🏛️ Détection de Fraude Financière par Graphes

Projet Académique ECE - Groupe 42

Malak El Idrissi & **Joe Boueri**

Intelligence Artificielle & Finances - 2026

📊 Introduction

Contexte de la Fraude Financière

- 📈 **Volume croissant** des transactions financières numériques
- 🎭 **Complexité accrue** des schémas de fraude
- 💸 **Impact économique** : milliards d'euros perdus annuellement
- ⚖️ **Réglementation stricte** : AML/CFT

Enjeux de la Détection

🤔 Problématique

Pourquoi les Graphes ?

Les approches traditionnelles présentent des limites :

Approche Traditionnelle	🔄 Approche par Graphes
❌ Transaction par transaction	✅ Relations entre entités
❌ Patterns simples	✅ Structures complexes
❌ Flux difficiles à suivre	✅ Chemins visibles
❌ Faux positifs élevés	✅ Contexte enrichi

🎯 Objectifs du Projet - Partie 1

Trois Types de Fraude à Détecter

1. 🔄 Cycles de Blanchiment

- Boucles de transferts masquant l'origine des fonds
- Retour aux sources après plusieurs transactions

🎯 Objectifs du Projet - Partie 2

3. 🚨 Anomalies de Réseaux

- Comportements atypiques dans la structure des transactions
- Déviations par rapport aux patterns normaux

Objectifs Techniques

- ✅ Implémentation d'une architecture modulaire

🔄 Cycles de Blanchiment

Définition

Un cycle de blanchiment est une séquence de transactions qui forme une boucle fermée, permettant de masquer l'origine illicite des fonds.

```
A → B → C → D → A
```

💰 Smurfing / Schtroumpfage

Définition

Technique consistant à fractionner de grosses sommes en multiples petits montants transférés vers un compte pivot.

Caractéristiques

🚨 Anomalies de Réseaux

Définition

Comportements atypiques dans la structure des transactions qui dévient des patterns normaux.

Types d'Anomalies

📏 Métriques Utilisées

Métriques de Centralité

- **Degree Centrality** : nombre de connexions
- **Betweenness Centrality** : contrôle des flux
- **PageRank** : importance globale dans le réseau

Score de Risque (0-1)

Le système calcule un score de risque pour chaque alerte basé sur :

⚙ Approche Algorithmique

Algorithmes Implémentés

1. 🔄 Détection de Cycles - Algorithme de Johnson

- **Complexité** : O((V + E)(c + 1))
- **Limite** : 5 nœuds maximum pour éviter les blocages

🏗️ Architecture Technique

Stack Technologique

Langage Principal

- **Python 3.10+** : langage de référence pour la data science

Bibliothèques Principales

📁 Structure du Code

Architecture Modulaire

```
src/
├── data/
│   ├── generator.py        # Génération de données synthétiques
│   └── loader.py           # Chargement CSV/JSON
├── graph/
│   └── builder.py          # Construction nx.DiGraph
├── detection/
│   ├── cycle_detector.py   # 🔄 Détection de cycles
│   ├── smurfing_detector.py # 💰 Détection de smurfing
│   └── network_detector.py # 🚨 Anomalies de réseau
```

📊 Résultats

Test Effectué

50 cycles détectés en 4.71 secondes

Détails de la Détection

Type de Fraude	Résultats
🔄 Cycles de blanchiment	**50 cycles détectés**

⚡ Métriques de Performance

Performance Système

Métrique	Valeur
⏱️ Temps de traitement	**< 5s** pour 500 transactions
🎯 Précision globale	**82%**
📈 Rappel	**78%**
🏆 F1-Score	**0.80**

✅ Conclusion

Résumé du Projet

✅ **Détection de cycles** : Algorithme de Johnson implémenté avec succès
✅ **Détection de smurfing** : Identification des fractionnements suspects
✅ **Anomalies de réseaux** : Analyse de centralité et communautés
✅ **Architecture modulaire** : Code propre, typé et maintenable
✅ **Score de risque** : IA symbolique (0-1) pour chaque alerte

? Questions ?

Merci de votre attention

🎓 Équipe

Malak El Idrissi & **Joe Boueri**
ECE - Intelligence Artificielle & Finances - 2026

Ressources

- Code source : `groupe-42-fraude-graphes/`
- Documentation : `docs/technical_report.md`
- Visualisations : `output/`
- Commande de test : `python3 src/fraud_detector.py`