



EVALUATION ORIENTÉE DONNÉES DES SYSTÈMES DE DÉTECTION D'INTRUSION DANS LES RÉSEAUX

Soutenance de thèse

Lundi 26 mai 2025

Direction de thèse:

Sébastien Tixeuil
Gregory Blanc

Membres du jury:

Patrick Sondi, Romain Laborde, Maria Potop-Butucaru,
Gilles Guette, Houda Jmila

Solayman Ayoubi
LIP6 - Sorbonne Université

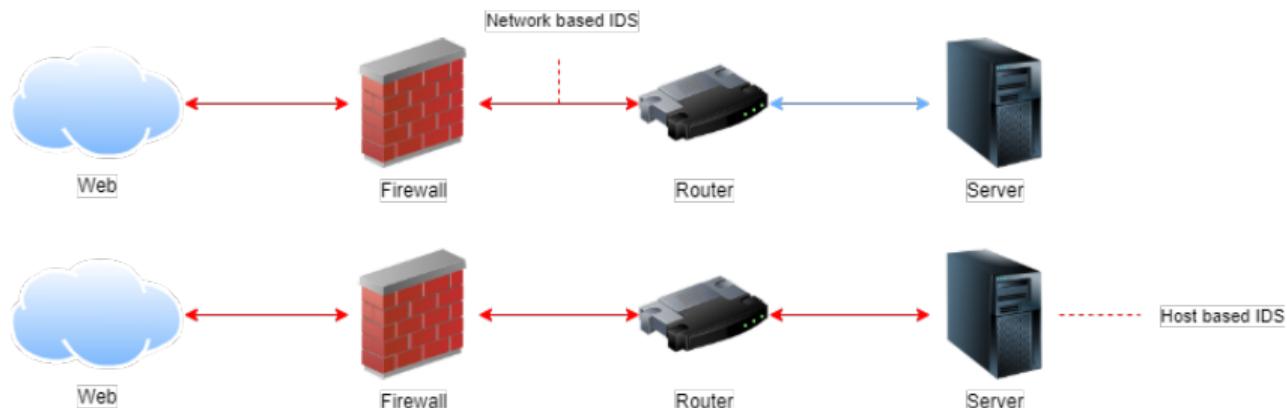
Evaluation orientée données des systèmes de détection d'intrusion dans les réseaux

Evaluation orientée données des systèmes de détection d'intrusion dans les réseaux

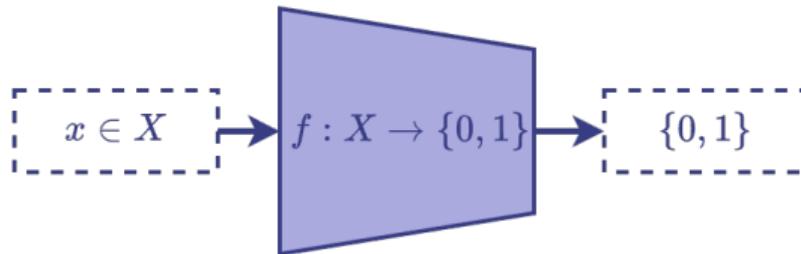
Système de detection d'intrusion

Définition d'un IDS

Outil déployé sous forme logicielle ou matérielle, conçu pour détecter et alerter les administrateurs en cas d'activité anormale ou potentiellement malveillante au sein d'un réseau ou d'un système d'exploitation.



Système de détection d'intrusion



Modèle basé sur les signatures

- Soit $S \subset X$ l'ensemble des états correspondant à des signatures connues d'intrusion.
- L'IDS peut alors être défini comme une fonction : $f(x) = \begin{cases} 0 & \text{si } x \in S \\ 1 & \text{sinon} \end{cases}$

Modèle basé sur l'analyse des anomalies

- Soit $N \subset X$ l'ensemble des états considérés comme normaux.
- L'IDS peut alors être défini comme :

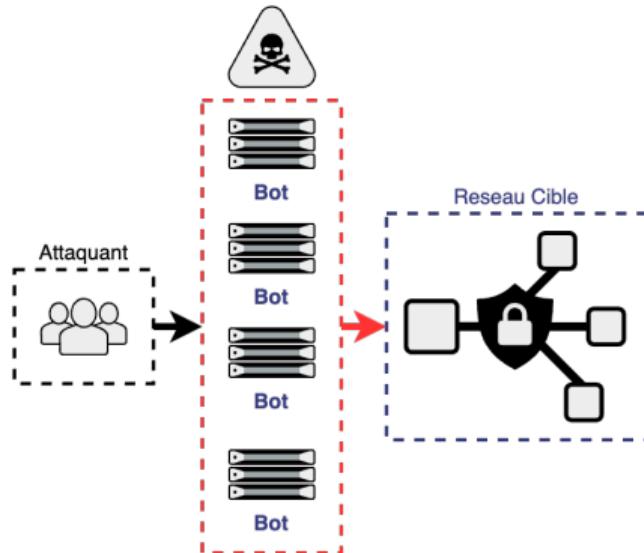
$$f(x) = \begin{cases} 0 & \text{si } x \notin N \\ 1 & \text{si } x \in N \end{cases}$$

Evaluation orientée données des systèmes de détection d'intrusion dans les réseaux

Les Réseaux

Vulnérabilités et Menaces Réseau

- DDoS : Surcharge un serveur ou un réseau avec un volume massif de requêtes pour le rendre indisponible..
- Brute Force : Devine des mots de passe ou des clés en essayant un grand nombre de combinaisons automatiquement.
- Scanning : Analyse systématique d'un réseau pour détecter les ports ouverts, services actifs ou vulnérabilités.
- Man-in-the-Middle : Intercepte et altère les communications entre deux parties sans leur consentement.



Les Réseaux



Représentation des données réseau

Les IDS surveillent le réseau et déclenchent une alerte dès qu'une attaque ou une activité suspecte est détectée. Plusieurs représentations des données peuvent être utilisées.

- Caractéristiques au niveau des paquets.
- Caractéristiques au niveau des flux.

Paquets Réseau			
	srcip	dstip	proto
x_1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$
x_2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$

Flux Réseau			
	dstport	durée du flux	paquets envoyé
x_1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$
x_2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$

Evaluation orientée données des systèmes de détection d'intrusion dans les réseaux

Evaluation orientée données

Définition

Approche où l'accent est mis sur l'utilisation de données pour évaluer les performances du système. L'évaluation est donc défini principalement par les données et non les métriques.

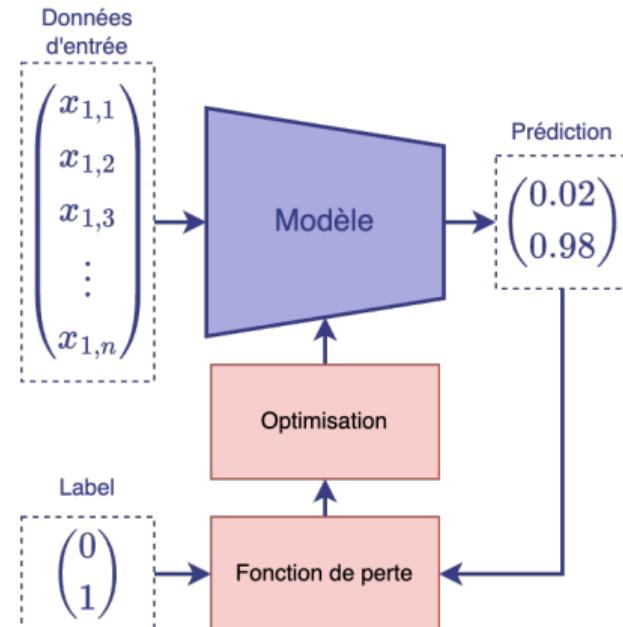
Apprentissage automatique (ML)

De nos jours le mécanisme de détection des IDS est très souvent un algorithme de ML, nous nous intéressons donc à l'évaluation orientée données des IDS basé sur ML.

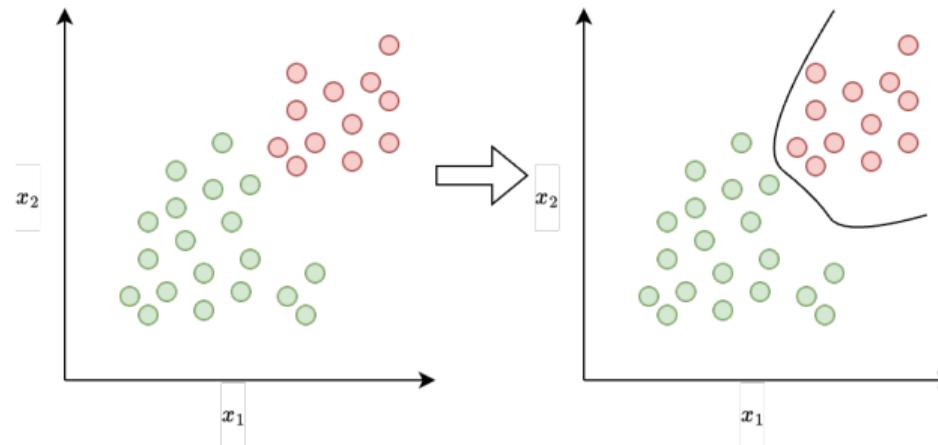
Evaluation orientée données

IDS basé sur le ML

Utilisent des algorithmes de ML pour détecter des comportements anormaux ou malveillants dans un réseau. Contrairement aux IDS traditionnels, ces systèmes apprennent à partir des données historiques.



