SR2I208 - Projet

Détection et classification de malwares en utilisant différents modèles d'intelligences artificielle

Soutenance - 29/06/2023

Nathanaël SIMON - Hamza Zarfaoui - Eddy Raingeaud

Sommaire

1. Introduction
2. Datasets
3. Modèles utilisés
4. Evaluation des modèles
5. Résultats obtenus
6. Détecteur

Introduction

But :

- Analyser les performances de différents modèles de feature selection et de machine learning
- Adaptation à la cybersécurité
  - Détection d'intrusion dans des logs

Problématiques :

- Classifier des logs en fonction de leur dangerosité
- Détecter des anomalies dans le fonctionnement normal d’un serveur

Préparation du dataset

Dataset détection

	0	1	2	3	4	5	6	7	8	9	...	92	93	94	95	96	97	98	99	100	101
0	hash	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	...	t_91	t_92	t_93	t_94	t_95	t_96	t_97	t_98	t_99	malware
1	071e8c3f8922e186e57548cd4c703a5d	112	274	158	215	274	158	215	298	76	...	71	297	135	171	215	35	208	56	71	1
2	33f8e6d08a6aae939f25a8e0d63dd523	82	208	187	208	172	117	172	117	172	...	81	240	117	71	297	135	171	215	35	1
3	b68abd064e975e1c6d5f25e748663076	16	110	240	117	240	117	240	117	240	...	65	112	123	65	112	123	65	113	112	1
4	72049be7bd30ea61297ea624ae198067	82	208	187	208	172	117	172	117	172	...	208	302	208	302	187	208	302	228	302	1

Préparation du dataset

Dataset Classification

Données originales

	0
0	ldrloaddll ldrgetprocedureaddress ldrloaddll l...
1	getsystemtimeasfiletime ntallocatevirtualmemor...
2	ldrgetdllhandle ldrgetprocedureaddress getsyst...
3	ldrloaddll ldrgetprocedureaddress ldrloaddll l...
4	ldrloaddll ldrgetprocedureaddress ldrgetproced...

