

# PROJET D'INFORMATIQUE

## DÉTECTION D'INTRUSION SUR LE RÉSEAU: APPRENTISSAGE AUTOMATIQUE Algorithme utilisé : Arbre de décision

1

Présenté par

**ELFRID TATIANA ATEMKENG KAMDOM**

Supervision: Yacine YADDADEN, PhD

# PLAN

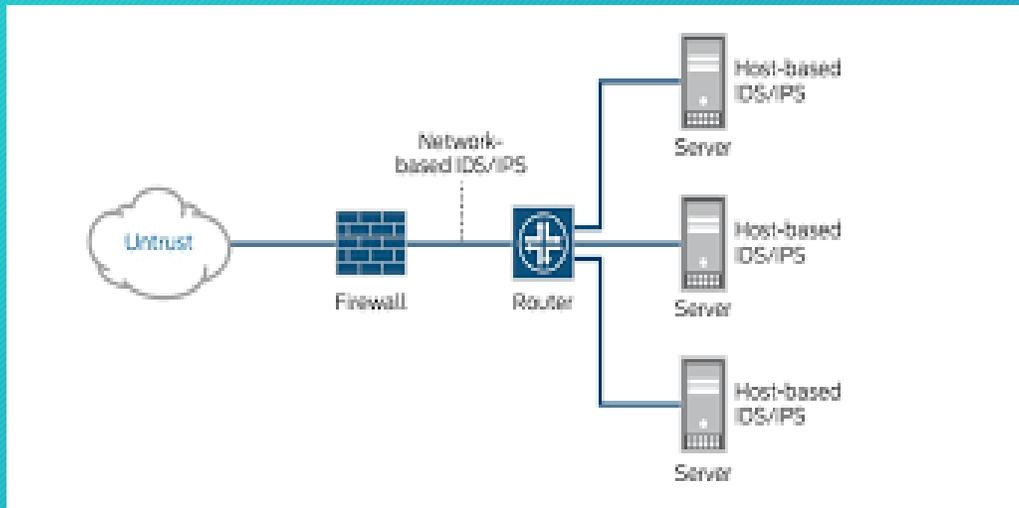
2

- 01 CONTEXTE ET PROBLEMATIQUE
- 02 OBJECTIF
- 03 METHODOLOGIE : ML.NET ET ARBRE DE DECISION
- 04 RESULTATS

# I. CONTEXTE ET PROBLEMATIQUE

## • CONTEXTE

La tâche de détection d'intrusion réseau consiste à analyser les données de trafic réseau pour identifier une activité suspecte pouvant indiquer une attaque sur le réseau.



## • PROBLEMATIQUE

- Fuites de données au sein des entreprises.
- Failles et problèmes de sécurité. Les méthodes de sécurité classiques ne suffisent plus.
- Les données incomplètes, inexactes ou incorrectement étiquetées, peuvent entraîner des faux positifs ou des faux négatifs, ce qui peut compromettre la capacité de l'IDS à détecter les menaces.



## II. OBJECTIFS

4

Faire une analyse des données d'un système de détection d'intrusion (IDS) afin de déterminer si un évènement futur est susceptible d'être une attaque ou non, en utilisant un algorithme d'apprentissage automatique en fonction des caractéristiques, des évènements et des modèles identifiés lors de l'entraînement.