Evaluation orientée données



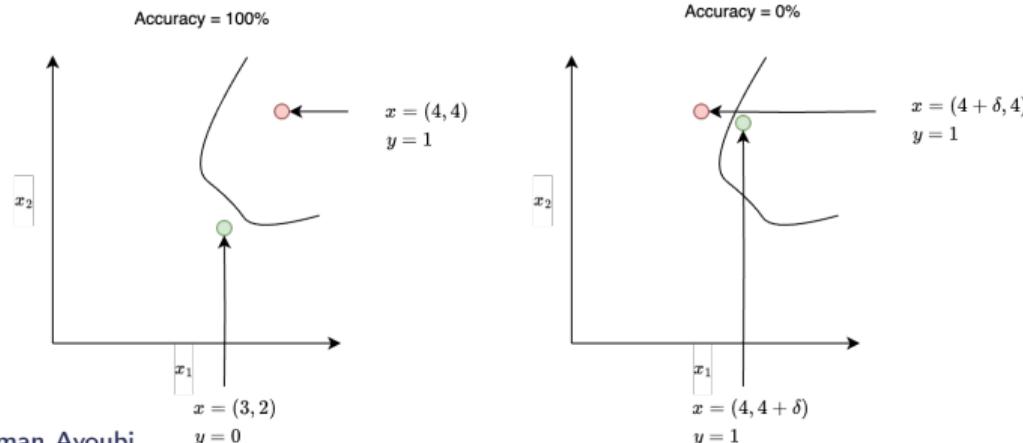
- On peut alors apprendre sur des données plutôt que définir des règles manuellement,
- et identifier des attaques inconnues ou nouvelles sans avoir besoin de signatures préalables.

Evaluation orientée données

Intuition

- 1 On détermine dans l'accuracy sur des données classique,
- 2 puis l'accuracy sur des données perturbés (δ).

Pour une même métrique mais des données différentes on obtiens des informations supplémentaires.



Evaluation orientée données

Problématique

Il y a un manque de méthodologie d'évaluation complète prenant en compte les spécificité du ML. De plus les approches actuelles manquent également de méthodologies d'évaluation standardisées.

Proposition

Nous proposons donc :

- une approche d'évaluation complète et standardisé,
- une approche orientée donnée permettant d'évaluer des aspects propre à l'utilisation du ML.

Contributions

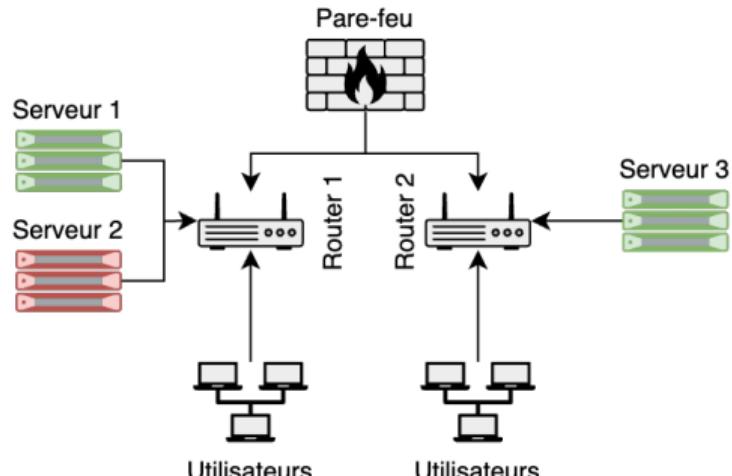
Cette thèse contribue à l'amélioration des méthodes d'évaluation des IDS basé sur ML :

- Architecture générale du cadre d'évaluation (Ayoubi et al. 2022)
- Formalisation du cadre d'évaluation
- Implémentation du cadre d'évaluation (Ayoubi et al. 2024b ; Ayoubi et al. 2024a ; Ayoubi et al. 2025)

Techniques d'évaluation traditionnelles

Les méthodes d'évaluation traditionnelles peuvent être classées en deux catégories :

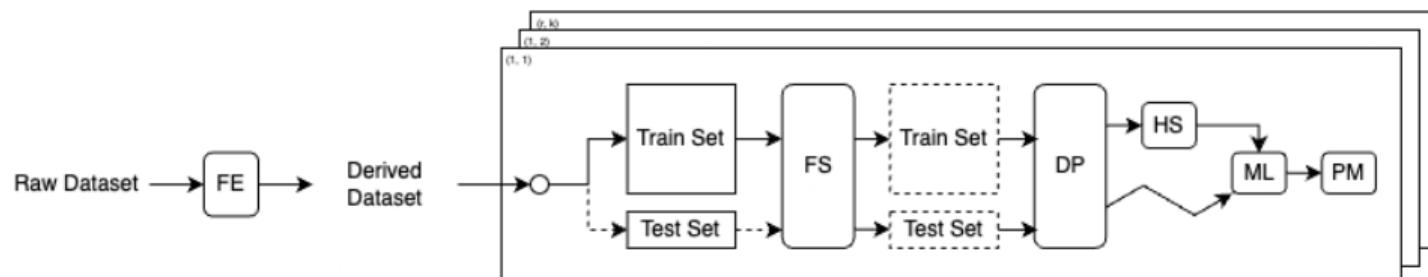
- Environnement de test (Testbed) :
 - Environnement statique (Lippmann et al. 2000b ; Lippmann et al. 2000a ; Rossey et al. 2002).
 - Environnement personnalisé (Singaraju, Teo et Yuliang Zheng 2004).
 - Environnement dynamiques et hybrides (Athanasiaades et al. 2003 ; Jadidbonab et al. 2021)
- Métriques spécialisées :
 - Analyse du coût (Gaffney et Ulvila 2000)
 - Intrusion Detection Capability (C_{ID}) (Gu et al. 2006 ; Imoize et al. 2018)



Évaluation spécifique aux IDS basés sur le ML

Les méthodes d'évaluation spécifique peuvent être classées en deux catégories principales :

- Approches standard de l'apprentissage automatique :
 - Standardisation (Magán-Carrión et al. 2020)
 - Automatisation (Zoppi, Ceccarelli et Bondavalli 2019)
- Approches orientées données :
 - Espace de conception pour l'évaluation (Milenkoski et al. 2015)
 - Mutation de trafic (Gaffney et Ulvila 2000)
 - Dérive conceptuelle (Andresini et al. 2021 ; Chua et Salam 2022)



Revue des méthodes d'évaluation dans l'état de l'art

	UNSW	Private	CIC-IDS	KDD99	Kyoto-Honeypot	Others	NSL-KDD	Accuracy	Precision / Recall	F-measure	ROC Curve / AUC	Confusion matrix / FAR	Anomaly Score (AS)	ERR	DR	EIR	MCC	Squared reconstruction error	Specificity	Efficiency	Classification tasks	Environment	Dataset manipulation	Resampling test set	Generate data	Multi-label evaluation	Model architecture	
Dataset																												
Metrics																												
Yu, Long et Cai 2017						✓		✓	✓	✓	✓	✓	✓	✓	✓						✓							
Tang et al. 2016								✓	✓	✓	✓	✓	✓	✓	✓								✓					
Al-Qatf et al. 2018						✓		✓	✓	✓	✓	✓	✓	✓	✓								✓					
Abbas et al. 2022			✓					✓														✓	✓					
Zixu, Liyanage et Gurusamy 2020	✓							✓	✓	✓																		
Khan 2021		✓							✓	✓	✓	✓	✓	✓	✓	✓	✓											
Zhang, Ran et Mi 2019								✓																				
Kim et al. 2016				✓																								
Nti et al. 2021						✓																						

Synthèse

- Les articles n'évaluent que l'efficacité de détection.
- Les chercheurs n'utilisent qu'un seul jeu de données et des métriques inadaptées
 - L'exactitude (accuracy) est la métrique la plus utilisée.

Problèmes émergents dans l'évaluation des IDS

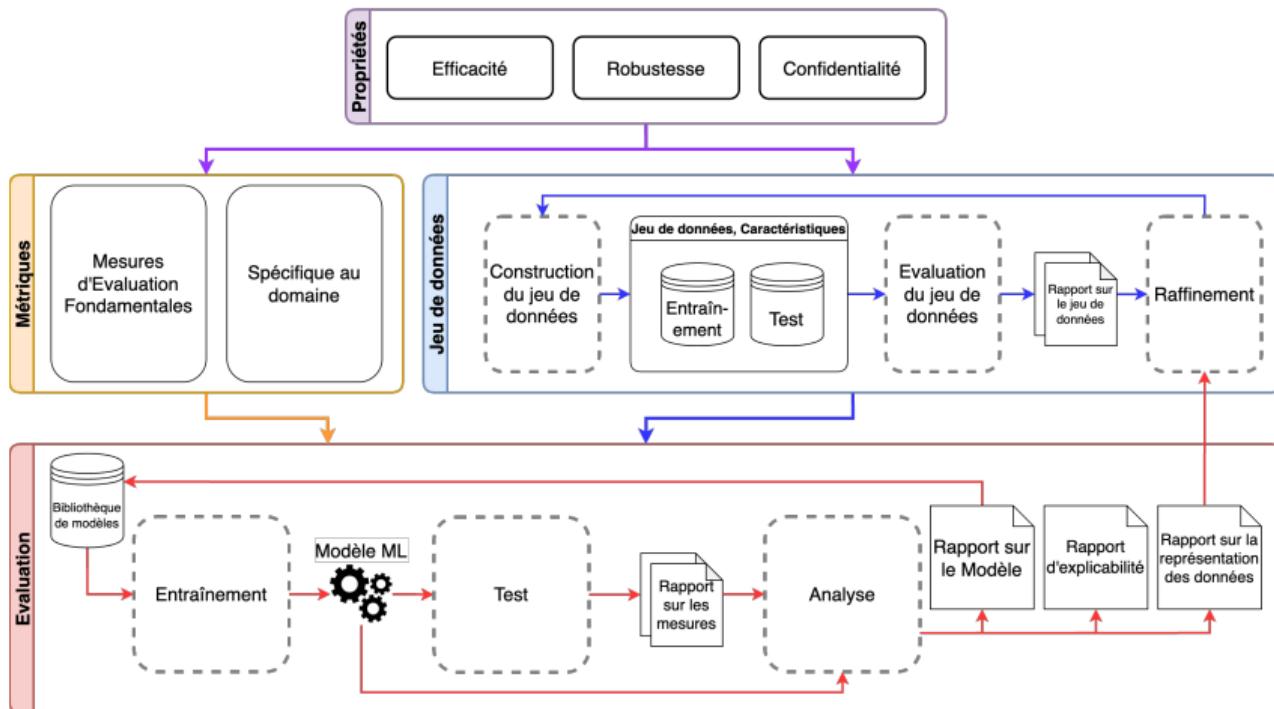
■ Problèmes liés aux jeux de données :

- Manque de données issues du monde réel (Chou et Jiang 2020)
- Erreurs de conception (Engelen, Rimmer et Joosen 2021; Rosay et al. 2022; Lanvin et al. 2023)
- Absence d'un ensemble de caractéristiques standardisé (Sarhan, Layeghy et Portmann 2022).
- Déséquilibre des jeu de données (Chou et Jiang 2020; Walling et Lodh 2022)

■ Problèmes liés à la méthodologie

- Pratiques expérimentales insuffisantes et manque de reproductibilité (Tavallaei, Stakhanova et Ghorbani 2010 ; Magán-Carrión et al. 2020).
- Nécessité d'une méthodologie plus pragmatique et structurée (Apruzzese, Laskov et Schneider 2023).