	0
0	Trojan
1	Trojan
2	Backdoor
3	Backdoor
4	Trojan

Données vectorisées

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 3, 0, ..., 0, 0, 0],
       [0, 1, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```
{'ldrloaddll': 132,
 'ldrgetprocedureaddress': 131,
 'regopenkeyexa': 221,
 'ntopenkey': 169,
 'ntqueryvaluekey': 181,
 'ntclose': 150,
 'ntqueryattributesfile': 176,
 'loadstringa': 136,
 'ntallocatevirtualmemory': 149,
 'ldrgetdllhandle': 130,
 'ldrunloaddll': 133,
 'findfirstfileexw': 52,
 'copyfilea': 13,
 'regcreatekeyexa': 209,
 'regsetvalueexa': 227,
 'regclosekey': 208,
 'createprocessinternalw': 21,
 'ntfreevirtualmemory': 164,
 'ntterminateprocess': 190,
 'getsystemtimeasfiletime': 92,
 'setunhandledexceptionfilter': 252,
 'ntcreatemutant': 153,
 'getsysteminfo': 90,
 'getsystemdirectoryw': 89,
 '__exception__': 1,
...
 'rtlcreateuserthread': 234,
 'setinformationjobobject': 249,
 'cryptprotectmemory': 37,
 'cryptunprotectmemory': 39,
 'findfirstfileexa': 51}
```

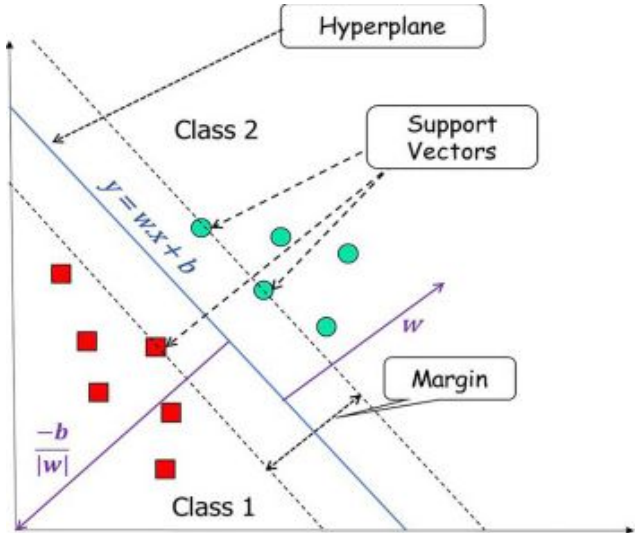
Préparation du dataset

- Pour la détection
  - Train / test = 80/20
  - 2 labels
    - 1 = sain
    - 0 = malware

- Pour la classification
  - Train / test = 80/20
  - 8 labels
    - Trojan, Adware, Dropper, Downloader, Worms, Backdoor, Spyware et Virus.

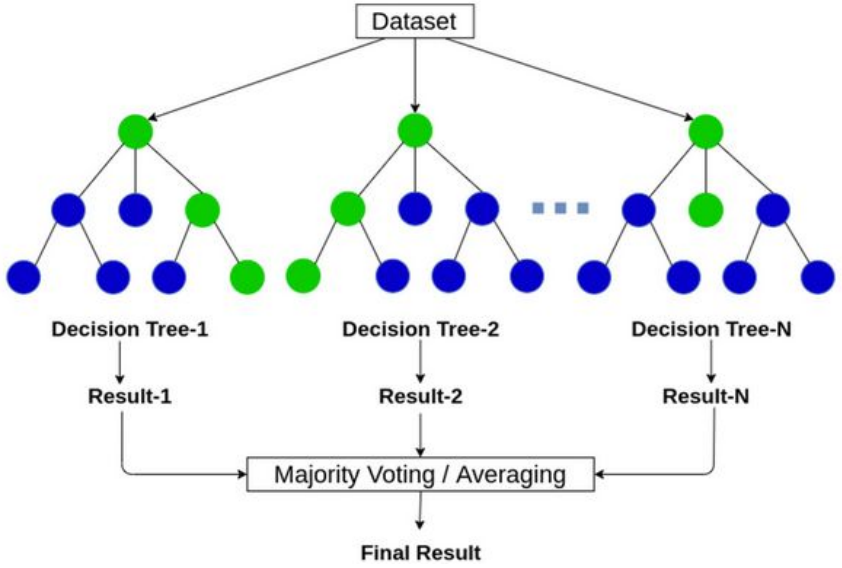
Machine Learning

