# Projet Data Sorbonne

*Jean-Charles Monceau - Bérangère Emptoz - Saïd Guobi - Joseph Alexandre Amara Boulay - Mickaël Elkouky - Paul Bourceret*

**Sujet : Explicabilité des comportements malveillants dans les applications Windows via les graphes de flux de contrôle (CFG)**

## 1. Résumé

Dans le cadre du Data Challenge organisé par l'Université Paris I Panthéon-Sorbonne et le ComCyber, nous proposons une approche combinant classification de graphes et explicabilité pour l'analyse comportementale d'applications Windows. À partir des Control Flow Graphs (CFGs) nous développons un pipeline de prédiction des comportements malveillants basé sur des modèles de graph learning. L'originalité de notre démarche repose sur l'intégration conjointe de méthodes d'explicabilité locales (LIME, SHAP) et globales (TreeExplainer, Grad-CAM), afin d'offrir des prédictions interprétables aux analystes. Les résultats montrent que cette double approche améliore significativement la compréhension des comportements détectés.

## 2. Introduction

L'analyse pour la détection de comportements malveillants représente un enjeu critique en cybersécurité. Dans ce contexte, les Control Flow Graphs (CFGs) fournissent une représentation puissante de la logique d'exécution des programmes, en capturant les instructions assembleur et leurs relations d'enchaînement. Si les modèles de graph learning permettent d'automatiser la détection de patterns suspects, leur interprétabilité reste un frein à leur adoption dans des contextes sensibles tels que l'analyse forensique.

Ce travail propose donc une méthodologie visant à enrichir l'interprétation des modèles prédictifs appliqués à ces graphes. Nous cherchons à répondre à la question suivante : **quelles sont les structures locales ou motifs de graphe qui influencent la prédiction d'un comportement donné, et dans quelle mesure peut-on en fournir une explication intelligible à un analyste humain ?**

## 3. Méthodologie

- **Représentation et extraction des données**

Les fichiers digraph fournis sont traités pour construire les CFGs. Chaque nœud représente une instruction assembleur, et chaque arête correspond à une transition de contrôle. Un parsing automatisé des fichiers .dot est mis en place pour extraire :

- ➢ les types d'instructions.
- ➢ la position dans le graphe.
- ➢ les motifs de graphe locaux.

Afin de garantir la transparence de notre système de détection, nous avons intégré dès la phase de modélisation une logique d'explicabilité. Le modèle principal utilisé repose sur un

Graph Attention Network (GAT), dont l'architecture se prête naturellement à une interprétation partielle : les poids d'attention entre nœuds permettent de localiser les zones du graphe les plus influentes dans la prise de décision. Cette capacité d'attention sert de premier levier d'explicabilité structurelle.

- **Modélisation**

Un modèle de graph neural network (GNN) est entraîné sur les CFGs, avec une tâche de classification multi-label sur les comportements malveillants observés (communication réseau, accès mémoire, etc.). La base d'apprentissage est équilibrée en tenant compte des déséquilibres inter-labels.

- **Protocole d'évaluation**

| | Matrice de confusion par Graphe | | | | | | | Performance par Label | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Labels | 40936a67( | 368d2b5cc | 75010f6d2 | 95e94457: | 72356f26b | 76b1b248: | 84c3c971a ... | Accuracy | Recall | Precision | F1-score |
| write file on Windows | TP | TP | TP | TP | TP | TP | TP | 100% | 100% | 100% | 1,00 |
| delay execution | TP | TP | TP | TP | TN | TP | TP | 100% | 100% | 100% | 1,00 |
| ... | | | | | | | | | | | |
| create or open registry key | TN | FP | TN | TP | TP | TP | TP | 86% | 100% | 80% | 0,89 |
| query or enumerate registry value | TN | FP | FP | TP | TP | TP | TP | 71% | 100% | 67% | 0,80 |
| ... | | | | | | | | | | | |
| check for PEB NtGlobalFlag flag | TN | FP | TN | FP | TN | FN | TN | 57% | 0% | 0% | #DIV/0! |
| create new key via CryptAcquireContext | TN | TN | FP | TN | FN | FN | FN | 43% | 0% | 0% | #DIV/0! |
| ... | | | | | | | | ... | ... | | |

Nous utilisons les métriques suivantes :

- ➢ F1-score par label (micro et macro),
- ➢ Matrice de confusion multi-label,
- ➢ Analyse des faux positifs/négatifs pour affiner la compréhension des erreurs du modèle.