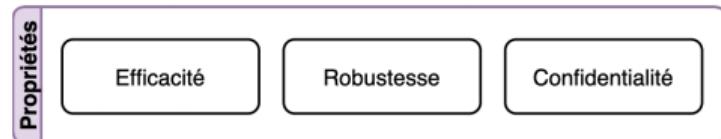
Proposition d'architecture d'évaluation



Proposition d'architecture d'évaluation

Propriétés

- Ne pas se limiter à l'efficacité : il faut aussi évaluer d'autres propriétés.
- Nous proposons de nouvelles propriétés influencées à la fois par :
 - des travaux dans le domaine de la détection des intrusions, tels que ceux d'Axelsson (Axelsson 2000),
 - et par des problèmes liés aux données dans le domaine du ML



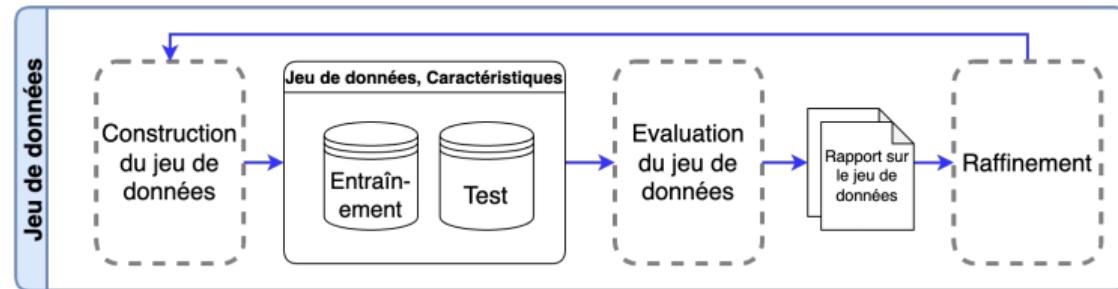
Proposition d'architecture d'évaluation

Métriques

- Il est essentiel de choisir les métriques appropriés afin d'évaluer correctement une propriété.
- Il existe plusieurs groupes de métriques :
 - Mesures d'évaluation fondamentales (accuracy, precision, recall).
 - Mesures d'évaluation combinées (g-mean, balanced accuracy, mcc).
 - Évaluation graphique des performances (ROC, AUC).
 - Spécifique au domaine (C_{ID})



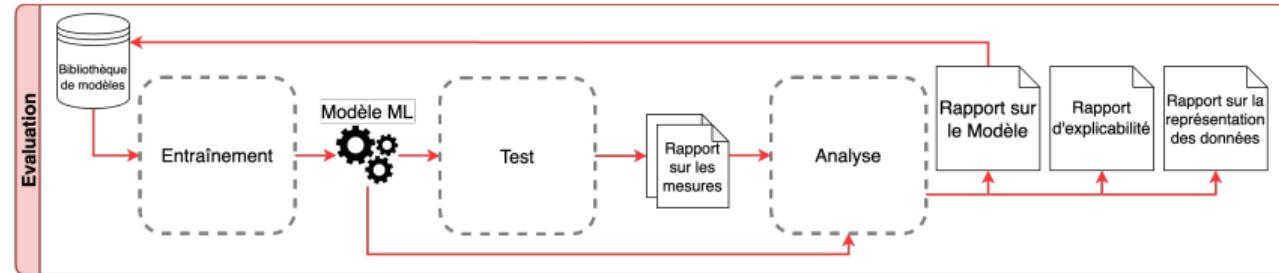
Proposition d'architecture d'évaluation



Jeu de données

- 1 Construction du jeu de données : inclut l'ensemble des étapes nécessaires, ainsi que celles spécifiques à la propriété.
- 2 Séparation du jeu de données : sépare si nécessaire le jeu de données en deux.
- 3 Evaluation du jeu de données : évalue la qualité du jeu de donnée produit selon plusieurs critères.
- 4 Raffinement du jeu de données : corrige et améliore le jeu de données.

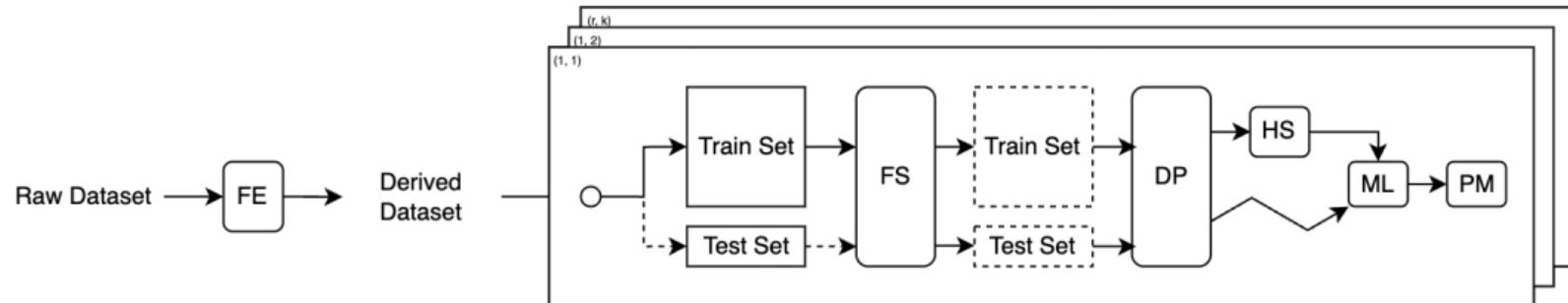
Proposition d'architecture d'évaluation



Évaluation

- Une fois le jeu de données construit et les métriques adaptées choisies, nous pouvons évaluer un ou plusieurs algorithmes :
 - 1 Entrainement et Test : entraîne simplement l'algorithme choisi avec le jeu de données, puis utilise le jeu de données de test pour calculer le résultat de chaque métrique.
 - 2 Analyse : permet d'obtenir un certain nombre de rapports très utiles pour l'amélioration de l'IDS.

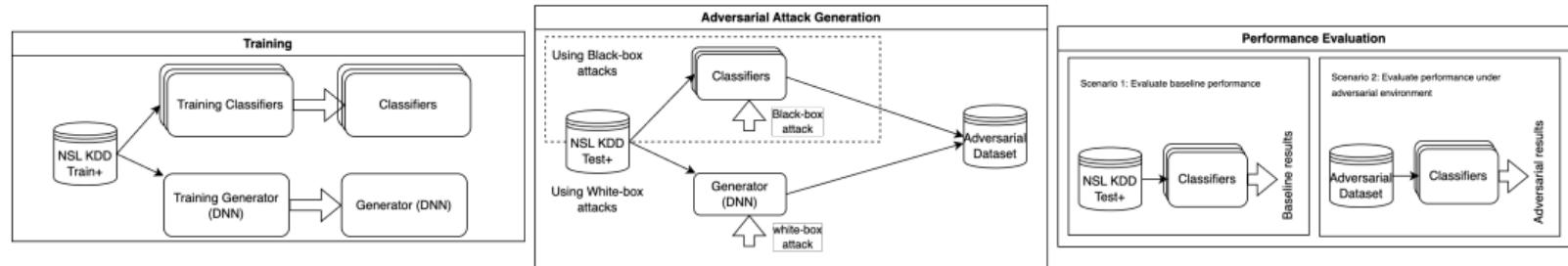
Schéma général du processus d'évaluation



Processus d'évaluation de l'efficacité (Magán-Carrión et al. 2020)

- Préparation des données : sélection de caractéristiques, normalisation, puis évaluation (et éventuelle amélioration) de la qualité du dataset.
- Utilisation de métriques adaptées : les performances sont mesurées avec des métriques pertinentes pour les données déséquilibrées (ex : F1-score, ROC-AUC, Balanced Accuracy)

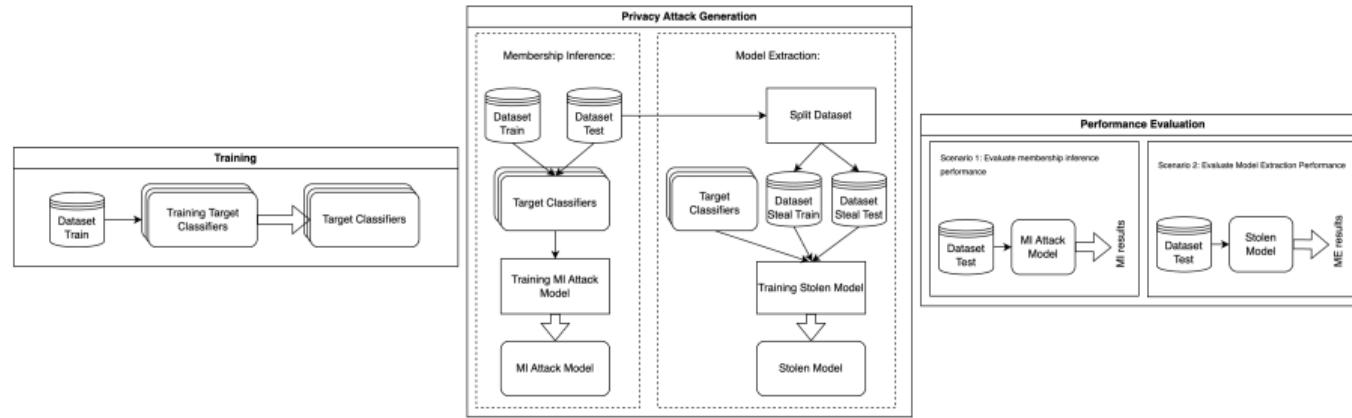
Schéma général du processus d'évaluation



Processus d'évaluation de la robustesse (Jmila et Khedher 2022)

- Attaques white-box et black-box : on entraîne d'abord les modèles, puis on génère des attaques adverses sur le jeu de test pour créer un adversarial dataset.
- Attaques par empoisonnement (poisoning) : on modifie le jeu d'entraînement au lieu du jeu de test, en y introduisant des attaques.

