, industriel ou issu de l'activité utilisateur — mais elle a également mis en évidence la complexité et la profondeur de chacun de ces axes. Plusieurs prolongements naturels de cette recherche peuvent ainsi être envisagés.

Tout d'abord, la modélisation de l'activité humaine constitue un axe qu'il conviendrait d'approfondir. L'introduction de la théorie de l'activité dans le domaine de l'analyse réseau a permis d'ouvrir une voie nouvelle et prometteuse pour contextualiser les traces numériques. Ce cadre pourrait être étendu et enrichi afin de proposer une formalisation plus systématique de l'intentionnalité et des objectifs sous-jacents aux flux observés. Une piste intéressante consisterait à associer à chaque jeu de données des métadonnées décrivant précisément le scénario de génération, les objectifs de l'utilisateur et le contexte technique. Une telle démarche permettrait d'améliorer la transparence, la comparabilité, et la réutilisabilité des jeux de données, tout en facilitant le choix des modèles d'analyse les plus adaptés à une situation donnée.

En parallèle, les outils développés pour l'enregistrement, l'étiquetage et la classification du trafic réseau pourraient faire l'objet d'améliorations substantielles. L'automatisation du processus d'étiquetage reste un enjeu majeur. Des efforts devraient être portés sur la précision et la rapidité de ces méthodes, en explorant notamment des approches hybrides combinant règles expertes, apprentissage automatique, et génération de labels via des modèles de langage (LLM). L'intégration de ces derniers pourrait permettre une interprétation plus fine des contextes d'exécution, et ainsi améliorer la qualité sémantique de l'étiquetage.

Un autre axe de perspective concerne l'analyse des différents types de trafic. Si cette thèse a permis d'explorer une diversité de cas, un approfondissement spécifique de chacun de ces types s'impose. Pour le trafic industriel, par exemple, l'étude pourrait être élargie à d'autres protocoles que S7COMM, en particulier ceux utilisés par d'autres fabricants ou dans des environnements critiques spécifiques. En ce qui concerne le trafic chiffré, de nouvelles expérimentations pourraient viser l'analyse de protocoles de chiffrement plus variés, y compris ceux utilisés dans des contextes de confidentialité renforcée, comme TOR, qui n'a pas été abordé ici. L'enjeu serait alors de mieux caractériser les comportements chiffrés sans recourir au contenu, mais en s'appuyant uniquement sur des propriétés de flux, de temporalité, et de métadonnées.

Enfin, le dernier axe de perspective majeur concerne l'architecture du switch intelligent proposée dans ce travail. Si les expérimentations menées jusqu'ici ont permis de valider les principes de classification et de détection dans un cadre contrôlé, une mise en œuvre en environnement opérationnel reste indispensable pour en évaluer pleinement la robustesse, la pertinence et les éventuelles limites. La collaboration avec l'entreprise Interface Concept offre un terrain favorable pour poursuivre ce travail, en intégrant progressivement les briques logicielles développées dans un prototype fonctionnel de commutateur. Il s'agira alors d'analyser le comportement du

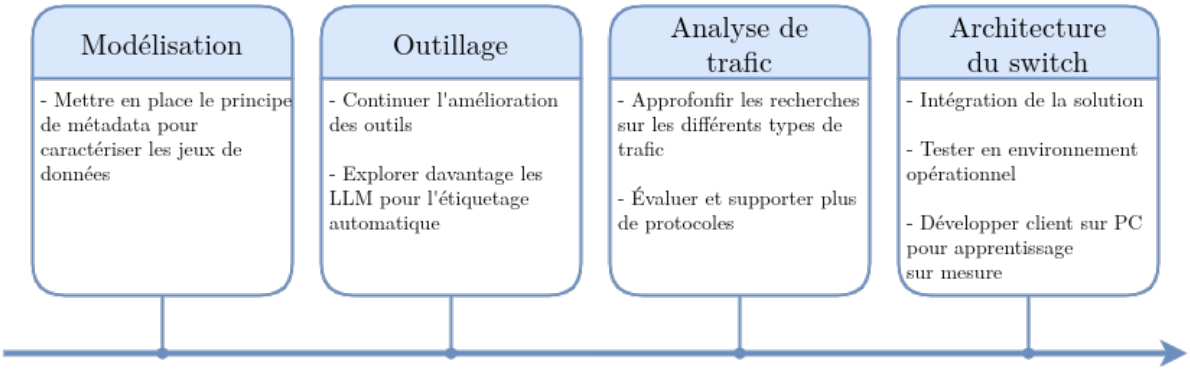



FIGURE 4.19 – Résumé des perspectives

système en conditions réelles, d'évaluer les performances, l'impact sur les flux et la compatibilité avec les contraintes industrielles, afin de confirmer ou d'ajuster les choix d'architecture retenus.

Par ailleurs, l'extension du dispositif côté postes clients représente un prolongement naturel de l'approche initiée. L'objectif serait de déployer, sur différentes machines du réseau, des modules légers capables d'enregistrer, d'étiqueter et d'apprendre localement les habitudes d'utilisation. Cela permettrait non seulement de raffiner les modèles d'analyse en les adaptant à chaque poste, mais aussi d'explorer la possibilité de reconnaître automatiquement le profil utilisateur ou la machine à l'origine des activités. Cette approche distribuée de l'apprentissage ouvrirait la voie à une détection plus fine, personnalisée et potentiellement plus résiliente face aux comportements malveillants. La mise en œuvre de ces extensions pourrait faire l'objet d'un travail de recherche appliqué au sein d'Interface Concept, ou bien constituer le cœur d'une future thèse dédiée à l'intégration, au déploiement et à l'évaluation des architectures de supervision intelligentes embarquées.

En somme, cette thèse a posé les fondations d'une approche intégrée de la classification et de la contextualisation du trafic réseau en environnement industriel. Les perspectives ouvertes, que l'on peut voir sur la figure 4.19, laissent entrevoir des prolongements aussi bien théoriques qu'opérationnels, et appellent à une exploration continue de ces thématiques à la croisée des sciences humaines, de l'intelligence artificielle et de la cybersécurité.

Un dernier point mérite d'être souligné : la mise en œuvre d'une architecture de supervision intelligente, qu'elle soit embarquée ou distribuée soulève des questions cruciales en matière de protection des données personnelles. Si les données collectées dans un contexte industriel sont souvent techniques, l'enregistrement de comportements utilisateurs ou de scénarios d'activité impose de réfléchir à des garanties fortes de confidentialité, notamment en lien avec les exigences du RGPD[115]. La mise en place de mécanismes de privacy-by-design[116], d'anonymisation ou de contrôle explicite sur les données collectées, devra accompagner tout déploiement à l'échelle réelle. Par ailleurs, une piste de recherche complémentaire pourrait consister à évaluer

la robustesse de l'architecture face à des tentatives d'évasion ou d'empoisonnement, dans le but de perturber la classification ou de biaiser les modèles d'apprentissage. L'étude de techniques adverses (ex. : génération de trafic camouflé, injection de patterns anormaux lors de l'entraînement) permettrait de mieux comprendre les limites du système et d'anticiper les contournements potentiels dans des contextes hostiles. Intégrer ces dimensions défensives et offensives dans les futurs travaux renforcerait à la fois la résilience, la légitimité et la portée scientifique de l'approche proposée.

