CYBERML 2026

Lucas Collemare, Antoine Malmezac, Cyprien Deruelle
Dataset : CIC IoT-DIAD 2024 (Packet-Based Features)

28 janvier 2026

Table des matières

1 Abstract

Ce rapport présente une chaîne batch complète de traitement de données réseau pour la détection d'anomalies (non supervisée) et la classification d'attaques dans un contexte IoT. L'étude est réalisée sur CIC IoT-DIAD 2024 (Packet-Based Features) au format CSV. Pour limiter les fuites de données, l'évaluation utilise un split par fichier (group) : les échantillons d'un même CSV ne peuvent pas apparaître simultanément en apprentissage et en test.

En détection d'anomalies binaire (Benign vs Attack), trois approches non supervisées (Isolation Forest, One-Class SVM, PCA par erreur de reconstruction) sont comparées et atteignent des AUPRC comprises entre 0.713 et 0.773 sur notre protocole. En classification supervisée binaire, trois modèles complémentaires (régression logistique, Random Forest, XGBoost) sont évalués ; XGBoost obtient la meilleure AUPRC (0.950). Au niveau familles d'attaque, Random Forest atteint une balanced accuracy de 0.825, et au niveau sous-types, Random Forest atteint 0.806 de balanced accuracy.

2 Introduction

Ce projet vise à la conception et l'évaluation d'une chaîne de données pour l'analyse cybersécurité, incluant : trois méthodes non supervisées et trois classificateurs supervisés, évalués avec des métriques adaptées (matrices de confusion, précision, rappel, AUPRC, balanced accuracy, MCC).

3 Présentation et caractérisation du dataset

3.1 Description générale

Le dataset étudié est CIC IoT-DIAD 2024 (Packet-Based Features). Les données sont fournies sous forme de fichiers CSV regroupés par familles d'attaques et sous-types (sous-dossiers). Chaque ligne représente un enregistrement de trafic décrit par un ensemble de caractéristiques (features) extraites du trafic réseau. Les colonnes comprennent un mélange de variables numériques (ex. tailles, entropies, compteurs, statistiques temporelles) et de champs catégoriels (ex. identifiants de protocole, chaînes applicatives).

3.2 Organisation en familles et sous-types

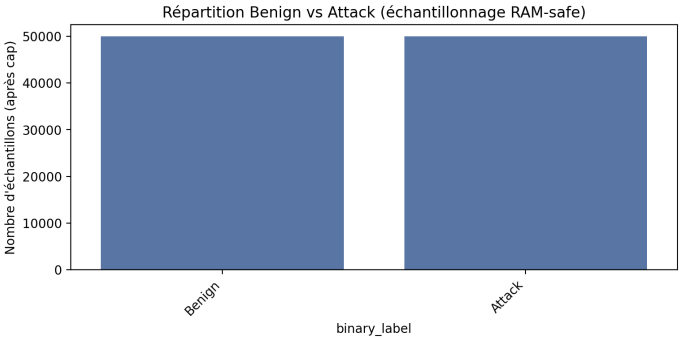
Table 1: Nombre de fichiers CSV par famille (répertoire Packet-Based Features).

family	n_csv
DDoS	101
DoS	38
Mirai	22
Web-Based	6
Recon	5
Benign	4
Spoofing	3
BruteForce	1