Schéma général du processus d'évaluation



Processus d'évaluation de la confidentialité

- Membership inference : on utilise un classifieur pour déterminer si des échantillons étaient dans le jeu d'entraînement du modèle.
- Model extraction : on tente de reproduire le modèle cible en lui envoyant des requêtes via un jeu de données de transfert.

Formalisation des étapes clés



Construction du jeu de données

- **Entrée :** $D := \{(x_i, y_i) \mid x_i \in \mathcal{X}^P, y_i \in \mathcal{Y}\}_{i=1}^N$ où $\mathcal{Y} = \{0, n\}$, avec $n \geq 1$.
- **Fonction :** $Construction : (\mathcal{X}^P \times \mathcal{Y}) \rightarrow (\mathcal{N}^{P'} \times \mathcal{Y})$.
- **Sortie :** $D'_F := \{(x''_i, y_i) \mid x''_i \in \mathcal{N}^{P'}, y_i \in \mathcal{Y}\}_{i=1}^N$.

Formalisation des étapes clés

Sélection des caractéristiques

- **Entrée** : $D := \{(x_i, y_i) \mid x_i \in \mathcal{X}^P, y_i \in \mathcal{Y}\}_{i=1}^N$
- **Fonction** : $S : \mathcal{X}^P \rightarrow \mathcal{X}^{P'}, P > P'$
- **Sortie** : $D_F := \{(x'_i, y_i) \mid x'_i \in \mathcal{X}^{P'}, y_i \in \mathcal{Y}\}_{i=1}^N$

Prétraitement des données

- **Entrée** : $D_F := \{(x'_i, y_i) \mid x'_i \in \mathcal{X}^{P'}, y_i \in \mathcal{Y}\}_{i=1}^N$
- **Fonction** : $N : \mathcal{X}^{P'} \rightarrow \mathcal{N}^{P'}$
- **Sortie** : $D'_F := \{(x''_i, y_i) \mid x''_i \in \mathcal{N}^{P'}, y_i \in \mathcal{Y}\}_{i=1}^N$

Formalisation des étapes clés

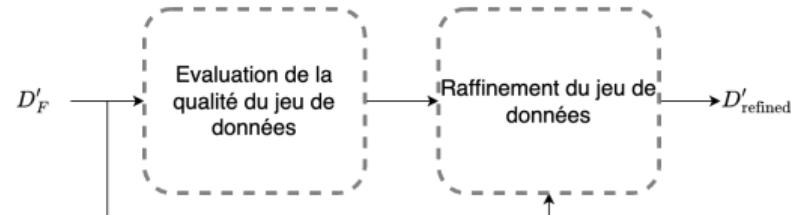
Augmentation avec des données adverses

- Entrée : D'_F
- Fonction :
 $A : \mathcal{N}^{P'} \times (\mathcal{N}^{P'} \rightarrow \mathcal{Y}) \rightarrow \mathcal{A}^{P'}$
- Sortie : $D_{\text{adv}} := \{(x_i^{\text{adv}}, y_i) \mid x_i^{\text{adv}} \in \mathcal{A}^P, y_i \in \mathcal{Y}\}_{i=1}^{N_{\text{adv}}}$

Augmentation avec des données d'attaques sur la confidentialité

- Entrée : D'_F
- Fonction :
 - MI : $(\mathcal{X}^P \times \mathcal{Y}) \times (\mathcal{X}^P \rightarrow \mathcal{Y}) \rightarrow (\mathcal{Y} \times \mathbb{B})$
 - ME : $(\mathcal{X}^P \times \mathcal{Y}) \times (\mathcal{X}^P \rightarrow \mathcal{Y}) \rightarrow (\mathcal{X}^P \times \mathcal{Y})$
- Sortie :
 - $D_{\text{attack}} := \{(\hat{h}_\theta(x_i), y_{\text{attack}}(x_i)) \mid \forall x_i \in D\}$
 - $D_{\text{attack}} = \{(x, h_\theta(x)) \mid x \in \mathcal{Q}\}$

Formalisation des étapes clés



Evaluation du jeu de données

- **Entrée :** D'_F .
- **Fonction :**
 $\text{Eval} : (\mathcal{N}^{P'} \times \mathcal{Y}) \rightarrow (\mathbb{B}^n \times \mathbb{R}^n)$.
- **Sortie :** $E := (c, s) \in \mathbb{B}^n \times \mathbb{R}^n$.

Raffinement du jeu de données

- **Entrée :** D'_F et E .
- **Fonction :** $\text{Refine} : (\mathcal{N}^{P'} \times \mathcal{Y}) \times (\mathbb{B}^n \times \mathbb{R}^n) \rightarrow (\mathcal{N}^{P'} \times \mathcal{Y})$.
- **Sortie :** D'_{refined} .

Formalisation des étapes clés



Séparation du jeu de données

- **Entrée :** $D'_{\text{refined}} := \{(x''_i, y_i) \mid x''_i \in \mathcal{N}^{P'}, y_i \in \mathcal{Y}\}_{i=1}^N$.
- **Fonction :** $\text{Split} : \mathcal{N}^{P'} \times \mathcal{Y} \rightarrow (\mathcal{N}^{P'} \times \mathcal{Y}) \times (\mathcal{N}^{P'} \times \mathcal{Y})$.
- **Sortie :** D_{train} et D_{test} .

Formalisation des étapes clés



Entrainement

- **Entrée** : D_{train} et h le modèle à entraîner.
- **Fonction** : Train : $(\mathcal{N}^{P'} \times \mathcal{Y}) \times (\mathcal{N}^{P'} \rightarrow \mathcal{Y}) \rightarrow (\mathcal{N}^{P'} \rightarrow \mathcal{Y})$.
- **Sortie** : h_θ le modèle entraîné et paramétré par θ .

Formalisation des étapes clés



Evaluation

- Entrée : D_{test} et h_θ .
- Fonction : Report : $(\mathcal{N}^{P'} \times \mathcal{Y}) \times (\mathcal{N}^{P'} \rightarrow \mathcal{Y}) \rightarrow \mathbb{R}^n$.
- Sortie : $R := (r_1, r_2, \dots, r_n) \in \mathbb{R}^n$ l'ensemble des métriques.

Comparaison avec les cadres d'évaluation existants

Reference	Properties	Dataset Construction	Dataset Evaluation	Refinement	Domain Specific Metrics	Domain Specific Metrics
Magán-Carrión et al. 2020	*	*				
Bermúdez-Edo et al. 2006	*	*				
Cárdenas, Baras et Seamon 2006	*				✓	
Milenkoski et al. 2015	✓	*			✓	
Our Proposal	✓	✓	✓	✓	✓	✓

Limites des approches traditionnelles

- Peu d'efforts des concepteurs de modèles : datasets obsolètes, métriques inadéquates, absence d'outils pour manipuler les données.
- Champ d'évaluation limité : rare prise en compte de propriétés comme la robustesse.
- Faible adoption des nouvelles méthodes : manque de solutions pratiques et reproductibles.

Conclusion

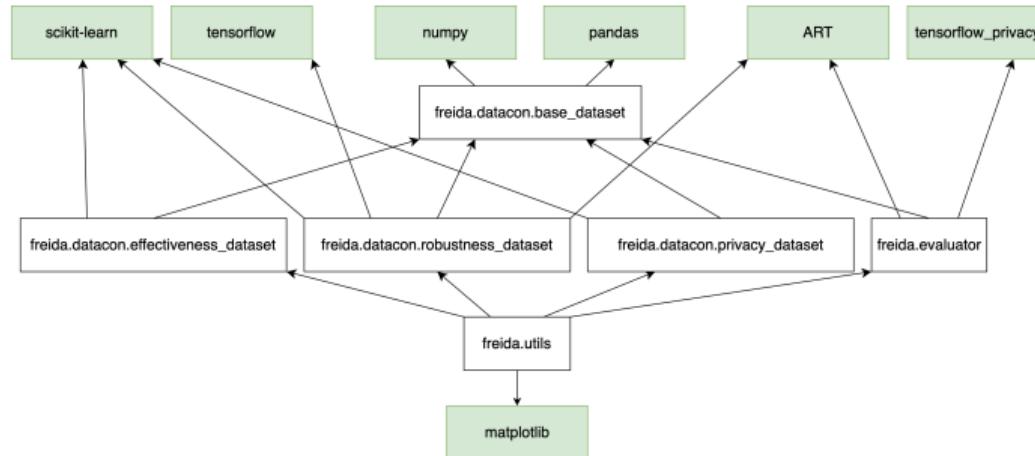
- Peu de travaux existants proposent une évaluation complète des IDS.
- Nous proposons une méthodologie d'évaluation systématique et modulaire des IDS.
- Notre cadre d'évaluation se concentre sur trois aspects principaux :
 - 1 l'efficacité de détection,
 - 2 la robustesse face aux attaques adversariales,
 - 3 et la confidentialité.
- Le formalisme proposé offre une base rigoureuse tout en restant flexible pour des extensions futures.
- La partie suivante détaille l'implémentation du framework en Python pour illustrer sa mise en pratique.