# Liste des publications

1. M. Lannuzel and D. Brosset, "A modelling approach for identifying normal activities in computer network communications," in *Advances in Information and Communication*, K. Arai, Ed. Cham : Springer Nature Switzerland, Lecture Notes in Networks and Systems (LNNS), vol. 1284, 2025, pp. 467–481.
2. M. Lannuzel and D. Brosset, "Network Traffic Generation Framework for Intrusion Detection," in *Cybersecurity*, submitted, 2025.

Annexe A

# Téléchargement des Datasets

## A.1 IDS

| Name | Status | Link |
|---|---|---|
| CIC DDoS dataset | OK | http ://205.174.165.80/CICDataset/CICDDoS2019/Dataset/ |
| CICIDS2018 | OK | Install the AWS CLI (https ://aws.amazon.com/cli/), available on Mac, Windows and Linux, Run :<br>`aws s3 sync --no-sign-request --region <your-region>`<br>`"s3://cse-cic-ids2018/" dest-dir`<br>Where **your-region** is your region from the AWS regions list (https ://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html) and **dest-dir** is the name of the desired destination folder in your machine. |
| CICIDS2017 | OK | http ://205.174.165.80/CICDataset/CIC-IDS-2017/Dataset/ |
| AFDA-IDS | CloudStor problem | https ://research.unsw.edu.au/projects/adfa-ids-datasets |
| Unified Network Dataset | OK | https ://csr.lanl.gov/data/2017/ |
| DDoSTB | Not find (maybe not accessible) | Not find |
| Boosters | Not find (maybe not accessible) | Not find |
| TUIDS Coordinated Scan | Not shared | Not find |
| TUIDS DDoS | Not shared | Not find |
| TUIDS Intrusion | Not shared | Not find |
| UNSW-NB15 | CloudStor problem | https ://research.unsw.edu.au/projects/unsw-nb15-dataset |

| | | |
|---|---|---|
| Botnet dataset | OK | http ://205.174.165.80/CICDataset/ISCX-Bot-2014/Dataset/ |
| STA2018 | OK | https ://research-portal.st-andrews.ac.uk/en/datasets/sta2018-full-thesis-data |
| CTU-13 | OK | https ://mcfp.felk.cvut.cz/publicDatasets/CTU-13-Dataset/ |
| ISCXIDS2012 | OK | http ://205.174.165.80/CICDataset/ISCX-IDS-2012/Dataset/ |
| Twente | Website problem | https ://www.simpleweb.org/wiki/index.php/Labeled_Dataset_for_Intrusion _Detection |
| Kyoto | OK | http ://www.takakura.com/Kyoto_data/ |
| Waikoto | Website problem | https ://labs.ripe.net/datarepository/data-sets/the-waikato-internet-traffic-storage-wits-passive-datasets/<br>https ://wand.net.nz/wits/ |
| ASNM-CDX | OK | `http://www.fit.vutbr.cz/\sim{}ihomoliak/asnm/ASNM-CDX-2009.html`<br>https ://www.westpoint.edu/centers-and-research/cyber-research-center/data-sets |
| CAIDA DDoS | Need access demand | `https://www.caida.org/catalog/datasets/ddos-20070804_dataset/` |
| LBNL | OK | https ://www.icir.org/enterprise-tracing/download.html (doesn't work)<br>https ://cs.princeton.edu/courses/archive/fall16/cos432/hw4/lbl-internal.20041004-1305.port002.dump.anon.pcap (sample) |
| DEFCON | OK | Link not accessible but DEFCON17 and DEFCON23 at https ://media.defcon.org/DEF%20CON%2017/DEF%20CON%2017%20ctf/ |
| NSL-KDD | OK | http ://205.174.165.80/CICDataset/NSL-KDD/Dataset/ |
| KDD'99 | OK | http ://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html |
| DARPA 2000 | OK | https ://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets |
| DARPA 1999 | OK | https ://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset |
| DARPA 1998 | OK | https ://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset |

Table A.1: Statuts et liens de téléchargement des Datasets d'IDS

## A.2 Analyse du trafic réseau

| | | |
|---|---|---|
| LASS2023 | OK | https ://traces.cs.umass.edu/index.php/Network/Network |
| CIC DoHBrw 2020 | OK | http ://205.174.165.80/CICDataset/DoHBrw-2020/Dataset/ |
| MAWI | OK | http ://mawi.wide.ad.jp/mawi |
| ISCXTor 2017 | OK | http ://205.174.165.80/CICDataset/ISCX-Tor-NonTor-2017/Dataset/ |
| ISCXVPN 2016 | OK | http ://205.174.165.80/CICDataset/ISCX-VPN-NonVPN-2016/Dataset/ |
| USTC-TFC2016 | OK | https ://github.com/echowei/DeepTraffic/tree/master/1.malware_traffic _classification |

| | | |
|---|---|---|
| CTU-2013 | OK | https ://mcfp.felk.cvut.cz/publicDatasets/CTU-13-Dataset/ |
| UJN traces | Website problem | http ://cilab.ujn.edu.cn/Dataset.htm (doesn't work) |
| UNIBS traces | Need access demand | http ://netweb.ing.unibs.it/ntw/tools/traces/ |
| Moore | Data not accessible anymore | https ://www.cl.cam.ac.uk/research/srg/netos/projects/archive/nprobe/ https ://www.cl.cam.ac.uk/research/srg/netos/projects/brasil/data/index.html |
| Auckland II | Website problem | https ://wand.net.nz/wits/auck/2/auckland_ii.php |

