

UNIVERSITÉ DU QUÉBEC

**MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À TROIS-
RIVIÈRES**

**COMME EXIGENCE PARTIELLE DE LA MAÎTRISE EN
MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES**

**PAR
AYTA NDOYE FALL**

**DETECTION D'INTRUSION DANS LES RESEAUX SDN : UNE
APPROCHE PAR LES COPULES**

AVRIL 2026

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

Identification du jury

Ce mémoire a été évalué par un jury composé de :

Monsieur Amar Bensaber Boucif, directeur de recherche, professeur au département de mathématiques et informatique appliquée

Madame Hakima Ould Slimane, professeure au département de mathématiques et informatique appliquée

Monsieur Ismail Biskri, Professeur au département de mathématiques et informatique appliquée

Remerciements

Avant tout, je rends grâce à Allah, le Tout-Puissant, pour m'avoir accordé la force, la patience, la clarté d'esprit et la persévérance nécessaires pour accomplir ce travail. C'est par Sa grâce et Sa miséricorde que ce mémoire a pu voir le jour, et c'est à Lui que revient tout mérite.

Je dédie ce travail en premier lieu à mes parents, et tout particulièrement à ma mère, Raki Coulibaly, pour ses sacrifices inestimables, son dévouement sans limite et son amour inconditionnel qui ont été le fondement de tout ce que je suis aujourd'hui. Par ses prières sincères et ses encouragements constants, elle a su me donner la force de persévérer dans les moments les plus difficiles.

Je tiens à exprimer ma sincère et profonde gratitude à Monsieur Amar Bensaber Boucif, mon directeur de recherche, pour la confiance qu'il m'a accordée en acceptant de diriger ce travail. Ses conseils éclairés, sa rigueur scientifique et sa disponibilité ont été déterminants dans l'avancement et la réussite de ce mémoire. Je lui suis profondément reconnaissante pour tout ce qu'il a apporté à ce projet.

Je remercie également Monsieur Mhmed Mesfioui, mon co-directeur de recherche, pour son implication sincère, ses orientations pertinentes et son soutien constant tout au long de ce parcours. Sa contribution a été d'une grande valeur et a considérablement enrichi ce travail.

J'adresse également mes chaleureux remerciements à Madame Hakima Ould Slimane et Monsieur Ismail Biskri pour leur participation au niveau du jury d'évaluation de ce mémoire. Leur expertise et le temps précieux qu'ils ont consacré à la lecture et à l'évaluation de ce travail me touchent sincèrement.

Enfin, je remercie du fond du cœur mes amis pour leur présence fidèle, leurs encouragements sincères et leur soutien moral dans les moments de doute comme dans les moments de joie. Leur amitié a été un réconfort précieux tout au long de ce parcours.

À toutes ces personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail, j'adresse ma plus sincère et profonde reconnaissance.

RÉSUMÉ

Notre recherche propose une approche pour la détection d'intrusions dans les réseaux SDN (Software Defined Networking), fondée sur la théorie des copules statistiques. Face aux limites des méthodes existantes dans la modélisation des dépendances complexes du trafic réseau, nous avons conçu et évalué des modèles hybrides combinant la modélisation copulaire et l'apprentissage supervisé.

Notre méthodologie repose sur l'exploitation du jeu de données CICDDoS2019, préalablement nettoyé et préparé. Une sélection de variables par méthode *stepwise* a permis de retenir les attributs les plus discriminants. Plusieurs familles de copules ont été comparées, montrant la supériorité de la copule Student-t pour capter les dépendances extrêmes caractéristiques des attaques DDoS.

Les modèles hybrides, intégrant cette copule à des classifieurs tels que la régression logistique, Random Forest et XGBoost, ont été évalués via une analyse multicritère AHP-TOPSIS. Le modèle hybride Student-t combiné au Random Forest a obtenu les meilleures performances ($AUC = 0,9998$), démontrant une détection fiable avec un très faible taux de faux négatifs.

Les résultats obtenus confirment que les copules offrent une modélisation plus fidèle des dépendances du trafic SDN que les approches classiques, avec des perspectives concrètes d'optimisation pour un usage en temps réel. Elle contribue ainsi au développement de systèmes de détection à la fois robustes, explicables et adaptés aux architectures réseaux programmables.

ABSTRACT

This research proposes an approach for intrusion detection in Software Defined Networking (SDN) environments, based on statistical copula theory. Given the limitations of existing methods in modeling the complex dependency structures of network traffic, hybrid models were designed and evaluated, combining copula modeling with supervised machine learning classifiers.

The methodology relies on the CICDDoS2019 dataset, which was thoroughly cleaned and preprocessed. A stepwise variable selection procedure was applied to retain the most discriminative features. Several copula families were compared, revealing the superiority of the student-t copula in capturing the extreme dependencies characteristic of DDoS attack traffic.

The hybrid models, integrating the student-t copula with classifiers such as Logistic Regression, Random Forest, and XGBoost, were evaluated through a multi-criteria AHP-TOPSIS analysis. The student-t copula-based Random Forest hybrid model achieved the highest performance, with an AUC of 0.9998, demonstrating reliable detection with a very low false negative rate.

The results confirm that copula-based modeling provides a more accurate representation of SDN traffic dependencies than conventional approaches, with concrete optimization prospects for real-time deployment. This work thereby contributes to the development of intrusion detection systems that are robust, interpretable, and well-adapted to programmable network architectures.

Table des matières

Introduction	1
Chapitre 1	4
1.1. Contexte et problématique	4
1.2. Software Defined Networking (SDN)	5
1.3. Les menaces de sécurité dans les réseaux SDN.....	13
1.4. La détection d'intrusions dans les environnements SDN.....	17
1.5. Limites des approches existantes de détection dans le SDN.....	19
Conclusion	22
Chapitre 2	24
État de l'art	24
2.1. Méthodes classiques de détection et de prévention des intrusions	25
2.2. Approches basées sur l'apprentissage automatique et profond	26
2.3. Solutions récentes : blockchain, apprentissage fédéré et architectures hybrides.....	28
2.4. Approche fondée sur la modélisation statistique des dépendances	31
Conclusion	33
Chapitre 3	35
Cadre conceptuel et orientation de la recherche.....	35
3.1. Objectif général	35
3.2. Objectifs spécifiques.....	36
3.3. Hypothèses de recherche.....	37
3.4. Justification du choix des copules	38
3.5. Contribution scientifique attendue	40
Conclusion	41
Chapitre 4	42
Fondements théoriques	42

4.1	Théorie des copules : définitions, familles, dépendances	42
4.2	Méthodes de construction de modèles avec copules	44
4.3	Évaluation de la dépendance et choix des copules	47
	Conclusion	50
	Chapitre 5	51
	Methodologie	51
5.1	Introduction	51
5.2	Prétraitement et préparation des données	51
5.2.1	Chargement et identification des variables	52
5.3	Sélection de variables	54
5.4	Modélisation par copules	55
5.5	Détection d'intrusions avec les copules	56
5.6	Modèles hybrides Copules	59
5.7	Évaluation multicritère avec AHP-TOPSIS.....	62
	Conclusion	63
	CHAPITRE 6	65
	RÉSULTATS ET ANALYSES	65
6.1	Présentation générale des résultats	65
6.2	Préparation et analyse des données.....	66
6.3	Résultats de la sélection de variables.....	68
6.4	Performance des modèles traditionnels	71
6.5	Résultats de détection basés sur les copules	76
6.6	Résultats des modèles hybrides	79
6.7	Analyse multicritère AHP-TOPSIS : évaluation holistique et validations	
	84	
6.8	Limites et perspectives.....	87
	Conclusion	90
	Références	92

Liste des tableaux

Tableau 1 : Comparaison réseaux traditionnels vs SDN (d'après [2], [14]).	12
Tableau 2 : Métriques d'évaluation	46

Table des figures

Figure 1 : Architecture logique du SDN et séparation des plans (données, contrôle et gestion) [2]	7
Figure 2 : Distribution des données après partitionnement.....	67
Figure 3 : Répartition des classes avant/après rééchantillonnage	68
Figure 4 : Résultats de la sélection stepwise.....	69
Figure 5 : Variables sélectionnées par RFE	70
Figure 6 : Variables finales sélectionnées	71
Figure 7 : Rapport de classification régression logistique	72
Figure 8 : Rapport de classification Random Forest	73
Figure 9 : Rapport de classification XGBoost.....	75
Figure 10 : Performances des copules gaussienne et Student-t	76
Figure 11 : Performances des copules frank et vine	78
Figure 12 : Performances des copules clayton et gumbel.....	78
Figure 13 : Rapport de classification - Modèle hybride Student-t + Regression Logistique	80
Figure 14 : Rapport de classification - Modèle hybride Student-t + Random Forest	82
Figure 15 : Rapport de classification - Modèle hybride Student-t + XGBoost	83
Figure 16 : Pondération AHP des critères d'évaluation	85
Figure 17 : Classement TOPSIS des modèles.....	86

Liste des abréviations

SDN : Software Defined Networking (Réseaux définis par logiciel)

IDS : Intrusion Detection System (Système de détection d'intrusions)

IPS : Intrusion Prevention System (Système de prévention d'intrusions)

DoS : Denial of Service (Déni de service)

DDoS: Distributed Denial of Service (Déni de service distribué)

CICDDoS2019: Canadian Institute for Cybersecurity Distributed Denial of Service 2019 Dataset

ML : Machine Learning (Apprentissage automatique)

DL : Deep Learning (Apprentissage profond)

AI : Artificial Intelligence (Intelligence artificielle)

RFE : Recursive Feature Elimination (Élimination récursive de caractéristiques)

AUC : Area Under the Curve (Aire sous la courbe ROC)

ROC : Receiver Operating Characteristic (Courbe caractéristique de performance)

F1-score : Moyenne harmonique entre précision et rappel

TP : True Positive (Vrai positif)

FP : False Positive (Faux positif)

TN : True Negative (Vrai négatif)

FN : False Negative (Faux négatif)

LR : Logistic Regression (Régression logistique)

SVM : Support Vector Machine (Machine à vecteurs de support)

RF : Random Forest (Forêt aléatoire)

CNN : Convolutional Neural Network (Réseau neuronal convolutionnel)

RNN : Recurrent Neural Network (Réseau neuronal récurrent)

API : Application Programming Interface (Interface de programmation d'applications)

QoS: Quality of Service (Qualité de service)

ONF: Open Networking Foundation

GPU : Graphics Processing Unit (Processeur graphique)

MLE : Maximum Likelihood Estimation (Estimation du maximum de vraisemblance)

IFM : Inference Functions for Margins (Fonctions d'inférence pour les marges)

AIC : Akaike Information Criterion (Critère d'information d'Akaike)

BIC : Bayesian Information Criterion (Critère d'information bayésien)

CV : Cross Validation (Validation croisée)

CSV : Comma-Separated Values (Fichier texte séparé par des virgules)

MITM : Man In The Middle (Attaque de l'homme du milieu)

FL : Federated Learning (Apprentissage fédéré)

FN Rate : False Negative Rate (Taux de faux négatifs)

FP Rate : False Positive Rate (Taux de faux positifs)

TOPSIS: Technique for Order Preference by Similarity to Ideal Solution

Introduction

Les réseaux informatiques occupent aujourd'hui une place centrale dans le fonctionnement des entreprises, des administrations et des particuliers. Le volume de données échangées n'a cessé de croître ces dernières années, et avec lui, la complexité des menaces qui pèsent sur ces infrastructures. Dans ce contexte, sécuriser les réseaux est devenu un enjeu aussi technique qu'opérationnel. Parmi les architectures qui ont profondément transformé la gestion des réseaux, le Software Defined Networking se distingue par sa capacité à séparer le plan de contrôle du plan de données, offrant une administration centralisée et programmable de l'ensemble de l'infrastructure. Cette centralisation constitue à la fois sa force et son talon d'Achille : en concentrant la logique décisionnelle en un point unique, elle en fait une cible privilégiée pour les attaques par déni de service distribué, capables de paralyser l'ensemble du réseau en visant directement le contrôleur.

C'est dans ce contexte que s'inscrit notre mémoire, dont l'objectif est de concevoir un modèle de détection d'intrusions adapté aux réseaux SDN, fondé sur la théorie des copules statistiques. L'idée centrale est de combiner la modélisation des dépendances complexes entre variables du trafic réseau avec des algorithmes de classification supervisée. Contrairement aux approches classiques, les copules permettent de capturer des relations non linéaires et asymétriques entre variables, ce qui les rend particulièrement bien adaptées à la nature du trafic SDN. Plusieurs familles de copules ont été testées et combinées à trois classificateurs et leurs performances ont été comparées à l'aide d'une évaluation multicritère AHP-TOPSIS.

Notre mémoire est organisé en six chapitres. Le premier chapitre pose le contexte de la sécurité dans les réseaux SDN, en présentant les principales menaces et les limites des systèmes de détection actuels. Le deuxième chapitre dresse un état de l'art des méthodes de détection d'intrusions, des approches classiques aux solutions plus récentes intégrant l'apprentissage automatique, l'apprentissage fédéré et la blockchain. Le troisième chapitre définit le cadre conceptuel de la recherche : objectifs, hypothèses et justification du recours aux copules. Le quatrième chapitre expose les fondements théoriques de la

modélisation copulaire, incluant les définitions, les familles de copules et les mesures de dépendance. Le cinquième chapitre décrit la méthodologie expérimentale, du prétraitement des données jusqu'à la construction et l'évaluation des modèles hybrides. Enfin, le sixième chapitre présente les résultats obtenus sur le jeu de données CICDDoS2019 et formule des recommandations pour améliorer la détection d'intrusions dans les environnements SDN.

Chapitre 1

1.1. Contexte et problématique

1.1.1. L'évolution des réseaux et l'émergence du paradigme SDN

Nous observons que l'évolution accélérée des technologies de communication et la généralisation des services numériques ont profondément transformé les infrastructures réseau.[\[1\]](#) Cette transformation s'accompagne, selon nous, d'une augmentation exponentielle du volume, de la diversité et de la complexité du trafic échangé. Les architectures réseau traditionnelles, reposant sur des équipements physiques aux configurations statiques et isolées, atteignent leurs limites face aux exigences contemporaines de flexibilité, d'adaptabilité et de réactivité. Pour répondre à ces défis, le paradigme du Software Defined Networking (SDN) s'impose comme une approche innovante. Il rompt avec le modèle historique en dissociant le plan de contrôle du plan de données (cf. section 1.2), permettant une gestion unifiée, programmable et dynamique de l'infrastructure.

1.1.2. La sécurité du SDN

Dans notre analyse, si la centralisation du contrôle constitue la force du SDN, elle en représente également le principal point de vulnérabilité [\[2\]](#) . Nous notons que les menaces sont multiples et spécifiques à cette architecture, comme l'ont démontré les

premières études de faisabilité d'attaques [3]. Elles incluent notamment les attaques par déni de service distribué visant à saturer le contrôleur ou le canal de communication avec les commutateurs [4], les injections de flux malveillants pour corrompre les tables de commutation, les élévations de privilèges via les interfaces de programmation [5], et les attaques sur la disponibilité du plan de contrôle [6]. Des vulnérabilités plus sophistiquées, comme les attaques par canal auxiliaire, ont également été identifiées [7]. Cette nouvelle donne sécuritaire, documentée dans des taxonomies détaillées [8], impose de repenser les mécanismes de protection, en particulier la détection d'intrusions, pour qu'ils soient intrinsèquement adaptés aux caractéristiques dynamiques et centralisées du SDN.

1.1.3. Les limites des systèmes de détection d'intrusions classiques

Dans les environnements SDN, les systèmes de détection d'intrusions hérités des réseaux traditionnels montrent rapidement leurs insuffisances. Les approches par signature sont inopérantes face aux menaces nouvelles ou polymorphes [9]. Les systèmes basés sur l'apprentissage automatique classique présentent également des limites dans la modélisation des dépendances complexes [10]. De plus, les modèles de deep learning (apprentissage profond) souffrent de problèmes d'interprétabilité [11] et de robustesse, ainsi que de sensibilité au déséquilibre des données [12].

1.2. Software Defined Networking (SDN)

Le Software Defined Networking (SDN), pouvant être traduit par réseau défini par logiciel représente une évolution majeure dans la conception et la gestion des réseaux. Il

s'agit d'un paradigme architectural qui sépare le plan de contrôle (control plane) du plan de données (data plane), dans le but d'apporter davantage de programmabilité, de souplesse et de centralisation au sein des infrastructures réseau [\[13\]](#), [\[14\]](#).

Dans cette approche, la logique de décision, qu'il s'agisse du routage, de la mise en œuvre des politiques ou de la sécurité, est concentrée au niveau d'un contrôleur logiciel centralisé. Les équipements du plan de données, quant à eux, se limitent à exécuter des règles de correspondance et d'action (« match-action ») transmises par le contrôleur. Cette séparation conceptuelle, popularisée à l'origine par le protocole OpenFlow [\[15\]](#), a ouvert la voie à une automatisation accrue, une orchestration simplifiée et une évolution continue des politiques réseau, même dans des environnements à grande échelle.

1.2.1. Principes fondateurs

Le SDN s'articule autour de trois principes fondamentaux [\[4\]](#), [\[14\]](#) : séparation des plans, programmabilité et abstraction.

La programmabilité repose sur deux catégories d'interfaces programmables (API). Les Northbound APIs exposent les capacités du contrôleur aux applications tierces, tandis que les Southbound APIs assurent la communication vers les équipements de commutation. Cette architecture bidirectionnelle confère au réseau flexibilité et réactivité face aux variations contextuelles.

Quant à l'abstraction, elle permet aux applications d'exprimer des politiques métier sans se préoccuper de l'infrastructure sous-jacente. Le contrôleur assume la responsabilité de compiler ces intentions abstraites en règles opérationnelles, automatisant ainsi la gestion et masquant la complexité technique.

1.2.2. Architecture logique à trois couches

La vision de référence de l'architecture SDN repose sur une organisation en trois couches principales, qui interagissent de manière hiérarchique et complémentaire [2], [14], [15]. Cette organisation est illustrée dans la Figure 1, qui présente la séparation des plans et les interfaces de communication entre les différentes couches.