Choix technologiques et méthodologies

Fonctionnalité	FREIDA Beta	FREIDA	MLflow Plugin
Implémentation	Python simple	Python simple	basé MLflow
Sans dépendance	✓	✓	✗
Propriétés évaluées	Efficacité	Efficacité, Robustesse, Confidentialité	Efficacité
Support de Scikit-Learn	✓	✓	✓
Support de TensorFlow	✗	✓	✗
Docker	✗	✓	✗
Gestionnaire de paquets	Poetry	pip	pip
Problème principal	Designs choices	WIP	Limited automation before training

- Code disponible sur GitLab, versionné selon les publications.
- Docker pour garantir une exécution stable et reproductible.
- Bibliothèques clés : pandas, numpy, scikit-learn, TensorFlow, Keras, Solara.
- Attaques & confidentialité : ART pour les attaques adversariales, tensorflow-privacy pour la protection des données.

Architecture logicielle



- Deux modules principaux : Dataset et Evaluation, organisant les étapes d'évaluation :
 - 1 Construction du dataset.
 - 2 Entraînement.
 - 3 Évaluation.
- Fichiers de configuration : centralisent tous les paramètres.

Utilisation de l'outil d'évaluation

Intégration en tant que module Python

- Installation : cloner le dépôt GitLab et installer le module Python.
- Évaluation :
 - Créer un fichier Python et importer le module.
 - Préparer une configuration.
 - Charger le dataset CSV et instancier les modèles.
 - Appeler la fonction d'évaluation de FREIDA avec : la liste de modèles, le jeu de données et la configuration.
- Exemple : avec ce fichier de config , on peut évaluer l'efficacité et la robustesse.

```
{  
  "seed": 42,  
  "properties": ["effectiveness", "robustness"],  
  "features_selection": "LASSO",  
  "alpha_lasso": 0.01,  
  "preprocessing": "StandardScaler",  
  "drop_columns": ["id", "attack_cat"],  
  "label_column": "label",  
  "class_to_remove": "Generic",  
  "attacks": ["FGSM", "ZOO", "Poisoning"],  
  "dataset_evaluation": ["null_values"],  
  "dataset_refinement": ["SimpleImputer"],  
  "splitting": "imbalanced",  
  "test_size": 0.2,  
  "is_pretrained": false  
}
```

Utilisation de l'outil d'évaluation

Application autonome avec interface graphique

1 Configuration :

- Définissez les paramètres via l'interface.
- Exportez/importez des configurations pour assurer la reproductibilité.

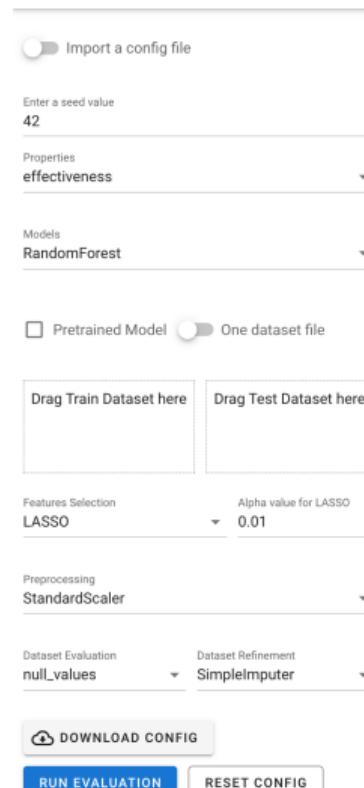
2 Propriétés à évaluer : Efficacité, Robustesse.

3 Choix des modèles :

- Liste limitée dans l'interface.
- Support complet de Scikit-learn et Keras via la version Python.

4 Chargez un dataset déjà splitté ou laissez FREIDA le faire.

5 Lancer l'évaluation.



The screenshot shows a configuration interface for an evaluation tool. It includes sections for importing configurations, setting seed values, choosing properties like 'effectiveness', selecting models like 'RandomForest', and managing datasets through 'Drag Train Dataset here' and 'Drag Test Dataset here' fields. It also features sections for feature selection ('LASSO'), preprocessing ('StandardScaler'), and dataset refinement ('SimpleImputer'). At the bottom are buttons for 'DOWNLOAD CONFIG', 'RUN EVALUATION', and 'RESET CONFIG'.

Import a config file

Enter a seed value
42

Properties
effectiveness

Models
RandomForest

Pretrained Model One dataset file

Drag Train Dataset here Drag Test Dataset here

Features Selection
LASSO

Alpha value for LASSO
0.01

Preprocessing
StandardScaler

Dataset Evaluation
null_values

Dataset Refinement
SimpleImputer

DOWNLOAD CONFIG

RUN EVALUATION

RESET CONFIG

Utilisation de l'outil d'évaluation

Freida

Import a config file

Enter a seed value
42

Properties
effectiveness, robustness

Models
RandomForest

Pretrained Model One dataset file

Attacks
FGSM

Drag Train Dataset here Drag Test Dataset here

Label Column
id

Multi-Label Column (Open World Scenario)

Drop Columns
id

Dataset Preview

Train Dataset

#	id	dur	proto	service	state	splts	dpkts	shbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	simpkt	dimpkt	sjlt	djlt	swin	stopb	dwpb	dwin	tcprrt	synack	ackdat	smmean	dmean	trans_depth	response_be
0	1	1.1e-05	udp	-	INT	2	0	496	0	90909.0902	254.0	180363632.0	0.0	0	0	0.011	0.0	0.00.00	0	0	0	0.0	0.0	0.0	248	0	0	0			
1	2	8e-06	udp	-	INT	2	0	1762	0	125000.0003	254.0	881000000.0	0.0	0	0	0.008	0.0	0.00.00	0	0	0	0.0	0.0	0.0	881	0	0	0			
2	3	5e-06	udp	-	INT	2	0	1068	0	200000.0051	254.0	854400000.0	0.0	0	0	0.005	0.0	0.00.00	0	0	0	0.0	0.0	0.0	534	0	0	0			
3	4	4e-06	udp	-	INT	2	0	900	0	166666.6608	254.0	600000000.0	0.0	0	0	0.006	0.0	0.00.00	0	0	0	0.0	0.0	0.0	450	0	0	0			
4	5	1e-05	udp	-	INT	2	0	2126	0	100000.0025	254.0	850400000.0	0.0	0	0	0.01	0.0	0.00.00	0	0	0	0.0	0.0	0.0	1063	0	0	0			

Rows per page: 10 ▾ 1-5 of 82332 < >

Test Dataset

#	id	dur	proto	service	state	splts	dpkts	shbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	simpkt	dimpkt	sjlt	djlt	swin	stopb	dwpb	dwin	tcprrt	synack	ackdat	smmean	dmean	trans_depth	response_be
0	1	0.121478	tcp	-	FIN	6	4	258	172	74.08749	252.254	14158.94238	8495.365234	0	0	24.2956	8.375	30.177547	11.830604	255	621772692	2202533631	:	:	:	:	:	:	:	:	
1	2	0.649902	tcp	-	FIN	14	38	734	42014	78.473372	62	252.8395.112305	503571.3125	2	17	49.915	15.432865	61.426934	1387.77833	255	1417884146	3077387971	:	:	:	:	:	:	:	:	
2	3	1.623129	tcp	-	FIN	8	16	364	13186	14.170161	62	252.1572.271851	60929.23047	1	6	231.87571	102.737203	17179.58686	11420.92623	255	2116150707	2963114973	:	:	:	:	:	:	:	:	
3	4	1.681642	tcp	ftp	FIN	12	12	628	770	13.677108	62	252.2740.178955	3358.62207	1	3	152.876547	90.235728	259.080172	4991.784669	255	1107119177	1047442890	:	:	:	:	:	:	:	:	
4	5	0.449454	tcp	-	FIN	10	6	534	268	33.373826	254.252	8561.499023	3987.059814	2	1	47.750333	75.659602	2415.837634	115.807	255	2436137549	19771514190	:	:	:	:	:	:	:	:	

Rows per page: 10 ▾ 1-5 of 175341 < >

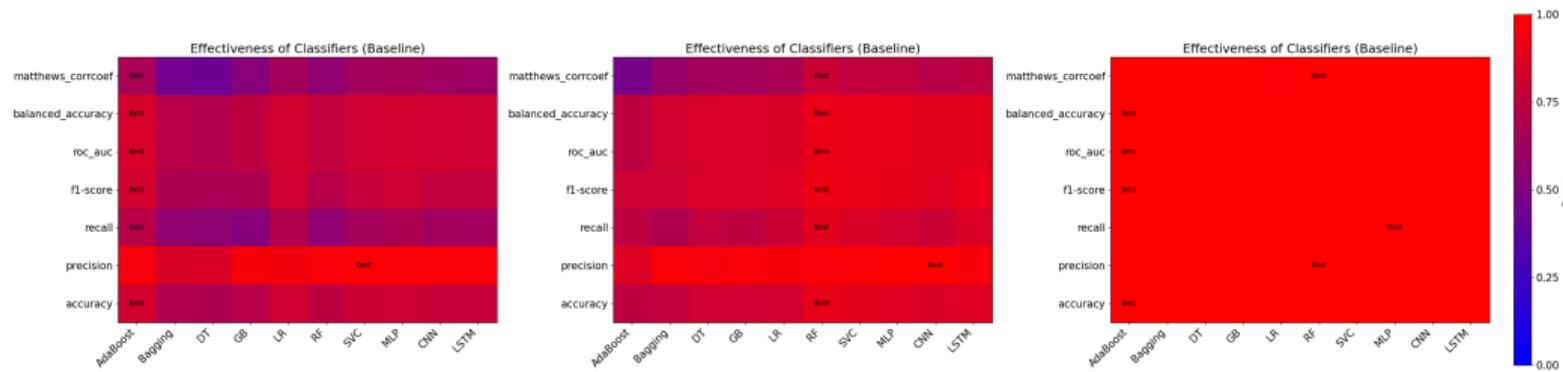
Evaluation Results

No evaluation in progress, please fill the configuration and launch the evaluation process.

Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données

- Objectif : évaluer la facilité d'utilisation de FREIDA en comparant la performance de différents modèles sur des jeux de données de référence.
- 10 modèles évalués :
 - Classiques (ML) : Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Classifier (SVC), AdaBoost, Bagging, Gradient Boosting (GB).
 - Profonds (DL) : Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM)
- 3 jeux de données utilisés :
 - NSL-KDD (Tavallaei et al. 2009)
 - UNSW-NB15 (Moustafa et Slay 2015)
 - CICIDS-2017 (Sharafaldin, Lashkari et Ghorbani 2018)
- Critères de sélection :
 - Modèles et datasets fréquemment utilisés dans la littérature IDS.
 - Citation élevée des jeux de données.