- Support Vector Machine (SVM)
  - Trouver un hyperplan qui sépare les différentes classes
  - Utilisation de kernels pour s'adapter à des modèles non linéaires
  - Complexité importante, et paramétrisation précise
  - Détection d'anomalie : OneClassSVM



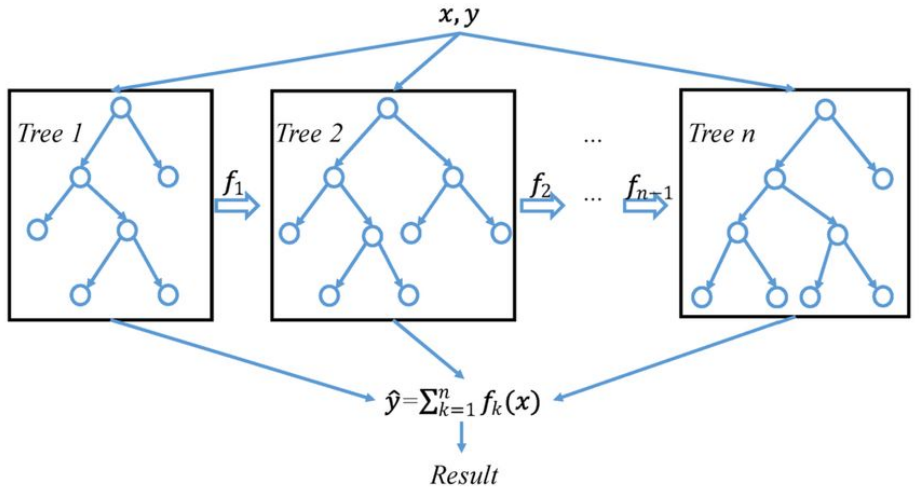
Machine Learning

- Random Forest
  - Construit un grand nombre d'arbre de prédiction indépendant, et combine leur prédiction
  - Moins sujet au sur-ajustement
  - Détection d'anomalie : IsolationForest



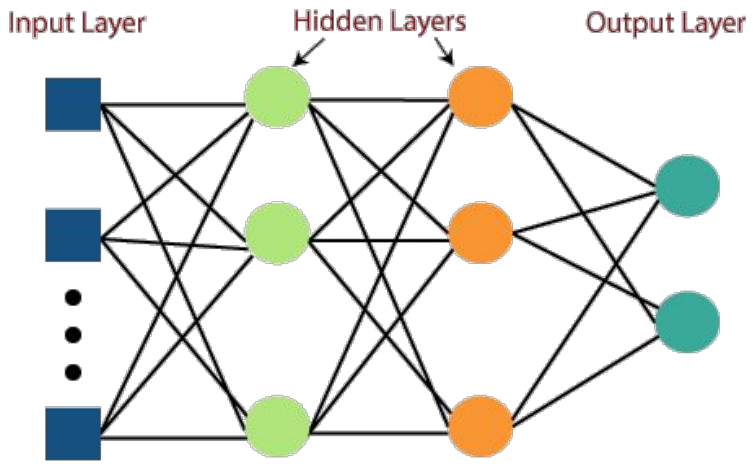
Machine Learning

- Extreme Gradient Boosting (XGBoost)
  - Construire un modèle prédictif en combinant plusieurs modèles de prédiction faible
  - Optimisation d'une fonction de perte en effectuant une descente de gradient
  - Modèles précis pouvant traiter un grand nombre de données
  - Pas d'adaptation pour la détection d'anomalie



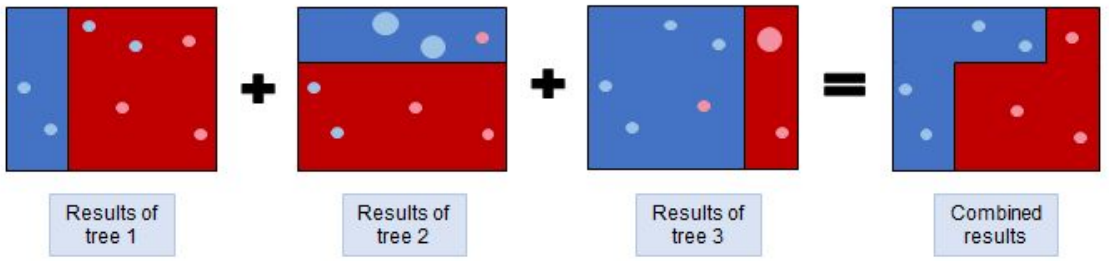
Machine Learning

- MLPC
  - Réseau de neuronnes à plusieurs couches pour la classificatoin
  - Rétropropagation du gradient, où les poids sont ajustés pour minismiser la perte
  - Réglages minutieux pour obtenir de bons résultats sans overfitting



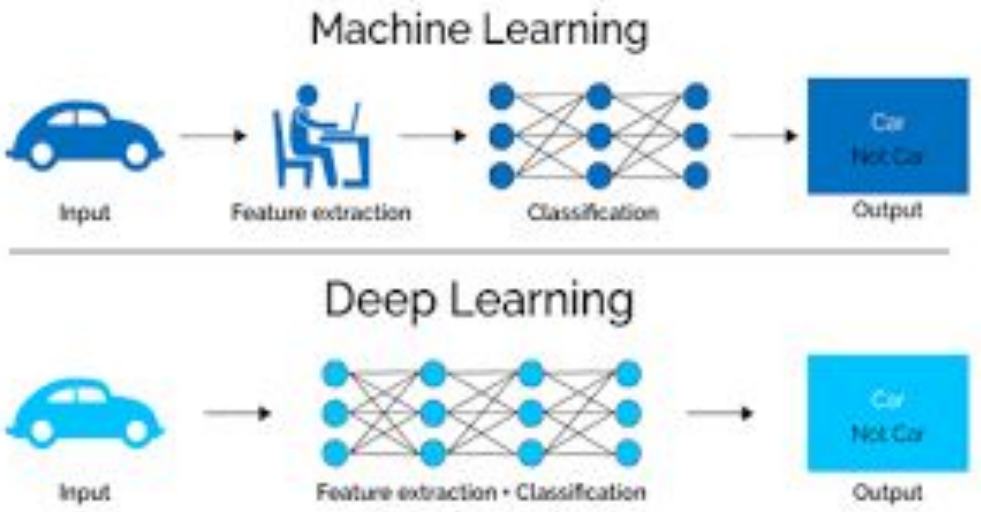
Machine Learning

- AdaBoost
  - Combien des classificateurs faibles pour obtenir un classificateur fort
  - Classificateurs faibles comme des arbres de décision
