



ÉCOLE NATIONALE SUPÉRIEURE
D'INFORMATIQUE ET D'ANALYSE DES SYSTÈMES
- RABAT

PROJET DE MACHINE LEARNING ET DEEP LEARNING

Approche de Machine Learning et Deep
Learning pour la Classification des
Trafic Réseau : BENIGN vs. DDoS

Élèves :

Asma EL IDRISSE
Aya LEBSIRE
Salma JENNANE
Salma OUAHIB

Enseignant :

Houda BENBRAHIM

6 janvier 2025

Résumé

Avec la montée des cyberattaques, notamment les attaques de type *DDoS*, ce projet vise à développer un modèle efficace pour détecter les anomalies dans le trafic réseau en utilisant des techniques de *Machine Learning* et de *Deep Learning*. La problématique se concentre sur la classification des flux en normaux et malveillants, tout en relevant les défis liés aux valeurs aberrantes, aux corrélations élevées et au déséquilibre des classes.

Plusieurs modèles, tels que **KNN**, **Random Forest**, **Naive Bayes**, **SVM** et un **MLP**, ont été testés en raison de leur capacité à traiter des données complexes. Les données ont été soigneusement préparées à travers le traitement des valeurs aberrantes et infinies, ainsi que la réduction des dimensions. Les modèles **SVM** et **Naive Bayes** ont montré des performances exceptionnelles, suivis par **Random Forest**, tandis que **KNN** a nécessité des ajustements pour réduire les erreurs. Par ailleurs, le modèle *K-Means* a été employé pour explorer les structures des données et identifier des regroupements naturels. Le **MLP** a démontré une grande robustesse grâce à une régularisation adaptée.

Les résultats obtenus confirment que ces approches permettent une détection proactive et fiable des anomalies, offrant ainsi des solutions concrètes pour renforcer la sécurité des réseaux informatiques.

Table des matières

Résumé	1
1 Introduction	4
1.1 Position du problème	4
1.2 Travaux connexes et état de l’art	4
2 Les données	5
2.1 Jeu de données	5
2.2 Preprocessing des données	6
2.2.1 Nettoyage des données	6
3 Application des algorithmes	8
3.1 Apprentissage supervisé	8
3.1.1 Random Forest	8
3.1.2 K Nearest Neighbor	9
3.1.3 Naive Bayes-Bernoulli	9
3.1.4 Support vector machine(SVM)	10
3.1.5 Analyse des résultats	10
3.1.6 Multilayer perceptron (MLP)	10
3.2 Apprentissage non supervisé	11
3.2.1 K-Means	11
Conclusion Générale	14

Table des figures

2.1	Présence d'enregistrements dupliqués	6
2.2	Supression d'enregistrements dupliqués	6
2.3	Distrubition des données	7
3.1	Performance de Random Forest.	8
3.2	Matrice de confusion de Random Forest.	8
3.3	Performance de KNN.	9
3.4	Matrice de confusion de KNN.	9
3.5	Performance de Naive Bayes-Bernoulli.	9
3.6	Matrice de confusion de Naive Bayes-Bernoulli.	9
3.7	Performance de SVM.	10
3.8	Matrice de confusion de SVM.	10
3.9	Performance de MLP.	11
3.10	Matrice de confusion.	11
3.11	le graphique de la méthode du coude.	12
3.12	Résultat du coefficient de silouhette	12
3.13	Résultat du coefficient de Davies-Bouldin	13
3.14	Résultats de k_{means}	13

Chapitre 1

Introduction

1.1 Position du problème

Le but de cette étude est de détecter les cybermenaces qui menacent la sécurité des réseaux informatiques. L'application de techniques de machine learning s'avère cruciale dans ce contexte, en mettant en œuvre divers modèles adaptés à ce problème et en les comparant afin d'évaluer leur efficacité. L'objectif final est d'identifier le modèle le plus performant, capable de détecter, et de catégoriser rapidement et avec précision les intrusions réseau,

L'importance de ces expériences réside dans la capacité à repérer les cyberattaques de manière proactive et rapide, en identifiant les facteurs clés qui facilitent cette détection. Cette approche représente une solution innovante et efficace pour renforcer la protection des infrastructures face aux menaces numériques croissantes.