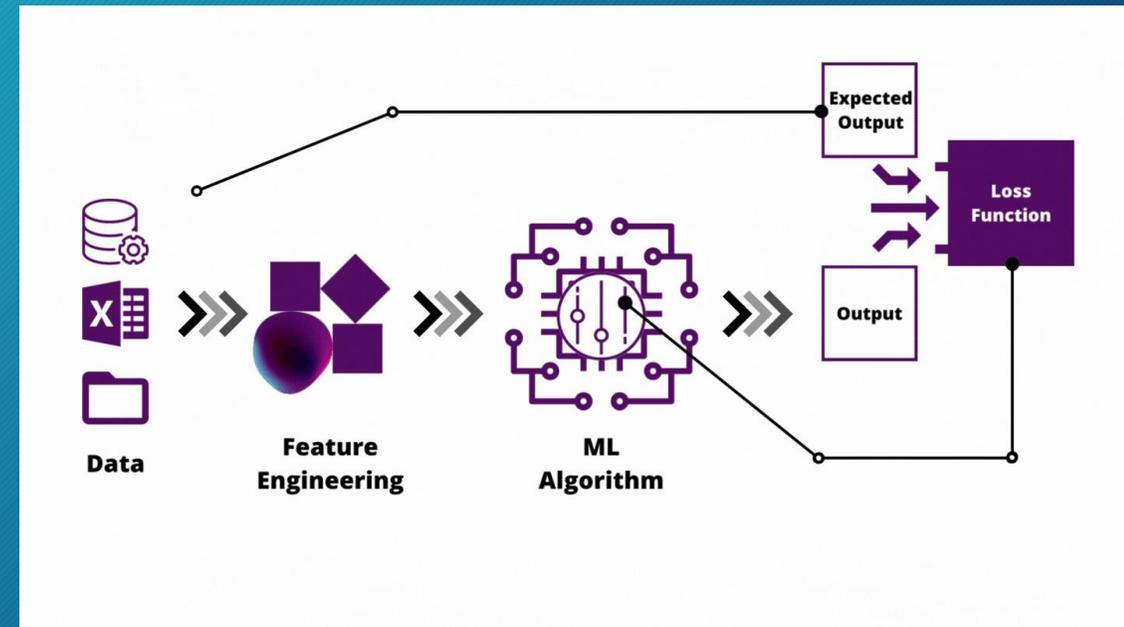
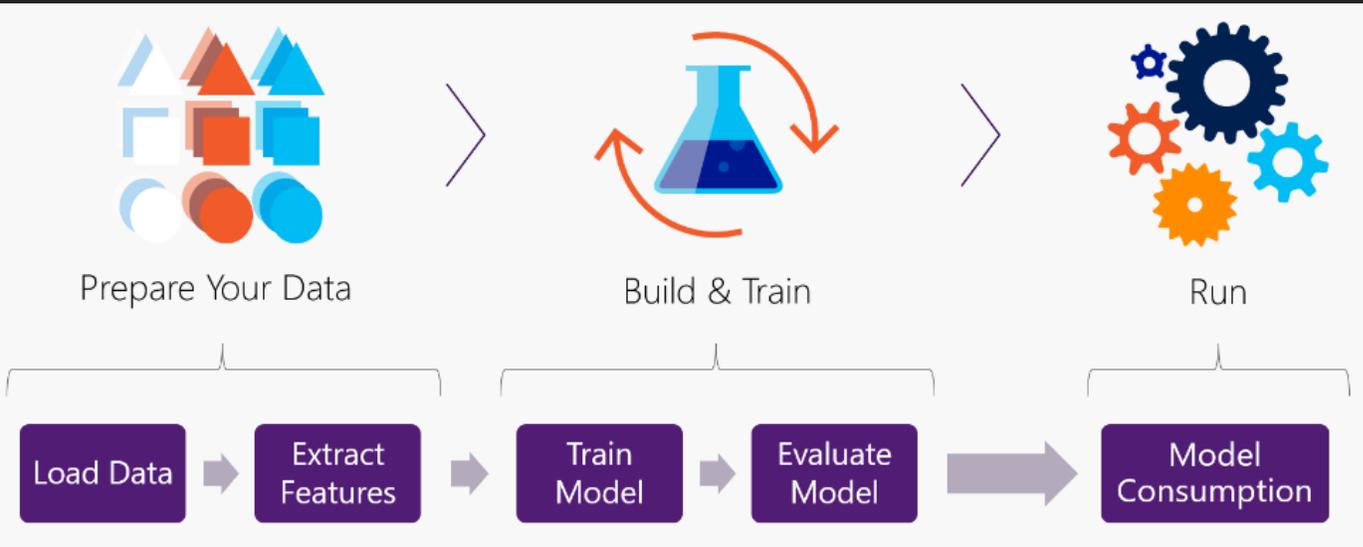
### JEUX DE DONNEES



Le jeu de données contient du trafic à la fois bénin et malveillant, dans le but de détecter et de classer différents types d'attaques.