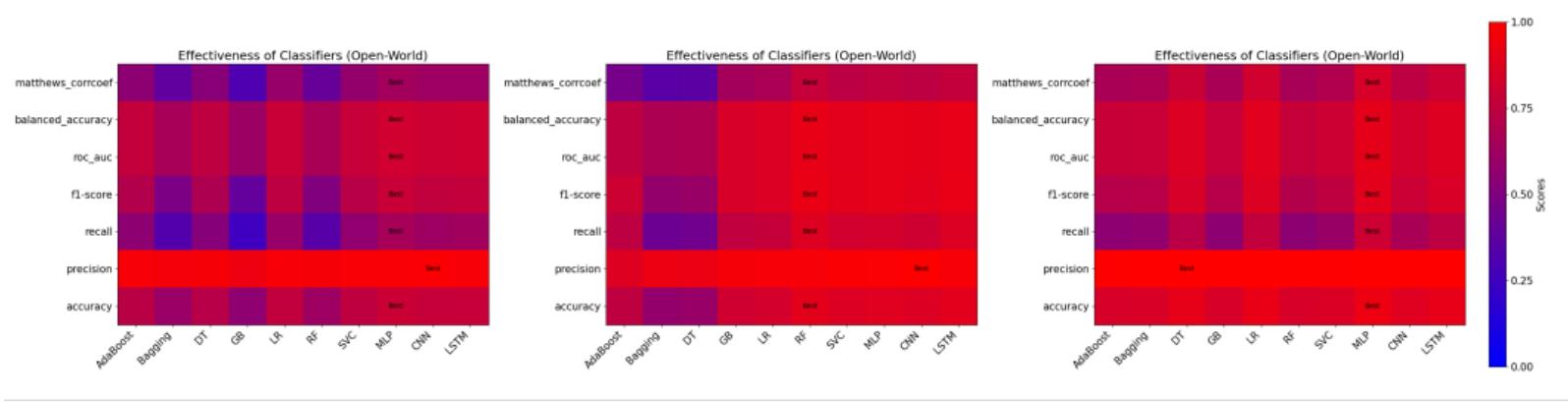
Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données



Efficacité (base)

Les performances des modèles varient selon les jeux de données, avec des résultats exceptionnels sur CIC-IDS2017, tandis que RF excelle sur UNSW-NB15 et AdaBoost sur NSL-KDD.

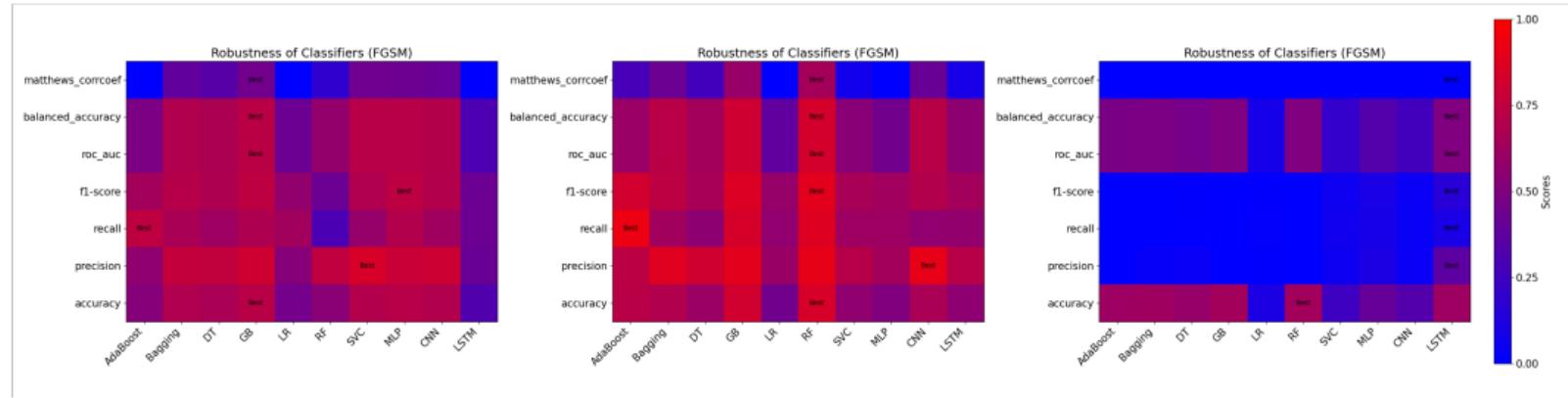
Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données



Efficacité (Open-World)

Les performances globales baissent, mais les meilleurs modèles restent les mêmes : RF reste solide sur UNSW-NB15, MLP surpassé les autres sur CIC-IDS2017, et les modèles de type réseau de neurones s'adaptent mieux sur NSL-KDD

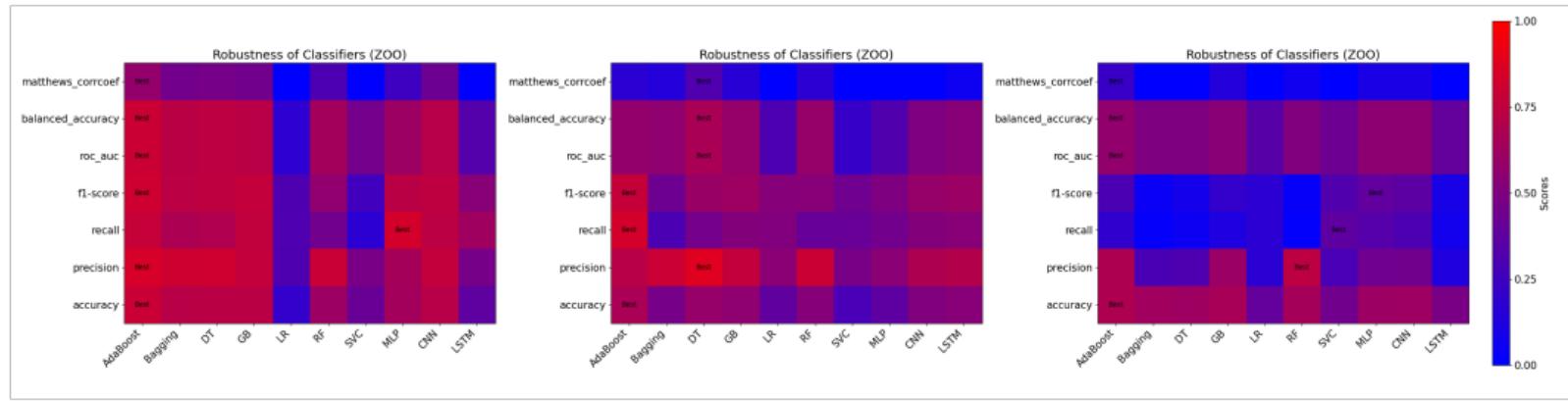
Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données



Robustesse (FGSM)

Les modèles montrent une robustesse variable : RF et Bagging résistent bien sur UNSW-NB15, Bagging et GB sont les plus robustes sur NSL-KDD, tandis que tous les modèles, échouent sévèrement sur CIC-IDS2017.

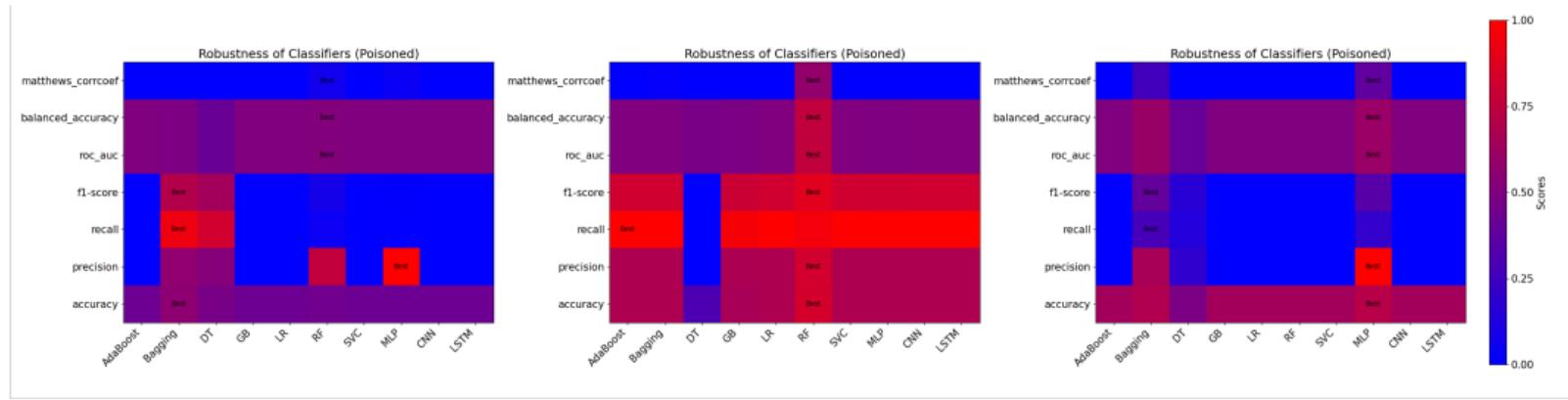
Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données



Robustesse (ZOO)

AdaBoost s'est révélé le plus robuste sur les trois jeux de données, tandis que RF, LR et les modèles à base de réseaux de neurones ont montré de fortes vulnérabilités, en particulier sur CIC-IDS2017 et NSL-KDD.