1.2 Travaux connexes et état de l'art

Plusieurs systèmes sont utilisés pour la détection d'intrusions avant l'application des algorithmes de machine learning et de deep learning, tels que les systèmes de détection d'intrusions basés sur les signatures (IDS) et les systèmes basés sur les anomalies. Ces systèmes ont longtemps été utilisés pour surveiller le trafic. Toutefois, ils présentent des limites, notamment en termes de capacité à détecter de nouvelles menaces, et leur gestion peut être complexe.

L'avènement du machine learning a révolutionné la détection des intrusions en permettant une détection plus précise et dynamique des attaques. De nombreuses études récentes ont exploré l'application de diverses techniques d'apprentissage supervisé et non supervisé pour améliorer la précision des systèmes de détection d'intrusions

Chapitre 2

Les données

2.1 Jeu de données

Suite à une collecte ciblée, nous avons obtenu un ensemble de données spécifiquement conçu pour l'analyse des flux réseau et la détection des anomalies, particulièrement dans le cadre des attaques DDoS (Distributed Denial of Service). Ce dataset inclut une large variété de paramètres classés en plusieurs catégories pour une meilleure compréhension des caractéristiques du trafic réseau.

Les données associées aux flux réseau incluent des paramètres cruciaux tels que la durée totale des flux (*Flow Duration*), qui mesure la durée d'un flux réseau, ainsi que le nombre total de paquets envoyés en avant et en arrière (*Total Fwd Packets* et *Total Backward Packets*). La taille des paquets est prise en considération en analysant les métriques de longueur totale des paquets envoyés (*Total Length of Fwd Packets* et *Total Length of Bwd Packets*), qui permettent d'avoir une vision détaillée sur le volume et la direction des flux des trafics réseaux.

Les caractéristiques statistiques des paquets fournissent des informations supplémentaires sur le comportement du trafic, incluant des mesures comme la longueur maximale, minimale, moyenne et l'écart type des paquets envoyés en avant et en arrière (*Fwd Packet Length Max, Min, Mean, Std*). Ces métriques permettent de détecter les variations inhabituelles dans les paquets, souvent associées à des comportements malveillants.

Enfin, des paramètres temporels liés à l'activité des flux permettent une évaluation fine des délais dans le trafic réseau. Cela inclut des métriques comme les temps actifs moyens, maximums et minimums (*Active Mean, Active Max, Active Min*), ainsi que les temps inactifs (*Idle Mean, Idle Max, Idle Min*). Ces données temporelles jouent un rôle clé dans l'identification des anomalies de comportement dans le réseau.

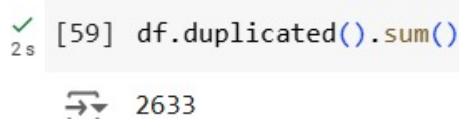
La variable cible (*Label*) catégorise chaque flux réseau comme normal ("BENIGN") ou anormal ("DDoS"), permettant ainsi de différencier les comportements légitimes des activités malveillantes. Ce dataset offre ainsi une base solide pour analyser, comprendre et détecter les anomalies réseau, tout en renforçant la capacité de sécurisation proactive.

2.2 Preprocessing des données

2.2.1 Nettoyage des données

Gestion des données dupliquées

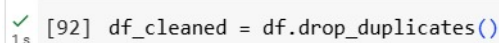
La présence de 2633 lignes dupliquées dans notre dataset souligne l'importance de les éliminer afin d'assurer la qualité et la fiabilité des données.



```
[59] df.duplicated().sum()
```

2633

FIGURE 2.1 – Présence d'enregistrements dupliqués



```
[92] df_cleaned = df.drop_duplicates()
```

1s

FIGURE 2.2 – Suppression d'enregistrements dupliqués

Gestion des données infinies

Les valeurs infinies issues de Flow Bytes et Flow Packets, souvent causées par des divisions par zéro, perturbent les calculs et faussent les analyses. Les supprimer garantit des données fiables, cohérentes et sans problèmes lors des traitements.

Gestion des colonnes corrélées

La présence d'un grand nombre de variables dans notre jeu de données (79 colonnes) nous a conduit à réaliser une réduction dimensionnelle basée sur l'analyse des corrélations. Nous avons développé une fonction qui calcule les coefficients de corrélation entre chaque paire de colonnes. En fixant un seuil de 80 % de corrélation, nous avons pu identifier et supprimer l'une des deux variables fortement corrélées dans chaque paire. Grâce à cette méthode, nous avons réussi à réduire considérablement le nombre de variables, passant de 79 à 40, tout en préservant l'essentiel de l'information contenue dans les données.