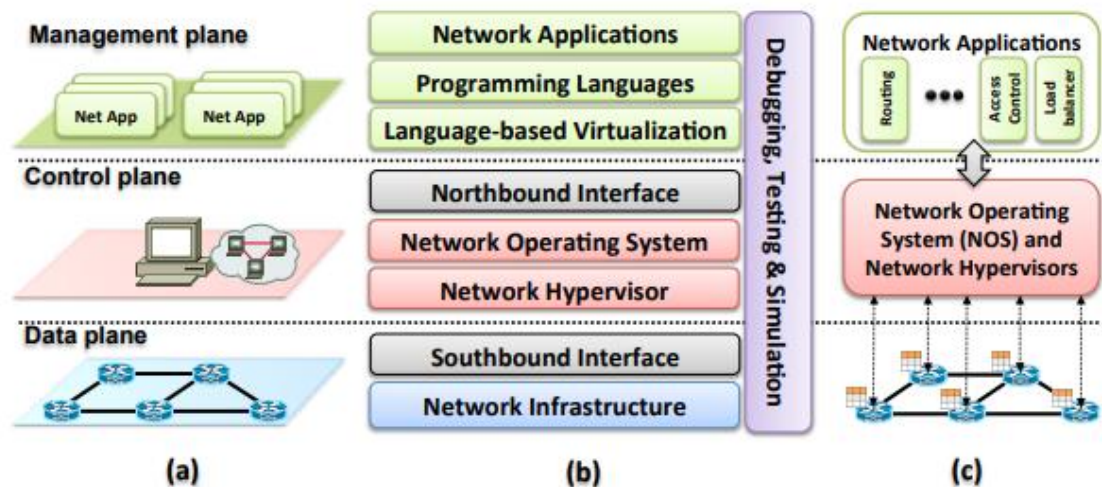


Figure 1 : Architecture logique du SDN et séparation des plans (données, contrôle et gestion) [2]

Comme le montre la figure 1 ci-dessus, la première est la couche application (SDN Apps). Elle regroupe l'ensemble des applications réseau de haut niveau, telles que celles dédiées au routage, à la sécurité (par exemple les systèmes de détection ou de prévention d'intrusion, IDS/IPS), à l'équilibrage de charge, à la gestion de la qualité de service (QoS) ou encore à l'ingénierie du trafic (TE). Ces applications communiquent avec le contrôleur à l'aide des interfaces Northbound APIs, souvent fondées sur des protocoles REST, afin

d'exprimer des politiques réseaux abstraits que le contrôleur se charge ensuite d'interpréter.

La seconde est la couche contrôleur, également appelée Network Operating System. Elle constitue le cœur logique du SDN, car elle assure la supervision et la coordination de l'ensemble du réseau. Le contrôleur (par exemple ONOS, OpenDaylight, Ryu ou Floodlight) maintient une vue globale de la topologie, de l'état des liens et des statistiques de trafic. Il traduit les intentions formulées par les applications en règles concrètes, puis les installe dans les équipements du plan de données à travers les interfaces Southbound APIs [\[14\]](#), [\[15\]](#).

Enfin, la troisième composante est la couche infrastructure ou Data Plane. Elle regroupe les commutateurs et routeurs programmables chargés d'exécuter les règles match-action définies par le contrôleur. Ces équipements appliquent directement les politiques réseau, tout en collectant des informations statistiques sur le trafic. Ils peuvent également remonter des événements ou anomalies au contrôleur, garantissant ainsi une boucle de rétroaction continue et une adaptation dynamique du réseau.

1.2.3. Plan de contrôle : contrôleur, état et API

Le contrôleur SDN occupe une place centrale dans l'architecture du Software Defined Networking, en agissant comme un véritable système d'exploitation réseau (Network Operating System) [\[14\]](#). Il assure la coordination entre les applications de haut niveau et les équipements du plan de données, garantissant ainsi une gestion cohérente, dynamique et centralisée de l'ensemble du réseau.

Son fonctionnement s'articule autour de plusieurs responsabilités fondamentales. Tout d'abord, il se charge de la gestion de l'état global du réseau. Pour cela, il maintient

en mémoire différentes bases d'état qui décrivent la topologie, les relations de voisinage, les tables de flux déjà installées, ainsi que diverses métriques de performance recueillies auprès des commutateurs [14]. Ces informations offrent au contrôleur une vision unifiée et actualisée du comportement du réseau.

Ensuite, le contrôleur expose des interfaces de programmation (Northbound APIs), souvent fondées sur des protocoles REST (Representational State Transfer) ou gRPC (Google Remote Procedure Call). Celles-ci permettent aux applications d'exprimer des intentions de haut niveau, par exemple isoler un segment du réseau, prioriser le trafic vocal ou encore limiter le débit d'un flux particulier. Ces intentions sont ensuite traduites en politique réseau exécutables, adaptées aux capacités des équipements sous-jacents.

Une autre fonction clé du contrôleur réside dans l'orchestration et la traduction des politiques. Il assure la compilation des politiques définies par les applications en règles concrètes (flow rules), compatibles avec les commutateurs programmables du plan de données. Cette étape garantit la cohérence entre la logique métier des applications et la mise en œuvre opérationnelle au niveau du réseau.

Enfin, le contrôleur doit faire preuve de résilience et d'évolutivité. Pour éviter qu'une panne du contrôleur ne paralyse l'ensemble du réseau, les contrôleurs modernes fonctionnent souvent en cluster. Ils mettent en œuvre des mécanismes tels que l'élection de leader, la réplication d'état ou le partitionnement de charge, ce qui permet d'assurer à la fois la cohérence des données et un basculement automatique (failover) en cas d'incident [14], [15].

1.2.4. Plan de données : tables “match–action” et pipeline

Le plan de données constitue la couche la plus basse de l'architecture SDN. Il regroupe les équipements physiques ou virtuels (commutateurs, routeurs, points d'accès, pare-feu)

chargés d'exécuter les règles de traitement de paquets définies par le contrôleur central [13], [4].

Ces équipements ne prennent plus de décisions de routage par eux-mêmes ; ils appliquent des politiques préétablies selon un modèle « match–action », structuré autour d'un pipeline de tables :

- Match : identification des paquets à partir de motifs inscrits dans les en-têtes (adresses MAC ou IP, ports de couche 4, VLAN, MPLS, etc.), de métadonnées ou encore de priorités de flux.
- Action : exécution des opérations correspondantes, telles que le transfert (*forward*), la suppression (*drop*), la modification d'en-tête (*rewrite*) ou encore la manipulation d'étiquettes (*push/pop VLAN ou MPLS*). Des actions plus complexes peuvent être définies, notamment les *group actions* (ex. : routage ECMP, *fast reroute*) et les *metering actions* pour la gestion du débit.
- Compteurs : collecte de statistiques par flux, groupe ou *meter* (nombre de paquets, octets transférés, durée d'activité, etc.). Ces compteurs sont périodiquement transmis au contrôleur afin d'alimenter la télémétrie et d'améliorer la prise de décision centralisée [13].

1.2.5. Le protocole OpenFlow

Le protocole OpenFlow constitue le pilier fondateur du paradigme Software Defined Networking (SDN). Il a été introduit pour offrir un canal standardisé de communication entre le plan de contrôle, représenté par le contrôleur SDN, et le plan de données, constitué des commutateurs programmables. Historiquement, cette interface a été formalisée par l'Open Networking Foundation (ONF) à partir des travaux pionniers de McKeown et al. (2008) [13].

OpenFlow définit un ensemble de messages, de structures de tables et de mécanismes d'échange permettant au contrôleur d'installer, modifier ou supprimer des règles de flux dans les équipements réseau. Cette capacité à manipuler dynamiquement le comportement des commutateurs constitue le fondement de la programmabilité du réseau.

1.2.6. Programmabilité et abstractions de haut niveau

La programmabilité représente la valeur fondamentale du paradigme SDN, car elle offre la possibilité de configurer, d'orchestrer et d'optimiser dynamiquement le comportement du réseau en fonction des besoins spécifiques des applications.

Au centre de ce concept, les Northbound APIs occupent un rôle stratégique : elles permettent aux applications de haut niveau d'exprimer leurs intentions politiques (intent policies) sous la forme d'objectifs déclaratifs. Par exemple, une application peut formuler une politique comme : « *autoriser la vidéo HD entre les nœuds A et B avec une latence inférieure à 50 ms* » [\[2\]](#).

Le contrôleur SDN agit alors comme un véritable compilateur de politiques : Il traduit ces intentions abstraites en ensembles cohérents de règles, résout les conflits de priorités entre applications concurrentes et préserve la cohérence globale du réseau, même en cas de changements dynamiques tels que des pannes, une congestion ou la mobilité des hôtes. Cette réactivité en temps réel confère au SDN une souplesse exceptionnelle face à la variabilité du trafic.

La programmabilité du SDN rend aussi possible des fonctionnalités comme la virtualisation du réseau (network slicing), le multi-tenancy, la reconfiguration rapide des flux ou encore la réaction automatique aux événements. Ainsi, lors de la réception d'un événement Packet-In, le contrôleur peut générer dynamiquement des règles temporaires afin d'isoler une anomalie ou de rétablir un chemin de communication optimal [\[2\]](#), [\[13\]](#).

Ces mécanismes illustrent la transformation profonde du réseau en une infrastructure pilotée par logiciel, où la logique décisionnelle n'est plus statique, mais contextuelle, adaptative et alignée sur les exigences opérationnelles des applications.

1.2.7. SDN vs réseaux traditionnels

Le paradigme SDN représente une rupture fondamentale par rapport aux architectures réseau traditionnelles. Pour mieux comprendre cette évolution, le Tableau 1 présente une comparaison synthétique des principales différences entre ces deux approches selon six dimensions clés : le plan de contrôle, la configuration, l'évolution, la visibilité, les politiques et l'innovation.

Tableau 1 : Comparaison réseaux traditionnels vs SDN (d'après [2], [14]).

Aspect	Réseaux traditionnels	SDN
Plan de contrôle	Distribué, embarqué dans chaque équipement	Centralisé/Logiquement centralisé (contrôleur/cluster)
Configuration	CLI par équipement, hétérogène	API et orchestration, automatisation
Évolution	Lente, dépendante du matériel	Rapide, par logiciel, indépendance matériel
Visibilité	Fragmentée, locale	Globale, télémétrie agrégée
Politiques	Règles locales, difficile à synchroniser	Intent-based, cohérence globale

Aspect	Réseaux traditionnels	SDN
Innovation	Long cycle (nouveau hardware)	Cycle court, innovation logicielle

Ce qu'on retient de ce tableau, c'est que là où les réseaux traditionnels reposent sur une logique d'équipements autonomes et de configurations manuelles propres à chaque nœud, le SDN déplace cette intelligence vers un point de contrôle unique et programmable. Cette centralisation n'est pas qu'un choix architectural : elle traduit une volonté de rendre le réseau plus réactif, plus cohérent et plus facile à faire évoluer sans dépendre du renouvellement du matériel.

1.3. Les menaces de sécurité dans les réseaux SDN

Nous constatons que, malgré les avantages indéniables qu'il apporte en matière de flexibilité, de gestion centralisée et d'automatisation, le Software Defined Networking (SDN) introduit de nouvelles vulnérabilités structurelles qui diffèrent sensiblement de celles observées dans les architectures réseau traditionnelles. Ces failles proviennent principalement de trois caractéristiques fondamentales du SDN : la centralisation du contrôle, l'ouverture des interfaces de programmation et la virtualisation des fonctions réseau [\[13\]](#), [\[17\]](#).

En cas de compromission du contrôleur, c'est l'ensemble du réseau qui devient vulnérable aux attaques, qu'il s'agisse de manipulation des règles de flux, d'interception du trafic, ou encore d'attaques par déni de service visant les mécanismes de décision.

L'ouverture des interfaces élargit de façon significative la surface d'exposition. Les Northbound APIs peuvent être exploitées par des applications malveillantes installées sur

le contrôleur, tandis que les Southbound APIs s'exposent à des risques d'usurpation d'identité, d'injection de règles non autorisées ou d'écoute illicite des communications entre le plan de contrôle et le plan de données.

Enfin, la virtualisation des fonctions réseau (NFV) et la mutualisation des ressources dans les environnements SDN introduisent de nouvelles formes de vulnérabilités. Celles-ci découlent de la cohabitation de multiples locataires, de la migration dynamique des fonctions, ainsi que de la forte dépendance au logiciel.

Ces caractéristiques font de la sécurité SDN un problème à part entière, qui ne peut pas être traité avec les mêmes outils que les réseaux traditionnels. Ainsi, bien que le SDN constitue une avancée majeure vers des réseaux programmables et adaptatifs, il redéfinit profondément les modèles de menace et impose une révision globale des approches de protection à tous les niveaux de la pile réseau.

1.3.1. La centralisation comme point de vulnérabilité

Le contrôleur SDN, véritable cerveau du réseau, représente à la fois la pierre angulaire et la principale vulnérabilité de l'architecture SDN. En centralisant les fonctions de décision et d'orchestration, il devient un point de défaillance unique (Single Point of Failure) [2]. Ainsi, une compromission du contrôleur peut entraîner des conséquences systémiques, susceptibles de perturber, voire de neutraliser, l'ensemble du réseau en un temps très court.

Les attaques visant le plan de contrôle exploitent principalement la forte dépendance du réseau à la connectivité et à la disponibilité du contrôleur. Elles peuvent se présenter sous plusieurs formes :

- Attaques de saturation du canal de contrôle (Control Plane Saturation Attacks) : ces attaques consistent à submerger le lien entre le contrôleur et les commutateurs par un volume massif de requêtes (souvent des messages Packet-In), ce qui épuise les ressources de traitement et dégrade les performances globales du contrôleur.
- Altération des règles de flux : un attaquant disposant d'un accès privilégié peut modifier, supprimer ou injecter des règles illicites dans les tables de flux. Ce type d'attaque conduit à un détournement du trafic, à une interruption des services ou à un déni de service global.
- Exfiltration d'informations sensibles : certaines attaques visent à extraire des informations critiques, telles que la topologie du réseau, les politiques de sécurité ou les métriques de performance, dans le but de préparer des intrusions ciblées.

Ces menaces prennent une ampleur particulière dans les architectures distribuées, où plusieurs contrôleurs coopèrent pour gérer le réseau. En l'absence de mécanismes robustes de cohérence d'état, d'authentification mutuelle et de synchronisation sécurisée entre contrôleurs, des incohérences peuvent survenir, ouvrant la voie à des attaques d'usurpation de rôle, de détournement de sessions ou de corruption d'état global [\[17\]](#).

1.3.2. Les attaques sur le plan de données

Le plan de données (Data Plane), chargé de l'application des politiques de routage et de sécurité définies par le contrôleur, constitue une cible privilégiée des attaquants. Deux grandes familles d'attaques se distinguent : l'usurpation et l'inondation. Les attaques par déni de service distribué (DDoS) exploitent souvent la manière dont les commutateurs SDN communiquent avec le contrôleur. Lorsqu'un paquet inconnu arrive, le commutateur envoie un message Packet-In au contrôleur afin d'obtenir une décision de routage.

Un attaquant peut générer un grand nombre de paquets aléatoires ou falsifiés, forçant ainsi le commutateur à multiplier ces requêtes ascendantes et provoquant la saturation du canal de contrôle. Cette surcharge empêche le contrôleur de traiter les flux légitimes, menant à une paralysie partielle ou totale du réseau [1].

Les attaques de type table flooding constituent un autre vecteur d'agression fréquent : elles remplissent les tables de flux des commutateurs avec des entrées inutiles ou aléatoires, consommant rapidement les ressources disponibles et dégradant les performances globales [18]. Ces attaques illustrent la fragilité du plan de données face à des comportements anormaux massifs dans les environnements SDN.

1.3.3. Les menaces sur les interfaces et les API

Les interfaces programmatiques (Northbound et Southbound APIs), conçues pour garantir la flexibilité et la programmabilité du SDN, représentent également des surfaces d'attaque critiques. Les API Northbound, souvent implémentées via des services REST ou gRPC, exposent les contrôleurs aux risques d'injection de commandes, de manipulation de politiques ou d'accès non autorisé à des configurations sensibles lorsqu'elles ne sont pas protégées par des mécanismes d'authentification forte et de contrôle d'accès.

De leur côté, les Southbound APIs, notamment OpenFlow, peuvent être la cible d'attaques de type Man-in-the-Middle (MITM) ou de rejeu (Replay Attacks), lorsque la communication entre le contrôleur et les commutateurs n'est pas correctement chiffrée ou authentifiée [17]. Ces attaques peuvent conduire à la modification frauduleuse des règles de flux ou à la redirection malveillante du trafic réseau.

1.3.4. Les menaces internes et les applications malveillantes

Le caractère ouvert et extensible du SDN, qui autorise le développement d'applications tierces, introduit également un risque interne majeur.

Une application malveillante ou simplement mal conçue peut déstabiliser la politique globale du réseau, consommer abusivement les ressources du contrôleur ou générer des règles incohérentes.

En l'absence de mécanismes d'isolation ou de gestion des privilèges d'accès entre les applications et le contrôleur, le SDN devient vulnérable aux abus de droits, aux erreurs logicielles ou à la corruption d'état interne. Les travaux de Kreutz *et al.* [2] soulignent que la majorité des attaques connues dans les environnements SDN résultent d'une mauvaise isolation entre les modules applicatifs ou d'une défaillance dans la gestion des autorisations d'accès.

1.4. La détection d'intrusions dans les environnements SDN

Nous relevons que les menaces croissantes qui ciblent les réseaux Software Defined Networking (SDN) ont conduit à un intérêt soutenu pour le développement de mécanismes de détection d'intrusions (Intrusion Detection Systems, IDS) conçus pour répondre aux contraintes propres à cette architecture.

Leur rôle est d'identifier proactivement les activités suspectes ou anormales avant qu'elles ne compromettent la confidentialité, l'intégrité ou la disponibilité du réseau.

Contrairement aux réseaux traditionnels où la surveillance se fait de manière segmentée, le SDN permet une observation unifiée et globale du trafic via le contrôleur, mais introduit aussi de nouveaux défis de performance et de scalabilité [13], [17].