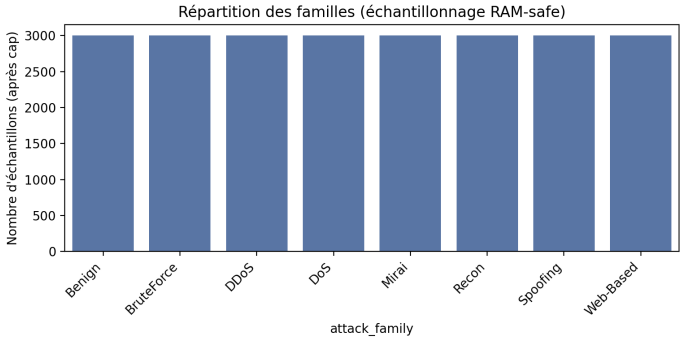
Au niveau des sous-types, les dossiers présents incluent notamment :
- DDoS : DDoS-UDP_Flood, DDoS-TCP_Flood, DDoS-HTTP_Flood, DDoS-SlowLoris, DDoS-ACK_Fragmentation, etc.
- DoS : DoS-UDP_Flood, DoS-TCP_Flood, DoS-SYN_Flood, DoS-HTTP_Flood.
- Recon : Recon-PortScan, Recon-PingSweep, Recon-OSScan, VulnerabilityScan, etc.
- Spoofing : DNS_Spoofing, MITM-ArpSpoofing.
- Web-Based : SqlInjection, XSS, CommandInjection, Uploading_Attack, etc.
- BruteForce : DictionaryBruteForce.
- Mirai : Mirai-greip_flood.

3.3 Déséquilibre des classes et échantillonnage

En pratique, les datasets de trafic réseau sont souvent déséquilibrés : le trafic bénin est majoritaire et les attaques sont rares (ce qui rend l'accuracy peu informative). Dans notre implémentation, un chargement RAM-safe est utilisé : lecture par chunks et plafonds (caps) pour limiter la taille des ensembles de travail. Cette stratégie modifie la distribution par rapport au dataset complet, mais conserve les propriétés importantes pour l'évaluation (variabilité inter-fichiers, diversité des familles et sous-types).



(a) Binaire : Benign vs Attack.



(b) Familles (plafond à 3000 par famille).

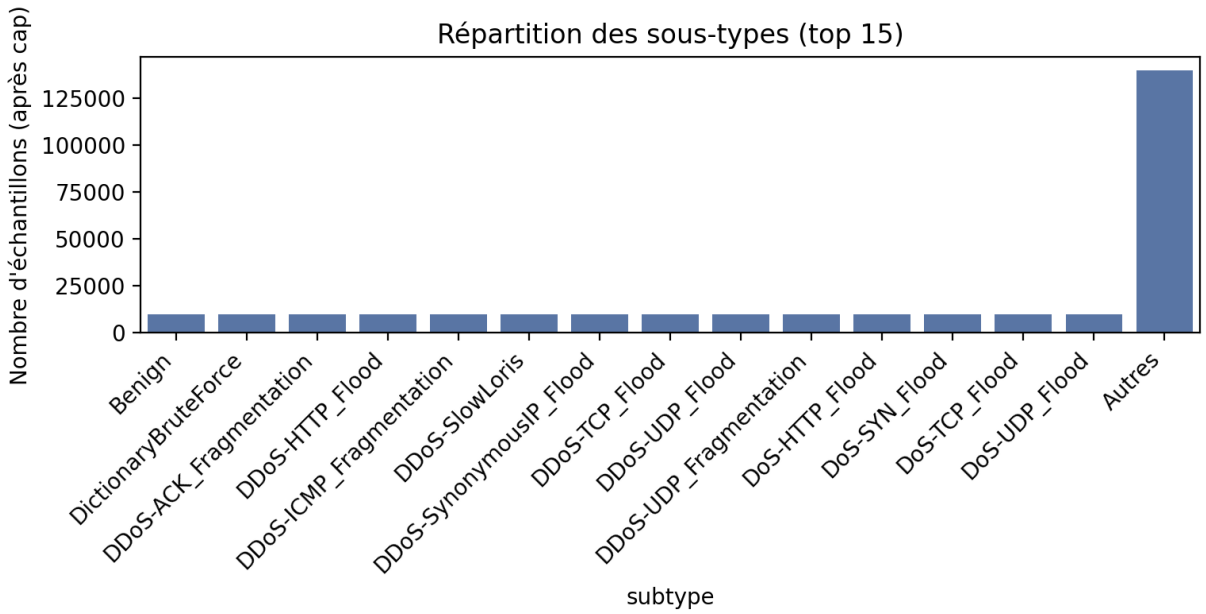


Figure 1: Répartition des classes après échantillonnage RAM-safe.