TABLE A.2: Statuts et liens de téléchargement des Datasets d'analyse de trafic

Annexe B

# Liens entre les différents auteurs des jeux de données

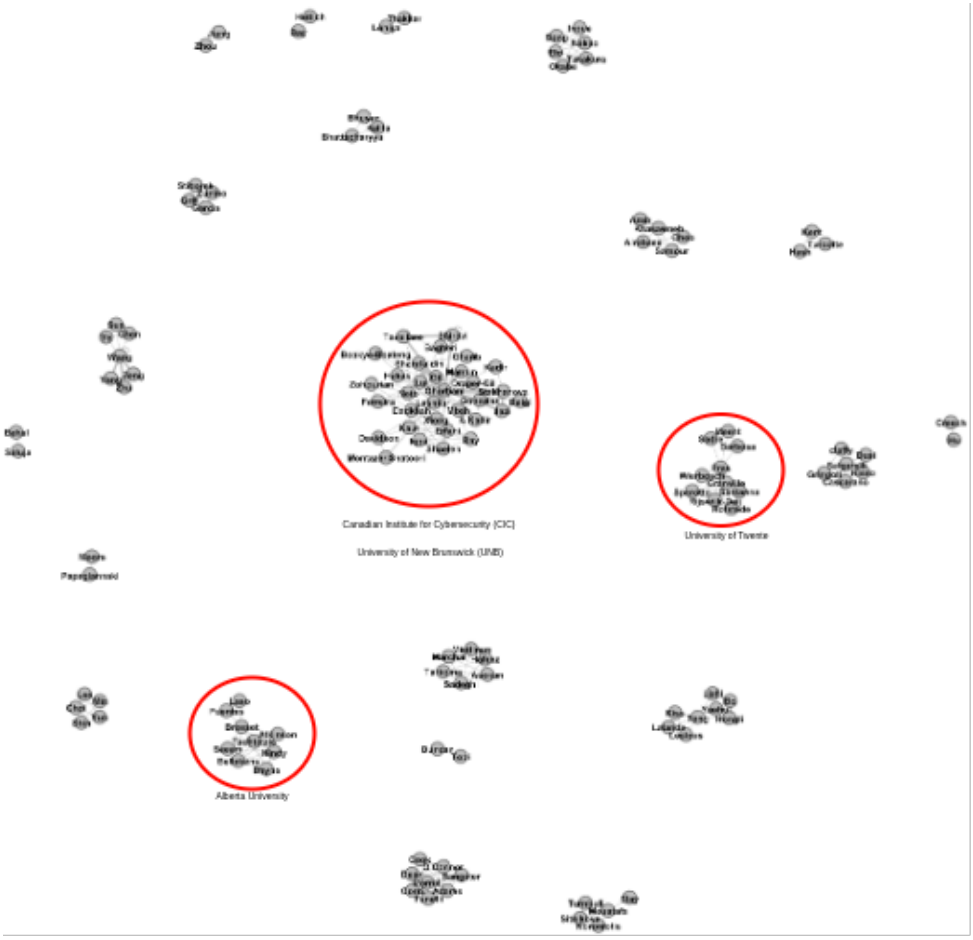



Figure B.1 – Liens entre les différents auteurs des jeux de données

Annexe C

# Captures d'écran

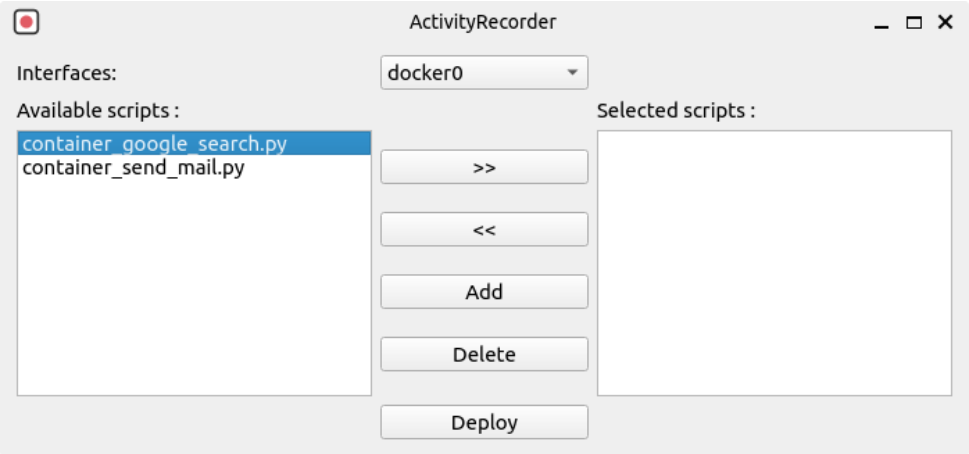



Figure C.1 – Menu du logiciel SALAD

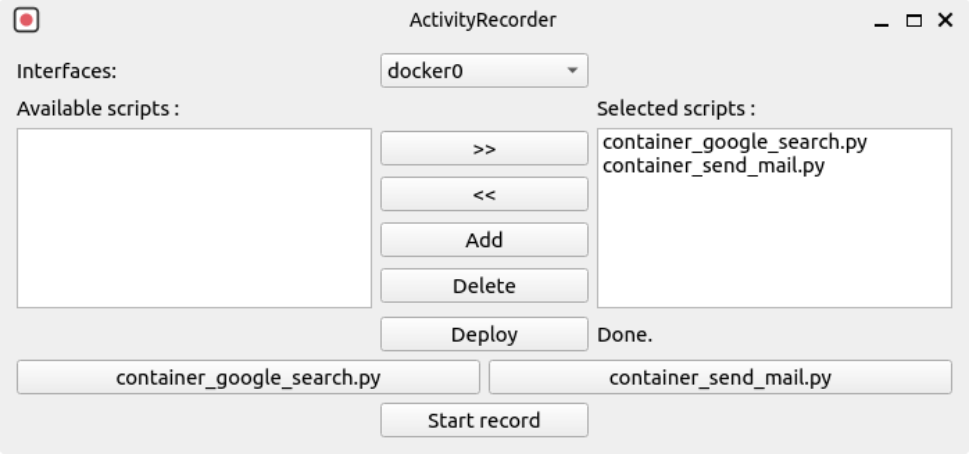



Figure C.2 – Menu du logiciel SALAD après déploiement du conteneur

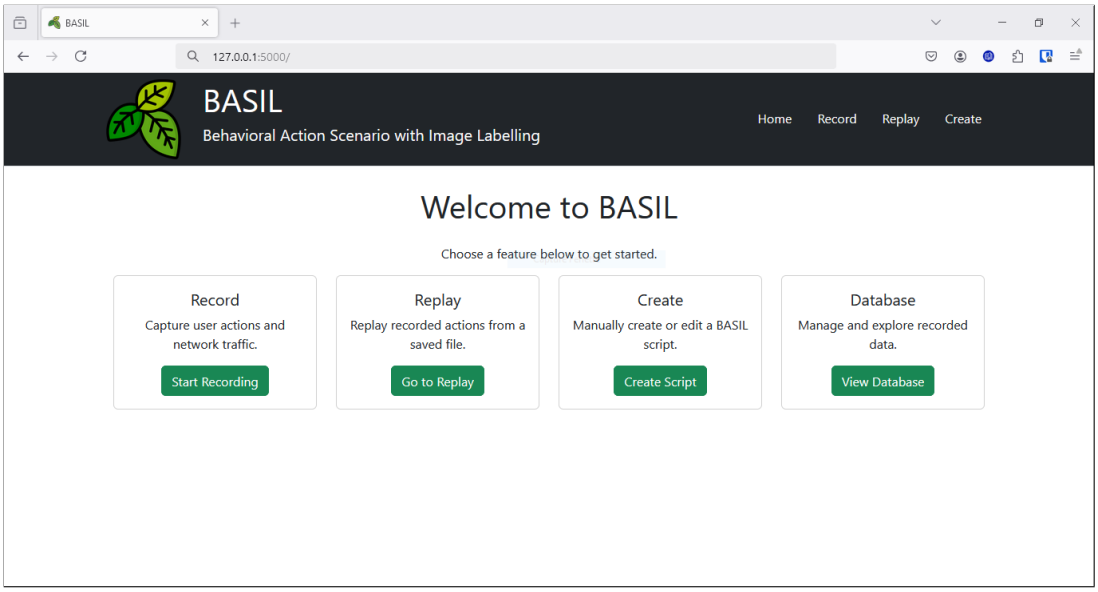



FIGURE C.3 – Interface WEB du logiciel BASIL

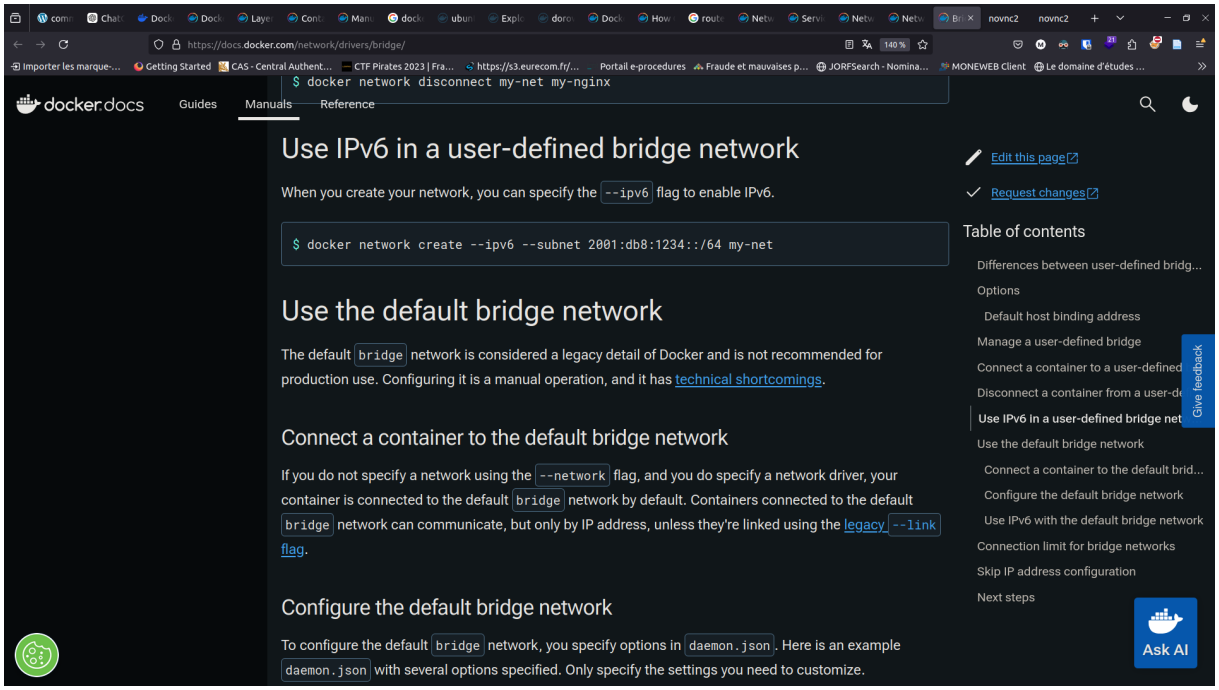



FIGURE C.4 – Capture d'écran de la documentation Docker

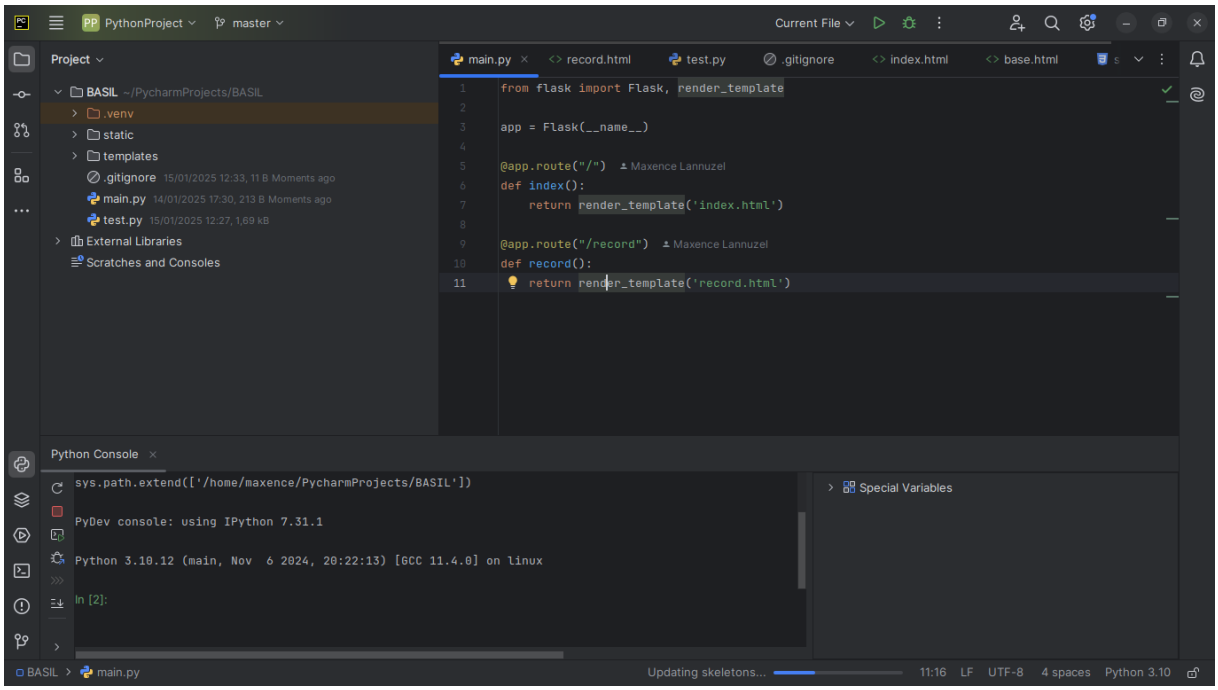



FIGURE C.5 – Capture d'écran de PyCharm

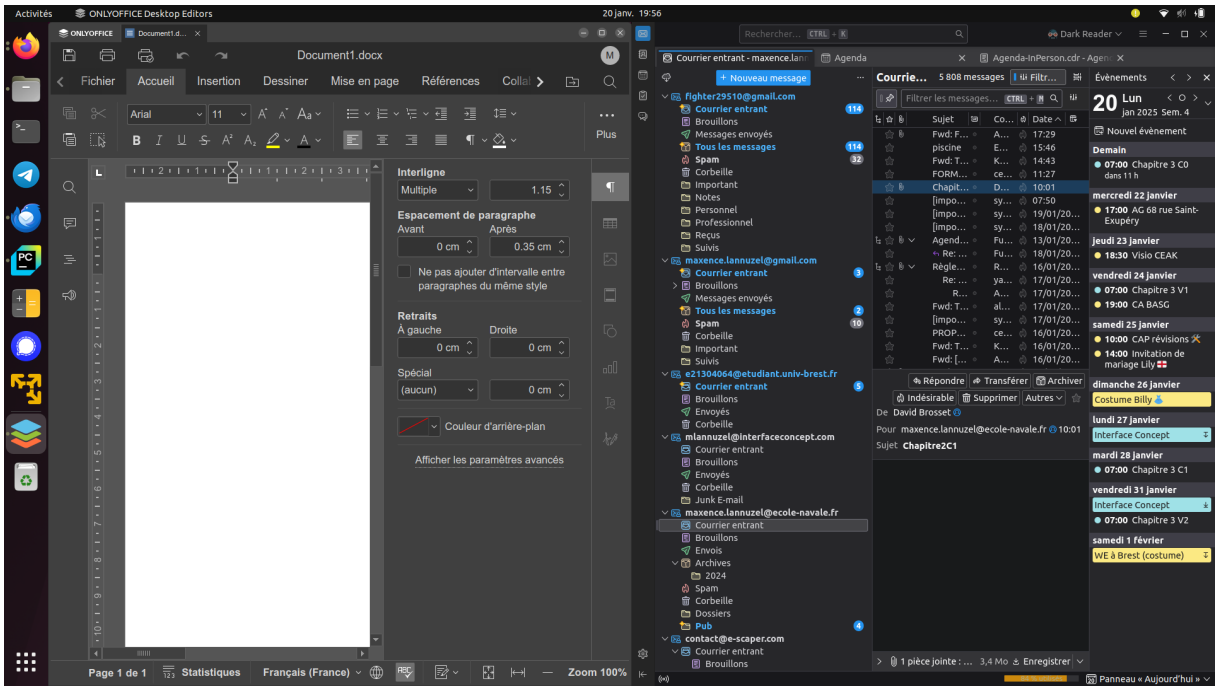



FIGURE C.6 – Capture d'écran de Thunderbird et OnlyOffice

# BIBLIOGRAPHIE


[1] T. Olzak, *The Purdue Model.* Tom Olzak, Jul 2021. [Online]. Available : https://www.spiceworks.com/it-security/cyber-risk-management/articles/securing-industrial-control-systems-from-modern-cyber-threats/

[2] "Introduction to ics security part 2," 06 2021. [Online]. Available : https://www.sans.org/blog/introduction-to-ics-security-part-2/

[3] Y. Engestrom, "Learning by expanding : An activity-theoretical approach to developmental research. helsinki : Orienta-konsultit," 1987.

[4] J. A. Hernández and M. Colom, "Repeatability, reproducibility, replicability, reusability (4r) in journals' policies and software/data management in scientific publications : A survey, discussion, and perspectives," 2023. [Online]. Available : https://hal.science/hal-04322522/document

[5] Agence de l'Union européenne pour la cybersécurité (ENISA), "NIS Investments 2024," Publications Office of the European Union, Tech. Rep., Nov. 2024. [Online]. Available : https://www.enisa.europa.eu/publications/nis-investments-2024