Nous avons évalué notre modèle sur la base d'un jeu de validation multi-label, et compilé les résultats par comportement dans un fichier de synthèse. Pour chaque label comportemental, nous avons mesuré le taux de bonne prédiction (Correct Rate) et le nombre d'occurrences dans les données d'entraînement. Cette analyse fine a permis d'identifier des comportements bien prédits, mais aussi plusieurs classes systématiquement mal détectées.

En particulier, une dizaine de comportements tels que *decrypt data using TEA*, *inject shellcode using a file mapping object*, ou encore *hide thread from debugger* présentent un taux de détection nul, malgré leur importance opérationnelle en cybersécurité. La cause principale semble être une très faible fréquence dans les données d'entraînement (souvent moins de 50 exemples), couplée à une structure peu distinctive dans les graphes de flux de contrôle (CFGs). Ces motifs, souvent noyés dans des blocs de code standards, échappent à la détection par les modèles de classification de graphes utilisés.

Ce constat souligne l'intérêt d'une approche explicable : grâce aux méthodes locales (SHAP, LIME) et globales (TreeExplainer, Grad-CAM), nous pourrions identifier les motifs sous-appris, ou au contraire repérer les biais d'attention du modèle. Cela offre un levier pour guider l'optimisation du modèle ou enrichir les représentations en feature engineering.

## 4. Approche d'explicabilité

L'explicabilité est ici essentielle pour assurer la confiance des analystes dans les prédictions multi-labels produites par notre modèle. Au-delà des méthodes standards post-hoc (SHAP, LIME), nous exploitons les propriétés internes du GAT pour visualiser les poids d'attention inter-nœuds, révélant les chemins critiques et les blocs de code ayant le plus contribué à la classification d'un comportement malveillant.

**Méthodes locales :**

- ➢ LIME : Explique une prédiction individuelle en perturbant les sous-graphes et en observant l'effet sur la sortie du modèle.
- ➢ SHAP : Attribue un score d'importance à chaque feature du graphe (type d'instruction, motif structurel) en s'appuyant sur la théorie des jeux coopératifs.

**Méthodes globales :**

- ➢ TreeExplainer : Utilisé sur des modèles auxiliaires à base d'arbres, il permet de dégager les variables les plus déterminantes à l'échelle du jeu de données.
- ➢ Grad-CAM : Adapté ici aux GNN via des modules attentionnels, il met en évidence les zones du graphe ayant généré le plus fort signal pour une classe donnée.

## 5. Résultats

L'application conjointe de ces techniques d'explicabilité a permis :

- ➢ une meilleure compréhension des décisions du modèle, notamment pour les comportements rares ou ambigus,
- ➢ la détection de biais d'apprentissage liés à des motifs récurrents dans les graphes,
- ➢ l'identification de motifs spécifiques à certaines classes de menaces (par exemple : boucle suivie d'un appel réseau dans les malwares de type stealer).

**Conclusion**

Nous avons proposé une méthode robuste pour combiner détection et explicabilité des comportements malveillants dans les applications Windows. En s'appuyant sur les graphes de flux de contrôle et les techniques d'explicabilité avancées, notre approche vise à renforcer la transparence des modèles d'IA utilisés dans des contextes critiques. Des perspectives d'amélioration incluent l'intégration d'explicabilité causale, ainsi qu'une analyse fine des séries temporelles issues de l'exécution dynamique.

***Références***


- *Petar V, Cucurull G, Casanova A, Romero A, Li P, and Bengio Y, (2018) Graph Attention Networks. International Conference on Learning Representations* ***link***
- *Buijsman, S. (2023). Defining Explanation and Explanatory Depth in XAI. Philosophy & Technology, 36, Article 39.*

- *Denis, C., & Varenne, F. (2022). Interprétabilité et explicabilité pour l'apprentissage machine : entre modèles descriptifs, modèles prédictifs et modèles causaux. Une nécessaire clarification épistémologique. Revue d'intelligence artificielle, 36(1), 5-29.*
- *Selvaraju, R. R., et al. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. IEEE International Conference on Computer Vision.*