4 Chaîne complète de traitement des données

4.1 Prétraitement et prévention des fuites

Les fichiers CSV contiennent des colonnes potentiellement identifiantes (adresses IP, ports, adresses MAC, timestamps, identifiants de flux). Ces variables peuvent introduire un biais fort (apprentissage de signatures spécifiques à un capture) et amplifier les risques de fuite d'information entre apprentissage et test. Nous supprimons donc ces colonnes via :
- Une liste de colonnes candidates (src_ip, dst_ip, src_port, dst_port, timestamp, etc.),
- Des motifs (patterns) sur les noms de colonnes.

4.2 Chargement RAM-safe et harmonisation des schémas

Les CSV présentent des schémas hétérogènes. La chaîne de chargement procède comme suit :

1. Scan des en-têtes : constitution de l'union des colonnes observées.
2. Sélection des caractéristiques : suppression des colonnes identifiantes et de la colonne Label.
3. Lecture par chunks (20 000 lignes) et limitation à 1 chunk par fichier (pour contrôler la RAM).
4. Conversion numérique : conversion des colonnes retenues en numérique (errors=coerce), remplacement des infinis par NaN.
5. Gestion des valeurs manquantes : remplissage à 0 (nan_to_num) avant apprentissage.
6. Plafonds d'échantillons :
- binaire : 50 000 Benign et 50 000 Attack,
- familles : 3000 par famille,
- sous-types : 10 000 par sous-type.

4.3 Séparation train/test par fichier

Pour limiter la fuite d'information, le split train/test est réalisé par fichier (group). un fichier CSV est entièrement assigné au train ou au test. Concrètement, on sélectionne environ 20 % des fichiers par classe pour constituer le test. En cas de groupe mixte (un même fichier contenant plusieurs labels) ou de dégénérescence (train/test vide), un split stratifié standard est utilisé en repli.

4.4 Normalisation et mise en forme

Les modèles sont entraînés sur les caractéristiques normalisées via StandardScaler (moyenne nulle, variance unité) ajusté sur l'ensemble d'entraînement puis appliqué au test. Ce choix :
- Stabilise la régression logistique et l'OCSVM,
- Met les caractéristiques sur une échelle comparable.

5 Méthodes de classification et de détection d'anomalies

5.1 Détection d'anomalies (non supervisée)

La détection d'anomalies est évaluée dans un cadre classique : le modèle est appris uniquement sur le trafic Benign, puis appliqué au test contenant Benign et Attack. Le détecteur produit soit une étiquette (anomalie vs normal) soit un score (anomalie croissante), utilisé pour l'AUPRC.

Isolation Forest Isolation Forest isole des observations via des partitions aléatoires ; les points rares nécessitent moins de splits. Nous réglons un taux de contamination (clippé dans $[0.01, 0.30]$) et entraînons uniquement sur Benign.

One-Class SVM (RBF) One-Class SVM apprend une frontière séparant la masse de données (Benign) de l'extérieur. Le noyau RBF capture des frontières non linéaires ; le paramètre ν contrôle la fraction d'outliers.

PCA par erreur de reconstruction La PCA (variance expliquée 95 %) modélise un sous-espace principal du trafic Benign. Le score d'anomalie est l'erreur de reconstruction (MSE). Un seuil est choisi au quantile 95 % des erreurs sur Benign (train).

5.2 Classification d'anomalies (supervisée)

Pour la classification, le modèle est entraîné sur un ensemble annoté et prédit directement une classe. Nous comparons trois modèles supervisés complémentaires afin de couvrir des hypothèses inductives différentes.

Régression logistique Modèle linéaire entraîné par optimisation convexe. Il sert de référence robuste et rapide, et fournit un score probabiliste exploitable pour tracer des courbes précision-rappel.

Random Forest Ensemble d'arbres entraînés sur des sous-échantillons et sous-ensembles de variables. Cette approche réduit la variance, capture des interactions non linéaires et se comporte bien en tabulaire.