# III. METHODOLOGIE : ML.NET

5

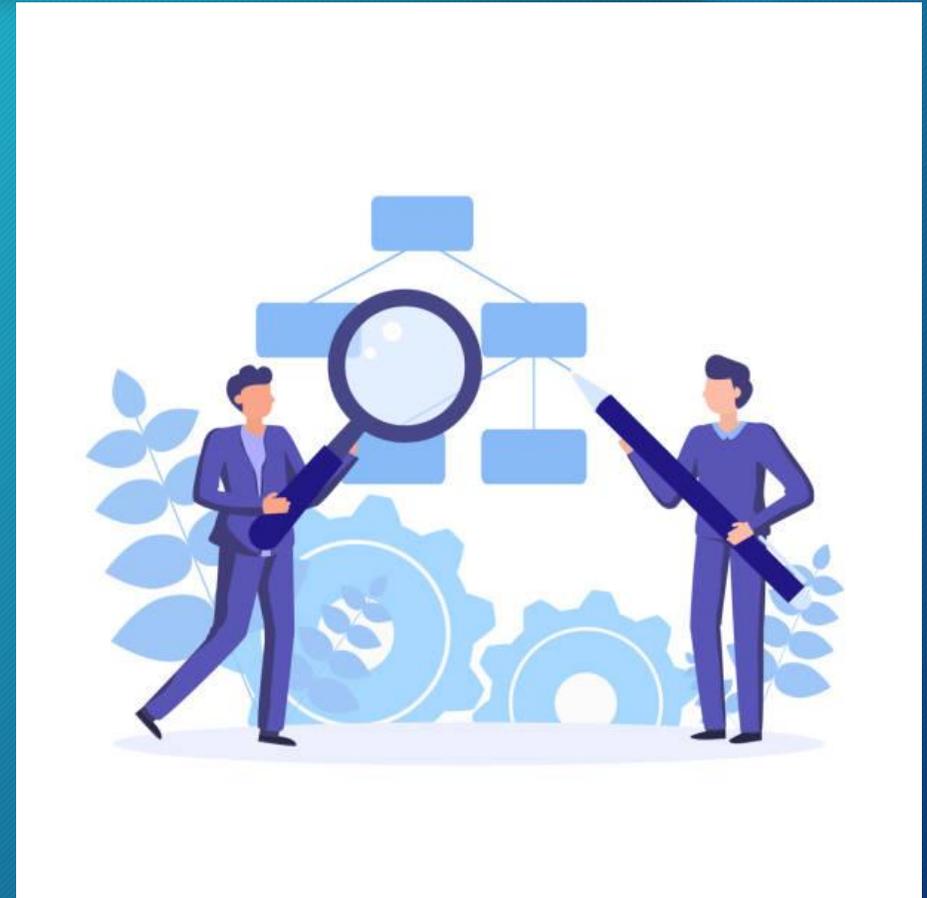


# III. METHODOLOGIE : ARBRE DE DECISION

6

- Algorithme utilisé pour représenter visuellement et explicitement les décisions et la prise de décision pour des problèmes de classification et problèmes de régression.
- Utilisée dans les domaines de l'intelligence artificielle, de la science des données et de l'analyse des données,
- Elle ressemble à une structure descendante composée de nœuds et une fois qu'elle est construite, on distingue: ID3, C4.5, CART, CHAID, Random Forest, FastTree...

Nous allons dans notre cas utiliser le Fast tree pour l'implémentation de notre implémentation.



# IV. RESULTATS

```
C:\Users\DELL\Desktop\mes codes sources\TP01\TP01\bin\Debug\net7.0> TP01.exe -e test_file.csv -t train_file.csv
```

## 1. Classification

```
Model quality metrics evaluation
-----
Accuracy: 99,07 %
Auc: 99,91 %
F1Score: 99,14 %
Positive Precision: 0,9942260715078837
```

## 2. matrice de confusion

```
Positive Precision: 0,9955237242614146
===== End of model evaluation =====
TEST POSITIVE RATIO: 0,5339 (13449,0/(13449,0+11743,0))
Confusion table
||=====
PREDICTED || positive | negative | Recall
TRUTH     ||=====
positive  || 13 344 | 105 | 0,9922
negative  || 60 | 11 683 | 0,9949
||=====
Precision || 0,9955 | 0,9911 |
```

**MERCI POUR VOTRE  
AIMABLE ATTENTION !!!**