  - Peut gérer les données bruités



Machine Learning

- Deep Learning
  - Réseaux de neuronnes de nombreuses couches intermédiaires
  - Couches organisées en séquences
  - Capture les motifs complexes et non linéaires
  - Nécessite un grand nombre de données d'entrainement.



Evaluation

- Mesures de bases
  - TP – True Positifs
  - TN – True Negatifs
  - FP – False Positifs
  - FN – False Negatifs

- Accuracy
  - Mesure des prédictions justes sur le total de prédiction
  - Utilisation de la matrice de confusion dans un cas où classes > 2

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Class / Prediction	Class1	Class2	Class3
Class1	T1	F1	F1
Class2	F2	T2	F2
Class3	F3	F3	T3

$$Accuracy = \frac{\sum_{i=1}^{N} a_{ii}}{\sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij}}$$

Evaluation

Evaluation pour une classe :

- Précision
  - proportion de TP par rapport à toutes les prédictions
  - Permet de mesurer la confiance dans dans une prédiction positive

$$Prec = \frac{TP}{TP + FP}$$

- Recall
  - Proportion de TP par rapport à toutes les entrées positives du dataset
  - Permet de mesurer la capacité du modèle à trouver les entrées positives initiales

$$Rec = \frac{TP}{TP + FN}$$

- F1-Score
  - Moyenne harmonique de la précision et du recall
  - 1 si precision et recall parfait
  - 0 si precision ou recall nul