XGBoost Boosting d'arbres avec régularisation et optimisation efficace. Il offre souvent les meilleures performances sur données tabulaires, au prix d'un réglage plus fin.

5.2.1 Tâches et niveaux de granularité

Nous étudions trois niveaux de tracking :
- Binaire : supervisé (classification) et non supervisé (détection d'anomalies).
- Familles : uniquement supervisé (classification multi-classes).
- Sous-types : uniquement supervisé (classification multi-classes, Benign conservé).

6 Benchmark expérimental

6.1 Métriques et définitions

On note TP, FP, TN, FN les composantes de la matrice de confusion binaire. Les métriques rapportées sont :
- Précision : Prec $= \frac{TP}{TP+FP}$ (avec zero_division=0)
- Rappel : Rec $= \frac{TP}{TP+FN}$
- Balanced accuracy : moyenne des rappels par classe, BAcc $= \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$
- MCC (Matthews Correlation Coefficient) : corrélation entre prédictions et vérité terrain, robuste au déséquilibre
- AUPRC : aire sous la courbe précision-rappel, adaptée aux scénarios où la classe positive est rare

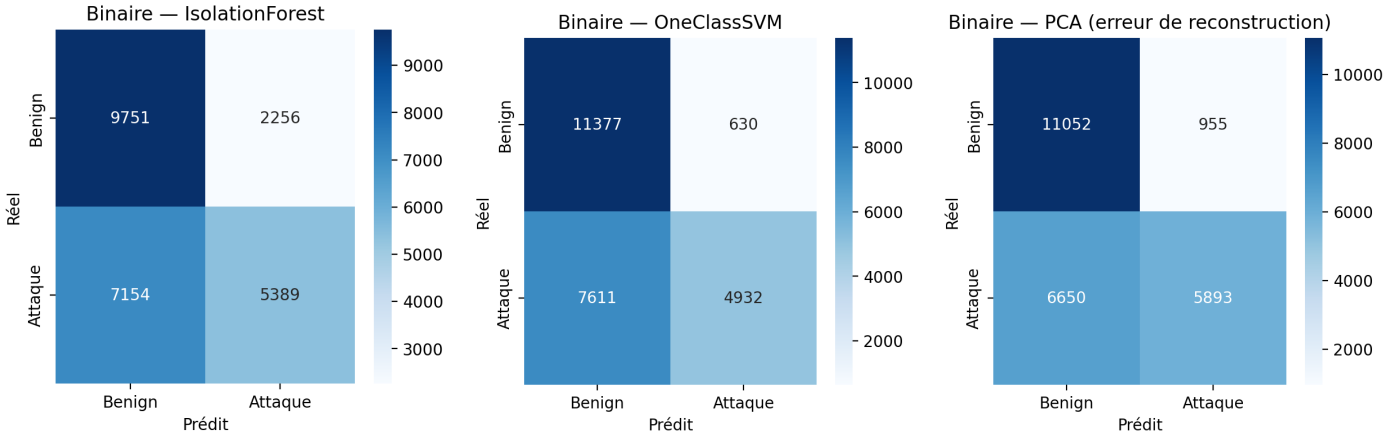
En multi-classes, la précision et le rappel sont rapportés en macro-moyenne sur les classes présentes en test ; la balanced accuracy est la moyenne des rappels par classe, et l'AUPRC est une macro-moyenne en one-vs-rest quand des probabilités sont disponibles.

6.2 Détection d'anomalies : résultats non supervisés (binaire)

Les résultats suivants comparent trois détecteurs non supervisés entraînés uniquement sur le trafic Benign et évalués sur un test Benign vs Attack.