[6] J. L. H. Ramos, G. Karopoulos, I. N. Fovino, R. Spigolon, L. Sportiello, G. Steri, S. Gorniak, P. Magnabosco, R. Atoui, and C. C. Martinez, "Cyber resilience act requirements standards mapping," Publications Office of the European Union, Tech. Rep., 2024. [Online]. Available : https://publications.jrc.ec.europa.eu/repository/handle/JRC137340

[7] Naval Group, "Chantier charles de gaulle : la refonte à mi-vie," 2018, consulté le 18 mars 2025. [Online]. Available : https://www.naval-group.com/fr/chantier-charles-de-gaulle-la-refonte-mi-vie

[8] EU Automation, "Prolongez la durée de vie de vos automates programmables," 2025, consulté le 18 mars 2025. [Online]. Available : https://assets.euautomation.com/automated/guides/fr/PLCsLifespanFR.pdf

[9] PLC Talk Forum Users, "Life span of a plc," 2025, consulté le 20 mars 2025. [Online]. Available : https://www.plctalk.net/threads/life-span-of-a-plc.103625/page-2

[10] B. Meindl and J. Mendonça, "Mapping industry 4.0 technologies : From cyber-physical systems to artificial intelligence," *arXiv preprint arXiv :2111.14168*, 2021. [Online]. Available : https://arxiv.org/abs/2111.14168

[11] A. N. de la Sécurité des Systèmes d'Information (ANSSI), "Mémento de l'exposant privé," 2023, accessed : 2025-03-21. [Online]. Available : https://cyber.gouv.fr/sites/default/files/IMG/pdf/mementodep-v1-1.pdf

[12] S. AG, "Q4 2024 press release," 2024, accessed : 2025-02-07. [Online]. Available : https://assets.new.siemens.com/siemens/assets/api/uuid:df33f4cd-3bb5-4cd1-9ff8-79314ee62e0f/HQCOPR202411047033EN.pdf

[13] R. Automation, "Rockwell automation reports fourth quarter and full year 2024 results, introduces fiscal 2025 guidance," 2024, accessed : 2025-02-07. [Online]. Available : https://www.rockwellautomation.com/en-us/company/news/press-releases/Rockwell-Automation-Reports-Fourth-Quarter-and-Full-Year-2024-Results-Introduces-Fiscal-2025-Guidanc
html