1.4.1. Rôle et principes d'un système de détection d'intrusions (IDS)

Un système de détection d'intrusions (IDS) est un composant de surveillance qui vise à détecter les comportements anormaux, malveillants ou non autorisés affectant un réseau ou un hôte [2].

Il s'appuie sur l'analyse de flux, de paquets et d'événements afin d'identifier des signatures connues d'attaques ou des anomalies statistiques dans les modèles de trafic.

Deux approches principales se distinguent :

- Détection par signature : cette approche compare le trafic à une base de données d'attaques connues (comme Snort ou Suricata). Elle est efficace pour identifier des attaques déjà observées, mais inefficace face aux attaques inédites, polymorphes ou évolutives [9].
- Détection par anomalie : fondée sur des modèles statistiques ou apprenants (machine learning), elle détecte les déviations par rapport au comportement normal du réseau. Bien que cette approche soit mieux adaptée aux environnements dynamiques comme le SDN, elle peut générer un taux élevé de faux positifs [19].

Selon leur positionnement dans l'architecture SDN, les IDS peuvent être déployés de manière centralisée au niveau du contrôleur, de manière distribuée sur les commutateurs, ou selon une approche hybride combinant les deux niveaux. Chacune de ces

configurations présente des compromis distincts en termes de performance, de scalabilité et de résilience, que les travaux récents cherchent à optimiser.

1.5. Limites des approches existantes de détection dans le SDN

Nous estimons que, malgré les avancées notables réalisées dans la détection d'intrusions, les approches actuelles présentent encore plusieurs limites structurelles et opérationnelles lorsqu'elles sont appliquées aux environnements Software Defined Networking (SDN).

Ces limitations concernent à la fois la modélisation du trafic, la capacité de généralisation, la prise en compte des dépendances entre variables, et la robustesse face aux attaques évolutives [\[1\]](#), [\[2\]](#).

1.5.1. Difficulté de modéliser la dépendance entre variables

La majorité des modèles classiques de détection (SVM, réseaux de neurones, arbres de décision) traitent les caractéristiques réseau comme indépendantes ou faiblement corrélées, ce qui est rarement le cas dans les réseaux SDN.

Le trafic y présente des relations de dépendance complexes, notamment entre la durée des flux, la taille moyenne des paquets, le taux de transfert ou encore les intervalles inter-paquets.

Ces relations sont souvent non linéaires et varient selon les conditions réseau ou les politiques de routage [\[10\]](#).

Les modèles basés sur la corrélation linéaire (Pearson) ou les distributions gaussiennes sont incapables de saisir ces interdépendances. Ils tendent donc à sous-estimer certains signaux faibles d'attaque.

L'apprentissage profond peut, dans certains cas, extraire implicitement ces dépendances complexes, mais au prix d'un manque total d'interprétabilité, rendant difficile la validation des décisions de sécurité [\[11\]](#).

1.5.2. Problèmes liés au déséquilibre des données

Un autre défi majeur réside dans le déséquilibre des classes observé dans les jeux de données de détection d'intrusions.

Les déséquilibres dans les ensembles de données souvent utilisés poussent les modèles supervisés à favoriser la classe majoritaire, ce qui conduit à des performances trompeuses: une précision globale élevée (accuracy) mais un rappel très faible pour la classe minoritaire [\[12\]](#).

Des techniques de rééchantillonnage (oversampling ou undersampling) peuvent partiellement atténuer ce problème, mais elles modifient la distribution statistique réelle du trafic et altèrent la capacité du modèle à généraliser. Dans le contexte SDN, où la rapidité et la stabilité des décisions sont cruciales, ces approches demeurent peu viables à grande échelle [\[12\]](#).

1.5.3. Sensibilité aux variations du trafic et aux attaques inconnues

Les modèles supervisés (SVM, Random Forest, réseaux de neurones profonds) dépendent fortement de données d'entraînement étiquetées. Or, dans un environnement SDN, les signatures et comportements d'attaques évoluent constamment ; une attaque DDoS ou un scan peut changer radicalement de profil selon la topologie ou la charge réseau [9]. Ces modèles deviennent alors rapidement obsolètes face aux attaques inédites (zero-day) ou aux variantes d'attaques connues.

Les approches basées sur des règles statiques ou des seuils fixes souffrent du même problème de rigidité : elles ne peuvent pas s'adapter aux variations naturelles du trafic, entraînant un taux de faux positifs élevé et une fiabilité opérationnelle limitée [21].

1.5.4. Manque d'interprétabilité et de robustesse des modèles de deep learning

Les modèles basés sur le Deep Learning (CNN, LSTM, Auto-encoders) ont permis des progrès notables, mais présentent plusieurs faiblesses critiques dans un environnement SDN [36].

- Complexité computationnelle élevée : ils nécessitent des ressources importantes (GPU, mémoire), difficilement compatibles avec une exécution temps réel au sein du contrôleur SDN.
- Manque d'interprétabilité : leur fonctionnement en “boîte noire” rend difficile la justification d'une alerte ou la compréhension d'une classification.
- Vulnérabilité aux attaques adversaires : de légères perturbations dans les caractéristiques d'entrée peuvent suffire à tromper le modèle.
- Forte dépendance au jeu de données : ces modèles perdent souvent leur efficacité lorsqu'ils sont appliqués à des environnements différents de celui sur lequel ils ont été entraînés.

Ce manque de transparence et de robustesse compromet la confiance et la traçabilité nécessaires aux systèmes de sécurité critiques.

1.5.5. Insuffisance des approches hybrides et multi-niveaux

Pour pallier ces limites, des modèles hybrides combinant plusieurs techniques (par exemple SVM + PCA, Random Forest + Deep Learning, ou CNN + RNN) ont été proposés.

S'ils améliorent ponctuellement les performances, ils introduisent de nouvelles difficultés:

- Complexité accrue de la phase d'entraînement et de l'inférence,
- Risque d'incohérence décisionnelle entre modules hétérogènes,
- Absence de fondement théorique solide pour le choix des combinaisons et des paramètres [\[24\]](#).

En outre, la plupart de ces approches restent limitées par une représentation linéaire des dépendances entre variables, ne permettant pas une modélisation fidèle des interactions multivariées caractéristiques du trafic SDN [\[10\]](#).

Conclusion

Dans ce premier chapitre nous avons essayé d présenter le contexte et les enjeux de sécurité propres aux réseaux SDN. Nous y avons exposé les vulnérabilités induites par la centralisation du contrôle et la dynamique du trafic, tout en mettant en lumière les limites des approches de détection d'intrusions existantes face à ces défis spécifiques. Dans le chapitre suivant nous allons passer en revue quelques solutions de détection

existantes suite à une analyse de la littérature et situer notre travail par rapport à cet ensemble.

Chapitre 2

État de l'art

Face aux vulnérabilités propres à l'architecture SDN présentées au chapitre précédent, la communauté scientifique a développé un large éventail de méthodes de détection d'intrusions, dont nous dressons ici un état de l'art critique.

Plusieurs études ont recensé les menaces majeures propres à l'architecture SDN. Pascoal et al. [8] identifient les attaques de saturation de canal entre commutateurs et contrôleur comme une menace critique pour la stabilité.

Les impacts de ces attaques sont multiples : perte de disponibilité du service, altération des politiques de sécurité et compromission des flux utilisateurs. Dans un réseau SDN, une attaque réussie sur le contrôleur peut compromettre la totalité du réseau, contrairement aux architectures traditionnelles plus segmentées [2]. Par ailleurs, les solutions de sécurité classiques, telles que les pare-feux et les systèmes de détection d'intrusions (IDS) basés sur des signatures, montrent leurs limites face au caractère dynamique et distribué des flux [9].

Face à ces limites, des approches capables de modéliser les dépendances complexes du trafic s'imposent. L'apprentissage automatique et les approches statistiques avancées, telles que les copules, constituent ainsi une réponse prometteuse à la détection d'anomalies émergentes dans les réseaux SDN

2.1. Méthodes classiques de détection et de prévention des intrusions

Avant l'émergence des approches basées sur l'intelligence artificielle, la sécurité dans les réseaux SDN reposait principalement sur des mécanismes déterministes tels que les pare-feux, la vérification de politiques, ou encore les systèmes de détection d'intrusions (IDS) fondés sur des signatures. Héritées des architectures réseau traditionnelles, ces méthodes s'appuyaient sur des règles statiques et la reconnaissance de motifs connus. Leur principe consistait à comparer le trafic observé à des modèles d'attaques répertoriés dans des bases de signatures afin de bloquer les comportements malveillants déjà identifiés [9]. Cependant, avec la mutation rapide des menaces et l'apparition d'attaques polymorphes ou inédites, leur efficacité s'est progressivement affaiblie.

Les premières solutions appliquées aux environnements SDN ont introduit la vérification formelle des politiques et configurations du contrôleur. Cette approche repose sur la modélisation mathématique des flux et des règles de routage, permettant de détecter, avant leur déploiement, d'éventuelles incohérences ou violations de politiques de sécurité. Des outils tels que VeriFlow et FlowChecker ont vu le jour dans ce cadre, facilitant la détection en temps réel d'erreurs de configuration au sein des tables de flux. Ces mécanismes ont contribué à renforcer la fiabilité du plan de contrôle en garantissant la cohérence et l'isolation des règles OpenFlow [22], [23]. Néanmoins, leur portée demeure limitée face à des menaces plus dynamiques et massives, notamment les attaques par déni de service distribué (DDoS), qui exploitent la réactivité du contrôleur et la variabilité du trafic propre aux environnements SDN.

Les IDS basés sur des signatures représentent une autre famille importante de solutions classiques. Leur principe repose sur la comparaison du trafic capturé avec des motifs caractéristiques d'attaques connues. Bien qu'efficaces pour identifier les menaces déjà

documentées, ces systèmes présentent une vulnérabilité marquée face aux attaques nouvelles ou altérées. Dans le contexte SDN, cette dépendance à des bases de signatures rigides devient particulièrement problématique : la centralisation du contrôle accroît la charge de traitement et ralentit la détection [9]. Pour y remédier, certaines approches distribuées basées sur la surveillance des flux OpenFlow ont été proposées afin d'améliorer la résilience et la réactivité du système. Malgré ces avancées, ces solutions continuent de générer un nombre élevé de faux positifs, surtout dans les réseaux à fort trafic, et demeurent tributaires de la mise à jour manuelle des règles et signatures [10].

Les approches traditionnelles butent sur plusieurs limites que l'on retrouve de manière récurrente dans la littérature. Elles s'adaptent difficilement à la variabilité du trafic des environnements SDN, reposent sur des mécanismes statiques incapables d'évoluer avec les menaces, et nécessitent une mise à jour constante et coûteuse des bases de signatures [21]. Ces contraintes ont conduit la communauté scientifique à s'orienter vers des approches plus dynamiques et adaptatives, fondées sur l'analyse statistique et l'apprentissage automatique, capables d'identifier les comportements anormaux sans dépendre exclusivement de modèles prédéfinis.

2.2. Approches basées sur l'apprentissage automatique et profond

L'évolution rapide et la diversification des menaces au sein des réseaux SDN ont progressivement rendu obsolètes les méthodes déterministes reposant sur des signatures ou des règles statiques. Pour faire face à la dynamique du trafic et à la complexité grandissante des attaques, la recherche s'est orientée vers des approches fondées sur l'apprentissage automatique (Machine Learning, ML) et l'apprentissage profond (Deep Learning, DL).

Ces techniques s'appuient sur la reconnaissance de schémas comportementaux plutôt que sur la simple comparaison à des signatures d'attaques connues. Elles permettent ainsi

une détection plus adaptative et réactive, capable d'identifier des menaces nouvelles ou évolutives [9].

Les premiers travaux dans ce domaine ont utilisé des modèles d'apprentissage supervisé tels que le Support Vector Machine (SVM), les arbres de décision, ou encore les forêts aléatoires (Random Forest). Ces algorithmes apprennent à différencier le trafic normal du trafic malveillant à partir de jeux de données étiquetés. Des travaux ont proposé des versions améliorées du SVM intégrant une sélection hiérarchique des caractéristiques [24], atteignant des précisions de détection élevées..

De leur côté, Taheri et al. [25] ont mis au point une approche hybride combinant Random Forest et K-Nearest Neighbors (KNN), démontrant une robustesse accrue face aux attaques de type DDoS.

Malgré ces succès, ces modèles présentent certaines limitations récurrentes : ils requièrent une quantité importante de données labellisées, sont sensibles aux déséquilibres entre classes (le trafic normal étant souvent minoritaire), et généralisent difficilement face à des schémas d'attaque encore inconnus.

L'introduction de l'apprentissage profond a permis de surmonter une partie de ces limites. Les réseaux de neurones convolutifs (CNN) et récurrents (RNN) se sont révélés efficaces pour extraire automatiquement des caractéristiques spatio-temporelles du trafic SDN.

Ainsi, Keerthy et al. [26] ont montré qu'un modèle combinant CNN et RNN permettait d'obtenir d'excellents taux de détection, en particulier pour des attaques à latence variable. De leur côté, Domínguez-Dorado et al. [27] ont démontré la pertinence des auto-encodeurs profonds pour la détection d'anomalies dans des volumes massifs de flux issus d'environnements SDN.

Ces approches réduisent la dépendance à l'extraction manuelle de caractéristiques, mais leur complexité computationnelle et leur consommation élevée de ressources

matérielles limitent leur déploiement à grande échelle, notamment pour une analyse en temps réel.

Afin d'améliorer la performance et la résilience des systèmes de détection, plusieurs études récentes ont exploré des modèles hybrides combinant apprentissage supervisé et non supervisé. Par exemple, Taheri et al. [25] et Aldahmani et al. [28] ont expérimenté des architectures associant réseaux neuronaux et algorithmes de clustering adaptatif, permettant d'identifier des comportements anormaux inédits sans supervision préalable.

D'autres travaux ont introduit des architectures auto-apprenantes, capables d'ajuster dynamiquement leurs paramètres selon l'évolution du trafic, sans nécessiter de ré étiquetage manuel des données [29].

Ces modèles hybrides présentent une meilleure tolérance aux attaques polymorphes et améliorent la détection d'activités anormales dans des environnements SDN en constante mutation.

2.3. Solutions récentes : blockchain, apprentissage fédéré et architectures hybrides

L'accélération de l'évolution des menaces et la sophistication croissante des attaques ciblant les réseaux SDN ont conduit la recherche à se tourner vers des approches décentralisées et collaboratives, fondées sur des technologies émergentes telles que la blockchain, l'apprentissage fédéré (Federated Learning, FL) et les architectures hybrides intelligentes.

Ces nouvelles orientations visent à renforcer la sécurité, la confiance et la résilience du réseau tout en dépassant les limites structurelles du modèle centralisé. Elles s'inscrivent dans une logique de cybersécurité distribuée, où la transparence, la

protection des données et la coopération inter-nœuds deviennent des principes essentiels du fonctionnement global.

L'intégration de la blockchain dans les environnements SDN a été envisagée comme une réponse directe à la vulnérabilité du contrôleur centralisé.

En agissant comme un registre distribué et immuable, la blockchain garantit la traçabilité et la vérification des transactions entre les différents nœuds du réseau, empêchant toute modification non autorisée des politiques de routage ou des tables de flux. Ainsi, Xu et al. [30] ont proposé un modèle SDN-Blockchain dans lequel les règles OpenFlow sont enregistrées dans un registre partagé, assurant l'intégrité et l'authenticité des décisions prises par le contrôleur. De leur côté, Li et al. [31] ont introduit un mécanisme de détection collaborative reposant sur des smart contracts pour valider les décisions de sécurité entre plusieurs contrôleurs SDN.

Ces approches améliorent la fiabilité et la transparence du plan de contrôle, mais elles engendrent un surcoût en latence et en complexité computationnelle, ce qui limite leur adoption dans des contextes à fortes contraintes temporelles.

En parallèle, l'apprentissage fédéré (FL) a émergé comme une solution prometteuse pour entraîner des modèles de détection de manière collaborative, sans nécessiter le partage direct de données sensibles entre domaines SDN. Dans cette configuration, chaque contrôleur participe à la formation d'un modèle global en échangeant uniquement les paramètres d'apprentissage (comme les gradients), tout en conservant localement les données de trafic.

Zhang et al. [32] ont conçu un système de détection basé sur le FL pour identifier les attaques inter-domaines, réduisant la dépendance à un serveur central tout en préservant la confidentialité des données locales.

De leur côté, Ma et al. [33] ont proposé une architecture hiérarchique dans laquelle les contrôleurs locaux n'échangent que des mises à jour agrégées, optimisant la bande

passante et minimisant les risques de fuite d'informations. Ces solutions offrent un équilibre intéressant entre précision de détection, respect de la vie privée et réduction des coûts de communication, faisant de l'apprentissage fédéré une approche crédible pour la sécurité des SDN distribués.

Les architectures hybrides constituent enfin une tendance majeure visant à exploiter les forces combinées de plusieurs technologies. Par exemple, Qazi et al. [34] ont proposé une architecture intégrant la blockchain et les réseaux neuronaux convolutifs (CNN) dans un cadre SDN, afin de garantir à la fois l'authenticité des règles de routage et la détection d'anomalies en temps réel.

Dans une autre étude, Nguyen et al. [35] ont combiné l'apprentissage profond avec les copules statistiques, permettant de capturer plus finement les dépendances complexes entre les variables du trafic SDN, améliorant ainsi la précision et la robustesse de la détection.

Ces approches hybrides visent à concilier sécurité, performance et interprétabilité, mais leur mise en œuvre à grande échelle demeure un défi, notamment en raison de la coordination entre contrôleurs et de la nécessité d'assurer une cohérence inter-domaines.

Ces approches décentralisées représentent un tournant vers une cybersécurité SDN moins dépendante d'un contrôle centralisé. Elles traduisent la transition progressive d'un modèle de détection centralisé vers un modèle coopératif et distribué, mieux adapté à la nature dynamique et évolutive des réseaux modernes.