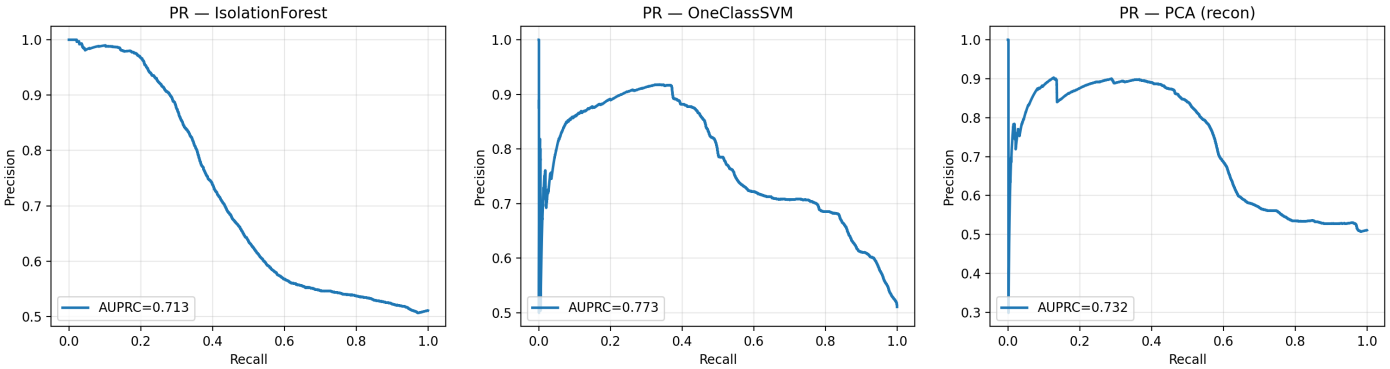
Table 2: Benchmark binaire non supervisé (détection d'anomalies).

scenario	title	precision	recall	auprc	balanced acc	mcc
Binaire (non supervisé)	IsolationForest	0.7049	0.4296	0.7127	0.6209	0.2610
Binaire (non supervisé)	OneClassSVM	0.8867	0.3932	0.7727	0.6704	0.4069
Binaire (non supervisé)	PCA-recon	0.8605	0.4698	0.7323	0.6951	0.4350



(a) Isolation Forest (b) One-Class SVM (c) PCA (recon.)

Figure 2: Matrices de confusion – binaire non supervisé.



(a) Isolation Forest (b) One-Class SVM (c) PCA (recon.)

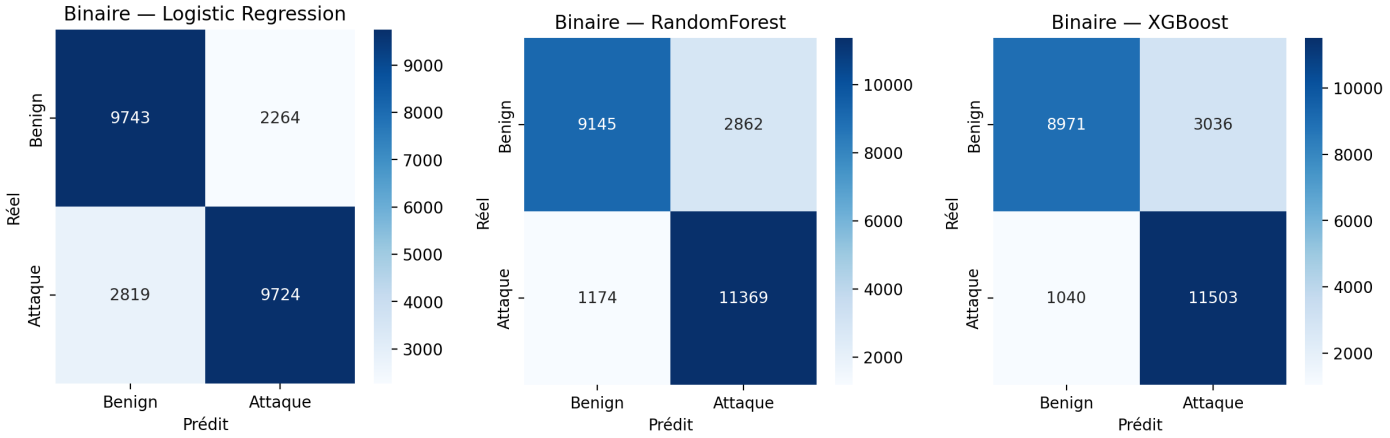
Figure 3: Courbes précision-rappel (PR) – binaire non supervisé.