[14] M. E. Corporation, "Mitsubishi electric integrated report 2024," 2024, accessed : 2025-02-07. [Online]. Available : https://www.mitsubishielectric.com/investors/library/integrated_report/pdf/2024/integrated_report2024_en.pdf

[15] S. Electric, "Pre-q4 2024 release consensus based on forecasts for schneider electric," 2024, accessed : 2025-02-07. [Online]. Available : https://www.se.com/ww/en/assets/564/document/502702/release-pre-q4-2024-consensus.pdf?p_enDocType=EDMS&p_File_Name=Pre-Q4%202024%20release%20consensus%20based%20on%20forecasts%20for%20Schneider%20Electric

[16] O. Corporation, "Omron integrated report 2024," 2024, accessed : 2025-02-07. [Online]. Available : https://www.omron.com/global/en/integrated_report/pdf_2024_en.html

[17] A. Group, "Q4 2024 press release," 2024, accessed : 2025-02-07. [Online]. Available : https://new.abb.com/docs/librariesprovider51/aktieinfo/abb-q4-2024-press-release-english.pdf?sfvrsn=9580ba0c_1

[18] M. Ahmed, A. Mahmood, and J. Hu, "A survey on network flow classification techniques," *Procedia Computer Science*, vol. 110, pp. 502–509, 2017. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S187770581730276X

[19] Y. Wang, Y. Zhang, Y. Liu, Y. Liu, and Y. Zhang, "The deep flow inspection framework based on horizontal federated learning," in *2022 31st International Conference on*

*Computer Communications and Networks (ICCCN)*. IEEE, 2022, pp. 1–9. [Online]. Available : https://ieeexplore.ieee.org/document/9919969

[20] BSS, "Recommandations de sécurité concernant l'analyse des flux https," Oct 2014. [Online]. Available : https://www.ssi.gouv.fr/uploads/IMG/pdf/NP_TLS_NoteTech.pdf

[21] A. Habibi Lashkari, G. Draper Gil, M. Mamun, and A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related features," 02 2016.

[22] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features." in *ICISSp*, 2017, pp. 253–262.

[23] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 104 650–104 675, 2020.