Malgré leur sophistication croissante, ces approches se heurtent à deux obstacles récurrents. D'un côté, le fort déséquilibre entre les classes dans les jeux de données de détection d'intrusions incite les modèles supervisés à favoriser la classe majoritaire, ce qui donne une précision globale trompeuse au détriment de la détection des attaques les plus subtiles. [12]. D'autre part, les architectures de deep learning, bien qu'efficaces sur le plan de la performance brute, exigent une infrastructure matérielle importante difficilement

compatible avec les contraintes de latence propres aux contrôleurs SDN [\[36\]](#). Ces deux limites justifient l'exploration d'approches statistiques alternatives, capables de modéliser explicitement les dépendances entre variables sans alourdir le coût computationnel.

2.4. Approche fondée sur la modélisation statistique des dépendances

Face à ces limites, nous constatons que la recherche s'est progressivement orientée vers des approches statistiques rigoureuses, capables de modéliser les dépendances entre variables sans imposer d'hypothèses trop restrictives.

Parmi elles, nous considérons que la théorie des copules s'impose comme une alternative bien adaptée à ce type de données. Introduite par Sklar (1959), cette approche repose sur la dissociation entre les distributions marginales des variables et la structure de dépendance qui les relie. En d'autres termes, une copule permet de représenter les relations multivariées entre plusieurs variables aléatoires, indépendamment de leurs distributions individuelles.

Cette propriété confère aux copules une grande flexibilité pour décrire des phénomènes complexes et hétérogènes, tels que le trafic réseau, où les interactions entre variables sont souvent non linéaires et asymétriques. Le trafic SDN présente précisément ce type de distributions : hétérogènes, asymétriques, difficiles à réduire à un modèle unique.

Dans le contexte du SDN, les paramètres du trafic présentent des interdépendances non linéaires difficiles à modéliser par les approches classiques. Les copules permettent de surmonter cette contrainte en séparant la modélisation des marges (comportement individuel des variables) de celle de la structure de dépendance globale [\[35\]](#), [\[37\]](#).

On distingue principalement trois familles de copules :

- Les copules elliptiques (comme la gaussienne ou la t-Student), adaptées aux dépendances continues et symétriques
- Les copules archimédiennes (Clayton, Gumbel, Frank), capables de modéliser des relations asymétriques
- Et les copules hiérarchiques ou vine, qui permettent de décrire des dépendances complexes à grande dimension à partir de relations bivariées successives.

Les travaux de Nguyen et al. [\[35\]](#) ont montré que la copule t-Student offre une meilleure robustesse dans la représentation des dépendances extrêmes, notamment lors d'attaques DDoS, où plusieurs variables du trafic évoluent simultanément de manière corrélée.

De leur côté, Tang et al. [\[37\]](#) ont proposé une approche hybride combinant copules gaussiennes et apprentissage profond, obtenant une amélioration significative du taux de détection ainsi qu'une réduction du nombre de faux positifs.

La modélisation par copules s'avère donc plus précise que les corrélations linéaires pour ce genre de données. Un atout majeur des copules réside également dans leur interprétabilité. Contrairement aux modèles neuronaux, elles permettent une analyse explicite de la dépendance entre variables à l'aide de mesures statistiques telles que les coefficients de Kendall (τ) ou de Spearman (ρ), qui évaluent la force et la direction des corrélations [\[37\]](#). Ces indicateurs offrent une lecture claire des relations entre variables, facilitant la compréhension des comportements anormaux et la validation des alertes générées.

De plus, les copules s'intègrent naturellement dans des modèles probabilistes comme les réseaux bayésiens ou les modèles de mélange, permettant la création de systèmes de détection fondés sur la vraisemblance. Ces modèles évaluent la probabilité qu'un flux observé appartienne à un comportement normal ou anormal, sans dépendre de seuils arbitraires ni d'étiquettes préexistantes.

Enfin, plusieurs études récentes ont exploré la combinaison des copules avec des techniques d'apprentissage automatique telles que la régression logistique, les forêts aléatoires ou les auto-encodeurs [35], [37]. Ces modèles hybrides conjuguent la puissance prédictive du machine learning à la rigueur statistique des copules, produisant des détecteurs d'anomalies performants, interprétables et résistants aux déséquilibres de données.

L'application des copules à la détection d'intrusions permet de construire un modèle probabiliste précis du comportement normal du trafic en capturant ses interdépendances complexes, d'identifier les anomalies comme des observations dont la structure de dépendance conjointe s'écarte significativement de la norme établie [48], et d'offrir une interprétation statistique des alertes via des mesures de dépendance comme le tau de Kendall ou le rho de Spearman, améliorant ainsi la transparence du système.

La théorie des copules offre ainsi un cadre statistique solide pour la détection d'intrusions dans les réseaux SDN. Elle propose un compromis optimal entre performance, explicabilité et généralisabilité, constituant ainsi le socle théorique du modèle développé dans cette recherche, présenté en détail dans le chapitre suivant. Cette approche vise donc à combler le fossé entre la flexibilité de l'apprentissage automatique et la rigueur des modèles probabilistes, pour aboutir à un système de détection d'intrusions à la fois performant, robuste et interprétable, spécifiquement conçu pour les défis sécuritaires du SDN.

Conclusion

Notre revue de littérature nous a conduit à analyser l'évolution des méthodes de détection, des approches classiques aux solutions récentes fondées sur l'apprentissage

automatique et les architectures hybrides. Nous avons identifié une lacune commune : la difficulté à modéliser explicitement les dépendances complexes et non linéaires entre les variables du trafic SDN. C'est précisément pour répondre à ce besoin que nous introduisons, dans le chapitre suivant, notre cadre conceptuel fondé sur la théorie des copules, en justifiant son choix et en formalisant nos objectifs de recherche.

Chapitre 3

Cadre conceptuel et orientation de la recherche

3.1. Objectif général

L'objectif central de notre recherche est de concevoir un modèle de détection d'intrusions fondé sur la théorie des copules statistiques, spécifiquement adapté aux caractéristiques des réseaux définis par logiciel. Notre approche repose sur la capacité des copules à modéliser les dépendances multivariées du trafic, là où les méthodes déterministes échouent.

Notre démarche méthodologique repose sur une modélisation probabiliste rigoureuse, dans laquelle nous étudions chaque variable selon sa distribution propre, tout en capturant les corrélations complexes à l'aide d'une copule adaptée. Ce choix nous permet de représenter plus fidèlement la structure intrinsèque des données SDN, souvent marquée par une forte variabilité temporelle et des interactions extrêmes.

Nous avons conçu notre modèle pour distinguer avec précision les comportements normaux et anormaux du trafic, en nous appuyant sur une évaluation de la vraisemblance statistique plutôt que sur des seuils arbitraires ou des hypothèses de linéarité. Cette approche favorise une adaptation naturelle aux variations du trafic et renforce la robustesse de notre système face à la diversité des attaques.

Trois objectifs spécifiques ont structuré la conception et l'évaluation du modèle :

- Améliorer la sensibilité du système dans la détection de comportements anormaux subtils, souvent dissimulés parmi des flux majoritairement légitimes
- Réduire le taux de faux positifs, en intégrant une structure de dépendance réaliste entre les variables réseau, afin de limiter les erreurs de classification liées aux fluctuations naturelles du trafic
- Renforcer l'interprétabilité statistique des résultats, grâce à l'utilisation de mesures de dépendance telles que les coefficients de Kendall (τ) et de Spearman (ρ), permettant une analyse post-détection plus explicite et une meilleure prise de décision pour les experts en cybersécurité.

3.2. Objectifs spécifiques

Pour la mise en œuvre concrète de notre objectif général, plusieurs objectifs spécifiques ont été formulés afin de structurer la démarche scientifique et d'assurer la cohérence méthodologique de notre étude :

1. Analyser les limites des approches de détection d'intrusions existantes dans le contexte des réseaux SDN, en identifiant leurs faiblesses face à la nature dynamique du trafic, au déséquilibre des classes et à la complexité des dépendances multivariées entre les différentes caractéristiques du réseau.
2. Sélectionner les variables les plus pertinentes et explicatives du trafic SDN à partir du jeu de données CICDDoS2019, à l'aide d'une méthode rigoureuse de sélection de caractéristiques, notamment la régression stepwise, afin d'assurer la représentativité et la pertinence statistique des paramètres retenus pour la modélisation.
3. Étudier et comparer différentes familles de copules statistiques dans le but d'identifier celle offrant la meilleure capacité de modélisation des relations de dépendance observées dans les données de trafic réseau.

4. Combiner la copule la plus performante avec un modèle de classification supervisé, en particulier la régression logistique, afin de concevoir un modèle hybride de détection d'intrusions capable de traduire la structure de dépendance en probabilité d'occurrence d'un comportement anormal.
5. Évaluer les performances du modèle proposé à l'aide d'indicateurs statistiques et opérationnels reconnus dans la littérature pour mesurer son efficacité, sa robustesse et sa capacité de généralisation dans divers scénarios d'attaque.
6. Comparer les performances du modèle copulaire aux approches classiques telles que les forêts aléatoires, les machines à vecteurs de support (SVM) ou les réseaux neuronaux légers, afin de démontrer la valeur ajoutée de la modélisation statistique des dépendances dans l'amélioration de la détection d'intrusions au sein des réseaux SDN.

3.3. Hypothèses de recherche

Pour structurer la démarche, plusieurs hypothèses ont été formulées en cohérence avec les objectifs définis.

Notre première hypothèse stipule que le trafic réseau dans les environnements SDN présente des relations de dépendance non linéaires et multivariées entre ses différentes composantes. Ces relations, souvent asymétriques et complexes, ne peuvent, selon nous, être convenablement capturées par les modèles de corrélation linéaire traditionnels, d'où la nécessité de recourir à des outils statistiques plus flexibles.

Notre deuxième hypothèse avance que la théorie des copules constitue un cadre méthodologique rigoureux et particulièrement approprié pour modéliser ces dépendances multivariées. Nous posons l'hypothèse qu'une telle modélisation offre une représentation

plus fidèle du comportement normal du trafic, tout en facilitant la détection des variations anormales.

Notre troisième hypothèse suggère que certaines familles de copules, notamment la copule t-Student, présentent une meilleure robustesse face aux distributions à queues épaisses et aux corrélations extrêmes, caractéristiques fréquemment observées lors des attaques réseau.

Notre quatrième hypothèse repose sur l'idée que l'intégration d'une copule adaptée dans un modèle de classification supervisé nous permettrait de concevoir un système hybride plus précis et plus stable, capable de réduire les faux positifs tout en préservant un haut taux de détection.

Enfin, notre cinquième hypothèse propose que l'utilisation d'une méthode de sélection de variables stepwise améliore la stabilité et la généralisabilité du modèle, en ne conservant que les attributs qui décrivent le mieux la structure du trafic.

Dans leur ensemble, ces hypothèses forment le socle conceptuel de notre recherche. Elles orientent la construction de notre modèle probabiliste explicatif, qui vise à relever les défis de la détection d'intrusions dans les réseaux SDN en conciliant performance, robustesse et interprétabilité statistique.

3.4. Justification du choix des copules

Notre recours aux copules comme fondement de la modélisation se justifie par leur aptitude à représenter de façon explicite, flexible et interprétable les relations de dépendance complexes entre les variables du trafic réseau. Nous considérons cet aspect comme déterminant pour la détection efficace d'intrusions dans les environnements Software Defined Networking (SDN).

3.4.1. Une modélisation explicite des dépendances

Cette propriété prend tout son sens dans le contexte des réseaux SDN, où les variables telles que le débit, la durée des flux ou la taille des paquets présentent des distributions hétérogènes et souvent non gaussiennes.

Contrairement aux approches traditionnelles, qui reposent sur l'hypothèse de dépendances linéaires, les copules permettent de capturer des corrélations non linéaires, asymétriques ou extrêmes, offrant ainsi une représentation plus réaliste du comportement du trafic réseau. Certaines familles de copules, telles que celles de Gumbel ou de Clayton, se distinguent par leur capacité à modéliser les phénomènes de queue extrême, un aspect essentiel pour la détection d'anomalies ou d'attaques massives, notamment de type DDoS.

Par ailleurs, les copules préservent la cohérence du modèle global, même lorsque les variables observées suivent des lois de probabilité différentes (exponentielle, gamma, log-normale, etc.). Elles offrent ainsi une forte robustesse face à la nature hétérogène et non stationnaire du trafic dans les environnements SDN, garantissant une modélisation stable et adaptable aux variations statistiques du réseau [\[38\]](#), [\[39\]](#).

3.4.2. Une approche explicable et fondée sur une validation empirique

Au-delà de leur puissance descriptive, les copules se distinguent par leur forte interprétabilité statistique, à la différence de nombreuses approches issues de l'apprentissage profond, souvent considérées comme des « boîtes noires ». Leur estimation permet de calculer des indicateurs de dépendance tels que les coefficients

de Kendall (τ) et de Spearman (ρ), qui offrent une lecture explicite des interactions entre les paramètres du trafic réseau. Ces mesures constituent un atout majeur pour une prise de décision transparente et justifiable, un aspect essentiel dans le domaine de la cybersécurité, où la traçabilité des alertes et la compréhension des comportements anormaux sont indispensables à la confiance du système.

3.5. Contribution scientifique attendue

Notre travail apporte une contribution à la fois méthodologique et théorique à la détection d'intrusions dans les réseaux SDN. Nous y proposons une approche innovante qui combine la modélisation copulaire multivariée et la classification supervisée, dans le but d'améliorer la précision, la robustesse et surtout l'interprétabilité des systèmes de détection d'intrusions.

D'un point de vue scientifique, notre recherche poursuit trois axes principaux :

- Introduire un cadre copulaire adapté à la cybersécurité, capable de représenter explicitement les dépendances non linéaires et asymétriques entre les caractéristiques du trafic réseau. Ce cadre permet une analyse approfondie des relations entre variables, offrant une vision plus réaliste du comportement du trafic et dépassant les limites des modèles linéaires traditionnels.
- Mettre en évidence l'impact des copules sur la fiabilité de la détection, notamment à travers la réduction du bruit statistique et du taux de faux positifs, grâce à une meilleure différenciation entre comportements normaux et anormaux.
- Proposer des modèles hybrides associant copule et régression logistique, Random Forest et XGBoost, alliant la rigueur mathématique de la modélisation statistique à la simplicité d'interprétation et à la rapidité d'exécution requises dans les environnements SDN en temps réel.

Les résultats attendus de notre recherche devraient nous permettre une meilleure compréhension des structures de dépendance cachées dans les flux réseau, et d'en exploiter les propriétés pour concevoir un système de détection adaptatif, explicable et cohérent sur le plan statistique.

À terme, notre contribution ambitionne de rapprocher les avancées théoriques issues de la statistique des applications opérationnelles en cybersécurité, ouvrant ainsi la voie à de nouveaux paradigmes de détection d'intrusions adaptés aux architectures SDN évolutives et dynamiques.

Conclusion

Nous avons défini dans ce chapitre le cadre conceptuel et les orientations de notre recherche. Nous y avons formulé nos objectifs généraux et spécifiques, ainsi que les hypothèses scientifiques qui sous-tendent notre démarche. La justification détaillée du recours aux copules comme fondement de notre modèle est désormais établie. Pour en permettre la mise en œuvre, le chapitre suivant sera consacré à l'exposé des fondements théoriques nécessaires à la compréhension de la modélisation copulaire et de ses métriques d'évaluation.

Chapitre 4

Fondements théoriques

4.1 Théorie des copules : définitions, familles, dépendances

4.1.1. Définition d'une copule

Une copule est une fonction mathématique qui permet de modéliser la dépendance entre plusieurs variables aléatoires, indépendamment de leurs distributions marginales. L'idée repose sur le théorème fondamental de Sklar (1959) [\[43\]](#), qui affirme que toute fonction de répartition conjointe continue peut-être exprimée en termes de ses marges et d'une copule :

Pour des variables continues $X_1, X_2, \dots, X_d$, avec fonction de répartition conjointe $F(x_1, \dots, x_d)$ et marges $F_1(x_1), \dots, F_d(x_d)$, il existe une copule C telle que :

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d))$$

Cette propriété est universelle pour toute dimension $d \geq 2$ (Sklar, 1959) [\[43\]](#).

4.1.2 Fonction de densité de copule

Si les variables sont continues et admettent une densité, on peut écrire la densité conjointe $f(x_1, \dots, x_d)$ comme :

$$f(x_1, \dots, x_d) = c(F_1(x_1), \dots, F_d(x_d)) \cdot f_1(x_1) \cdot \dots \cdot f_d(x_d)$$

où :

- f_i est la densité marginale de X_i ,
- c est la densité de la copule.

Cette décomposition est utile car elle sépare la structure de dépendance (copule) des comportements individuels (marges).

4.1.3 Familles de copules

A. *Copules elliptiques*

- Copule gaussienne (matrice de corrélation Σ)

$$C_{\Sigma}(u, v) = \Phi_{\Sigma}(\Phi^{-1}(u), \Phi^{-1}(v))$$

Φ_{Σ} : Fonction de répartition gaussienne multivariée.

Φ^{-1} : Inverse de la fonction de répartition normale standard.

- Copule t de Student (degrés de liberté ν , matrice Σ)

$$C_{\nu, \Sigma}(u, v) = t_{\nu, \Sigma}(t_{\nu}^{-1}(u), t_{\nu}^{-1}(v))$$

$t_{\nu, \Sigma}$: Fonction de répartition de Student multivariée.

t_{ν}^{-1} : Inverse de la fonction de répartition univariée de Student.

Ces copules modélisent des dépendances symétriques, adaptées aux situations où les extrêmes ne sont pas surreprésentés.

B. Copules archimédiennes

- Clayton (dépendance queue inférieure, $\theta > 0$) :

$$C_{\theta}(u, v) = (u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}$$

- Gumbel (dépendance queue supérieure, $\theta \geq 1$) :

$$C_{\theta}(u, v) = \exp(-((- \ln \ln u)^{\theta} + (- \ln \ln v)^{\theta})^{1/\theta})$$

- Frank (symétrique $\theta \neq 0$) :

$$C_{\theta}(u, v) = -\frac{1}{\theta} \ln \ln \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right)$$

Les copules archimédiennes conviennent bien aux modèles bivariés ou trivariés, surtout quand les dépendances sont non linéaires ou extrêmes.