Gestion des outliers

Pour détecter les valeurs aberrantes dans nos données, nous avons utilisé la méthode du Z-score, qui évalue l'écart de chaque valeur par rapport à la moyenne en termes d'écart-type. Cette méthode nous a permis d'identifier 61 603 lignes, soit 27,61 % des données, comme étant des outliers. Pour corriger ces anomalies, nous avons choisi d'utiliser l'imputation par la médiane, afin de préserver la qualité des données et minimiser l'impact des valeurs aberrantes sur l'analyse.

Distrubition des données :

Le diagramme montre que le dataset est composé de 57.4% de flux malveillants (DDoS) et de 42.6% de flux normaux (BENIGN), mettant en évidence un déséquilibre des classes qui devra être pris en compte lors de l'analyse et la modélisation.

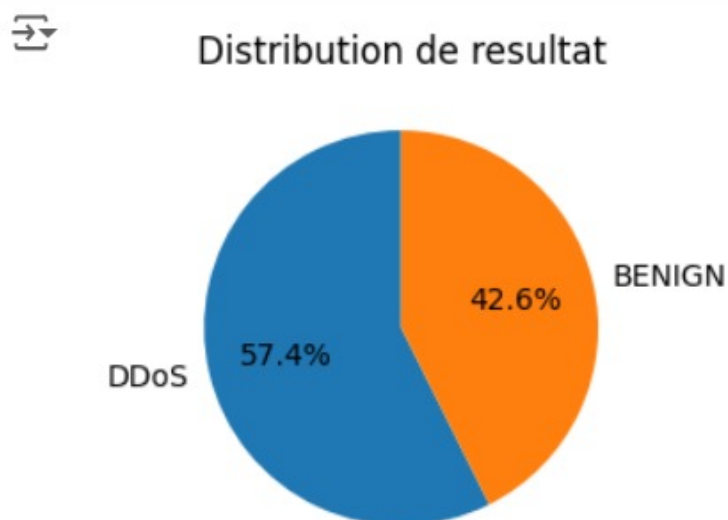


FIGURE 2.3 – Distrubition des données

Chapitre 3

Application des algorithmes

3.1 Apprentissage supervisé

Dans cette partie, nous avons mis en œuvre des algorithmes d'apprentissage supervisé adaptés à la classification, en tenant compte de la nature de notre problématique. Parmi les méthodes utilisées figurent : le **KNN**, le **Naive Bayes-Bernoulli**, le **Random Forest**, le **SVM**, ainsi que le **MLP**, un modèle de *Deep Learning*. Ces algorithmes ont été sélectionnés pour leur efficacité dans des contextes similaires et leur capacité à répondre à nos besoins spécifiques.

3.1.1 Random Forest

	precision	recall	f1-score	support
0	1.00	1.00	1.00	28491
1	1.00	1.00	1.00	38434
accuracy			1.00	66925
macro avg	1.00	1.00	1.00	66925
weighted avg	1.00	1.00	1.00	66925

FIGURE 3.1 – Performance de Random Forest.

```
print(confusion_matrix(Y_test, y_pred))
```

```
[[28490  1]
 [  3 38431]]
```

FIGURE 3.2 – Matrice de confusion de Random Forest.

3.1.2 K Nearest Neighbor

	precision	recall	f1-score	support
0	1.00	1.00	1.00	28491
1	1.00	1.00	1.00	38434
accuracy			1.00	66925
macro avg	1.00	1.00	1.00	66925
weighted avg	1.00	1.00	1.00	66925

FIGURE 3.3 – Performance de KNN.

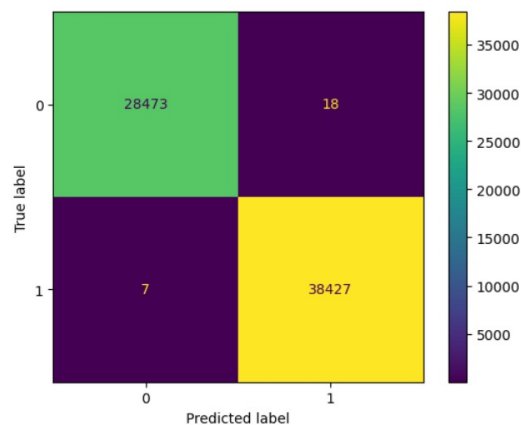


FIGURE 3.4 – Matrice de confusion de KNN.