$$F1score = \frac{2 * (Prec * Rec)}{Prec + Rec}$$

Evaluation

Evaluation Multi-Classes

- Macro F1-Score
  - Moyenne des F1-Scores

$$MacroAverageF1Score = \frac{\sum_{i=1}^{N} F1Score_{class(i)}}{N}$$

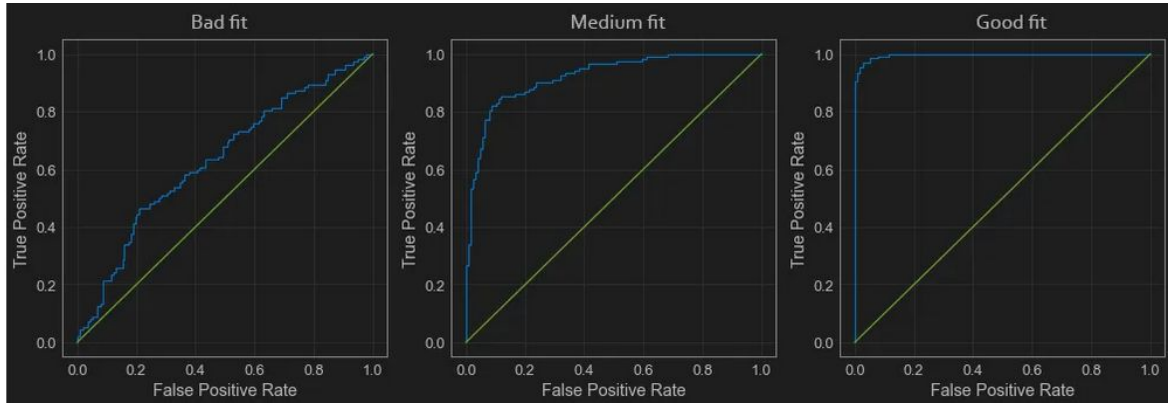
- Weighted F1-Score
  - Moyenne des F1-Scores pondérée par la taille de chaque classe

$$WeightedF1Score = \frac{2 * (Prc(Weight) * Rec(Weight)}{(Prc(Weight) + Rec(Weight)}$$

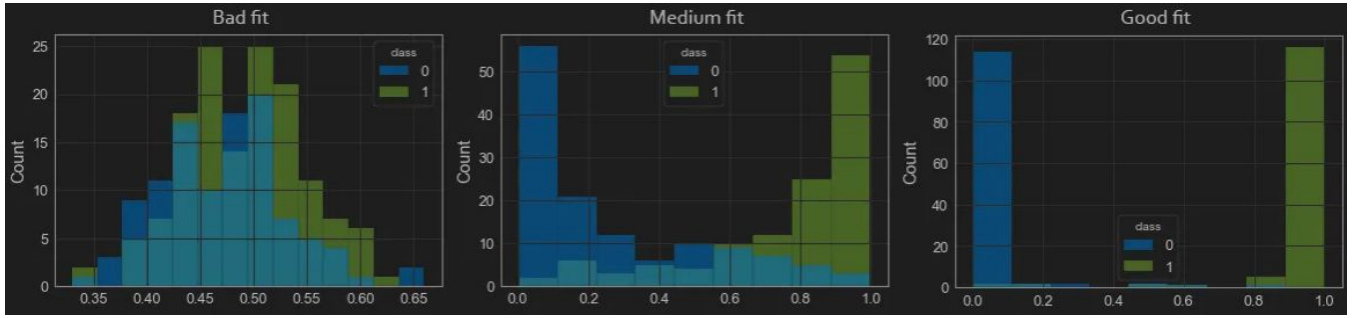
Evaluation

Courbe ROC et AUC Score

- Courbe ROC
  - True Positive Rate / False Positive Rate
  - Histogramme
  - OneVersusRest pour le multiclass



- AUC Score
  - Aire sous la courbe
  - 0,5 = aléatoire
  - 1 = parfait



Résultats

Détection

	Accuraccy	Macro F1-Score	Weighted F1-Score
Random Forest	0.98872	0.86297	0.98748
SVM	0.98621	0.81371	0.98374
XGBoost	0.99009	0.88147	0.98909
MLPC	0.98678	0.83640	0.98517
Adaboost	0.98325	0.78751	0.98095
DL	0.98564	0.81105	0.98332

Random Forest

	Precision	Recall	F1-Score
No Malware	0.94406	0.59735	0.73171
Malware	0.98946	0.99906	0.99424
Accuracy	0.98872		
Macro-average	0.96676	0.79820	0.86297
Weighted-average	0.98829	0.98872	0.98748

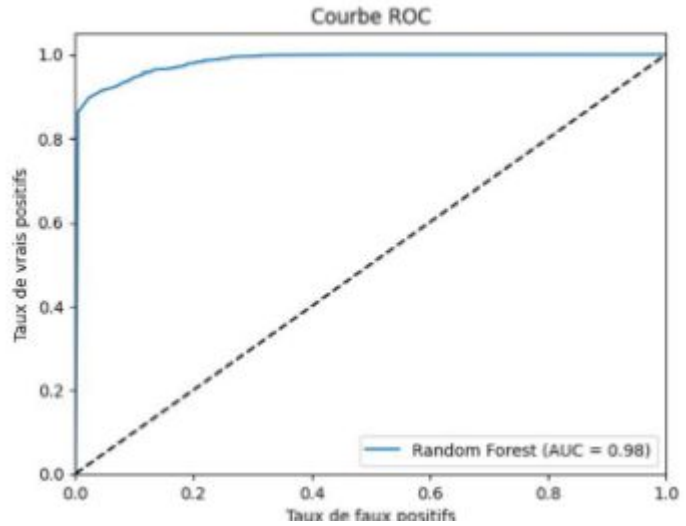
AdaBoost

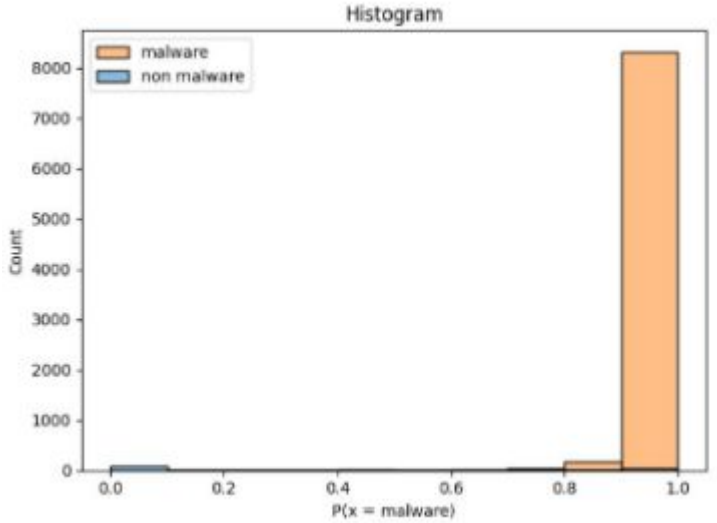
	Precision	Recall	F1-Score
No Malware	0.81102	0.45575	0.58357
Malware	0.98578	0.99719	0.99145
Accuracy	0.98325		
Macro-average	0.89840	0.72647	0.78751