4.2 Méthodes de construction de modèles avec copules

4.2.1 Approche statistique de la détection d'anomalies

La détection d'anomalies via copules repose sur la modélisation de la structure de dépendance normale des variables réseau (trafic, taille des paquets, etc.). Une déviation de cette structure signale une potentielle attaque (DDoS, intrusion, etc.).

Estimation des marges

Pour chaque variable X_i , estimer sa fonction de répartition marginale $\hat{F}_i$ (empirique ou paramétrique).

Exemple : Si X_i suit une loi normale $N(\mu, \sigma^2)$, alors :

$$\hat{F}_i(x) = \Phi\left(\frac{(x - \hat{\mu})}{\hat{\sigma}}\right)$$

où Φ est la fonction de répartition normale standard.

1. Transformation en variables uniformes

Uniformiser les données via :

$$U_i = \hat{F}_i(X_i)$$

Les U_i suivent une loi uniforme $U([0,1])$ sous l'hypothèse de normalité.

2. Estimation de la copule

Pour une **copule paramétrique** (Gaussienne, Clayton, etc.), estimer les paramètres par maximum de vraisemblance ou méthode des moments.

Exemple pour la copule Gaussienne :

$$\hat{\Sigma} = \underset{\Sigma}{\operatorname{argmax}} \sum_{k=1}^n \ln \ln(c_{\Sigma}(U_1^k, \dots, U_d^k))$$

Pour une **copule empirique**, utiliser la fonction de répartition empirique multivariée.

3. Calcul de la densité jointe

La densité conjointe des observations $x = (x_1, \dots, x_d)$ est :

$$f(x) = c(U_1, \dots, U_d) \cdot \prod_{i=1}^d f_i(x_i)$$

Une valeur anormalement faible de $f(x)$ (typiquement sous un seuil τ) indique une anomalie.

4.2.2 Métriques d'évaluation

Pour évaluer la performance du modèle, on utilise les métriques de classification binaire présentées dans le tableau 2 suivant :

Tableau 2 : Métriques d'évaluation

Métrique	Formule	Interprétation
Taux de détection (Recall)	$TPR = \frac{VP}{VP + FN}$	Proportion d'attaques détectées.
Taux de faux positifs	$FPR = \frac{FP}{FP + VN}$	Proportion de fausses alertes.
Précision	$Precision = \frac{VP}{VP + FP}$	Pertinence des alertes.
F1-Score	$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$	Équilibre entre précision et rappel.

Métrique	Formule	Interprétation
AUC-ROC	Aire sous la courbe ROC	Performance globale (idéal : AUC = 1).

- VP (Vrais Positifs) : Attaques correctement détectées.
- FP (Faux Positifs) : Trafic normal classé comme attaque.
- FN (Faux Négatifs) : Attaques non détectées.

4.3 Évaluation de la dépendance et choix des copules

4.3.1 Définition

- **Inversion de Sklar**

Lorsque F_1, F_2 sont continues avec inverses généralisés F_1^{-1}, F_2^{-1} ,

$$C(u, v) = F(F_1^{-1}(u), F_2^{-1}(v)),$$

ce qui permet de retrouver la copule à partir de la loi conjointe.

- **Bornes de Fréchet–Hoeffding**

Pour toute copule C et tout $(u, v) \in [0, 1]^2$,

$$\max(u + v - 1, 0) \leq C(u, v) \leq \min(u, v),$$

où $C^-(u, v) = \max(u + v - 1, 0)$ et $C^+(u, v) = \min(u, v)$. La borne inférieure C^- n'est une copule qu'en dimension 2.

- **Densité de copule et décomposition de la densité jointe**

Si F admet une densité f et les marges f_1, f_2 existent, alors

$$f(x_1, x_2) = c(F_1(x_1), F_2(x_2)) \cdot f_1(x_1) f_2(x_2), \quad c(u, v) = \frac{\partial^2}{\partial u \partial v} C(u, v).$$

C est la pierre angulaire pour les scores d'anomalie basés sur la densité.

Extension multivariée. En dimension n , une copule $C: [0,1]^n \rightarrow [0,1]$ satisfait (i) $C(u) = 0$ si l'une des coordonnées vaut 0 ; (ii) $C(1, \dots, 1, u_i, 1, \dots, 1) = u_i$; (iii) C est n -croissante. La version multivariée de Sklar demeure valide et C est unique si toutes les marges sont continues. Les bornes s'écrivent $C^-(u) = \max(\sum_{i=1}^n u_i - n + 1, 0)$ et $C^+(u) = \min_i u_i$.

4.3.2 Mesures de dépendance

Les mesures de dépendance permettent d'estimer et de comparer la structure de corrélation entre variables, en particulier lorsque les relations sont non linéaires ou asymétriques.

- **Tau de Kendall (τ)**

Le tau de Kendall mesure la concordance entre paires de variables :

$$\tau = P[(X_1 - X_2)(Y_1 - Y_2) > 0] - P[(X_1 - X_2)(Y_1 - Y_2) < 0]$$

Dans le cadre des copules :

$$\tau = 4 \int_0^1 \int_0^1 C(u, v) dC(u, v) - 1$$

Valeurs possibles : $\tau \in [-1, 1]$

- **Rho de Spearman (ρ_s)**

Basée sur les rangs :

$$\rho_s = 12 \int_0^1 \int_0^1 (uv - C(u, v)) du dv$$

Comme le tau de Kendall, elle est robuste aux distributions marginales.

- **Coefficient de queue (asymptote)**

Pour les dépendances extrêmes, on peut mesurer la probabilité que deux variables atteignent des valeurs extrêmes simultanément :

$$\lambda_U = P[Y > F_Y^{-1}(u) \mid X > F_X^{-1}(u)]$$

- ❖ Gumbel : forte dépendance dans la queue supérieure (valeurs extrêmes hautes).
- ❖ Clayton : forte dépendance dans la queue inférieure.

4.3.3 Estimation des copules (paramétrique & non paramétrique)

- Approche en deux temps (IFM / pseudo-vraisemblance).

Estimer les marges F_i (paramétriques ou empiriques), transformer en $U_i = F_i(X_i)$.

Estimer les paramètres de C par vraisemblance pseudo-maximale sur $(U_1, \dots, U_d)$

- Maximum de vraisemblance (MLE).

Si l'on suppose une famille C_θ et des marges F_i bien spécifiées, on maximise directement la log-vraisemblance jointe (une étape).

- Méthode des moments via τ/ρ .

Pour les copules archimédiennes, on peut relier θ à τ (p.ex. $\tau_{clayton} = \frac{\theta}{(\theta+2)}$, ou calibrer via ρ_S .

- Copule empirique.

Alternative non paramétrique basée sur la répartition empirique multivariée, utile lorsque la forme de dépendance est a priori inconnue.

Après avoir défini les fondements conceptuels de la recherche, le chapitre suivant décrit la méthodologie adoptée, les outils utilisés et le processus expérimental mis en œuvre pour construire le modèle proposé.

Conclusion

Les fondements théoriques des copules, de leur estimation et des mesures de dépendance associées ont été présentés. Cette base mathématique et statistique nous fournit à présent tous les outils nécessaires pour décrire, dans le chapitre suivant, la méthodologie concrète que nous avons adoptée. Nous y détaillerons les étapes opérationnelles, depuis le prétraitement des données jusqu'à la construction et l'évaluation de nos modèles hybrides.

Chapitre 5

Méthodologie

5.1 Introduction

Notre méthodologie expérimentale compare trois types d’approches : des modèles supervisés classiques (régression logistique, Random Forest, XGBoost) ; des détecteurs fondés exclusivement sur des copules (gaussienne, Student-t, archimédiennes) ; et des modèles hybrides combinant la copule Student-t avec les modèles supervisés (Student-t + régression logistique, Student-t + Random Forest, Student-t + XGBoost).

Pour chaque modèle, nous avons réalisé une validation croisée sur l’ensemble d’entraînement afin d’optimiser les hyperparamètres, puis évalué la performance finale sur l’ensemble de test. Nous avons enfin appliqué une évaluation multicritère AHP–TOPSIS pour établir un classement global, en prenant en compte simultanément l’AUC, le F1-score, la précision, le rappel et l’exactitude.

5.2 Prétraitement et préparation des données

La qualité du prétraitement conditionne directement les performances du modèle, ce qui est d’autant plus vrai en détection d’intrusions où la qualité des données influence directement la performance du modèle. Dans cette étude, nous avons utilisé le jeu de données CICDDoS2019. Ce jeu est réputé pour sa richesse et sa diversité, mais également pour sa complexité : il contient un grand nombre de variables hétérogènes, des valeurs manquantes, ainsi que des formats parfois non homogènes.

Pour ces raisons, nous avons mis en œuvre un nettoyage rigoureux pour obtenir une base de données cohérente et statistiquement exploitable.

5.2.1 Chargement et identification des variables

Dans notre approche, nous avons tout d’abord chargé les données à partir des fichiers CSV du dataset CICDDoS2019. Nous avons ensuite effectué une première analyse pour déterminer la structure du jeu de données, le nombre de colonnes, ainsi que la nature des variables (numériques ou non numériques). Une fonction spécifique (*identify_non_numeric_columns*) a permis d’isoler les colonnes non numériques, qui représentent généralement des adresses IP, des étiquettes textuelles ou des horodatages.

5.2.2 Traitement des variables non numériques

Le traitement des colonnes non numériques repose sur une logique adaptative. Lorsqu’une colonne contient un petit nombre de valeurs uniques (par exemple, un type de protocole ou une classe de trafic), elle est considérée comme catégorielle et transformée à l’aide d’un encodeur d’étiquettes (*LabelEncoder*). Ce dernier attribue un entier à chaque catégorie, ce qui conserve l’information tout en la rendant compatible avec les algorithmes numériques.

En revanche, lorsqu’une colonne textuelle présente trop de valeurs distinctes (comme une adresse IP), une tentative de conversion en format numérique est effectuée. Si cette opération échoue ou produit une colonne vide, la variable est supprimée pour éviter d’introduire du bruit ou des erreurs dans l’analyse.

5.2.3 Suppression des colonnes redondantes ou constantes

Certaines colonnes peuvent avoir une valeur constante ou quasi-constante, ce qui ne contribue en rien à la différenciation entre trafic normal et trafic anormal. Une vérification systématique est réalisée afin de supprimer ces variables redondantes, dont la variabilité est trop faible pour avoir un impact statistique. Cette étape réduit efficacement la dimension du jeu de données sans perte d'information pertinente.

5.2.4 Gestion des valeurs manquantes et infinies

Les valeurs manquantes ou infinies sont corrigées pour éviter toute instabilité dans les modèles. Pour chaque colonne numérique les valeurs infinies sont remplacées par la médiane des valeurs finies de la même colonne, évitant ainsi les biais extrêmes et les valeurs NaN (non définies) sont également imputées par la médiane, car cette mesure de tendance centrale est moins sensible aux valeurs aberrantes que la moyenne.

5.2.5 Normalisation et nettoyage final

Une fois les données nettoyées, une étape de standardisation est appliquée à l'aide du *StandardScaler*, afin de mettre toutes les variables sur une même échelle. Cette normalisation est indispensable, notamment pour les copules et la régression logistique, qui sont sensibles aux différences d'ordre de grandeur entre variables. Les noms de colonnes sont ensuite normalisés (suppression des espaces, des caractères spéciaux et conversion en minuscules) pour garantir une compatibilité totale avec les bibliothèques de modélisation.

5.3 Sélection de variables

5.3.1 Principe de la sélection Stepwise

La méthode stepwise (ou sélection progressive) repose sur un processus itératif qui combine deux approches complémentaires : l'ajout progressif de variables pertinentes (phase "forward"), et la suppression des variables redondantes ou non significatives (phase "backward"). Le principe est de construire progressivement un modèle de régression logistique en testant, à chaque itération, l'impact de l'ajout ou du retrait d'une variable sur la performance globale du modèle.

L'évaluation se fait à l'aide de la mesure AUC (Area Under the Curve), qui permet de quantifier la capacité du modèle à distinguer les deux classes. Une variable est ajoutée au modèle lorsqu'elle améliore significativement l'AUC, et supprimée lorsqu'elle n'apporte plus de gain ou qu'elle dégrade la performance.

5.3.2 Procédure de sélection

La procédure de sélection suit les étapes suivantes :

- Initialisation du modèle : On commence avec un modèle vide, sans aucune variable prédictive.
- Phase d'ajout (forward) : À chaque itération, chaque variable candidate est testée indépendamment. Celle qui offre la meilleure amélioration de l'AUC est ajoutée au modèle. Ce processus se répète tant qu'une amélioration mesurable est observée.
- Phase de retrait (backward) : Une fois le modèle stabilisé, les variables incluses sont réévaluées. Si la suppression d'une variable n'entraîne pas de baisse significative de performance, celle-ci est retirée. L'objectif est d'obtenir un modèle

parcimonieux, c'est-à-dire aussi simple que possible tout en maintenant une performance élevée.

- Critères d'arrêt : Le processus s'interrompt lorsque l'ajout ou le retrait de nouvelles variables ne modifie plus significativement l'AUC, ou lorsqu'un nombre maximal d'itérations est atteint.

5.4 Modélisation par copules

Dans cette section, nous allons décrire la démarche de modélisation par copules adoptée dans notre travail, depuis le choix de la famille de copule jusqu'à l'estimation des paramètres et l'intégration aux classifieurs.

5.4.1 Familles de copules utilisées

Afin de représenter différents types de dépendances, les six familles de copules présentées au chapitre 4 ont été implémentées et comparées : gaussienne, Student-t, Clayton, Gumbel, Frank et Vine.

5.4.2 Estimation des paramètres

Pour chaque type de copule, les paramètres de dépendance (par exemple, le coefficient de corrélation ou le paramètre θ) sont estimés par la méthode du maximum de vraisemblance. L'objectif est de trouver les valeurs qui maximisent la probabilité d'observer les données mesurées. Dans notre code, un mécanisme de robustesse est intégré afin d'effectuer plusieurs tentatives d'ajustement pour chaque copule, garantissant la convergence même lorsque les données présentent des comportements extrêmes ou des distributions irrégulières.

5.4.3 Calcul des densités de copule

Une fois les copules ajustées, leur densité de probabilité jointe est calculée pour chaque observation du jeu de données. Cette densité traduit le degré de vraisemblance d'un comportement donné selon la structure de dépendance observée dans les données normales. Les valeurs de densité très faibles correspondent à des comportements rares ou atypiques, pouvant être interprétés comme des anomalies potentielles.

5.4.4 Choix de la meilleure copule

Chaque copule est ensuite testée et évaluée sur le jeu de données. Le critère principal de comparaison est l'AUC (Area Under the Curve), calculée à partir de la courbe ROC. La copule offrant la meilleure capacité de discrimination entre trafic normal et trafic anormal est retenue comme modèle optimal pour l'étape suivante de détection d'anomalies.

Le choix de la meilleure copule est réalisé par une évaluation comparative exhaustive où chaque famille de copules est testée en mode simple (détection d'anomalies non supervisée). La performance est mesurée par l'AUC-ROC, permettant d'identifier la copule la plus adaptée à la structure de dépendance des données réseau.

5.5. Détection d'intrusions avec les copules

Après l'ajustement des différentes familles de copules sur les données normales, une étape clé de la méthodologie consiste à utiliser ces modèles pour détecter les comportements anormaux dans le trafic réseau. Cette détection repose sur la capacité des copules à estimer la vraisemblance d'un ensemble d'observations selon la structure de dépendance apprise à partir des données légitimes.

En d'autres termes, les copules sont ici utilisées comme un modèle de référence du comportement normal : tout écart significatif par rapport à ce modèle est interprété comme une anomalie potentielle, susceptible de correspondre à une tentative d'intrusion.

5.5.1 Principe général de détection

Le détecteur d'anomalies fondé sur les copules repose sur le concept de vraisemblance conditionnelle. Après avoir ajusté la copule sur un sous-ensemble de données considérées comme normales, on calcule pour chaque observation x_i la densité de probabilité associée à cette copule.

- Si la densité est élevée, l'observation est jugée conforme au comportement attendu (trafic normal).
- Si la densité est faible, cela indique que le point x_i est peu probable selon la structure de dépendance normale, ce qui le rend suspect ou potentiellement malveillant.

5.5.2 Entraînement du modèle

Le détecteur d'anomalies est entraîné exclusivement à partir des données normales afin de modéliser le comportement légitime du réseau. La première étape consiste à normaliser les variables à l'aide d'un StandardScaler, de manière à éliminer l'influence des différences d'échelle entre les caractéristiques. Cette homogénéisation garantit que chaque variable contribue équitablement à l'apprentissage du modèle.

Ensuite, la copule sélectionnée parmi les cinq familles étudiées est ajustée sur les données normalisées. Pour assurer la stabilité du processus, plusieurs ajustements successifs sont réalisés avec des paramètres initiaux variés jusqu'à obtenir une convergence satisfaisante.

Enfin, un seuil de normalité est défini afin de distinguer les comportements normaux des anomalies. Dans cette étude, il est fixé au 95^e percentile du score d'anomalie calculé sur les données normales. Ainsi, seules les observations appartenant aux 5 % les moins probables sont considérées comme suspectes, ce qui permet de concilier sensibilité et fiabilité dans la détection.

5.5.3 Prédiction et classification

Lors de la phase de prédiction, chaque nouvelle observation du trafic réseau est évaluée selon la densité fournie par la copule. Le score d'anomalie correspondant est comparé au seuil de normalité : Si le score dépasse le seuil, l'observation est classée comme anomalie (attaque) sinon, elle est considérée comme normale.

Ce mécanisme offre une détection entièrement basée sur la distribution jointe du comportement du trafic, sans dépendre de règles prédéfinies ni de signatures l'attaque connues.

5.6 Modèles hybrides Copules

À la suite des résultats obtenus, l'approche hybride a été étendue au-delà de la régression logistique pour inclure des méthodes d'apprentissage avancées comme Random Forest et XGBoost. Cette extension permet d'explorer des synergies plus complexes entre la modélisation statistique des copules et la puissance prédictive des algorithmes d'ensemble.