3.1.3 Naive Bayes-Bernoulli

Rapport de classification (Naive Bayes - Bernoulli, meilleur modèle):

	precision	recall	f1-score	support
0	1.00	0.98	0.99	28491
1	0.99	1.00	0.99	38434
accuracy			0.99	66925
macro avg	0.99	0.99	0.99	66925
weighted avg	0.99	0.99	0.99	66925

Matrice de confusion (Naive Bayes - Bernoulli, meilleur modèle):

```
[[28040 451]
 [ 53 38381]]
```

FIGURE 3.5 – Performance de Naive Bayes-Bernoulli.

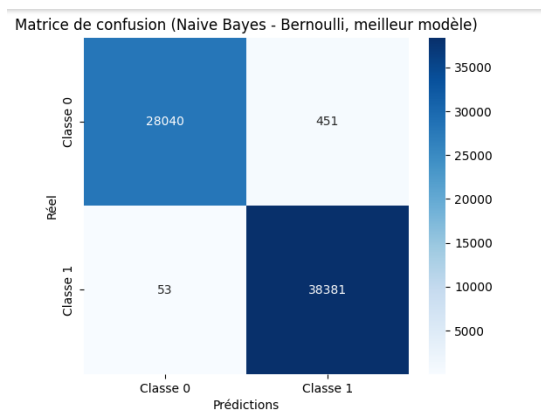


FIGURE 3.6 – Matrice de confusion de Naive Bayes-Bernoulli.

3.1.4 Support vector machine(SVM)

Rapport de classification (SVM, meilleur modèle):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	28491
1	1.00	1.00	1.00	38434
accuracy			1.00	66925
macro avg	1.00	1.00	1.00	66925
weighted avg	1.00	1.00	1.00	66925

Matrice de confusion (SVM, meilleur modèle):

```
[[28471  20]
 [ 22 38412]]
```

FIGURE 3.7 – Performance de SVM.

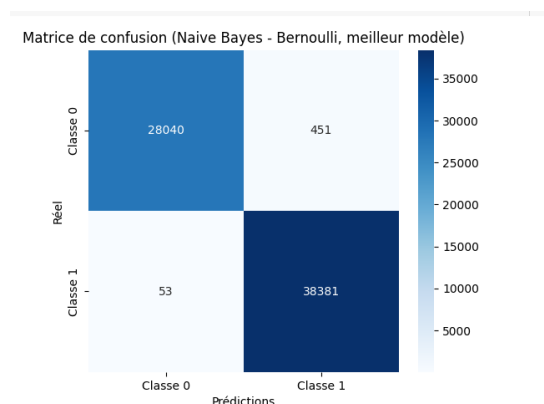


FIGURE 3.8 – Matrice de confusion de SVM.

3.1.5 Analyse des résultats

Les algorithmes **KNN**, **Random Forest**, **Naive Bayes** et **SVM** affichent des performances remarquables, avec une précision, un rappel et un F1-score proches de 100 %. Ces résultats exceptionnels indiquent un phénomène de sur-ajustement (*overfitting*), qui peut s'expliquer par la présence d'un dataset de grande taille, bien équilibré, et absence de valeurs manquantes et aberrantes.

Pour atténuer ce phénomène, il serait pertinent d'adopter des stratégies telles que la validation croisée, l'ajustement des hyperparamètres, ou encore l'introduction d'une régularisation plus stricte. Par ailleurs, réduire la complexité des modèles ou effectuer une réduction de dimensionnalité, comme avec PCA, pourrait également contribuer à améliorer leur capacité de généralisation.

3.1.6 Multilayer perceptron (MLP)

Cette approche appartient au domaine du Deep Learning, qui repose sur l'utilisation de réseaux neuronaux artificiels composés de plusieurs couches.

Accuracy globale: 0.9913

Rapport de classification complet :

	precision	recall	f1-score	support
0	1.00	0.98	0.99	28491
1	0.99	1.00	0.99	38434
accuracy			0.99	66925
macro avg	0.99	0.99	0.99	66925
weighted avg	0.99	0.99	0.99	66925

FIGURE 3.9 – Performance de MLP.