Weighted-average	0.98128	0.98325	0.98095

Courbes ROC

Random Forest

- Bon fit
- Séparation distinctes des classes , même avec le dataset déséquilibré

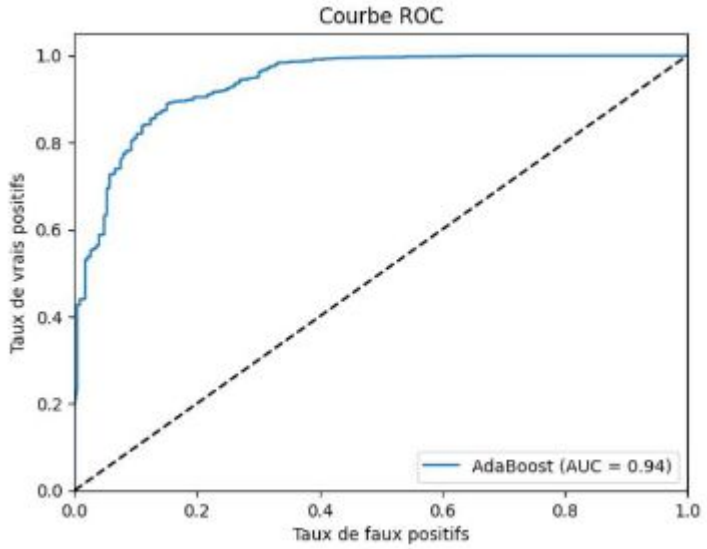


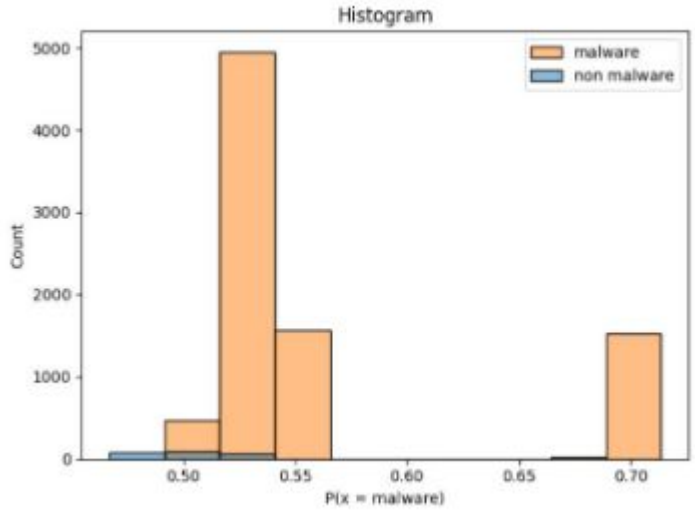


Courbes ROC

AdaBoost

- Fit moyen
- Séparation des classes mauvaise
- Bonne accuracy due au grand déséquilibre du dataset.





Résultats

Classification

	Accuraccy	Macro F1-Score	Weighted F1-Score
Random Forest	0.69691	0.70931	0.69644
Adaboost	0.44655	0.43540	0.43306
XGBoost	0.69691	0.71109	0.69970
MLPC	0.48594	0.50256	0.48286
SVM	0.17581	0.09710	0.10536
DL	F3	F3	

XGBoost

	Precision	Recall	F1-Score
Adware	0.93651	0.83099	0.88060
Backdoor	0.74737	0.69951	0.72265
Downloader	0.84066	0.76884	0.80315
Dropper	0.61929	0.67403	0.64550
Spyware	0.51741	0.64198	0.57300
Trojan	0.52284	0.51500	0.51889
Virus	0.83085	0.85641	0.84343
Worms	0.73822	0.66825	0.70149
Accuracy		0.69691	
Macro-average	0.71914	0.70688	0.71109
Weighted-average	0.70588	0.69691	0.69970

MLPC

	Precision	Recall	F1-Score
Adware	0.87719	0.70423	0.78125
Backdoor	0.46222	0.51232	0.48598
Downloader	0.66667	0.62312	0.64416
Dropper	0.40000	0.47514	0.43434
Spyware	0.34426	0.25926	0.29577
Trojan	0.33880	0.31000	0.32376
Virus	0.54098	0.67692	0.60137
Worms	0.47895	0.43128	0.45387
Accuracy		0.48594	
Macro-average	0.51363	0.49903	0.50256
Weighted-average	0.48612	0.48594	0.48286