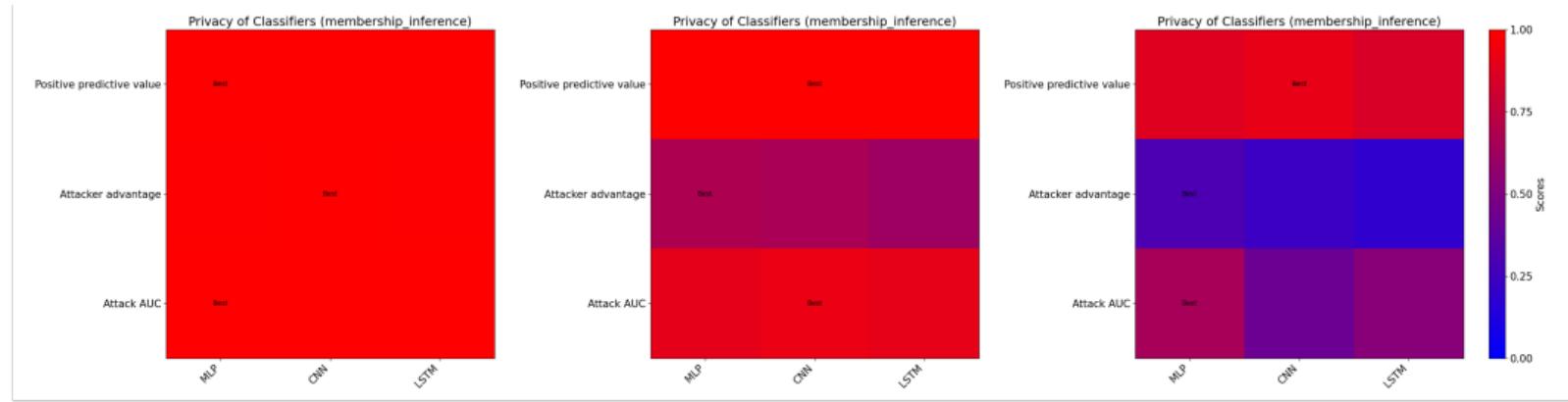
Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données



Robustesse (Empoisonnement)

La plupart des modèles restent relativement robustes sur UNSW-NB15, alors qu'ils s'effondrent sur NSL-KDD, tandis que sur CIC-IDS2017, seul MLP montre une certaine efficacité, les autres échouant à détecter les instances positives.

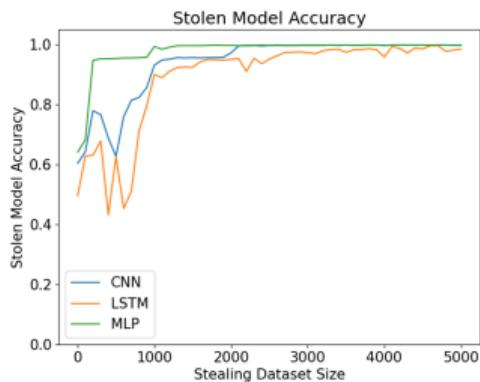
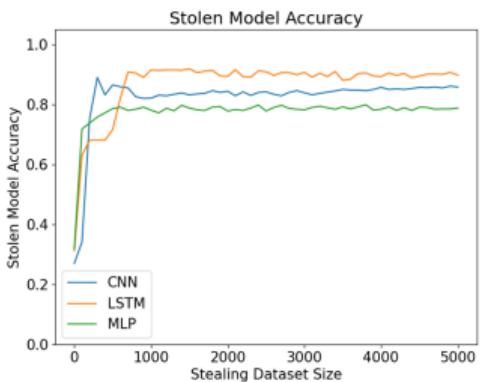
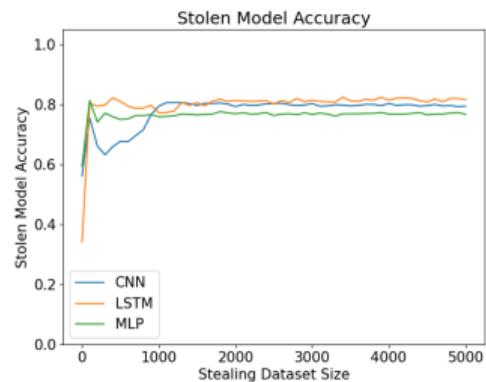
Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données



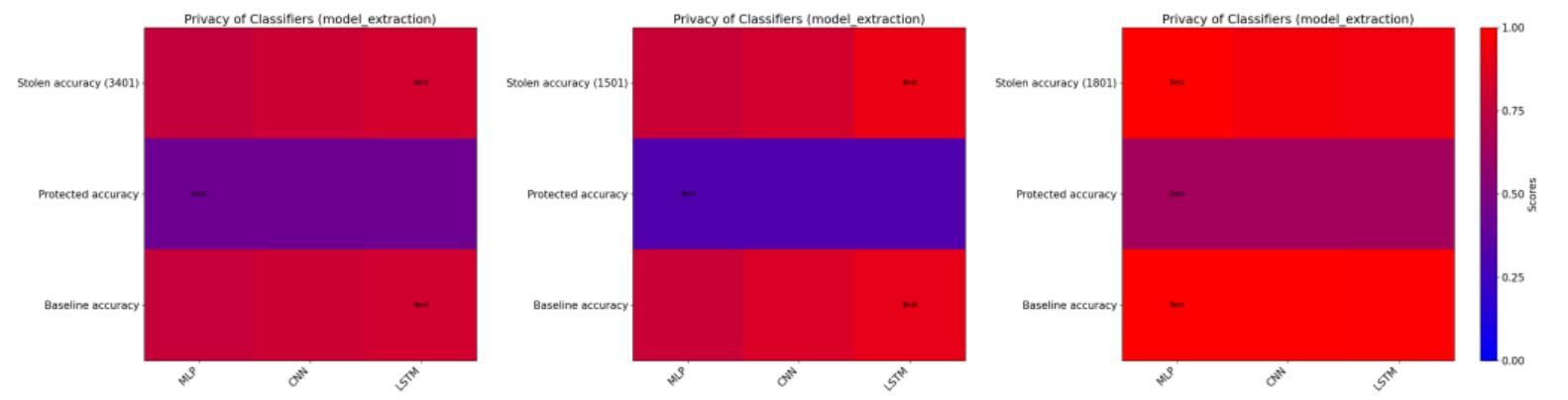
Confidentialité (Membership Inference)

L'évaluation révèle que la vulnérabilité à cette attaque varie selon le dataset et l'architecture du modèle, les CNN étant les plus exposés, surtout sur NSL-KDD et UNSW-NB15.

Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données



Cas d'utilisation : évaluation de plusieurs IDS avec différents jeux de données



Confidentialité (Model Extraction)

L'évaluation montre que les modèles extraits peuvent reproduire fidèlement les performances des modèles attaqués avec peu de requêtes, mais que la protection Reverse Sigmoid permet de bloquer efficacement ces attaques sans nuire à la précision du modèle.

Comparaison avec RELOAD

Outil	Extensible	Interopérable	Efficacité	Robustesse	Confidentialité
RELOAD basé sur Java	*	*			
FREIDA basé sur Python	✓	✓	✓	✓	✓

- RELOAD offre une extensibilité limitée en raison de son architecture basée sur Java. :
 - Manque de support pour les méthodes modernes de ML basés sur Python.
 - Pas d'évaluation de la robustesse ou de la confidentialité.
- FREIDA est disponible sous forme d'interface graphique et de paquet Python pour une intégration flexible. :
 - Compatible avec Scikit-Learn et Keras.
 - Facilement extensible et interopérable avec des outils modernes.
 - RELOAD dispose d'une interface graphique plus soignée et plus conviviale.
 - FREIDA offre une solution plus complète, plus souple et plus moderne.

Conclusion

- Évaluations Reproductibles et Comparables :
 - Permet de comparer plusieurs IDS ou configurations avec les mêmes données, favorisant la reproductibilité.
- Support Étendu des Tâches IDS :
 - Gère la classification binaire, multiconfiantes et la détection d'anomalies sur des données tabulaires.
- Manipulation des Données pour Évaluation Généralisée :
 - Évite les biais liés au "closed world" en modifiant les jeux de données pour tester la robustesse.
- Cadre Rigoureux et Pratique.
- Ouverture à l'Évolution :
 - Prêt pour l'intégration de nouveaux modèles, métriques et fonctionnalités futures (explicabilité, représentations).

Conclusion générale

- Notre cadre d'évaluation comble des lacunes critiques des méthodologies existantes pour les IDS basés sur le ML.
- FREIDA propose une approche systématique d'évaluation des IDS selon plusieurs propriétés :
 - efficacité de la détection,
 - robustesse face aux attaques adversariales,
 - et vulnérabilités en matière de confidentialité.
- Notre implémentation et la formalisation garantissent un benchmarking rigoureux des IDS.
- Ce cadre d'évaluation soutient ainsi le développement d'IDS plus robustes et adaptatives face aux menaces cybernétiques en constante évolution.

Perspectives générales

- Étendre l'évaluation à d'autre algorithme et évaluer des méthodes plus poussé de l'état de l'art tels que Kitsune. (Mirsky et al. 2018)
- Développer une interface graphique (GUI) plus complète, permettant aux utilisateurs de créer des modèles Keras personnalisés directement depuis l'interface.
- Ajoutée une étape d'extraction des caractéristiques pour adapter le cadre aux spécificités des IDS, en transformant les fichiers PCAP en formats exploitables pour l'évaluation.
- Intégrer de nouvelles propriétés d'évaluation tels que l'équité (fairness).

Merci de votre attention
Des questions ?

References |

-  Abbas, Adeel et al. (2022). "A New Ensemble-Based Intrusion Detection System for Internet of Things". In : *Arabian Journal for Science and Engineering* 47.2, p. 1805-1819. issn : 2191-4281. doi : 10.1007/s13369-021-06086-5. url : <https://doi.org/10.1007/s13369-021-06086-5>.
-  Al-Qatf, Majzed et al. (2018). "Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection". In : *IEEE Access* 6, p. 52843-52856. doi : 10.1109/ACCESS.2018.2869577.
-  Andresini, Giuseppina et al. (2021). "INSOMNIA : Towards Concept-Drift Robustness in Network Intrusion Detection". In : *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*. AISec '21. Virtual Event, Republic of Korea : Association for Computing Machinery, 111–122. isbn : 9781450386579. doi : 10.1145/3474369.3486864. url : <https://doi.org/10.1145/3474369.3486864>.
-  Apruzzese, Giovanni, Pavel Laskov et Johannes Schneider (2023). "SoK : Pragmatic Assessment of Machine Learning for Network Intrusion Detection". In : *arXiv preprint arXiv :2305.00550*.