[24] A. Thakkar and R. Lohiya, "A Review of the Advancement in Intrusion Detection Datasets," *Procedia Computer Science*, vol. 167, pp. 636–645, 2020, international Conference on Computational Intelligence and Data Science. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S1877050920307961

[25] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Software Networking*, vol. 2017, no. 1, pp. 177–200, 01 2017.

[26] A. Özgür and H. Erdem, "A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015," 04 2016.

[27] N. Moustafa and J. Slay, "Unsw-nb15 : a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[28] I. Sharafaldin, A. Habibi Lashkari, and A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," 01 2018, pp. 108–116.

[29] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, Oct. 2019, pp. 1–8.

[30] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling," *Computer Networks*, vol. 121, pp. 25–36, Jul. 2017. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S1389128617301172

[31] G. Creech and J. Hu, "Generation of a new IDS test dataset : Time to retire the KDD collection," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, Apr. 2013, pp. 4487–4492.

[32] M. J. M. Turcotte, A. D. Kent, and C. Hash, *Unified Host and Network Data Set*. WORLD SCIENTIFIC (EUROPE), nov 2018, ch. Chapter 1, pp. 1–22. [Online]. Available : https://www.worldscientific.com/doi/abs/10.1142/9781786345646_001

[33] S. Behal and K. Saluja, "Measuring the impact of ddos attacks on web services -a realtime experimentation," *International Journal of Computer Science and Information Security*, vol. 14, pp. 323–330, 10 2016.

[34] J. J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Z. Granville, and A. Pras, "Booters 2014 ; an analysis of ddos-as-a-service attacks," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, May 2015, pp. 243–251.

[35] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Towards generating real-life datasets for network intrusion detection," *Int. J. Netw. Secur.*, vol. 17, pp. 683–701, 2015. [Online]. Available : https://api.semanticscholar.org/CorpusID:18636791

[36] N. Moustafa and J. Slay, "Unsw-nb15 : a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*. IEEE, Nov. 2015, pp. 1–6.

[37] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *2014 IEEE Conference on Communications and Network Security*. IEEE, Oct. 2014, pp. 247–255.

[38] A. Al Tobi and I. Duncan, "Improving intrusion detection model prediction by threshold adaptation," *Information*, vol. 10, no. 5, pp. 1–42, Apr. 2019, this research was supported and funded by the Government of the Sultanate of Oman represented by the Ministry of Higher Education and the Sultan Qaboos University.

[39] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, Sep. 2014. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S0167404814000923

[40] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, May 2012. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S0167404811001672

[41] R. Barbosa, R. Sadre, A. Pras, and R. van de Meent, *Simpleweb/University of Twente Traffic Traces Data Repository*, ser. CTIT Technical Report Series. Netherlands : Centre for Telematics and Information Technology (CTIT), Apr. 2010, no. TR-CTIT-10-19.

[42] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao, "Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation," in *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, ser. EuroSys '11. ACM, Apr. 2011, pp. 29–36.

[43] RIPE Network Coordination Center , "The waikoto internet trace storage project dataset," https ://labs.ripe.net/datarepository/data-sets/the-waikato-internet-traffic-storage-wits-passive-datasets, 2009.

[44] B. Sangster, T. J. O'Connor, T. Cook, R. Fanelli, E. Dean, W. J. Adams, C. Morrell, and G. Conti, "Toward instrumenting network warfare competitions to generate labeled datasets," in *Proceedings of the 2nd Conference on Cyber Security Experimentation and Test*, ser. CSET'09. USA : USENIX Association, 2009, p. 9.

[45] V. S. Shirsath, "Caida ucsd ddos 2007 attack dataset," 2007. [Online]. Available : https://dx.doi.org/10.21227/dvp9-s124

[46] LBNL/ICSI, "Lbnl/icsi enterprise tracing project," https ://www.icir.org/enterprise-tracing/download.html, 2004.

[47] The Shmoo Group, "Capture the flag contest-defcon datasets," http ://cctf.shmoo.com/data/, 2000.

[48] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, Jul. 2009, pp. 1–6.

[49] S. Hettich and S. D. Bay, "The uci kdd archive," http ://kdd.ics.uci.edu, 1999.

[50] Lincoln Laboratory, "Mit lincoln laboratory : Darpa intrusion detection evaluation," https ://www.ll.mit.edu/ideval/data, 2000.

[51] N. Kiss, J.-F. Lalande, M. Leslous, and V. Viet Triem Tong, "Kharon dataset : Android malware under a microscope," 05 2016.

[52] A. F. Abdul Kadir, N. Stakhanova, and A. A. Ghorbani, *Android Botnets : What URLs are Telling Us*. Springer International Publishing, 2015, pp. 78–91.

[53] Y. Zhou and X. Jiang, "Dissecting android malware : Characterization and evolution," in *2012 IEEE Symposium on Security and Privacy*. IEEE, May 2012, pp. 95–109.

[54] K. Boakye-Boateng, A. A. Ghorbani, and A. H. Lashkari, "Securing substations with trust, risk posture, and multi-agent systems : A comprehensive approach," in *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*. IEEE, 08 2023, pp. 1–12.

[55] H.-K. Shin, W. Lee, S. Choi, J.-H. Yun, , and B.-G. Min, "Hai security datasets," 2023. [Online]. Available : https://github.com/icsdataset/hai

[56] P. M. Laso, D. Brosset, and J. Puentes, "Dataset of anomalies and malicious acts in a cyber-physical subsystem," *Data in Brief*, vol. 14, pp. 186–191, 2017. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S2352340917303402

[57] NETRESEC, "Scada/ics pcap files from 4sics," *4SICS*, 2015. [Online]. Available : https://www.netresec.com/?page=PCAP4SICS

[58] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023 : A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023.

[59] M. Erfani, F. Shoeleh, S. Dadkhah, B. Kaur, P. Xiong, S. Iqbal, S. Ray, and A. A. Ghorbani, "A feature exploration approach for iot attack type classification," in *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, Oct. 2021.

[60] N. Moustafa, "A new distributed architecture for evaluating ai-based security systems at the edge : Network ton_iot datasets," *Sustainable Cities and Society*, vol. 72, p. 102994, Sep. 2021. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S2210670721002808

[61] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics : Bot-iot dataset," Nov. 2018.

[62] A. Azab, M. Khasawneh, S. Alrabaee, K.-K. R. Choo, and M. Sarsour, "Network traffic classification : Techniques, datasets, and challenges," *Digital Communications and Networks*, Sep. 2022. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S2352864822001845

[63] Laboratory for Advanced System Software, "Umasstracerepository," https ://-traces.cs.umass.edu/index.php/Network/Network, 2023.

[64] A. Zohaib, J. Sheffey, and A. Houmansadr, "Investigating traffic analysis attacks on apple icloud private relay," in *Proceedings of the ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '23. ACM, Jul. 2023.

[65] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, "Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, Aug. 2020, pp. 63–70.

[66] MAWI, "Mawi working group traffic archive," *The New Pacific Diplomacy*, 2015. [Online]. Available : http://mawi.wide.ad.jp/mawi/

[67] A. Habibi Lashkari, G. Draper Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - ICISSP*, P. Mori, S. Furnell, and O. Camp, Eds., INSTICC. [Setúbal] : SCITEPRESS - Science and Technology Publications, 2017, pp. 253–262, literaturangaben.