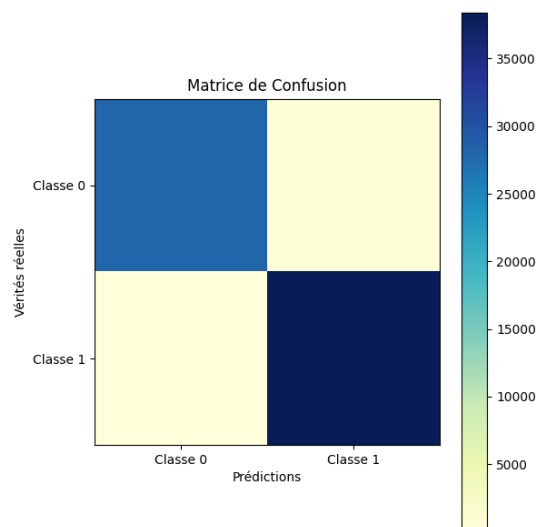


FIGURE 3.10 – Matrice de confusion.

Les hyperparamètres optimaux identifiés à l'aide de la méthode de (*Random Search*) sont : une taille de batch de **128**, une fonction d'activation de type **tanh**, un taux de dropout de **0,1362**, un taux d'apprentissage (*learning rate*) de **0,0063**, un optimiseur **RMSprop**, et un nombre d'unités fixé à **86**. Ces paramètres ont été sélectionnés pour optimiser les performances du modèle en fonction des critères définis ainsi l'utilisation de la régularisation pour remédier au problème de l'overfitting.

3.2 Apprentissage non supervisé

3.2.1 K-Means

Determination du nombre de clusters

Méthode du Coude (Elbow Method)

d'après ce graphique, le coude semble se situer autour de $k = 3$ ou 4 , alors le Nombre optimal de clusters : Probablement 3 ou 4.

le Coefficient Silhouette

Le score de silhouette continue d'augmenter à mesure que le nombre de clusters augmente, atteignant un maximum avec 10 clusters. Cela peut indiquer que les données ont une structure complexe et nécessitent un nombre plus élevé de clusters pour capturer leur diversité.

Coefficient de Davies-Bouldin

Le score minimal est 1.1265 pour 6 clusters, indiquant une meilleure qualité des regroupements par rapport aux autres nombres de clusters.

Bien que les indices suggèrent un nombre plus élevé de clusters, nous avons choisi de travailler avec 2 clusters. Cette décision peut être justifiée par le fait que ces 2 clusters