6.3 Résumé des résultats supervisés (binaire)

Pour le cas binaire, les modèles supervisés atteignent des performances nettement supérieures aux détecteurs non supervisés.

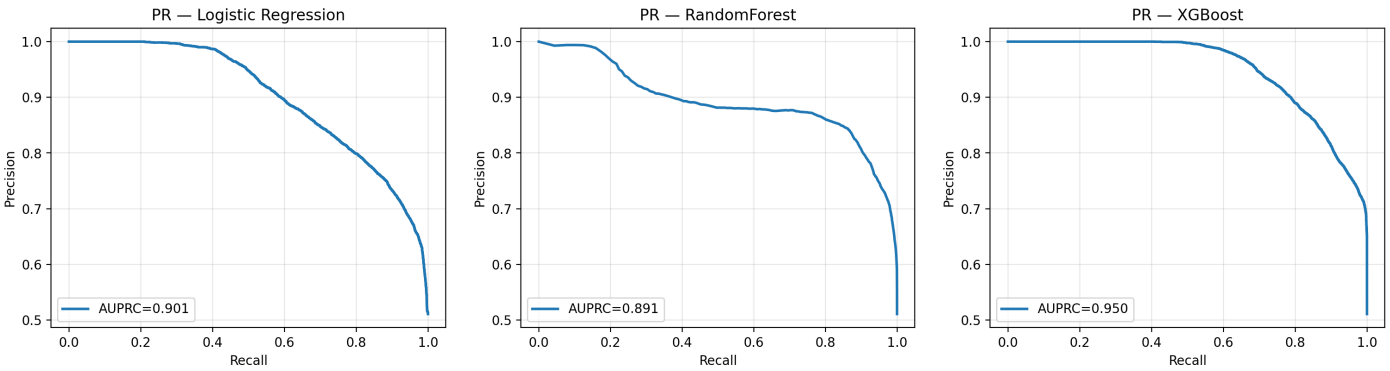
Table 3: Benchmark binaire supervisé (détection d'anomalies).

scenario	Modèle	precision	recall	auprc	balanced acc	mcc
Binaire (supervisé)	LogisticRegression	0.8111	0.7753	0.9014	0.7933	0.5867
Binaire (supervisé)	RandomForest	0.7989	0.9064	0.8914	0.8340	0.6765
Binaire (supervisé)	XGBoost	0.7912	0.9171	0.9495	0.8321	0.6757



(a) LogReg (b) RF (c) XGB

Figure 4: Matrices de confusion – binaire supervisé.



(a) LogReg (b) RF (c) XGB

Figure 5: Courbes précision-rappel (PR) – binaire supervisé.

6.4 Résultats binaire : non supervisé vs supervisé

On observe un gain net des modèles supervisés (AUPRC, MCC), ce qui est attendu lorsque des labels sont disponibles.