References II

-  Athanasiades, Nicholas et al. (2003). "Intrusion detection testing and benchmarking methodologies". In : *First IEEE International Workshop on Information Assurance, 2003. IWIAS 2003. Proceedings*. IEEE, p. 63-72.
-  Axelsson, Stefan (2000). "The base-rate fallacy and the difficulty of intrusion detection". In : *ACM Transactions on Information and System Security (TISSEC) 3.3*, p. 186-205.
-  Ayoubi, Solayman et al. (2022). "Data-driven evaluation of intrusion detectors : a methodological framework". In : *International Symposium on Foundations and Practice of Security*. Springer, p. 142-157.
-  Ayoubi, Solayman et al. (2024a). "FREIDA : A Concrete Tool for Reproducible Evaluation of IDS using a Data-driven Approach". In : *International Conference on Risks and Security of Internet and Systems*. Springer.
-  Ayoubi, Solayman et al. (2024b). "Towards Reproducible Evaluations of ML-Based IDS Using Data-Driven Approaches". In : *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, p. 5081-5083.

References III

-  Ayoubi, Solayman et al. (2025). "Privacy Benchmarking of Intrusion Detection Systems". In : *Advanced Information Networking and Applications*. Sous la dir. de Leonard Barolli. Cham : Springer Nature Switzerland, p. 406-417.
-  Bermúdez-Edo, María et al. (2006). "Proposals on assessment environments for anomaly-based network intrusion detection systems". In : *International Workshop on Critical Information Infrastructures Security*, p. 210-221.
-  Cárdenas, A.A., J.S. Baras et K. Seamon (2006). "A framework for the evaluation of intrusion detection systems". In : *2006 IEEE Symposium on Security and Privacy (S&P'06)*. Berkeley/Oakland, CA, p. 15-77. isbn : 978-0-7695-2574-7.
-  Chou, Dylan et Meng Jiang (2020). "Data-Driven Network Intrusion Detection : A Taxonomy of Challenges and Methods". In : ArXiv abs/2009.07352. url : <https://api.semanticscholar.org/CorpusID:221739085>.
-  Chua, Tuan-Hong et Iftekhar Salam (2022). "Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection System". In : ArXiv abs/2203.05232.

References IV

-  Engelen, Gints, Vera Rimmer et Wouter Joosen (2021). "Troubleshooting an Intrusion Detection Dataset : the CICIDS2017 Case Study". In : *2021 IEEE Security and Privacy Workshops (SPW)*, p. 7-12. doi : 10.1109/SPW53761.2021.00009.
-  Gaffney, John E et Jacob W Ulvila (2000). "Evaluation of intrusion detectors : A decision theory approach". In : *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*. IEEE, p. 50-61. doi : 10.1109/SECPRI.2001.924287.
-  Gu, Guofei et al. (2006). "Measuring intrusion detection capability : An information-theoretic approach". In : *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, p. 90-101.
-  Imoize, Agbotiname L. et al. (2018). "Software Intrusion Detection Evaluation System : A Cost-Based Evaluation of Intrusion Detection Capability". In : *Communications and Network* 10.4, p. 211-229. issn : 1949-2421, 1947-3826. doi : 10.4236/cn.2018.104017. url : <http://www.scirp.org/journal/doi.aspx?doi=10.4236/cn.2018.104017> (visité le 25/03/2021).

References V

-  Jadidbonab, Hesamaldin et al. (2021). "A Real-Time In-Vehicle Network Testbed for Machine Learning-Based IDS Training and Validation". In : *AI-Cybersec@SGAI*. url : <https://api.semanticscholar.org/CorpusID:248453976>.
-  Jmila, Houda et Mohamed Ibn Khedher (2022). "Adversarial machine learning for network intrusion detection : a comparative study". In : *Computer Networks* 214, 109073 :1-109073 :14.
-  Khan, Muhammad Ashfaq (2021). "HCRNNIDS : Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System". In : *Processes* 9.5. issn : 2227-9717. doi : 10.3390/pr9050834. url : <https://www.mdpi.com/2227-9717/9/5/834>.
-  Kim, Jihyun et al. (2016). "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection". In : *2016 International Conference on Platform Technology and Service (PlatCon)*, p. 1-5. doi : 10.1109/PlatCon.2016.7456805.
-  Lanvin, Maxime et al. (2023). "Errors in the CICIDS2017 Dataset and the Significant Differences in Detection Performances It Makes". In : *Risks and Security of Internet and Systems*. Sous la dir. de Slim Kallel et al. Cham : Springer Nature Switzerland, p. 18-33. isbn : 978-3-031-31108-6.

References VI

-  Lippmann, Richard et al. (2000a). "The 1999 DARPA off-line intrusion detection evaluation". In : *Computer Networks* 34.4. Recent Advances in Intrusion Detection Systems, p. 579-595. issn : 1389-1286. doi : [https://doi.org/10.1016/S1389-1286\(00\)00139-0](https://doi.org/10.1016/S1389-1286(00)00139-0). url : <https://www.sciencedirect.com/science/article/pii/S1389128600001390>.
-  Lippmann, R.P. et al. (2000b). "Evaluating intrusion detection systems : the 1998 DARPA off-line intrusion detection evaluation". In : *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*. T. 2, 12-26 vol.2. doi : 10.1109/DISCEX.2000.821506.
-  Magán-Carrión, Roberto et al. (2020). "Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches". In : *Applied Sciences* 10.5. Publisher : Multidisciplinary Digital Publishing Institute, p. 1775. doi : 10.3390/app10051775.
-  Milenkoski, Aleksandar et al. (2015). "Evaluating computer intrusion detection systems : A survey of common practices". In : *ACM Computing Surveys (CSUR)* 48.1. Publisher : ACM New York, NY, USA, p. 1-41. doi : 10.1145/2808691.
-  Mirsky, Yisroel et al. (2018). "Kitsune : An Ensemble of Autoencoders for Online Network Intrusion Detection". In : *ArXiv* abs/1802.09089.

References VII

-  Moustafa, Nour et Jill Slay (2015). "UNSW-NB15 : a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". In : *2015 Military Communications and Information Systems Conference (MilCIS)*, p. 1-6.
-  Nti, Isaac Kofi et al. (2021). "Network Intrusion Detection with StackNet : A phi coefficient Based Weak Learner Selection Approach". In : *2021 22nd International Arab Conference on Information Technology (ACIT)*, p. 1-11. doi : 10.1109/ACIT53391.2021.9677338.
-  Rosay, Arnaud et al. (2022). "Network intrusion detection : A comprehensive analysis of CIC-IDS2017". In : *8th International Conference on Information Systems Security and Privacy*. SCITEPRESS-Science et Technology Publications, p. 25-36.
-  Rossey, L.M. et al. (2002). "LARIAT : Lincoln adaptable real-time information assurance testbed". In : *Proceedings, IEEE Aerospace Conference*. T. 6, p. 6-6. doi : 10.1109/AERO.2002.1036158.
-  Sarhan, Mohanad, Siamak Layeghy et Marius Portmann (2022). "Towards a standard feature set for network intrusion detection system datasets". In : *Mobile networks and applications*, p. 1-14.

References VIII

-  Sharafaldin, Iman, Arash Habibi Lashkari et Ali A. Ghorbani (2018). "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization". In : *International Conference on Information Systems Security and Privacy*.
-  Singaraju, G., L. Teo et Yuliang Zheng (2004). "A testbed for quantitative assessment of intrusion detection systems using fuzzy logic". In : *Second IEEE International Information Assurance Workshop, 2004. Proceedings*. Second IEEE International Information Assurance Workshop, 2004. Charlotte, NC, USA : IEEE, p. 79-93. isbn : 978-0-7695-2117-6. doi : 10.1109/IWIA.2004.1288040. url : <http://ieeexplore.ieee.org/document/1288040/> (visité le 25/03/2021).
-  Tang, Tuan A et al. (2016). "Deep learning approach for Network Intrusion Detection in Software Defined Networking". In : *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, p. 258-263. doi : 10.1109/WINCOM.2016.7777224.

References IX

-  Tavallaei, M., N. Stakhanova et A. A. Ghorbani (2010). "Toward Credible Evaluation of Anomaly-Based Intrusion-Detection Methods". In : *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.5, p. 516-524. doi : 10.1109/TSMCC.2010.2048428.
-  Tavallaei, Mahbod et al. (2009). "A detailed analysis of the KDD CUP 99 data set". In : *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, p. 1-6. doi : 10.1109/CISDA.2009.5356528.
-  Walling, Supongmen et Sibesh Lodh (2022). "A survey on intrusion detection systems : Types, datasets, machine learning methods for NIDS and challenges". In : *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, p. 1-7.
-  Yu, Yang, Jun Long et Zhiping Cai (2017). "Network Intrusion Detection through Stacking Dilated Convolutional Autoencoders". In : *Security and Communication Networks* 2017, p. 4184196. issn : 1939-0114. doi : 10.1155/2017/4184196. url : <https://doi.org/10.1155/2017/4184196>.

References X

-  Zhang, Xiaoxuan, Jing Ran et Jize Mi (2019). "An Intrusion Detection System Based on Convolutional Neural Network for Imbalanced Network Traffic". In : *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, p. 456-460. doi : [10.1109/ICCSNT47585.2019.8962490](https://doi.org/10.1109/ICCSNT47585.2019.8962490).
-  Zixu, Tian, Kushan Sudheera Kalupahana Liyanage et Mohan Gurusamy (2020). "Generative Adversarial Network and Auto Encoder based Anomaly Detection in Distributed IoT Networks". In : *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, p. 1-7. doi : [10.1109/GLOBECOM42002.2020.9348244](https://doi.org/10.1109/GLOBECOM42002.2020.9348244).
-  Zoppi, Tommaso, Andrea Ceccarelli et Andrea Bondavalli (2019). "Evaluation of Anomaly Detection Algorithms Made Easy with RELOAD". In : *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, p. 446-455. doi : [10.1109/ISSRE.2019.00051](https://doi.org/10.1109/ISSRE.2019.00051).