Projet de Machine Learning

Détection d'attaques DDoS dans un environnement IoT

ML - 4IIR

Décembre 2025

Contexte & Problématique

Les dispositifs de l'Internet des Objets (IoT) sont de plus en plus ciblés par des cyberattaques, notamment les attaques de type **DDoS** (Distributed Denial of Service), qui visent à saturer les ressources réseau ou système. Étant donné les contraintes matérielles (CPU, mémoire, batterie) des appareils IoT, il est crucial de détecter ces attaques de manière **efficace** et **en temps réel**.

Ce projet propose d'explorer trois algorithmes classiques de **classification supervisée** — **KNN**, **Arbre de décision** et **Machine à Vecteurs de Support (SVM)** — pour détecter les attaques DDoS à partir de données réelles issues d'un environnement IoT.

Objectifs Pédagogiques

— Appliquer les étapes classiques de prétraitement sur un jeu de données de cybersécurité.

— Implémenter et comparer trois algorithmes de classification supervisée : KNN, Arbre de décision et SVM.

— Analyser l'impact du choix de l'algorithme sur les performances (précision, rappel, F1-score, temps de prédiction, robustesse au déséquilibre).

— Interpréter les résultats dans le contexte spécifique de la détection d'attaques dans l'IoT.

Jeu de données suggéré

— **TON_IoT Dataset** – University of New South Wales
Contient des traces de trafic IoT avec des étiquettes claires pour les attaques, y compris DDoS.
`https://research.unsw.edu.au/projects/toniot-datasets`

Ce dataset a été collecté sur un **environnement IoT/IIoT réel conçu pour la recherche**, comprenant des dispositifs physiques, des machines Windows et Linux, ainsi que du trafic réseau réel capturé dans un testbed expérimental.

Les données sont **labellisées** et contiennent notamment :

— des mesures de capteurs IoT et IIoT,

— des statistiques système,

— des données de trafic réseau,

— des comportements normaux et anormaux.

Étapes du Projet

1. Exploration des données

— Identifier les colonnes représentant les caractéristiques du trafic (features) et la colonne cible (label : normal / DDoS).

— Analyser la distribution des classes et visualiser les features pertinentes.

2. Prétraitement

— Supprimer ou imputer les valeurs manquantes.

— Encoder les variables catégorielles (ex. : protocole).

— Standardiser les données (z-score normalization).

— Gérer le déséquilibre de classe (ex. : pondération ou sous-échantillonnage).

3. Modélisation

— Implémenter les trois modèles suivants :

  — **K plus proches voisins (KNN)** – tester plusieurs valeurs de k

  — **Arbre de décision** – interprétable, pas besoin de standardisation

  — **SVM** – tester différents noyaux (linéaire, RBF)

— Utiliser une séparation stratiée en `train/validation/test` (ex. : 70%/15%/15%).

— (Optionnel) Faire un *grid search* ou *cross-validation* pour chaque modèle.

4. Évaluation

— Comparer les modèles sur les métriques suivantes : précision, rappel, F1-score, temps d'entraînement et de prédiction.

— Analyser les matrices de confusion.

— Discuter : quel modèle est le plus adapté à un usage IoT ?

Livrables, avant le 31 Décembre 2025

Dans un fichier : Nom1_Prenon1_Nom2_Prenon2_4IIRGnumeroDuGroupe.zip

- Code source Python(Nom1_Prenon1_Nom2_Prenon2_MLProject.ipynb)
- Rapport de 6 à 8 pages (PDF)
- préparer une présentation (10 Slides)

Critères d'évaluation

- Bonne application des algorithmes
- Qualité de l'analyse et de l'interprétation
- Comparaison claire entre les algorithmes
- Clarté du rapport et des figures