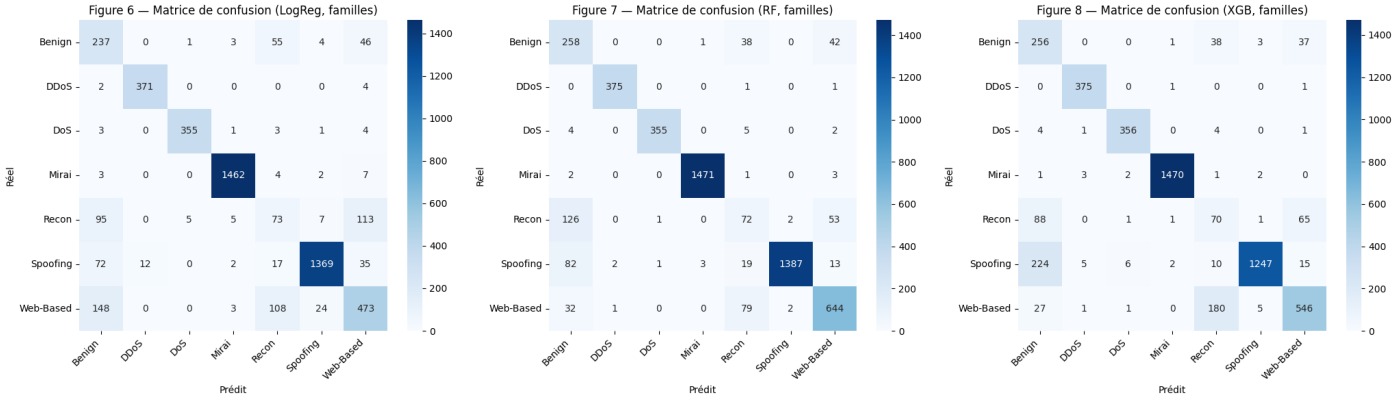
Table 4: Benchmark binaire (non supervisé + supervisé).

scenario	title	precision	recall	auprc	balanced acc	mcc
Binaire (non supervisé)	IsolationForest	0.7049	0.4296	0.7127	0.6209	0.2610
Binaire (non supervisé)	OneClassSVM	0.8867	0.3932	0.7727	0.6704	0.4069
Binaire (non supervisé)	PCA-recon	0.8605	0.4698	0.7323	0.6951	0.4350
Binaire (supervisé)	LogisticRegression	0.8111	0.7753	0.9014	0.7933	0.5867
Binaire (supervisé)	RandomForest	0.7989	0.9064	0.8914	0.8340	0.6765
Binaire (supervisé)	XGBoost	0.7912	0.9171	0.9495	0.8321	0.6757

6.5 Résultats multi-classes : familles

Table 5: Benchmark multi-classes (familles).

scenario	title	precision	recall	auprc	balanced acc	mcc
Familles	LogisticRegression	0.7590	0.7614	0.7802	0.7721	0.7897
Familles	RandomForest	0.8110	0.8056	0.8411	0.8250	0.8428
Familles	XGBoost	0.7733	0.7728	0.8140	0.7971	0.7885



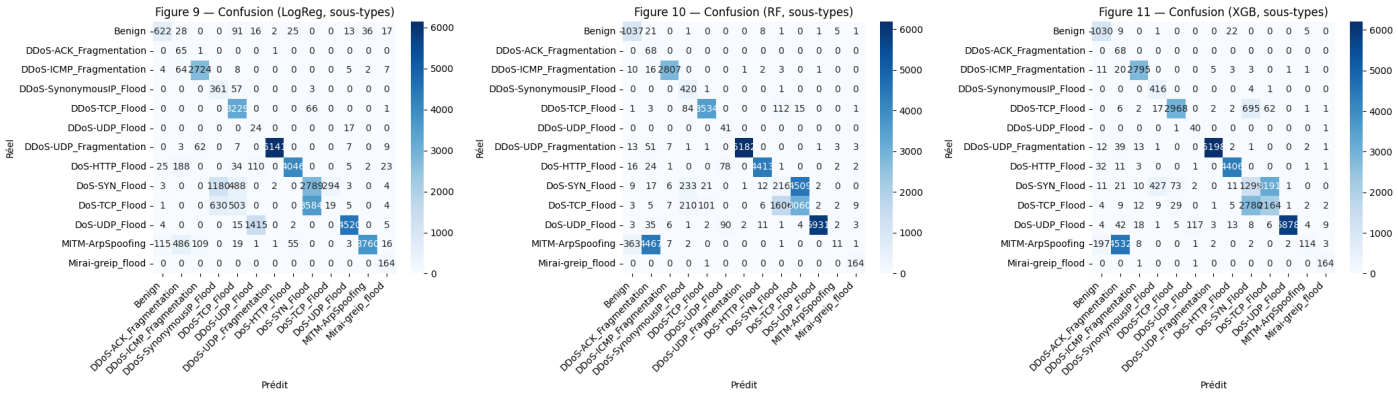
(a) LogReg (b) RF (c) XGB

Figure 6: Matrices de confusion – familles (classes présentes en test).

