



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Oran1, Faculté des Sciences Exactes et Appliquées Département d'Informatique

MASTER 1 SÉCURITÉ INFORMATIQUE

Module : IA Pour Cyber Security

Projet 5.3 : Attaques adversariales sur malware

SOMMAIRE

1. Résumé exécutif.....	4
2. Méthodologie.....	4
2.1 Environnement et topologie.....	4
2.2 Dataset utilisé.....	5
2.3 Outils utilisés.....	5
2.4 Entraînement du modèle (Baseline).....	5
2.5 Attaque adversariale (Evasion).....	5
5	
3. Résultats.....	6
3.1 Résultats quantitatifs.....	6
6	
3.2 Analyse des features sensibles.....	6
7	
4. Recommandations.....	7
5. Conclusion.....	8

Liste de Figures

<i>Figure 1 :Environnement de travail sous Google Colab.....</i>	<i>4</i>
<i>Figure 2 :Accuracy avant attaque.....</i>	<i>5</i>
<i>Figure 3 :Résultats de l'attaque adversariale (taux d'évasion = 34 %)......</i>	<i>6</i>
<i>Figure 4 : Top 10 des features les plus sensibles face au bruit gaussien.....</i>	<i>7</i>

1. Résumé exécutif

Ce projet a pour objectif d'évaluer la robustesse d'un modèle de classification de malwares face à des attaques adversariales.

Un classifieur Random Forest a été entraîné à partir d'un sous-ensemble du **dataset EMBER**, puis soumis à une attaque par évasion basée sur l'ajout de bruit gaussien.

Les résultats montrent qu'après l'attaque, la précision du modèle chute de plus de **30 %**, ce qui démontre la vulnérabilité du modèle face à des perturbations légères mais ciblées.

2. Méthodologie

2.1 Environnement et topologie

L'expérimentation a été réalisée dans un environnement sécurisé et contrôlé :

- **Système d'exploitation** : Linux (machine virtuelle **Google Colab**)
- **Environnement de développement** : Visual Studio Code (préparation locale)
- **Langage de programmation** : Python
- **Dataset** : EMBER 2018 (sous-ensemble)

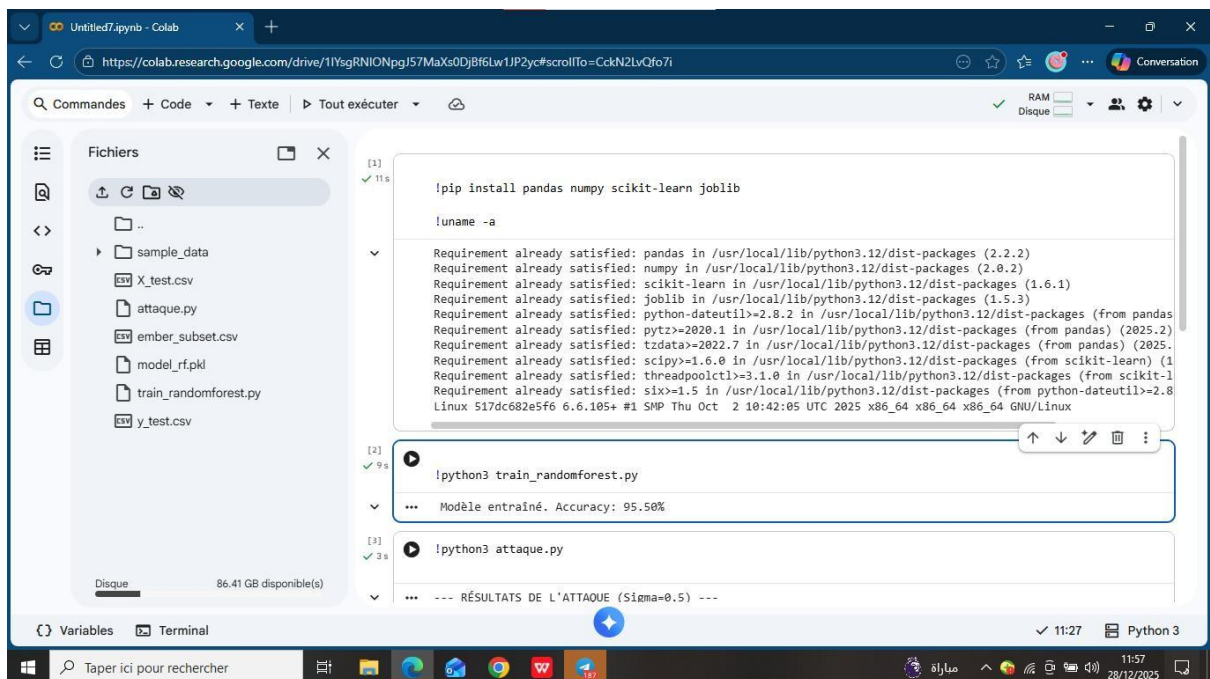


Figure 1: Environnement de travail sous Google Colab

2.2 Dataset utilisé:

Le dataset **EMBER** contient **2381 caractéristiques** extraites de fichiers exécutables Windows. En raison des limitations matérielles (4 Go de RAM), un sous-ensemble équilibré a été créé :

- **2500** échantillons malveillants.
- **2500** échantillons bénins.

2.3 Outils utilisés:

Outil	Utilisation
GoogleColab	Entraînement du modèle et exécution de l'attaque en ligne de commande.
Python	Langage utilisé pour les scripts train_randomforest.py et attaque.py
Pandas / Scikit-learn	Manipulation des données et création du modèle Random Forest .

2.4 Entraînement du modèle (Baseline):

Un modèle Random Forest a été entraîné sur les données non perturbées afin d'obtenir une performance de référence.