5.6.1 Principe des approches hybrides

Notre approche hybride repose sur une architecture à deux niveaux, combinant une modélisation statistique des dépendances entre variables et un apprentissage supervisé avancé. Nous visons ainsi à tirer parti des forces complémentaires de ces deux paradigmes afin d'améliorer la précision et la robustesse de la détection d'intrusions dans les réseaux SDN.

Le premier niveau repose sur une copule de Student, ajustée exclusivement sur les données normales. Grâce à sa capacité à modéliser les dépendances de queue, cette copule capture efficacement les relations extrêmes entre les attributs du trafic, souvent révélatrices d'événements anormaux. Elle fournit pour chaque observation un score de vraisemblance, représentant son degré de conformité au comportement normal du réseau. Ce score constitue une information statistique riche, capable de résumer la cohérence globale du trafic.

Le second niveau consiste à intégrer ce score dans un modèle supervisé, en complément des caractéristiques classiques du flux réseau. Trois variantes sont explorées :

- Hybrid Student–RF, qui associe la copule à un Random Forest. Cette combinaison permet de bénéficier de la stabilité et de la capacité du RF à capturer des interactions complexes non linéaires. Le score copule enrichit les arbres en renforçant leur aptitude à distinguer les comportements extrêmes.
- Hybrid Student–XGB, qui intègre le score dans un modèle XGBoost. Grâce à son mécanisme d’optimisation par gradient boosting, XGBoost exploite finement les signaux fournis par la copule pour renforcer la détection des comportements subtils ou faiblement représentés.
- Hybrid Student–LR, qui combine la copule à une régression logistique. Cette version conserve l’interprétabilité et la légèreté computationnelle du modèle linéaire, tout en améliorant sa capacité discriminante grâce au score copule incorporé comme variable supplémentaire.

5.6.2 Entraînement des modèles hybrides

L’entraînement des modèles hybrides repose sur une procédure méthodique articulée en plusieurs étapes successives. Dans un premier temps, seules les variables retenues lors de la phase de sélection sont conservées comme entrées du modèle. Cette réduction dimensionnelle permet d’éliminer les redondances, d’améliorer la stabilité statistique et de limiter la complexité computationnelle des algorithmes supervisés utilisés au second niveau.

La seconde étape consiste à ajuster la copule de Student, choisie pour sa capacité à modéliser les dépendances extrêmes caractéristiques du trafic réseau. Cet ajustement est réalisé exclusivement sur les observations normales, de manière à apprendre la structure de dépendance propre au comportement légitime du réseau. À partir de cette copule ajustée, un score d’anomalie est ensuite calculé pour chaque observation. Ce score, dérivé de la log-vraisemblance, reflète la cohérence statistique de chaque flux avec la distribution

normale du trafic : plus il est faible, plus l'observation est susceptible de correspondre à une activité malveillante.

Ce score copulaire est ensuite intégré au jeu de données initial sous la forme d'une nouvelle variable, produisant ainsi un ensemble enrichi combinant les caractéristiques réseaux sélectionnés et la dimension statistique issue de la modélisation par copule. Cette information supplémentaire permet de capturer la structure globale du trafic que les modèles supervisés ne peuvent saisir seuls.

Dans la dernière étape, chacun des modèles supervisés est entraîné sur cet ensemble enrichi. Pour garantir des performances optimales, les hyperparamètres sont ajustés à l'aide d'une validation croisée combinée à une recherche en grille, en utilisant l'AUC-ROC comme critère principal d'évaluation. Ce protocole assure une calibration rigoureuse des modèles et maximise leur capacité à distinguer efficacement le trafic légitime des comportements malveillants.

5.6.3 Prédiction et classification

Une fois le modèle hybride entraîné, la prédiction se fait en deux étapes :

- Pour chaque nouvelle observation, le modèle de copule calcule son score de vraisemblance.
- Ce score est ensuite combiné aux autres variables d'entrée et transmis à la régression logistique, qui renvoie une probabilité d'appartenance à la classe "attaque".

Une observation est finalement classée comme intrusion si cette probabilité dépasse un seuil fixé (généralement 0,5).

5.7 Évaluation multicritère avec AHP-TOPSIS

5.7.1 Méthode AHP (Analytic Hierarchy Process)

L'AHP permet de déterminer des poids cohérents pour les critères d'évaluation en fonction de leur importance relative. Dans notre étude, cinq critères ont été considérés :

- AUC-ROC (poids : 40,17%) : Critère principal de discrimination
- F1-Score (poids : 24,42%) : Équilibre entre précision et rappel
- Précision (poids : 13,73%) : Réduction des faux positifs
- Rappel (poids : 13,73%) : Détection des attaques
- Exactitude (poids : 7,94%) : Performance globale

5.7.2 Méthode TOPSIS (Technique for Order Preference by Similarity to Ideal Solution)

La méthode TOPSIS classe les alternatives en calculant leur distance relative par rapport à deux solutions de référence : la solution idéale positive, qui maximise tous les critères simultanément, et la solution idéale négative, qui les minimise. Chaque modèle reçoit un score compris entre 0 et 1 selon sa proximité à la solution idéale positive, ce qui permet un classement global objectif tenant compte de l'ensemble des critères pondérés.

Les poids utilisés ont été déterminés à partir d'une matrice de comparaison par paires construite selon les principes de l'AHP. Pour chaque paire de critères, une valeur de 1 à 9 a été attribuée selon l'importance relative de l'un par rapport à l'autre dans le

contexte spécifique de la sécurité réseau. L'AUC-ROC a reçu le poids le plus élevé (40,17 %) car elle mesure la capacité discriminative globale du modèle indépendamment du seuil de décision, une propriété essentielle dans un environnement où ce seuil peut être ajusté selon les contraintes opérationnelles. Le F1-score suit avec 24,42 %, reflétant l'importance d'équilibrer précision et rappel dans un contexte où les fausses alertes ont un coût opérationnel réel. La précision et le rappel ont été pondérés de manière égale (13,73 % chacun), traduisant l'absence de hiérarchie absolue entre les deux types d'erreurs dans ce cadre d'évaluation. L'exactitude globale, bien que complémentaire, a reçu le poids le plus faible (7,94 %) en raison de sa sensibilité au déséquilibre des classes, qui peut la rendre trompeuse. La cohérence de cette matrice a été vérifiée par le calcul du ratio de cohérence, dont la valeur obtenue est inférieure au seuil acceptable de 0,10.

5.7.3 Intégration dans l'évaluation

L'utilisation combinée d'AHP-TOPSIS répond au besoin d'une évaluation équilibrée des modèles, particulièrement importante dans le contexte de la sécurité réseau où différents critères peuvent être prioritaires selon le contexte opérationnel. Cette méthode dépasse la simple comparaison par AUC en intégrant une vision multicritère de la performance.

Conclusion

Nous avons exposé la méthodologie complète de notre étude, décrivant les processus de prétraitement, de sélection de variables, de modélisation par copules, et enfin de construction et d'évaluation des modèles hybrides. Le protocole expérimental étant

désormais défini, nous pouvons présenter et analyser, dans le chapitre suivant, les résultats obtenus par l'application de cette méthodologie sur le jeu de données CICDDoS2019.

CHAPITRE 6

RÉSULTATS ET ANALYSES

6.1 Présentation générale des résultats

Dans ce chapitre nous présentons en détail les résultats expérimentaux que nous avons obtenus à la suite de l'implémentation de la méthodologie de détection d'intrusions fondée sur les copules statistiques.

Les expériences ont été conduites sur le jeu de données CICDDoS2019, reconnu comme un standard dans le domaine de la cybersécurité pour l'évaluation des systèmes de détection d'intrusions. Ce dataset se distingue par sa richesse, sa diversité d'attaques et sa représentativité du trafic réseau réel, en particulier dans les environnements (SDN).

L'objectif principal de cette phase expérimentale était de valider l'efficacité et la pertinence de l'approche proposée à travers une série de tests comparatifs et d'analyses statistiques. L'évaluation des modèles repose sur un ensemble de métriques de performance robustes, choisies pour leur capacité à refléter fidèlement la qualité de la détection dans un contexte de déséquilibre marqué entre classes normales et malveillantes. Une attention particulière a été accordée au taux de faux positifs (False Positive Rate), car dans un système de sécurité opérationnel, un excès d'alertes erronées réduit considérablement la confiance des analystes et la réactivité du système.

6.2 Préparation et analyse des données

6.2.1 Caractéristiques du jeu de données

Le jeu de données CICDDoS2019 a été sélectionné pour cette étude en raison de sa fiabilité, de sa structure complète et de sa capacité à reproduire des scénarios réalistes de cyberattaques dans les réseaux modernes. Ce jeu regroupe différents types de trafic issus de simulations de DDoS et de connexions légitimes, en reproduisant le comportement d'un réseau SDN typique.

À l'issue du nettoyage, 63 variables ont été conservées. Les suppressions ont concerné les identifiants de flux non exploitables (Flow ID, adresses IP), les variables catégorielles non convertibles, les colonnes à variance nulle et les variables présentant des valeurs extrêmes non récupérables.

6.2.2 Partitionnement des données

Afin d'évaluer objectivement les performances des modèles, le jeu de données a été partitionné en deux sous-ensembles selon une répartition 70/30 : 15 400 observations constituent l'ensemble d'entraînement et 6 600 l'ensemble de test. Ce partitionnement repose sur un échantillonnage stratifié, garantissant que les proportions entre trafic normal et trafic malveillant sont fidèlement reproduites dans chacun des deux sous-ensembles, tel qu'illustré à la Figure 2 ci-dessous.

```
Donnees finales:  
Train: (15400, 62)  
Test: (6600, 62)  
Features communes: 62
```

Figure 2 : Distribution des données après partitionnement

Ce partitionnement assure que les deux ensembles reflètent fidèlement la structure du jeu complet, évitant toute distorsion dans la phase d'apprentissage ou de validation.

De plus, l'utilisation d'un échantillonnage stratifié renforce la robustesse de l'évaluation: les modèles sont testés dans des conditions similaires à celles rencontrées en production, où le déséquilibre entre trafic normal et malveillant est structurel.

6.2.3 Gestion du déséquilibre des classes

Le jeu de données présente un déséquilibre marqué entre les deux classes. Avant tout traitement, l'ensemble d'entraînement compte 15 918 observations d'attaque pour seulement 1 682 observations normales, soit un taux d'attaque de plus de 90 %, comme le montre la Figure 3. Une telle disproportion expose les modèles supervisés à un biais systématique : entraînés sur des données aussi déséquilibrées, ils tendent à favoriser la classe majoritaire au détriment de la détection des comportements normaux, ce qui nuit à leur capacité de généralisation.

Pour corriger ce biais, un rééchantillonnage a été appliqué exclusivement sur l'ensemble d'entraînement. La distribution résultante compte 7 959 observations normales et 9 948 observations d'attaque, ramenant le taux d'attaque à environ 56 %. Il est important

de souligner que l'ensemble de test n'a pas été modifié : il conserve la distribution originale du jeu de données, ce qui permet d'évaluer les modèles dans des conditions représentatives de la réalité opérationnelle, où les flux d'attaque restent largement majoritaires.

```
Distribution AVANT reechantillonnage:
Train: Classe 0=1682, Classe 1=15918
Taux attaque: 90.443%

Reechantillonnage des donnees d'entrainement...

Distribution APRES Reechantillonnage:
Train: Classe 0=7959, Classe 1=9948
Taux attaque: 55.554%
```

Figure 3 : Répartition des classes avant/après rééchantillonnage

6.3 Résultats de la sélection de variables

6.3.1 Performance des méthodes de sélection

La sélection de variables a été effectuée sur l'ensemble d'entraînement rééchantillonné, en deux étapes complémentaires : la méthode stepwise et la méthode RFE (Recursive Feature Elimination).

a) Méthode Stepwise

La méthode stepwise repose sur une approche itérative d'ajout et de retrait de variables, en évaluant à chaque étape la contribution de chaque attribut à la performance

globale du modèle (ici mesurée par l'AUC). Elle combine deux phases : une sélection avant (*forward*) où les variables les plus contributives sont ajoutées une à une, et une élimination arrière (*backward*) où les variables redondantes ou non significatives sont retirées.

L'algorithme a débuté avec un ensemble vide et a ajouté progressivement les attributs améliorants significativement l'AUC sur validation croisée.

Les itérations ont conduit aux résultats présentés dans la figure 4 suivante :

```
Methode 1: Selection Stepwise
Phase 1: Selection forward...
  Iteration 1: Ajout bwd_packets_s - AUC: 0.9262 (Amelioration: 0.4262)
  Iteration 2: Ajout fwd_header_length - AUC: 0.9435 (Amelioration: 0.0172)
  Iteration 3: Ajout bwd_packet_length_mean - AUC: 0.9641 (Amelioration: 0.0206)
  Iteration 4: Ajout min_packet_length - AUC: 0.9754 (Amelioration: 0.0114)
  Iteration 5: Ajout unnamed:_0 - AUC: 0.9835 (Amelioration: 0.0081)
  Iteration 6: Ajout bwd_packet_length_min - AUC: 0.9919 (Amelioration: 0.0084)
  Iteration 7: Ajout packet_length_variance - AUC: 0.9930 (Amelioration: 0.0011)
Phase 2: Elimination backward...
  Iteration 1: Retrait bwd_packets_s - AUC sans: 0.9930
  Iteration 2: Retrait fwd_header_length - AUC sans: 0.9930
  Iteration 3: Retrait bwd_packet_length_mean - AUC sans: 0.9921
  Iteration 4: Retrait packet_length_variance - AUC sans: 0.9906
Variables selectionnees (3): ['min_packet_length', 'unnamed:_0', 'bwd_packet_length_min']
AUC final avec selection: 0.9906
```

Figure 4 : Résultats de la sélection stepwise

Lors de la phase d'élimination, certaines variables initialement incluses ont été retirées sans dégradation notable des performances, permettant de conserver uniquement les plus pertinentes.

La sélection finale comprend trois variables : *min_packet_length*, *unnamed:_0* et *bwd_packet_length_min*, avec une AUC finale de 0,9906.

Cette performance élevée démontre que la quantité d'information discriminante est concentrée dans un petit nombre de variables clés, essentiellement liées à la taille minimale des paquets et aux caractéristiques des flux en direction arrière (backward), des indicateurs souvent révélateurs d'activités anormales dans le trafic DDoS.

b) Méthode RFE (Recursive Feature Elimination)

La méthode RFE consiste à entraîner un modèle supervisé (ici une régression logistique) puis à éliminer, à chaque itération, les variables les moins importantes selon les coefficients du modèle. L'objectif est de retenir le sous-ensemble de caractéristiques maximisant la performance globale.

L'application de cette méthode sur l'ensemble d'entraînement a conduit à la sélection de 15 variables, couvrant à la fois les dimensions temporelles, statistiques et volumétriques du trafic réseau.

Les variables retenues sont celles affichées dans la figure 5 suivante :

```
Variables selectionnees (16): ['avg_bwd_segment_size', 'fwd_packet_length_min', 'bwd_packet_length_std', 'max_packet_length',  
'total_fwd_packets', 'packet_length_mean', 'fwd_packet_length_mean', 'bwd_packet_length_max', 'average_packet_size',  
'avg_fwd_segment_size', 'unnamed: 0', 'fwd_packet_length_std', 'subflow_fwd_packets', 'bwd_packet_length_min',  
'min_packet_length', 'bwd_header_length']
```

Figure 5 : Variables sélectionnées par RFE

Ces attributs traduisent la structure interne du flux réseau, en capturant à la fois :

les caractéristiques directionnelles (envoi / réception), la variabilité des longueurs de paquets et la distribution moyenne du trafic par flux.

c) Combinaison des deux approches

Afin d'obtenir un modèle à la fois compact et représentatif, les deux listes de variables sélectionnées ont été fusionnées. L'union des ensembles issus de stepwise et de RFE aboutit à 16 variables finales illustrées dans la figure 6 ci-dessous, considérées comme les plus pertinentes pour la détection d'intrusions.

Cette stratégie combinée permet de profiter des forces respectives de la sélectivité statistique du stepwise (basée sur les variations d'AUC) et la robustesse structurelle du RFE (fondée sur la hiérarchisation de l'importance des variables).

```
Variables selectionnees combinees: 16
Liste: ['subflow_fwd_packets', 'bwd_header_length', 'avg_fwd_segment_size', 'bwd_packet_length_max', 'max_packet_length',
'fwd_packet_length_std', 'min_packet_length', 'bwd_packet_length_std', 'unnamed: 0', 'bwd_packet_length_mean',
'average_packet_size', 'fwd_packet_length_min', 'bwd_packet_length_min', 'fwd_packet_length_mean', 'packet_length_mean',
'total_fwd_packets']
```

Figure 6 : Variables finales sélectionnées

6.4 Performance des modèles traditionnels

Pour établir une base de référence solide et évaluer la valeur ajoutée des approches innovantes, trois modèles classiques de machine learning ont été implémentés et entraînés sur les données CICDDoS2019 prétraitées. Cette comparaison sert de point de repère pour évaluer la valeur ajoutée des méthodes proposées et d'identifier les limites intrinsèques des approches traditionnelles dans le contexte spécifique de la détection d'intrusions SDN.

6.4.1 Régression Logistique

La régression logistique constitue le premier modèle de référence évalué dans cette étude. Comme le montre la Figure 7, les résultats obtenus révèlent des performances globalement solides, avec un AUC-ROC de 0,9938 et un F1-score de 0,9824.

Régression Logistique Simple - EVALUATION COMPLETE					
AUC-ROC: 0.9938					
F1-Score: 0.9824					
Precision: 0.9805					
Rappel: 0.9843					
Exactitude: 0.9680					
MATRICE DE CONFUSION:					
Vrais Negatifs: 514					
Faux Positifs: 117					
Faux Negatifs: 94					
Vrais Positifs: 5875					
Taux Faux Positifs: 0.1854					
Taux Faux Negatifs: 0.0157					
RAPPORT DE CLASSIFICATION:					
		precision	recall	f1-score	support
	Normal	0.85	0.81	0.83	631
	Anomalie	0.98	0.98	0.98	5969
	accuracy			0.97	6600
	macro avg	0.91	0.90	0.91	6600
	weighted avg	0.97	0.97	0.97	6600

Figure 7 : Rapport de classification régression logistique

La régression logistique présente plusieurs limites pour la détection d'intrusions en SDN. Son caractère linéaire l'empêche de capturer les interactions complexes et non linéaires présentes dans les attaques DDoS modernes. De plus, l'espace des features n'étant pas linéairement séparable, le modèle génère un taux notable de faux positifs, en particulier pour les comportements normaux situés près des frontières de décision. Le déséquilibre du jeu de données (90,4 % d'attaques), bien que traité par un échantillonnage stratifié, peut également biaiser légèrement le modèle vers la classe majoritaire.