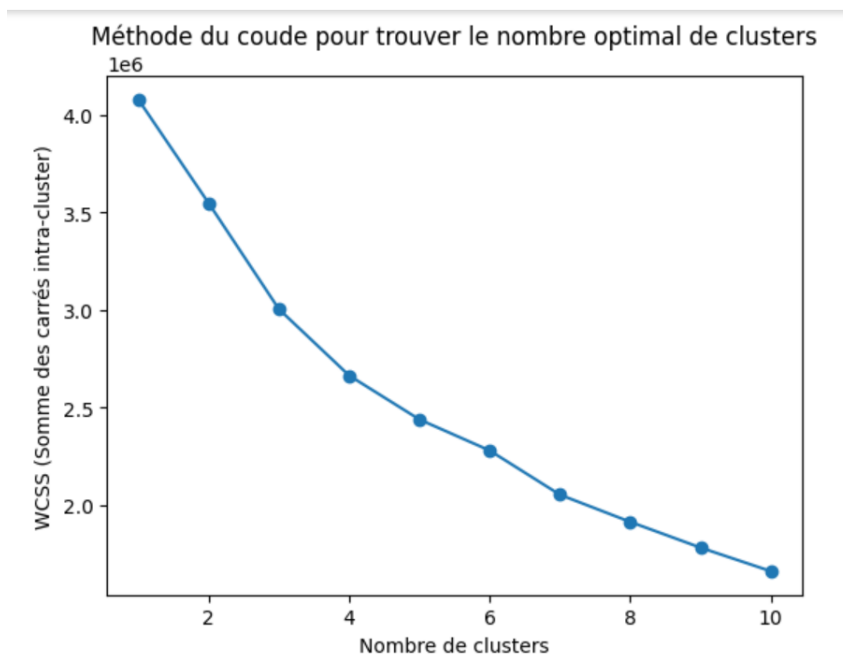


FIGURE 3.11 – le graphique de la méthode du coude.

```
Silhouette Score pour nombre de clusters 2 : 0.31907906343113207
Silhouette Score pour nombre de clusters 3 : 0.3284256416808819
Silhouette Score pour nombre de clusters 4 : 0.37059000707548617
Silhouette Score pour nombre de clusters 5 : 0.3885735668695945
Silhouette Score pour nombre de clusters 6 : 0.40351984675074537
Silhouette Score pour nombre de clusters 7 : 0.45351273217408256
Silhouette Score pour nombre de clusters 8 : 0.4622203371417872
Silhouette Score pour nombre de clusters 9 : 0.4689419809058217
Silhouette Score pour nombre de clusters 10 : 0.4811883298539905
```

FIGURE 3.12 – Résultat du coefficient de silhouette

contiennent probablement des sous-groupes internes, ce qui explique pourquoi les indices recommandent un découpage plus détaillé.

Visualiser des clusters avec k-means (k=2)

```

Coefficient de Davies-Bouldin pour nombre de clusters 2 : 1.5520188400503399
Coefficient de Davies-Bouldin pour nombre de clusters 3 : 1.4756785372702
Coefficient de Davies-Bouldin pour nombre de clusters 4 : 1.3825760730833678
Coefficient de Davies-Bouldin pour nombre de clusters 5 : 1.2373373670636245
Coefficient de Davies-Bouldin pour nombre de clusters 6 : 1.1265883094836922
Coefficient de Davies-Bouldin pour nombre de clusters 7 : 1.4108230896301002
Coefficient de Davies-Bouldin pour nombre de clusters 8 : 1.267661585390814
Coefficient de Davies-Bouldin pour nombre de clusters 9 : 1.215229570713982
Coefficient de Davies-Bouldin pour nombre de clusters 10 : 1.1616578666731932

```

FIGURE 3.13 – Résultat du coefficient de Davies-Bouldin

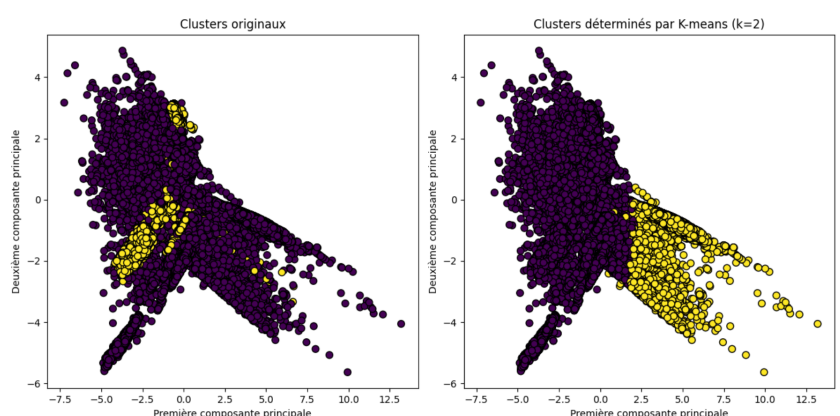


FIGURE 3.14 – Résultats de k_{means}

Conclusion Générale

Dans le cadre de ce projet, nous avons exploré l'utilisation de plusieurs algorithmes de machine learning et deep learning, notamment KNN, Random Forest, Naive Bayes et SVM et le MLP, pour la détection des cybermenaces.

L'importance de ce projet réside dans sa capacité à fournir une solution efficace face à l'augmentation constante des cyberattaques. En exploitant les modèles de machine learning, nous avons non seulement amélioré la précision et la rapidité de détection, mais aussi renforcé la compréhension des facteurs clés qui influencent ces menaces.

Le recours au machine learning est essentiel pour répondre aux défis posés par la complexité croissante des menaces numériques. Cette approche permet de renforcer la sécurité des infrastructures informatiques et de mieux anticiper les cyberattaques, offrant ainsi une contribution significative à la lutte contre les menaces numériques mondiales.

Finalement, comme perspectives futures on peut par la suite tester les modèles sur des données variées et renforcer leur robustesse face aux attaques adversariales constituerait également une avancée significative. Aussi on peut concevoir une solution intégrée combinant détection, alerte et réponse automatisée afin de renforcer la protection des réseaux face aux cybermenaces.