SVM

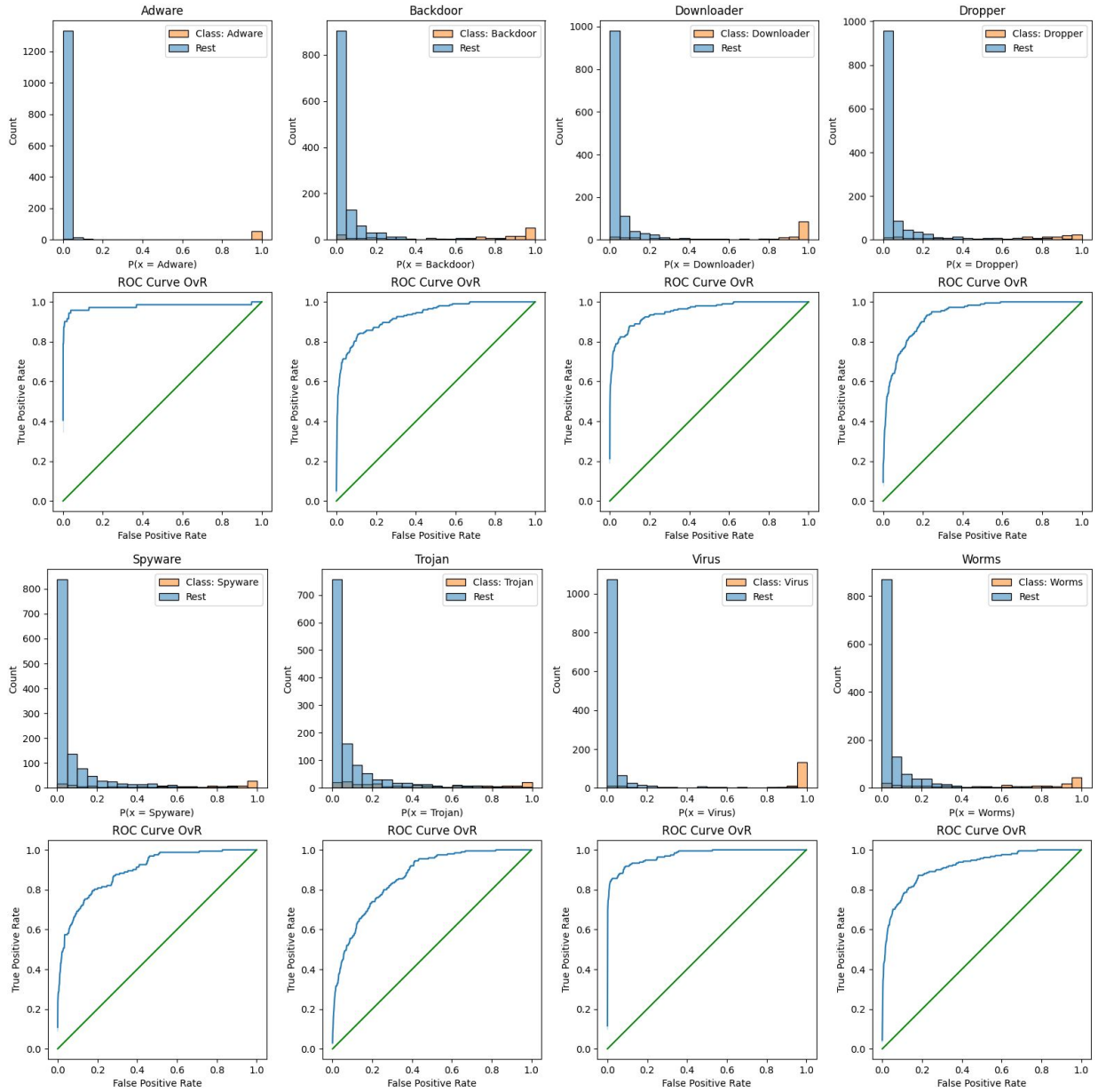
	Precision	Recall	F1-Score
Adware	0.00000	0.00000	0.00000
Backdoor	0.00000	0.00000	0.00000
Downloader	0.14565	0.97487	0.25343
Dropper	0.83333	0.02762	0.05348
Spyware	0.80000	0.07407	0.13559
Trojan	0.00000	0.00000	0.00000
Virus	0.82609	0.09744	0.17431
Worms	0.51282	0.09479	0.16000
Accuracy		0.17581	
Macro-average	0.38974	0.15860	0.09710
Weighted-average	0.40697	0.17581	0.10536

Courbes ROC

XGBoost

- Dinstiction dans les hitogrammes

- Bon fit ou moyen fit sur les courbes ROC

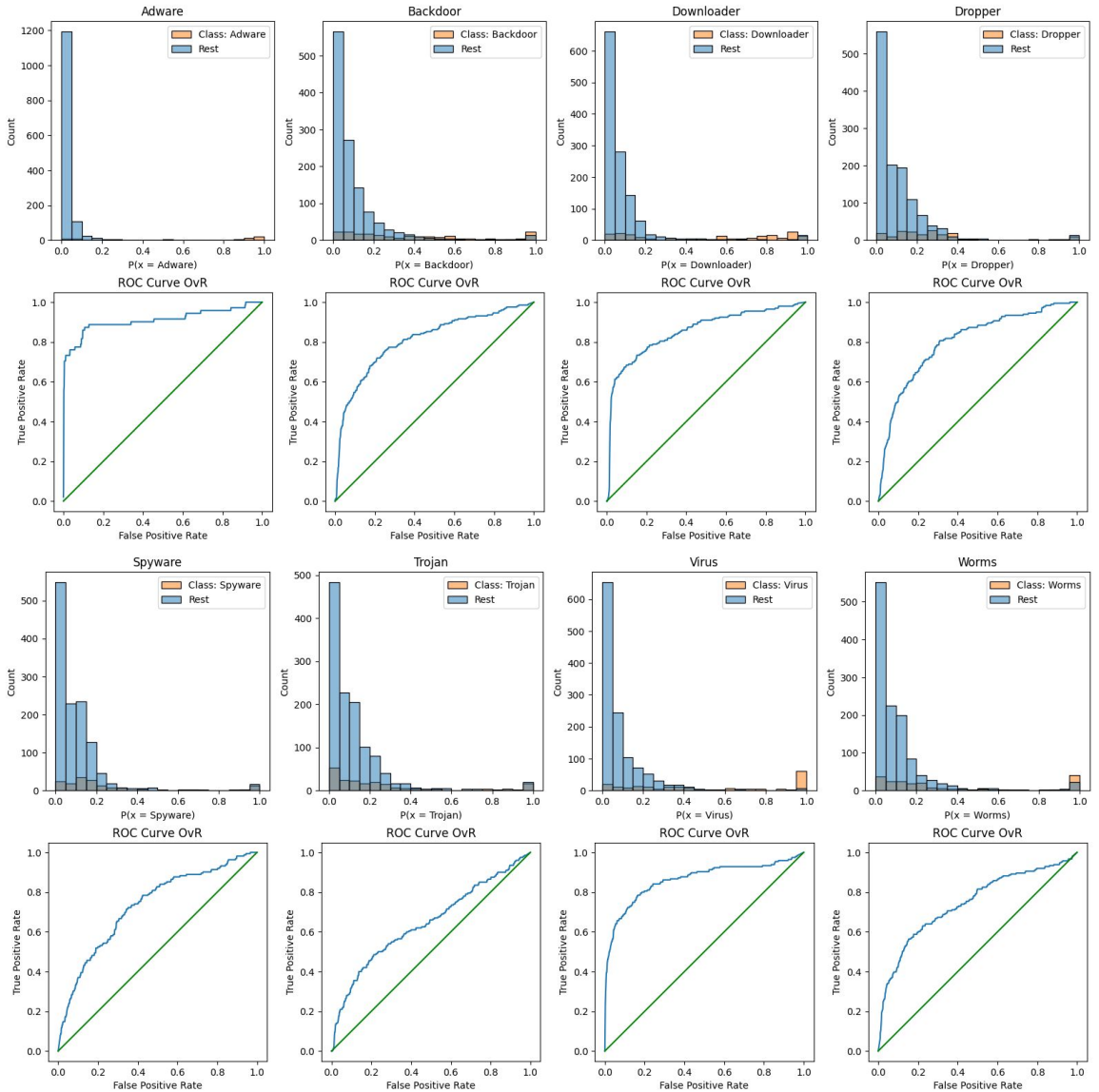


Class	AUC
Adware	0.9775
Backdoor	0.9308
Downloader	0.9547
Dropper	0.9329
Spyware	0.8939
Trojan	0.8633
Virus	0.9728
Worms	0.9156
Average	0.9302

Courbes ROC

MLPC

- Certaines classes ne sont pas bien distinctes
- Entre mauvais et moyen fit sur les courbes

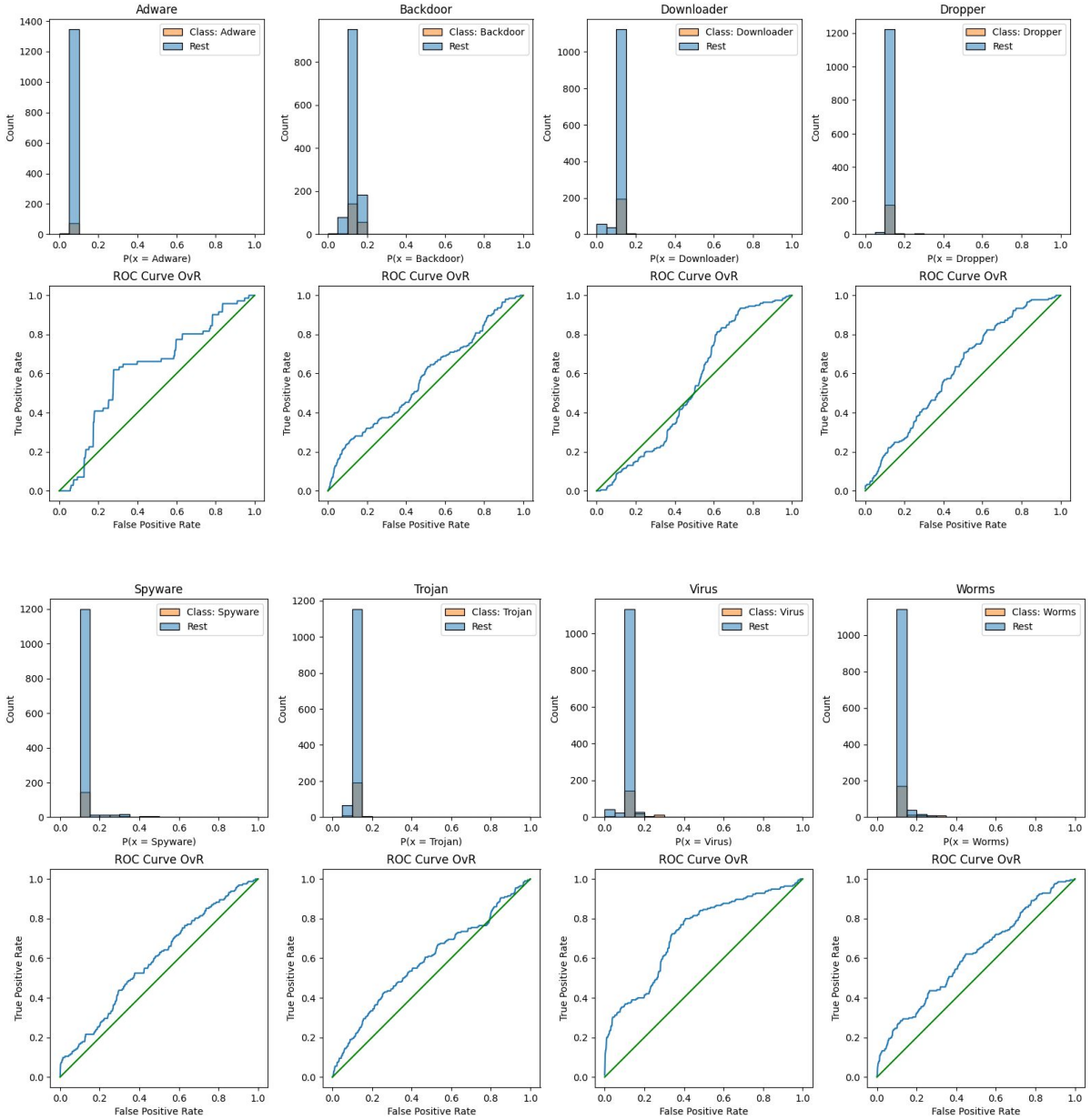


Class	AUC
Adware	0.9095
Backdoor	0.8096
Downloader	0.8529
Dropper	0.8032
Spyware	0.7297
Trojan	0.6454
Virus	0.8618
Worms	0.7436
Average	0.7945

Courbes ROC

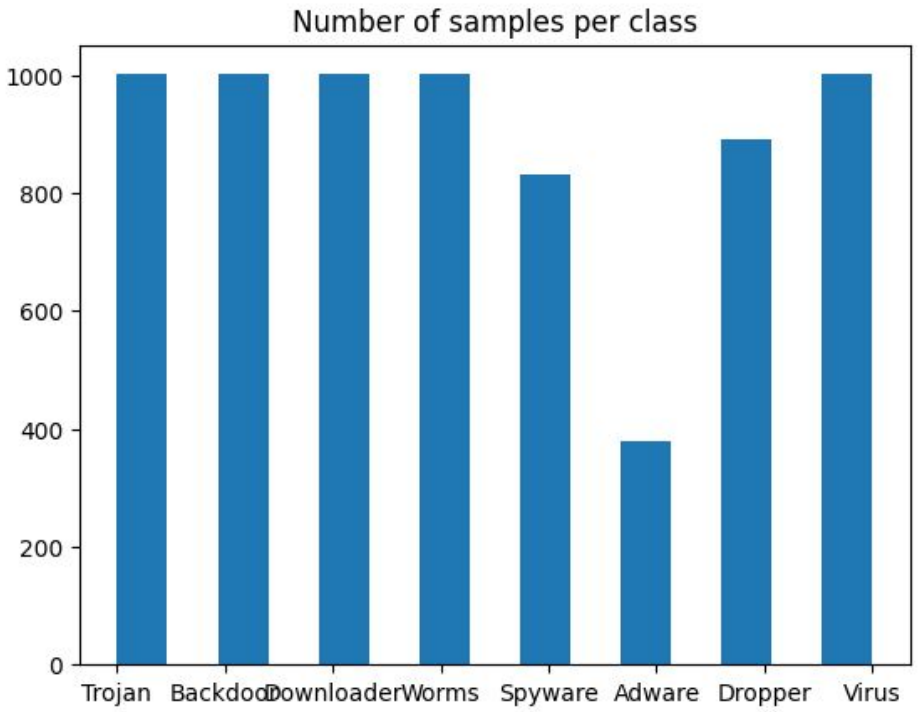
SVM

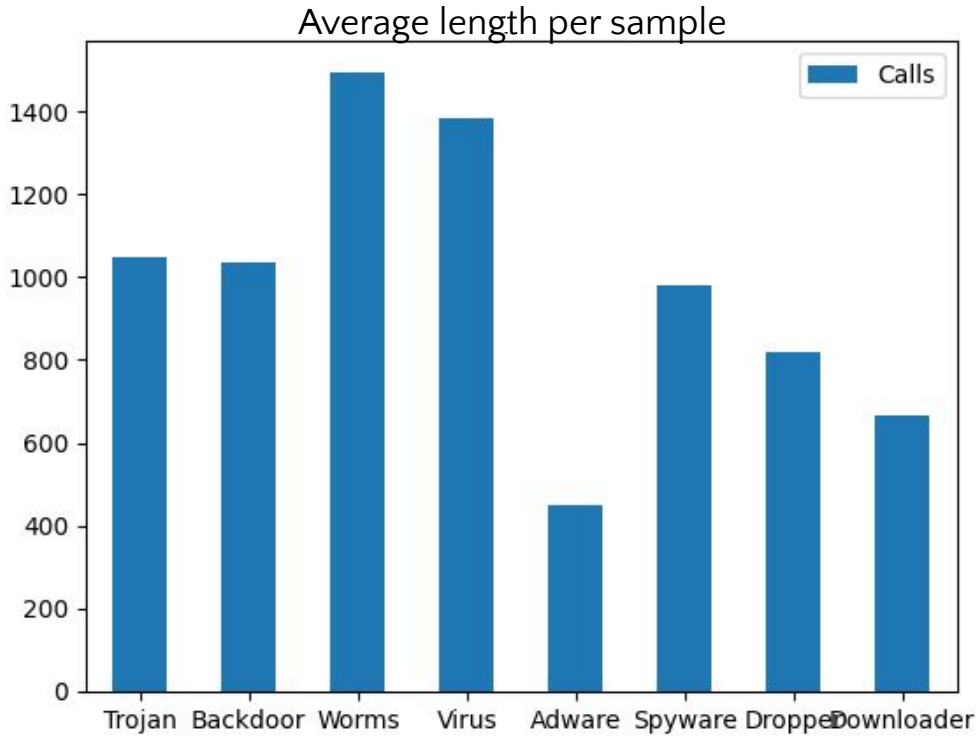
- Aucune dinstinction dans les classes
- Mauvais fit
- Score proche du hasard



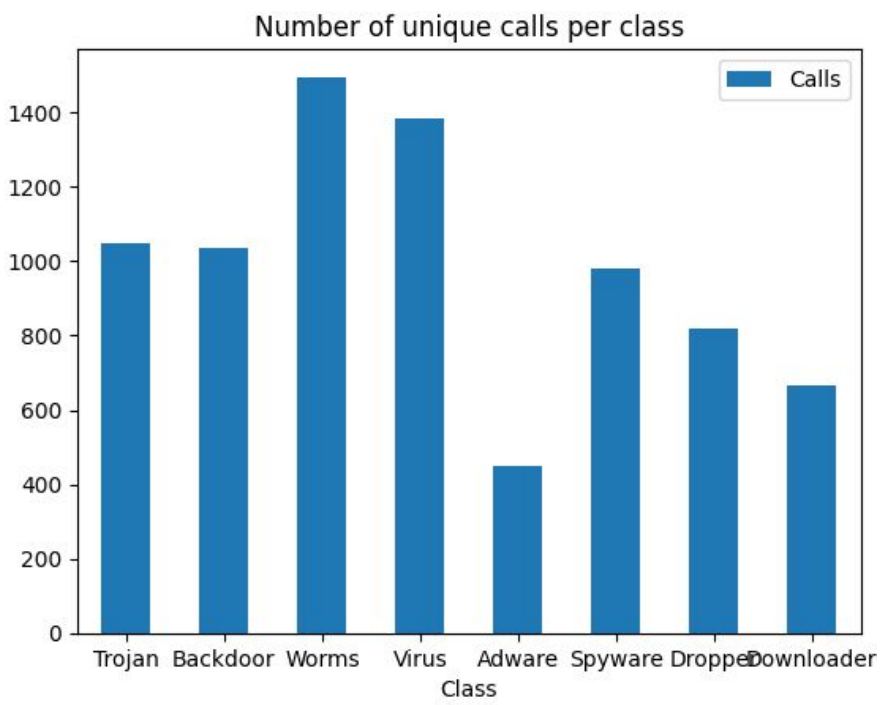
Class	AUC
Adware	0.6248
Backdoor	0.5752
Downloader	0.5345
Dropper	0.6159
Spyware	0.5904
Trojan	0.5856
Virus	0.7266