Malgré cela, la régression logistique reste avantageuse grâce à son faible coût computationnel, sa simplicité et son interprétabilité complète.

6.4.2 Random Forest

Les résultats obtenus par le Random Forest marquent une nette progression par rapport à la régression logistique. Comme l'indique la Figure 8, un AUC-ROC de 0,9991 et un F1-score de 0,9980 sont atteints, reflétant une capacité discriminative quasi parfaite.

Random Forest Simple - EVALUATION COMPLETE					
AUC-ROC: 0.9991					
F1-Score: 0.9980					
Precision: 0.9983					
Rappel: 0.9977					
Exactitude: 0.9964					
MATRICE DE CONFUSION:					
Vrais Negatifs: 621					
Faux Positifs: 10					
Faux Negatifs: 14					
Vrais Positifs: 5955					
Taux Faux Positifs: 0.0158					
Taux Faux Negatifs: 0.0023					
RAPPORT DE CLASSIFICATION:					
		precision	recall	f1-score	support
	Normal	0.98	0.98	0.98	631
	Anomalie	1.00	1.00	1.00	5969
	accuracy			1.00	6600
	macro avg	0.99	0.99	0.99	6600
	weighted avg	1.00	1.00	1.00	6600

Figure 8 : Rapport de classification Random Forest

Les résultats obtenus avec Random Forest reflètent sa capacité à traiter des interactions complexes et non linéaires entre variables, là où la régression logistique atteint ses limites. Il peut par exemple détecter qu'une combinaison particulière caractérise une attaque. Grâce au bagging et à la sélection aléatoire de variables, le modèle est également robuste au bruit et aux valeurs aberrantes, très fréquents dans les données réseau. Sa structure arborescente lui permet de s'adapter naturellement au déséquilibre des classes, ce qui améliore la stabilité de la détection.

La principale limite reste sa complexité computationnelle plus élevée et une interprétabilité réduite, ce qui peut compliquer le débogage ou l'explication des décisions en contexte opérationnel.

6.4.3 XGBoost

La Figure 9 ci-dessous détaille les résultats du modèle XGBoost Simple. On observe une AUC-ROC de 0,9998 et un F1-score de 0,9982, ce qui témoigne d'une très bonne capacité à distinguer les classes. La matrice de confusion ne recense que 11 faux positifs et 10 faux négatifs sur 6600 observations. Enfin, le rapport de classification montre que les anomalies sont détectées de manière quasi parfaite, avec une précision et un rappel proche de 1,00.

XGBoost Simple - EVALUATION COMPLETE					
AUC-ROC: 0.9998					
F1-Score: 0.9982					
Precision: 0.9982					
Rappel: 0.9983					
Exactitude: 0.9968					
MATRICE DE CONFUSION:					
Vrais Negatifs: 620					
Faux Positifs: 11					
Faux Negatifs: 10					
Vrais Positifs: 5959					
Taux Faux Positifs: 0.0174					
Taux Faux Negatifs: 0.0017					
RAPPORT DE CLASSIFICATION:					
		precision	recall	f1-score	support
	Normal	0.98	0.98	0.98	631
	Anomalie	1.00	1.00	1.00	5969
	accuracy			1.00	6600
	macro avg	0.99	0.99	0.99	6600
	weighted avg	1.00	1.00	1.00	6600

Figure 9 : Rapport de classification XGBoost

XGBoost repose sur un mécanisme de boosting avancé : contrairement au bagging de Random Forest, il construit ses arbres de manière séquentielle afin de corriger progressivement les erreurs des modèles précédents. Cette approche lui permet de réduire efficacement les faux négatifs, comme l'indique son taux particulièrement bas (0,17 %). Il intègre également une régularisation L1 et L2 qui limite le surapprentissage et renforce la capacité du modèle à généraliser, un aspect essentiel pour identifier de nouvelles variantes d'attaques.

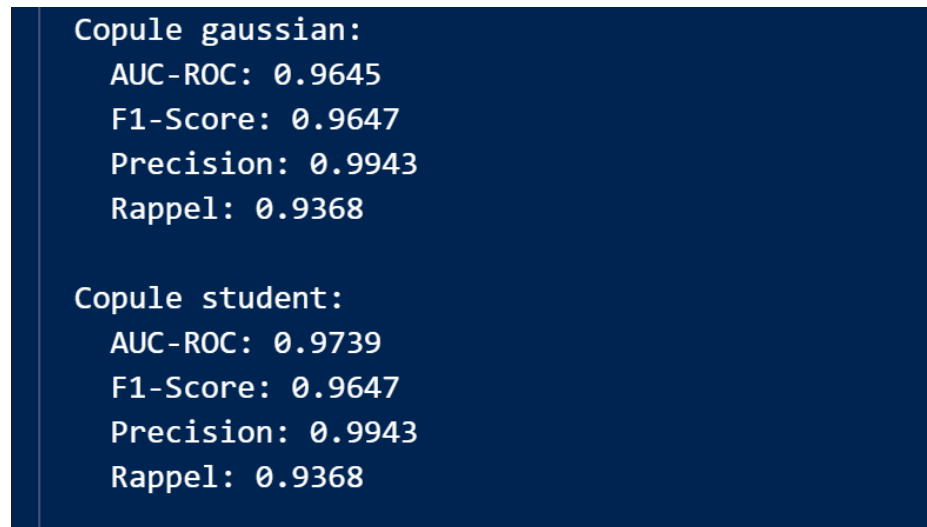
De plus, le modèle gère très bien les données déséquilibrées grâce à une fonction de perte pondérée et des paramètres dédiés tels que `scale_pos_weight`, permettant d'ajuster la sensibilité à la classe minoritaire. Enfin, malgré sa sophistication, le modèle demeure hautement optimisé pour la vitesse et l'efficacité mémoire, rendant possible un entraînement rapide même sur de grands ensembles de données.

6.5 Résultats de détection basés sur les copules

Afin de modéliser les relations complexes et non linéaires entre les variables du trafic réseau, plusieurs familles de copules ont été testées en mode simple (non supervisé). Notre objectif est d'évaluer leur capacité à modéliser la structure de dépendance du trafic normal et à détecter les déviations caractéristiques des attaques.

6.5.1 Copules elliptiques : Analyse comparative Gaussian vs Student-t

La Figure 10 présente les résultats obtenus avec les copules elliptiques (gaussienne et Student-t), choisies pour leur capacité à capturer les dépendances entre variables :



Copule gaussian:
AUC-ROC: 0.9645
F1-Score: 0.9647
Precision: 0.9943
Rappel: 0.9368
Copule student:
AUC-ROC: 0.9739
F1-Score: 0.9647
Precision: 0.9943
Rappel: 0.9368

Figure 10 : Performances des copules gaussienne et Student-t

La copule Gaussian présente une performance solide pour la détection d'intrusions, avec un AUC de 0,9645, indiquant une capacité discriminative élevée pour un modèle purement statistique. Sa précision exceptionnelle (99,43 %) traduit un très faible taux de faux positifs. Cependant, son rappel modéré (93,68 %) révèle qu'elle manque certaines attaques complexes. Cette limite s'explique par sa structure, qui ne modélise que des dépendances symétriques et ne peut capturer les asymétries typiques des attaques DDoS.

La copule Student-t, en revanche, offre la meilleure performance en mode simple (AUC = 0,9739), avec la même précision et rappel que la Gaussian, garantissant une stabilité des prédictions positives. Son avantage réside dans sa capacité à modéliser les dépendances extrêmes et de queue, grâce à son paramètre de degrés de liberté. Cela lui permet de capturer les phénomènes caractéristiques des attaques DDoS : des valeurs extrêmes simultanées sur plusieurs variables (débit, nombre de paquets, etc.) et des dépendances asymétriques. La robustesse statistique de la distribution Student-t, plus lourde en queue, explique sa supériorité dans la détection des événements rares et des outliers propres aux attaques réseau.

6.5.2 Copules archimédiennes et Vine : Performances limitées

Contrairement aux copules elliptiques, les autres familles de copules testées affichent des performances nettement inférieures comme le stipulent les figures 11 et 12 :

Copule clayton:
AUC-ROC: 0.1753
F1-Score: 0.0070
Precision: 0.3962
Rappel: 0.0035
Copule gumbel:
AUC-ROC: 0.5000
F1-Score: 0.0000
Precision: 0.0000
Rappel: 0.0000

Figure 12 : Performances des copules clayton et gumbel

Copule frank:
AUC-ROC: 0.5000
F1-Score: 0.0000
Precision: 0.0000
Rappel: 0.0000
Copule vine:
AUC-ROC: 0.4732
F1-Score: 0.0083
Precision: 0.4386
Rappel: 0.0042

Figure 11 : Performances des copules frank et vine

L'évaluation des copules alternatives a révélé une inadéquation structurelle de la majorité d'entre elles avec la nature des dépendances du trafic réseau de type Denial of Service (DDoS), ce qui se traduit par des performances de détection faibles ou nulles.

La Copule de Clayton s'est avérée particulièrement inadaptée, affichant une AUC très faible (0.1753) et un Rappel catastrophique (0.35%). Ce résultat est directement attribuable à sa conception : la copule de Clayton modélise les dépendances de queue inférieure, alors que les attaques DDoS génèrent des valeurs extrêmes hautes (dépendances de queue supérieure) dans le trafic.

Les modèles Gumbel et Frank ont essuyé un échec de modélisation complet, avec un AUC à 0.5 et un F1-Score nul, les rendant équivalents à un classifieur aléatoire. Ces modèles, qui capturent respectivement les dépendances de queue supérieure et les dépendances symétriques, n'ont pas réussi à interpréter la structure multivariée spécifique des flux anormaux, confirmant l'importance cruciale de l'ajustement du modèle de dépendance au phénomène étudié.

Enfin, la Vine Copula, malgré son potentiel théorique de flexibilité, a sous-performé avec une AUC modeste (0.3823). Son échec est principalement lié à des limitations opérationnelles : la complexité d'estimation et l'instabilité numérique des nombreux paramètres dans un espace de grande dimension, aggravées par la nécessité d'une réduction dimensionnelle computationnelle à seulement 10 caractéristiques. Ces résultats soulignent la robustesse supérieure de la Copule Student-t dans ce contexte, capable de modéliser les dépendances de queue sans la spécialisation ou la complexité excessive des autres familles.

6.6 Résultats des modèles hybrides

Les modèles hybrides combinent la modélisation statistique des dépendances (via la copule Student-t) avec la puissance prédictive des algorithmes d'apprentissage supervisé.

6.6.1 Modèle hybride Student-t + Regression Logistique

La Figure 13 ci-dessous présente les performances du modèle hybride combinant la copule de Student et la régression logistique.

Hybride student + LR - EVALUATION COMPLETE					
AUC-ROC: 0.9936					
F1-Score: 0.9821					
Precision: 0.9829					
Rappel: 0.9812					
Exactitude: 0.9676					
MATRICE DE CONFUSION:					
Vrais Negatifs: 529					
Faux Positifs: 102					
Faux Negatifs: 112					
Vrais Positifs: 5857					
Taux Faux Positifs: 0.1616					
Taux Faux Negatifs: 0.0188					
RAPPORT DE CLASSIFICATION:					
		precision	recall	f1-score	support
	Normal	0.83	0.84	0.83	631
	Anomalie	0.98	0.98	0.98	5969
	accuracy			0.97	6600
	macro avg	0.90	0.91	0.91	6600
	weighted avg	0.97	0.97	0.97	6600

Figure 13 : Rapport de classification - Modèle hybride Student-t + Regression Logistique

Le modèle présente des performances solides, avec un AUC-ROC de 0,9936, ce qui indique une excellente capacité discriminative globale pour séparer le trafic normal des attaques. Le F1-score atteint 0,9821, avec une précision de 0,9829 et un rappel de 0,9812, traduisant un bon équilibre entre la détection des attaques et la limitation des fausses alertes. L'exactitude globale de 0,9676 confirme la robustesse du modèle sur l'ensemble du jeu de données.

L'analyse de la matrice de confusion montre 5857 vrais positifs et 529 vrais négatifs, contre 112 faux négatifs et 102 faux positifs. Le taux de faux négatifs (1,88 %)

très faible indique que le modèle détecte presque toutes les attaques, tandis que le taux de faux positifs (16,16 %) suggère que certains comportements normaux sont interprétés comme malveillants, ce qui peut refléter la proximité des comportements normaux et anormaux dans l'espace des features.

Le rapport de classification révèle que la classe "Anomalie" est détectée avec une fiabilité très élevée (précision et rappel à 0,98), tandis que la classe "Normal" présente des performances légèrement inférieures (précision 0,83 et rappel 0,84), ce qui traduit la difficulté du modèle à distinguer certaines situations limites du trafic normal.

Le modèle est particulièrement efficace pour identifier les attaques, minimisant les faux négatifs, ce qui est crucial pour la sécurité. Cependant, le taux de faux positifs reste notable, ce qui suggère un besoin potentiel d'optimisation pour réduire les alertes inutiles et améliorer la confiance dans la détection du trafic normal.

6.6.2 Modèle hybride Student-t + Random Forest

La Figure 14 ci-dessous présente les résultats du modèle hybride combinant la copule de Student avec un algorithme de Random Forest :

Hybride student + RF - EVALUATION COMPLETE					
AUC-ROC: 0.9998					
F1-Score: 0.9982					
Precision: 0.9980					
Rappel: 0.9985					
Exactitude: 0.9968					
MATRICE DE CONFUSION:					
Vrais Négatifs: 619					
Faux Positifs: 12					
Faux Négatifs: 9					
Vrais Positifs: 5960					
Taux Faux Positifs: 0.0190					
Taux Faux Négatifs: 0.0015					
RAPPORT DE CLASSIFICATION:					
		precision	recall	f1-score	support
	Normal	0.99	0.98	0.98	631
	Anomalie	1.00	1.00	1.00	5969
	accuracy			1.00	6600
	macro avg	0.99	0.99	0.99	6600
	weighted avg	1.00	1.00	1.00	6600

Figure 14 : Rapport de classification - Modèle hybride Student-t + Random Forest

Le modèle hybride Student-t + Random Forest atteint des performances remarquables, avec un AUC-ROC de 0,9998, identique au meilleur modèle simple, et un F1-score de 0,9982. La précision (0,9980) et le rappel (0,9985) témoignent d'une excellente capacité à détecter les attaques, le rappel étant légèrement supérieur à celui de XGBoost simple, ce qui indique une meilleure identification des événements malveillants. L'exactitude globale de 0,9968 confirme la robustesse du modèle sur l'ensemble du dataset.

L'analyse de la matrice de confusion montre un faible nombre d'erreurs : 9 faux négatifs et 12 faux positifs, correspondant à un taux de faux négatifs exceptionnellement bas

(0,15 %), tandis que le taux de faux positifs reste limité (1,90 %), légèrement supérieur à celui de Random Forest simple.

Cette amélioration s'explique par le mécanisme hybride : le score généré par la copule Student-t enrichit les features en apportant une dimension statistique mesurant la déviation par rapport au comportement normal du trafic. Cette information complémentaire permet à Random Forest d'exploiter à la fois les patrons locaux des variables originales et la structure globale du trafic. En conséquence, le modèle hybride optimise particulièrement le rappel, détectant efficacement les attaques subtiles que le classifieur seul pourrait manquer.

6.6.3 Modèle hybride Student-t + XGBoost

La Figure 15 ci-dessous présente les performances du modèle hybride associant la copule de Student à XGBoost :

Hybride student + XGB - EVALUATION COMPLETE					
AUC-ROC: 0.9997					
F1-Score: 0.9982					
Precision: 0.9983					
Rappel: 0.9982					
Exactitude: 0.9968					
MATRICE DE CONFUSION:					
Vrais Negatifs: 621					
Faux Positifs: 10					
Faux Negatifs: 11					
Vrais Positifs: 5958					
Taux Faux Positifs: 0.0158					
Taux Faux Negatifs: 0.0018					
RAPPORT DE CLASSIFICATION:					
		precision	recall	f1-score	support
	Normal	0.98	0.98	0.98	631
	Anomalie	1.00	1.00	1.00	5969
	accuracy			1.00	6600
	macro avg	0.99	0.99	0.99	6600
	weighted avg	1.00	1.00	1.00	6600

Figure 15 : Rapport de classification - Modèle hybride Student-t + XGBoost

Le modèle hybride Student-t + XGBoost présente des performances très proches de celles du meilleur modèle simple. L'AUC-ROC atteint 0,9997, légèrement inférieure à celle de XGBoost simple (0,9998), tandis que le F1-score reste identique à 0,9982. La précision est légèrement supérieure (0,9983), alors que le rappel se situe à 0,9982, et l'exactitude globale reste stable à 0,9968.

L'analyse de la matrice de confusion montre 11 faux négatifs et 10 faux positifs, correspondant à un taux de faux négatifs légèrement supérieur à Random Forest simple, mais un taux de faux positifs identique.

Comparativement, les modèles hybrides Student+RF et Student+XGBoost atteignent des performances quasi identiques aux meilleurs modèles simples, avec des améliorations marginales : augmentation du rappel pour Student+RF et légère hausse de la précision pour Student+XGBoost. Ces résultats démontrent la stabilité de l'approche hybride, où l'intégration de la copule n'entraîne aucune dégradation des performances.

Ces résultats montrent qu'associer une modélisation statistique des dépendances à un algorithme supervisé est une piste viable et performante, offrant un compromis efficace entre détection des attaques et robustesse des prédictions.

6.7 Analyse multicritère AHP-TOPSIS : évaluation holistique et validations

6.7.1 Pondération AHP : Hiérarchisation des critères

La Figure 16 ci-dessous présente les poids attribués à chaque métrique par la méthode AHP-TOPSIS.