[68] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*. SCITEPRESS - Science and and Technology Publications, 02 2016.

[69] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, Jul. 2017.

[70] P. Lizhi, Z. Hongli, Y. Bo, C. Yuehui, and W. Tong, "Traffic labeller : Collecting internet traffic samples with accurate application information," *China Communications*, vol. 11, no. 1, pp. 69–78, Jan. 2014.

[71] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and k. c. claffy, "Gt : picking up the truth from the ground for internet traffic," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 12–18, Oct. 2009.

[72] A. W. Moore and K. Papagiannaki, *Passive and Active Network Measurement*, ser. SpringerLink, C. Dovrolis, Ed. Berlin, Heidelberg : Springer Berlin Heidelberg, 2005.

[73] U. of Aukland, "Aukland ii dataset," *Prismet*, vol. 71, no. 1, pp. 23–43. [Online]. Available : https://wand.net.nz/wits/auck/2/auckland_ii.php

[74] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescape, "Mirage : Mobile-app traffic capture and ground-truth creation," in *2019 4th International Conference on Computing, Communications and Security (ICCCS)*. IEEE, Oct. 2019.

[75] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019.

[76] R. Wang, Z. Liu, Y. Cai, D. Tang, J. Yang, and Z. Yang, "Benchmark data for mobile app traffic research," in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems : Computing, Networking and Services*, ser. MobiQuitous '18. ACM, Nov. 2018.

[77] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "Iot sentinel : Automated device-type identification for security enforcement in iot," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, Jun. 2017, pp. 2177–2184.

[78] D.-D. Orélie, *Théories de l'activité en sciences de l'information et de la communication et conception de dispositifs orientés utilisateurs*, vol. 2015/2, no. 28. [Online]. Available : https://www.cairn.info/revue-questions-de-communication-2015-2-page-43.htm

[79] S. Haggbloom, R. Warnick, J. Warnick, V. Jones, G. Yarbrough, T. Russell, C. Borecky, R. Mcgahhey, J. Powell, J. Beavers, and E. Monte, "The 100 most eminent psychologists of the 20th century," *Review of General Psychology*, vol. 6, pp. 139–152, 06 2002.

[80] J. Dolz and B. Schneuwly, "Le concept d'activité. quelques aspects qu'il rend visibles dans l'enseignement du français," *La Lettre de l'AIRDF*, vol. 21, no. 2, pp. 6–10, 1997, included in a thematic issue : Numéro spécial : Actes des journées d'étude de Montpellier. [Online]. Available : https://www.persee.fr/doc/airdf_1260-3910_1997_num_21_2_1279

[81] K. Stouffer, K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule *et al.*, *Guide to operational technology (ot) security*. US Department of Commerce, National Institute of Standards and Technology, 2023.

[82] CIGREF, "Convergence it-ot : Un rapprochement fructueux des systèmes industriels et des systèmes d'information," CIGREF, Tech. Rep., décembre 2019. [Online]. Available : https://www.cigref.fr/wp/wp-content/uploads/2019/12/Cigref-Convergence-IT-OT-Rapprochement-fructueux-Systemes-Information-et-Industriels-Decembre-2019-pdf

[83] Y. Maleh, "It/ot convergence and cyber security," *Computer Fraud & Security*, vol. 2021, no. 12, pp. 13–16, 2021.

[84] O. Jonathan, S. Misra, and V. Osamor, "Comparative analysis of machine learning techniques for network traffic classification," *IOP Conference Series : Earth and Environmental Science*, vol. 655, p. 012025, 02 2021.

[85] S. S. Al-Fedaghi, A. Alsaqa, and Z. Fadel, "Conceptual model for communication," *CoRR*, vol. abs/0912.0599, 2009. [Online]. Available : http://arxiv.org/abs/0912.0599

[86] International Organization for Standardization, *Information processing systems – Open Systems Interconnection – Basic Reference Model*, ser. ISO/IEC 7498-1 :1994. Geneva, Switzerland : International Organization for Standardization, 1994.

[87] V. Cerf and R. Kahn, "A protocol for packet network intercommunication," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 637–648, 1974.

[88] A. G. Blank, *TCP/IP Foundations*. John Wiley & Sons, 2006.

[89] J. Postel, "NCP/TCP transition plan," RFC 801, Nov. 1981. [Online]. Available : https://www.rfc-editor.org/info/rfc801

[90] R. T. Braden, "Requirements for Internet Hosts - Communication Layers," RFC 1122, Oct. 1989. [Online]. Available : https://www.rfc-editor.org/info/rfc1122

[91] W. Odom, *CCNA 200-301 Official Cert Guide, Volume 1*, I. Pearson Education, Ed. Hoboken, New Jersey : Cisco Press, 2020, vol. 1.

[92] J. Day, *Patterns in Network Architecture : A Return to Fundamentals*. Upper Saddle River, NJ : Prentice Hall, 2008. [Online]. Available : https://books.google.com/books/about/Patterns_in_Network_Architecture.html?id=k9sFgIM-z6UC

[93] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. C. Schmidt, and M. Wählisch, "Information-Centric Networking (ICN) Research Challenges," RFC 7927, Jul. 2016. [Online]. Available : https://www.rfc-editor.org/info/rfc7927

[94] M. Conti *et al.*, "Encrypted network traffic analysis and classification utilizing machine learning," *Sensors*, vol. 24, no. 11, p. 3509, 2023. [Online]. Available : https://www.mdpi.com/1424-8220/24/11/3509

[95] E. Papadogiannaki *et al.*, "A survey on encrypted network traffic analysis : Applications, techniques, and countermeasures," *ResearchGate*, 2021. [Online]. Available : https://www.researchgate.net/publication/353227879_A_Survey_on_Encrypted_Network_Traffic_Analysis_Applications_Techniques_and_Countermeasures

[96] M. Z. Shafiq *et al.*, "Network traffic classification using unsupervised learning : A survey," *HAL Archives*, 2022. [Online]. Available : https://hal.science/hal-04149117/document

[97] M. Akbari *et al.*, "Ai-powered internet traffic classification : Past, present, and future," *IEEE Communications Surveys & Tutorials*, 2022. [Online]. Available : https://ieeexplore.ieee.org/document/10355065

[98] T. Bujlow *et al.*, "Comparative analysis of classification methods and suitable datasets for protocol identification in it and ot networks," *Algorithms*, vol. 17, no. 5, p. 208, 2023. [Online]. Available : https://www.mdpi.com/1999-4893/17/5/208

[99] A. Risk, "4 ot/it network segmentation techniques : Selecting a cyber resilient configuration," *Applied Risk Resources*, 2018. [Online]. Available : https://applied-risk.com/resources/4-ot-it-network-segmentation-techniques-selecting-a-cyber-resilient-configuration

[100] P. Wang, X. Chen, F. Ye, and Z. Sun, "A Survey of Techniques for Mobile Service Encrypted Traffic Classification Using Deep Learning," *IEEE Access*, vol. 7, pp. 54 024–54 033, 2019.