6.6 Résultats multi-classes : sous-types

Table 6: Benchmark multi-classes (sous-types).

scenario	title	precision	recall	auprc	balanced acc	mcc
Sous-types	LogisticRegression	0.6027	0.7304	0.7329	0.7817	0.6652
Sous-types	RandomForest	0.6272	0.7825	0.8597	0.8057	0.6541
Sous-types	XGBoost	0.6849	0.7715	0.8604	0.7939	0.6408



(a) LogReg (b) RF (c) XGB

Figure 7: Matrices de confusion – sous-types (classes présentes en test).

6.7 Meilleurs modèles par tâche

Table 7: Meilleur modèle par tâche (selon la métrique indiquée).

Tâche	Meilleur	Métrique	Valeur
Binaire	XGBoost	AUPRC	0.9495
Familles	RandomForest	Balanced Acc	0.8250
Sous-types	RandomForest	Balanced Acc	0.8057

7 Analyse et discussion des résultats

7.1 Détection vs classification : rôle opérationnel

Les résultats confirment la complémentarité des approches : Les méthodes non supervisées sont adaptées au signal faible et à l'absence de labels, mais leurs performances restent limitées en présence d'attaques proches du trafic normal (Figures 3). À l'inverse, les méthodes supervisées exploitent des patterns discriminants et atteignent des AUPRC élevées en binaire (Table 4), au prix d'un besoin en données annotées.

7.2 Granularité du tracking (familles et sous-types)

Le passage du binaire à la multi-classe illustre un compromis : La classification par familles est un niveau pertinent pour le tracking SOC : elle réduit le nombre de classes tout en fournissant une sémantique opérationnelle (DoS/DDoS vs Recon vs Web). La classification par sous-types offre une granularité plus fine, mais amplifie le déséquilibre et la confusion entre attaques proches (Table 6, Figure 7).

8 Perspectives et limites

8.1 Limites

Plusieurs limites doivent être prises en compte avant interprétation opérationnelle. L'échantillonnage RAM-safe (un chunk par fichier et plafonds) peut sous-échantillonner certains régimes de trafic. De plus, la conversion numérique avec coercition transforme les valeurs non numériques en 0, ce qui peut réduire l'information utile. Enfin, la vérité terrain est déduite de l'arborescence des fichiers (famille/sous-type), sans vérification d'une colonne de label au niveau ligne.

8.2 Perspectives

- Enrichir le prétraitement (encodage des catégorielles, sélection de variables, détection de constantes).
- Renforcer le protocole (validation croisée par fichiers, split temporel si disponible).
- Étudier des mécanismes d'explicabilité (importance de variables, SHAP) pour un usage SOC.

9 Conclusion

Ce projet a mis en place une chaîne batch reproductible pour analyser des données réseau IoT, en combinant détection d'anomalies et classification supervisée à plusieurs granularités (binaire, familles, sous-types). Les résultats montrent : des performances limitées en non supervisé, des performances élevées en supervisé, avec un avantage pour XGBoost en binaire et Random Forest en multi-classes et une sensibilité aux variations de distribution et au bruit des caractéristiques.

Références

[1] Canadian Institute for Cybersecurity (CIC), CIC IoT-DIAD 2024 Dataset. University of New Brunswick. (Voir la description du cours et les liens fournis dans le sujet.)

[2] F. T. Liu, K. M. Ting, and Z.-H. Zhou, Isolation Forest. 2008.

[3] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, Estimating the Support of a High-Dimensional Distribution. 2001.

[4] L. Breiman, Random Forests. 2001.

[5] T. Chen and C. Guestrin, XGBoost: A Scalable Tree Boosting System. 2016.

[6] J. Davis and M. Goadrich, The Relationship Between Precision-Recall and ROC Curves. 2006.

[7] B. W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme. 1975.