- **Script** : `train_randomforest.py`
- **Métrique** : Accuracy
- **Résultat** : **95,50 %**

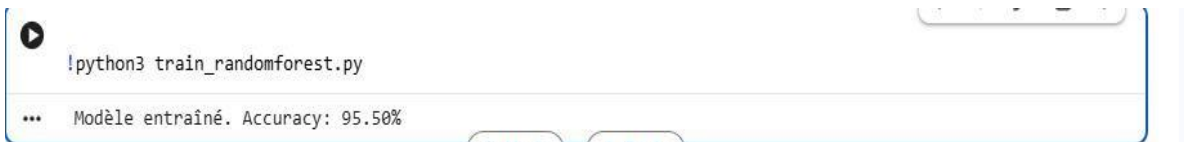


Figure 2:Accuracy avant attaque

2.5 Attaque adversariale (Evasion):

Une attaque par évasion a été mise en œuvre en ajoutant un bruit gaussien aux échantillons malveillants :

$$X_{adv} = X + \mathcal{N}(0, \sigma)$$

avec $\sigma=0.5$.

- **Script** : `attaque.py`
- **Objectif** : réduire la capacité de détection du modèle
- **Modèle ciblé** : `model_rf.pkl`

3. Résultats

3.1 Résultats quantitatifs :

```
!python3 attaque.py

... --- RÉSULTATS DE L'ATTAQUE (Sigma=0.5) ---
Nombre de malwares ayant trompé le modèle: 34 / 100
Taux d'évasion (Drop Accuracy): 34.0%
Figure(1000x600)
```

Figure 3:Résultats de l'attaque adversariale (taux d'évasion = 34 %)

- Le tableau suivant présente les performances du modèle avant et après l'attaque adversariale:

Indicateur	Valeur
Accuracy avant attaque	95,50 %
Accuracy après attaque	66 %
Taux d'évasion	34%

3.2 Analyse des features sensibles

Les résultats montrent qu'une simple perturbation gaussienne suffit à contourner le classifieur dans un nombre significatif de cas.

Cela met en évidence la nécessité d'intégrer des mécanismes de défense contre les attaques adversariales.

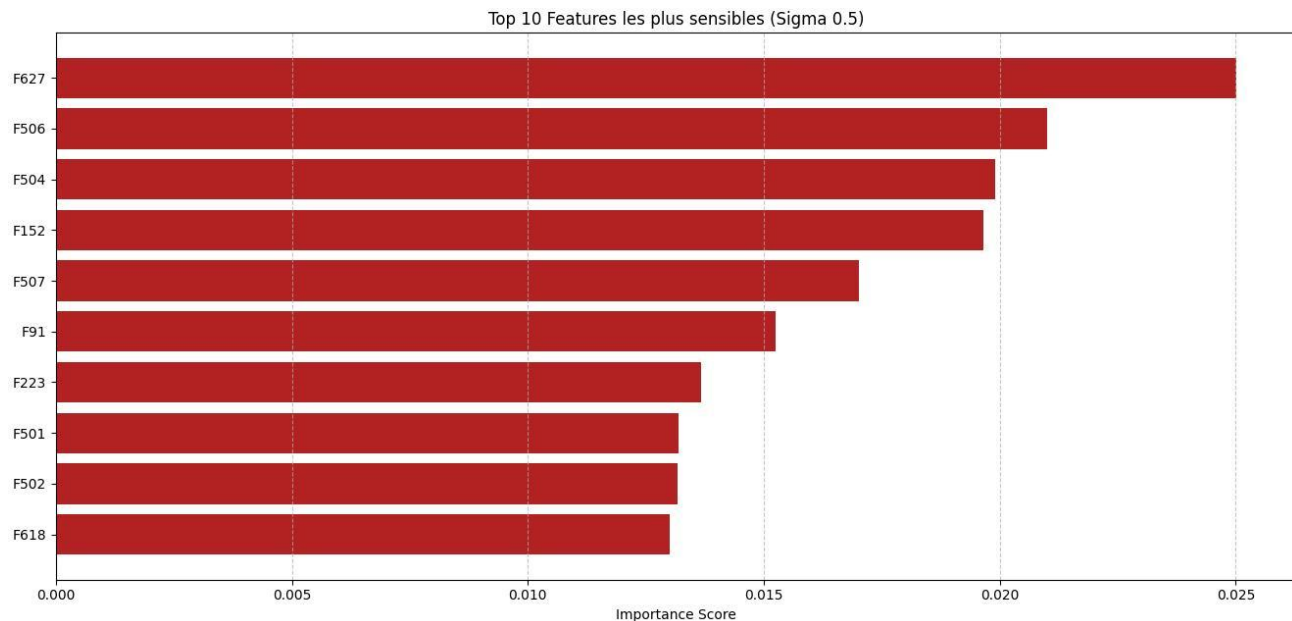


Figure 4: Top 10 des features les plus sensibles face au bruit gaussien

4. Recommandations

- **Formation à la cybersécurité :** Sensibiliser sur le fait que l'IA n'est pas infaillible.
- **Entraînement Adversarial :** Intégrer des échantillons bruités lors de l'entraînement pour renforcer le modèle.
- **Simulations régulières :** Tester périodiquement la robustesse du modèle face à de nouvelles formes de bruit.

5. Conclusion

Cette étude met en évidence la vulnérabilité des modèles de détection de malwares face aux attaques adversariales simples.

Elle souligne l'importance d'intégrer des mécanismes de défense, tels que l'entraînement adversarial, afin d'améliorer la robustesse des systèmes de cybersécurité. montre que les attaques exploitant l'**urgence restent très efficaces**. Le projet souligne l'importance d'une approche combinant **sensibilisation humaine et mesures techniques** pour renforcer la sécurité globale.