[101] Wireshark Developers, "Pcapng - wireshark wiki," 2025, consulté le 17 mars 2025. [Online]. Available : https://wiki.wireshark.org/Development/PcapNg

[102] J. L. Guerra, C. Catania, and E. Veas, "Datasets are not enough : Challenges in labeling network traffic," *arXiv preprint arXiv :2110.05977*, 2021.

[103] C. Catania, J. L. Guerra, and E. Veas, "Towards a better labeling process for network security datasets," *ResearchGate*, 2023. [Online]. Available : https://www.researchgate.net/publication/370469367_Towards_a_better_labeling_process_for_network_security_datasets

[104] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Iscx tor-nontor dataset," 2016, dataset for Tor and non-Tor network traffic analysis. [Online]. Available : https://www.unb.ca/cic/datasets/tor.html

[105] G. D. Gil, A. H. Lashkari, M. Mamun, and A. A. Ghorbani, "Iscx vpn-nonvpn dataset," 2016, dataset for VPN and non-VPN network traffic analysis. [Online]. Available : https://www.unb.ca/cic/datasets/vpn.html

[106] M. Nogues, D. Brosset, H. Hindy, X. Bellekens, and Y. Kermarrec, *Labelled Network Capture Generation for Anomaly Detection*, 04 2020, pp. 98–113.

[107] E. Snowden, “Tweet recommending qubes os,” Twitter, 2016, consulté le 17 mars 2025. [Online]. Available : https://x.com/Snowden/status/781493632293605376?lang=en

[108] Qubes OS Project, “Partnering with the freedom of the press foundation,” https://www.qubes-os.org/news/2018/05/24/partnering-with-the-freedom-of-the-press-foundation/, 2018, consulté le 17 mars 2025.

[109] ——, “Qubes os userbase statistics,” https://www.qubes-os.org/statistics/, consulté le 17 mars 2025.

[110] Y. Lee, H. Lee, K. Lee, and K. Yim, “Cognitive countermeasures against bad usb,” in *Advances on Broad-Band Wireless Computing, Communication and Applications*, L. Barolli, F. Xhafa, and K. Yim, Eds. Cham : Springer International Publishing, 2017, pp. 377–386.

[111] X. Zhou, V. Valls, A. Ghosal, and B. Campbell, “pforest : In-network inference with random forests,” *arXiv preprint arXiv :1909.05680*, 2019. [Online]. Available : https://arxiv.org/abs/1909.05680

[112] P. Sahu and S. Akashe, “Hybrid model for intrusion detection systems,” *arXiv preprint arXiv :2003.08585*, 2020. [Online]. Available : https://arxiv.org/abs/2003.08585

[113] M. Waseem, N. Muhammad, M. Farhan, and et al., “Adasyn-random forest based intrusion detection model,” *arXiv preprint arXiv :2105.04301*, 2021. [Online]. Available : https://arxiv.org/abs/2105.04301

[114] Siemens AG, “Simatic s7-300,” https://www.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-300.html, 2025, consulté le 23 avril 2025.

[115] “Règlement (ue) 2016/679 du parlement européen et du conseil du 27 avril 2016 relatif à la protection des personnes physiques à l’égard du traitement des données à caractère personnel et à la libre circulation de ces données (rgpd),” https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX:32016R0679, 2016, journal officiel de l’Union européenne, L 119, 4 mai 2016, p. 1–88.

[116] A. Cavoukian, “Privacy by design : The 7 foundational principles,” Information and Privacy Commissioner of Ontario, Canada, Tech. Rep., 2009. [Online]. Available : https://www.ipc.on.ca/wp-content/uploads/resources/7foundationalprinciples.pdf

COLLEGE MATHSTIC
DOCTORAL BRETAGNE
BRETAGNE OCEANE

Université de Bretagne Occidentale

**Titre :** Sécurisation de réseaux en environnement industriel : prototype de switch intelligent

**Mot clés :** Analyse de trafic réseau, Commutateur intelligent, Classification réseau, Traitement de flow

**Résumé :** L'Industrie 4.0 transforme les infrastructures industrielles en favorisant une convergence croissante entre les systèmes informatiques (IT) et opérationnels (OT). Cette interconnexion améliore la flexibilité et l'efficacité des processus, mais introduit également de nouvelles vulnérabilités. Les réseaux industriels sont désormais exposés à des cybermenaces comparables à celles des environnements classiques, tout en conservant des contraintes spécifiques telles que la criticité des communications, l'hétérogénéité des équipements ou la diversité des protocoles utilisés. Parallèlement, des réglementations récentes, comme la directive NIS2 ou le Cyber Resilience Act, imposent des exigences renforcées en matière de supervision, de détection et de réactivité pour les systèmes industriels connectés.

Dans ce contexte, cette thèse propose le développement d'un commutateur Ethernet intelligent intégrant des fonctions avancées de surveillance, de classification et de détection du trafic réseau. En déportant une partie de l'analyse directement au sein de l'infrastructure de communication, cette solution permet d'identifier en temps réel les flux échangés, de caractériser les comportements réseau, et d'orienter dynamiquement les communications selon leur nature (web, messagerie, supervision industrielle, etc.) vers des traitements spécialisés. L'approche repose sur l'observation passive, la modélisation statistique des échanges et l'usage de méthodes d'apprentissage automatique. Des jeux de données réalistes, issus d'environnements mixtes IT/OT, sont utilisés pour entraîner et évaluer les performances des modèles. Cette architecture vise à renforcer la cybersécurité industrielle, à soulager les systèmes centraux et à répondre efficacement aux nouvelles obligations réglementaires.

**Title:** Securing Networks in Industrial Environments: Prototype of an Intelligent Switch

**Keywords:** Network traffic analysis, Smart switch, Network classification, Flow processing

**Abstract:** Industry 4.0 is transforming industrial infrastructure by promoting an increasing convergence between Information Technology (IT) and Operational Technology (OT) systems. This interconnection improves process flexibility and efficiency but also introduces new vulnerabilities. Industrial networks are now exposed to cyber threats comparable to those found in traditional IT environments, while retaining specific constraints such as the criticality of communications, equipment heterogeneity, and protocol diversity. At the same time, recent regulations, such as the NIS2 directive and the Cyber Resilience Act, impose stricter requirements in terms of monitoring, detection, and responsiveness for connected industrial systems.

In this context, this thesis proposes the development of an intelligent Ethernet switch that integrates advanced features for traffic monitoring, classification, and detection. By offloading part of the analysis directly within the communication infrastructure, this solution enables real-time identification of network flows, characterisation of network behaviours, and dynamic routing of communications according to their nature (web, email, industrial supervision, etc.) to specialised processing systems. The approach relies on passive observation, statistical modelling of exchanges, and the use of machine learning techniques. Realistic datasets, derived from mixed IT/OT environments, are used to train and evaluate the performance of the models. This architecture aims to strengthen industrial cybersecurity, reduce the load on central systems, and effectively meet emerging regulatory requirements.