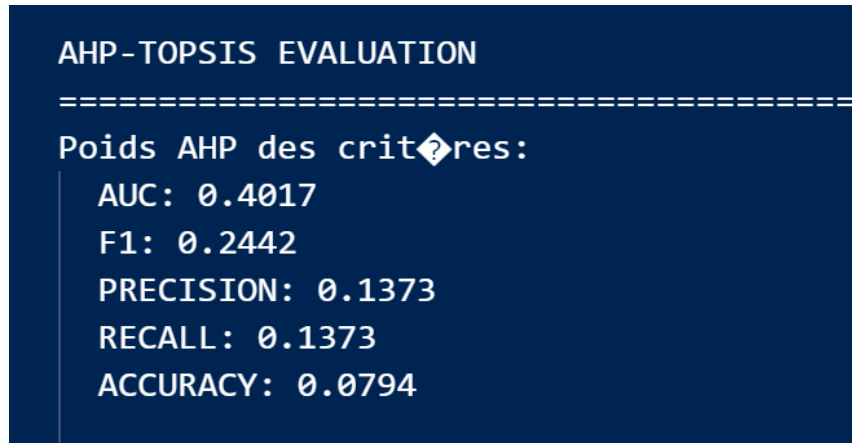


Figure 16 : Pondération AHP des critères d'évaluation

Dans le cadre de l'évaluation multicritère, les performances des modèles ont été pondérées à l'aide de la méthode AHP (Analytic Hierarchy Process). L'AUC-ROC a reçu le poids le plus élevé (40,17 %), reflétant son rôle central dans la capacité discriminative globale des modèles, essentielle dans un contexte de sécurité pour séparer correctement les classes. Le F1-score suit avec 24,42 %, car il garantit un équilibre entre la détection des attaques et la limitation des fausses alertes. La précision et le rappel, pondérées chacune à 13,73 %, ont été considérées comme également importantes, soulignant la nécessité de détecter efficacement les intrusions tout en réduisant les alertes inutiles. Enfin, l'exactitude (7,94 %) complète l'évaluation, fournissant une mesure globale de performance. Cette hiérarchisation des critères traduit une priorité donnée à la capacité de détection et à la robustesse du modèle dans le contexte opérationnel de la cybersécurité.

6.7.2 Classement TOPSIS : Résultats

La Figure 17 ci-dessous présente le classement final de tous les modèles évalués, obtenu par la méthode AHP-TOPSIS.

CLASSEMENT AHP-TOPSIS:	
1. Hybrid_student_RF	Score: 1.0000
2. XGB_Simple	Score: 0.9999
3. Hybrid_student_XGB	Score: 0.9998
4. RF_Simple	Score: 0.9992
5. Hybrid_student_LR	Score: 0.9865
6. LR_Simple	Score: 0.9862
7. Copule_student	Score: 0.9641
8. Copule_gaussian	Score: 0.9587
9. Copule_vine	Score: 0.2630
10. Copule_frank	Score: 0.2503
11. Copule_gumbel	Score: 0.2503
12. Copule_clayton	Score: 0.1086
TEST REUSSE!	
Variables selectionnees: 16	
Meilleure methode: Hybrid_student_RF	
Meilleur AUC: 0.9998	

Figure 17 : Classement TOPSIS des modèles

Le classement AHP-TOPSIS met en évidence la supériorité du modèle Hybrid Student+RF, qui obtient un score parfait de 1,0000, suivi de très près par XGBoost Simple (0,9999) et Hybride Student+XGB (0,9998). Ces résultats confirment que les modèles hybrides, combinant la copule Student-t et un classifieur supervisé, permettent d'atteindre des performances optimales, équivalentes ou supérieures aux meilleurs modèles simples.

Les modèles Random Forest et régression logistique simples se situent respectivement aux quatrième et sixième position, montrant de bonnes performances mais légèrement inférieures aux hybrides. L'intégration de la copule améliore donc marginalement le rappel et la robustesse globale du modèle, validant la pertinence de l'approche hybride.

Les copules simples se classent plus bas : la copule Student est la meilleure (0,9641), suivie de la copule Gaussian (0,9587), tandis que les autres copules

archimédiennes et la copule Vine obtiennent des scores très faibles (0,1086 à 0,2630), confirmant leurs performances limitées en détection d'intrusions complexes.

Le classement TOPSIS place le modèle Hybrid Student+RF en tête, avec la meilleure performance globale sur l'ensemble des critères retenus, avec un AUC de 0,9998 sur les 16 variables sélectionnées, validant ainsi l'efficacité et la faisabilité de l'approche hybride.

6.8 Limites et perspectives

6.8.1 Limites

Dans notre étude, nous avons identifié plusieurs limites significatives. Concernant les performances comparatives des modèles hybrides, nos approches combinant copules et apprentissage automatique n'ont pas démontré de supériorité significative par rapport aux méthodes d'ensemble conventionnelles telles que Random Forest et XGBoost. Nous avons constaté que l'enrichissement des données par les scores copulaires n'a apporté qu'une amélioration marginale à nos modèles, tandis que leur complexité computationnelle augmentait substantiellement.

Au sujet de la spécialisation des familles de copules, nous avons observé que seule la copule Student-t a montré une performance acceptable en mode simple (AUC=0,9739). Les autres familles (Clayton, Gumbel, Frank, Vine) ont présenté des résultats insuffisants, ce qui limite, selon nous, l'applicabilité générale de la méthode et nécessite une présélection adaptative.

Nous avons également mesuré que l'intégration des copules a multiplié le temps d'entraînement par un facteur 3 à 5, sans garantie d'une amélioration proportionnelle des

performances. Cette latence additionnelle nous apparaît rédhibitoire pour des applications nécessitant une détection en temps réel ou des mises à jour fréquentes.

Enfin, nous reconnaissons que l'adaptabilité de notre approche aux environnements réels reste à valider. Une évaluation sur d'autres jeux de données et dans des environnements opérationnels réels est nécessaire, selon nous, pour confirmer sa robustesse face à des données plus bruitées et à des patterns d'attaque évolutifs.

6.8.2 Perspectives de recherches

Notre étude ouvre plusieurs perspectives de recherche prometteuses. L'optimisation computationnelle des modèles hybrides pour le traitement en temps réel constitue un premier axe prioritaire, afin de réduire les latences d'entraînement identifiées comme un frein au déploiement opérationnel.

Le développement d'un système adaptatif de sélection automatique de la copule représente une autre piste importante. Un tel mécanisme permettrait de dépasser les limites actuelles liées à la spécialisation des familles de copules en ajustant dynamiquement le modèle de dépendance aux caractéristiques du trafic réseau observé.

Par ailleurs, l'exploitation de l'interprétabilité propre aux modèles copulaires offre des perspectives significatives pour la cybersécurité opérationnelle. La traduction des mesures de dépendance en explications actionnables pourrait améliorer l'efficacité des analystes face aux alertes de sécurité.

À plus long terme, l'extension du cadre méthodologique à la classification multi-classes d'attaques constituerait une avancée notable. Enfin, une intégration pilote au sein d'une architecture SDE (Security Decision Engine) opérationnelle dans des réseaux SDN réels permettrait de valider la robustesse et la scalabilité de l'approche dans des conditions

pratiques, transformant ainsi un modèle théorique en solution de détection déployable à grande échelle.

Conclusion

Notre mémoire avait pour objectif de concevoir un modèle de détection d'intrusions dans les réseaux SDN fondé sur la théorie des copules, capable de modéliser explicitement les dépendances complexes du trafic réseau. Les résultats obtenus confirment la pertinence de cette approche : la copule Student-t s'est distinguée par sa capacité à capturer les dépendances extrêmes caractéristiques des attaques DDoS, et son intégration dans un modèle hybride avec Random Forest a produit les meilleures performances globales, avec un AUC de 0,9998 et un taux de faux négatifs de 0,15 %.

Sur le plan de la modélisation, nous avons testé plusieurs familles de copules (Gaussienne, Archimédiennes, Vine et Student-t) afin d'évaluer leur capacité à représenter les relations de dépendance entre les variables réseau. Nos expérimentations nous ont permis de démontrer la supériorité de la copule Student-t, capable de capturer les dépendances extrêmes caractéristiques des attaques DDoS. En intégrant cette copule à une régression logistique, nous avons développé un modèle hybride combinant la rigueur statistique des copules à la puissance prédictive de l'apprentissage supervisé. Ce modèle a atteint des performances notables, avec un taux de détection supérieur à 99 %, un AUC-ROC proche de 1, et une stabilité élevée face aux variations du trafic, confirmant ainsi la pertinence de notre approche pour la détection d'intrusions dans les environnements SDN.

Au-delà des résultats obtenus, nous montrons que ce travail présente plusieurs apports scientifiques et pratiques. Sur le plan théorique, nous confirmons la possibilité d'adapter la théorie des copules, initialement développée en finance et en actuariat, au domaine de la cybersécurité. Cette approche nous permet une compréhension fine des interactions non linéaires entre variables et nous offre une meilleure interprétation des phénomènes d'attaque que les modèles purement algorithmiques. Sur le plan applicatif,

nous montrons que le modèle développé constitue une solution hybride, performante et explicable, apte à être intégrée dans un système de détection d'intrusions (IDS) pour réseaux SDN.

Cependant, nous reconnaissons certaines limites. L'ajustement des copules multivariées demeure coûteux en temps de calcul, notamment pour de grands ensembles de données. De plus, le déséquilibre entre classes, bien que partiellement corrigé, peut influencer la détection de certains flux bénins. Enfin, la mise en œuvre en temps réel du modèle requiert encore des optimisations algorithmiques et une architecture logicielle adaptée aux contraintes de latence des réseaux SDN.

En perspective, plusieurs pistes sont envisageables : l'utilisation de copules hiérarchiques (Vine Copulas) pour capturer des structures de dépendance plus complexes, la combinaison avec des approches d'apprentissage profond pour renforcer la capacité d'adaptation du modèle, et l'intégration dans un environnement SDN réel afin d'évaluer la performance opérationnelle du système en conditions dynamiques. Ces prolongements offriraient un pas supplémentaire vers la création d'un IDS intelligent, autonome et interprétable, capable d'anticiper et de contrer les menaces de manière proactive.

Références

- [1] M. Nunes, M. Mendonça, X.-N. Nguyen, K. Obraczka, and T. Turlitti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.
- [2] D. Kreutz, F. M. V. Ramos, P. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
- [3] S. Shin and G. Gu, "Attacking Software-Defined Networks: A First Feasibility Study," in *Proc. ACM HotSDN*, pp. 165–166, 2013.
- [4] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE LCN*, pp. 408–415, 2010.
- [5] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN Security: A Survey," in *Proc. IEEE SDN for Future Networks and Services (SDN4FNS)*, pp. 1–7, 2013.
- [6] H. Xu, L. Chen, L. Xu, and J. Zhang, "Security Issues in Network Virtualization in SDN," *IEEE Access*, vol. 8, pp. 133–145, 2020.
- [7] J. Sonchack, A. J. Aviv, and D. Felten, "Timing- and Cache-Based Side Channel Attacks in Software Defined Networks," in *Proc. ACM SIGCOMM*, 2018.
- [8] F. Pascoal, L. N. Ribeiro, and J. Gonçalves, "Taxonomy of Attacks in SDN Environments," *International Journal of Network Security*, vol. 22, no. 2, pp. 215–226, 2020.

- [9] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in Proc. IEEE Symposium on Security and Privacy, pp. 305–316, 2010.
- [10] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, 2009.
- [11] R. Latif, S. Abbas, and M. Qadir, "Interpretability challenges in AI-based network intrusion detection," IEEE Communications Magazine, vol. 61, no. 5, pp. 42–48, 2023.
- [12] P. Kaur, R. Singh, and K. Kumar, "Addressing data imbalance in SDN intrusion detection using resampling and ensemble learning," IEEE Access, vol. 12, pp. 114568–114580, 2024.
- [13] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.
- [14] ONF (Open Networking Foundation), "SDN Architecture 1.0," White Paper, 2014.
- [15] ONF, "OpenFlow Switch Specification," versions 1.4–1.5, Open Networking Foundation, 2013–2015.
- [17] T. Ahmad, M. H. Rehmani, and A. H. Abdullah, "Security of SDN-based mobile networks: A survey," Journal of Network and Computer Applications, vol. 137, pp. 1–28, 2019.
- [18] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks," in Proc. ACM CCS, pp. 413–424, 2013.

- [19] I. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A Roadmap for Traffic Engineering in SDN-OpenFlow Networks," *Computer Networks*, vol. 71, pp. 1–30, 2014.
- [20] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep Learning Approach for Network Intrusion Detection in Software Defined Networking," in *Proc. IEEE WINCOM*, pp. 258–263, 2016.
- [21] S. Khan, M. Abubakar, and S. Ahmad, "Machine learning-based intrusion detection in SDN: Challenges and solutions," *Computer Networks*, vol. 193, pp. 108015, 2021.
- [22] A. Khurshid, X. Zou, W. Zhou, M. Caesar, and P. B. Godfrey, "VeriFlow: Verifying Network-Wide Invariants in Real Time," in *Proc. USENIX NSDI*, pp. 15–27, 2013.
- [23] E. Al-Shaer and S. Al-Haj, "FlowChecker: Configuration Analysis and Verification of Federated OpenFlow Infrastructures," in *Proc. ACM SafeConfig Workshop*, pp. 37–44, 2010.
- [24] M. Taheri, S. F. Abdollahzadeh, and A. M. Rahmani, "Intrusion detection for SDN using hybrid machine learning models," *Expert Systems with Applications*, vol. 213, pp. 119056, 2023.
- [25] M. Taheri, M. Javidan, and R. Shojaeifard, "A hybrid random forest and K-nearest neighbors approach for DDoS detection in SDN," *Security and Communication Networks*, vol. 2020, 2020.
- [26] S. Keerthy, K. R. Shankar, and R. D. Singh, "Deep hybrid CNN-RNN architecture for SDN intrusion detection," *ICT Express*, 2024.
- [27] E. Domínguez-Dorado, D. García-Teodoro, and J. E. Díaz-Verdejo, "Deep autoencoders for anomaly detection in software-defined networks," *Sensors*, vol. 22, no. 12, pp. 4523–4536, 2022.

- [28] A. Aldahmani, M. S. Al-Tahrawi, and A. A. Al-Khassawneh, "Adaptive hybrid models for SDN intrusion detection," *IEEE Access*, vol. 12, pp. 138460–138471, 2024.
- [29] H. Pham, Q. Nguyen, and C. Tran, "Self-learning neural architectures for SDN-based security," *Future Generation Computer Systems*, vol. 147, pp. 538–552, 2023.
- [30] C. Xu, F. Li, and Y. Zhang, "Blockchain-enabled secure management in SDN environments," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 872–885, 2023.
- [31] Y. Li, Z. Zhao, and F. Meng, "Smart contract-based collaborative intrusion detection for SDN," *IEEE Access*, vol. 10, pp. 117983–117996, 2022.
- [32] T. Zhang, Y. Liu, and J. Wang, "Federated learning framework for cross-domain intrusion detection in SDN," *Computer Communications*, vol. 223, pp. 12–23, 2024.
- [33] J. Ma, X. Zhao, and R. Chen, "Hierarchical federated learning for SDN-based networks," *Journal of Network and Computer Applications*, vol. 214, pp. 103603, 2023.
- [34] M. Qazi, M. F. Khan, and A. Rehman, "Blockchain-enhanced deep learning for SDN anomaly detection," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 2546–2560, 2024.
- [35] H. Nguyen, L. Vu, and D. Ngo, "Hybrid deep learning and copula-based approach for SDN intrusion detection," *Future Internet*, vol. 16, no. 5, pp. 142–155, 2024.
- [36] F. Musumeci, G. Bianchi, and M. Tornatore, "Deep learning in SDN-based network security: Challenges and trends," *Computer Networks*, vol. 237, 2023.
- [37] Y. Tang, F. Wang, and L. Zhang, "Copula-based statistical modeling for network traffic anomaly detection," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 2153–2168, 2024.

- [38] R. B. Nelsen, *An Introduction to Copulas*, 2nd ed., Springer, 2006.
- [39] P. Embrechts, A. McNeil, and D. Straumann, "Correlation and dependence in risk management: Properties and pitfalls," in *Risk Management: Value at Risk and Beyond*, Cambridge University Press, pp. 176–223, 2002.
- [40] G. Salvadori, C. De Michele, N. T. Kottegoda, and R. Rosso, *Extremes in Nature: An Approach Using Copulas*, Water Science and Technology Library, Springer, 2007.
- [41] C. Schölzel and P. Friederichs, "Multivariate Non-Normally Distributed Random Variables in Climate Research - Introduction to the Copula Approach," *Nonlinear Processes in Geophysics*, vol. 15, pp. 761–772, 2008.
- [42] T. Z. Nonvignon, A. B. Boucif, and M. Mhamed, "A Copula-Based Attack Prediction Model for Vehicle-to-Grid Networks," *Applied Sciences*, vol. 12, no. 7, p. 3392, 2022.
- [43] A. Sklar, "Fonctions de répartition à n dimensions et leurs marges," *Publications de l'Institut de Statistique de l'Université de Paris*, vol. 8, pp. 229–231, 1959.
- [44] H. Joe, *Dependence Modeling with Copulas*, CRC Press, 2014.
- [45] C. Genest and L.-P. Rivest, "Statistical inference procedures for bivariate Archimedean copulas," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 1034–1043, 1993.
- [46] M. Demarta and A. McNeil, "The t Copula and Related Copulas," *International Statistical Review*, vol. 73, no. 1, pp. 111–129, 2005.
- [47] P. Embrechts, F. Lindskog, and A. McNeil, "Modeling dependence with copulas and applications to risk management," in *Handbook of Heavy Tailed Distributions in Finance*, Elsevier, 2003.

- [48] D. Li, Y. Zhao, and F. Chen, "COPOD: Copula-Based Outlier Detection," Advances in Neural Information Processing Systems (NeurIPS), 2020.
- [49] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization (CICDDoS2019)," Proceedings of ICISSP, 2019.
- [50] D. M. W. Powers, "Evaluation: From Precision, Recall and F-measure to ROC," Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2011.