Worms	0.6110
Average	0.6080

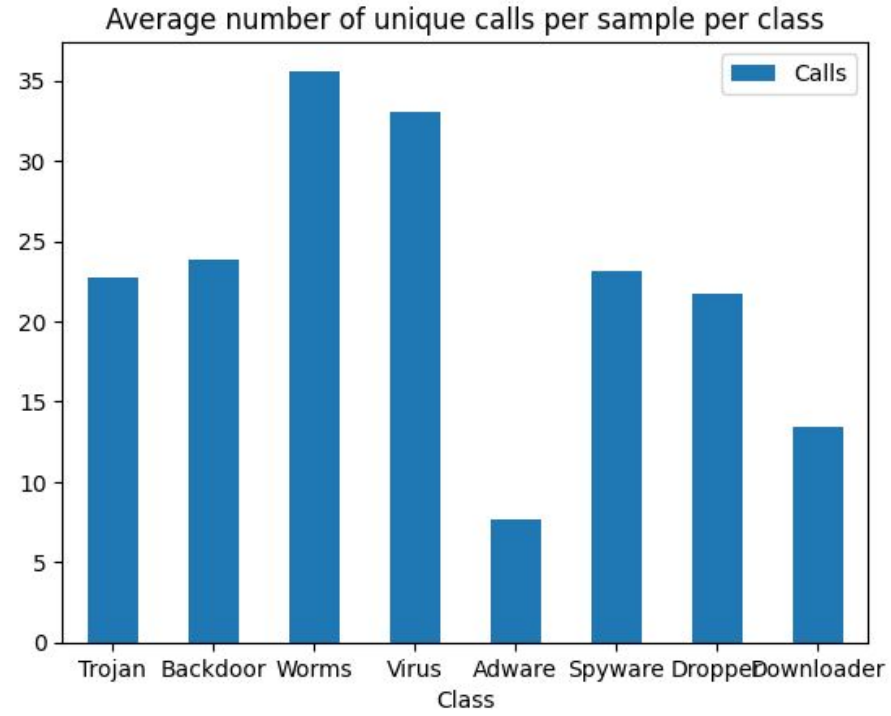
Analyse des données





Analyse des données





	Class	most_commons
0	Trojan	{1, 2, 3, 5, 10, 23, 119, 345}
1	Backdoor	{1, 2, 3, 12, 18001}
2	Worms	{1, 2, 3, 4, 201, 173, 1206, 23, 30}
3	Virus	{1, 2, 3, 5, 6, 11, 86321}
4	Adware	{1, 2, 3}
5	Spyware	{1, 2, 3, 100, 6, 8, 10, 11}
6	Dropper	{1, 2, 3, 4, 6, 18001, 150, 152}
7	Downloader	{1, 2, 3, 4, 6, 206, 244, 20}

Application

- Réalisation de deux applications permettant respectivement de faire de la classification et de la détection.

- Trois sections : Informations, Performances des modèles, Prédiction

- Trois étapes :
  - Choisir le modèle
  - Charger le fichier contenant les appels API
  - Analyser

Application

Démonstration

Conclusion

Analyse de deux problématiques

- Classification binaire
- Classification multiclasse

Mise en place de différentes techniques de machine learning

Comparaison des résultats

- Détection
  - Bons résultats, avec certaines disparités
  - Dataset grandement déséquilibré
- Classification
  - Résultats très disparates entre les différents modèles
  - XGBoost et Random Forest présent des résultats satisfaisant pour une classification à 8 labels
  - AdaBoost et MLPC et ne semblent pas performants
  - SVM et DL présentent des résultats proche de l'aléatoire

Conclusion

Merci pour votre attention.