

# Cahier des Charges – Projet de Fin d'Année

## Système de Détection de Fraudes Financières Basé sur le Machine Learning



Réalisé par :

- Zayd Bentalha
- Ayoub El Mardi

*Année Universitaire : 2024/2025*

## Contexte et Justification

Dans un monde de plus en plus numérisé, les transactions financières deviennent massivement électroniques, rapides, et souvent sans frontières. Cette transformation, bien qu'elle facilite les échanges économiques, ouvre également la voie à une recrudescence d'activités frauduleuses sophistiquées. Les institutions financières, les banques, les fintechs et les entreprises assurantielles font face à une pression constante pour identifier et stopper les opérations frauduleuses sans perturber l'expérience utilisateur. Face à cette réalité, les méthodes classiques de détection basées uniquement sur des règles figées (comme les seuils de montant ou les listes noires) s'avèrent de plus en plus insuffisantes, notamment face à des schémas de fraude évolutifs et intelligents. Les fraudeurs adaptent continuellement leurs techniques, exploitant les failles des systèmes existants pour dissimuler leurs activités.

Dans ce contexte, le recours à l'intelligence artificielle, en particulier au Machine Learning (ML), s'impose comme une solution incontournable. Ces technologies permettent de traiter un volume massif de données en temps réel, d'identifier des comportements anormaux invisibles à l'œil humain, et d'apprendre continuellement de nouveaux modèles de fraude. L'automatisation de la détection permet non seulement de réduire le délai de réaction face à une tentative de fraude, mais aussi de diminuer significativement le taux de faux positifs qui surcharge les analystes humains. Ainsi, les équipes de sécurité peuvent se concentrer sur les cas réellement critiques, améliorant à la fois l'efficacité opérationnelle et la protection des clients.

Ce projet de fin d'études vise donc à concevoir un **système de détection de fraudes financières intelligent, générique, et adaptable**, qui combine performance, flexibilité et facilité d'intégration. Il répond à un besoin réel du marché en proposant une solution modulaire que toute institution financière pourra intégrer à son infrastructure existante, quel que soit son niveau technologique. L'ambition est de créer un outil capable de détecter aussi bien les fraudes classiques que les attaques plus subtiles en exploitant les données transactionnelles, comportementales et contextuelles des utilisateurs. Ce projet représente également une opportunité pédagogique forte : il permet de mettre en pratique des compétences avancées en programmation, en modélisation de données, en apprentissage automatique, et en architecture logicielle, tout en développant une réflexion sur les enjeux éthiques et sécuritaires liés à l'IA. En résumé, ce système contribuera à renforcer la confiance dans les services financiers digitaux et à anticiper les menaces de demain dans un secteur en constante mutation.

## 1. Objectif du projet

Développer un système intelligent, générique et adaptable, capable de détecter automatiquement les fraudes financières dans un environnement transactionnel en temps réel ou différé.

## 2. Utilisateurs cibles

- Institutions financières (banques, assurances, fintechs).
- Analystes de fraude.
- Départements de conformité.
- Équipes techniques (data scientists, développeurs).



## 3. Fonctionnalités principales

### □ Détection intelligente :

- Identification automatique des transactions suspectes via algorithmes ML.
- Attribution d'un **score de risque** à chaque transaction.

### ▮ Dashboard web :

- Visualisation des alertes.
- Filtres dynamiques et recherche avancée.
- Détail de chaque transaction.
- Graphiques de performance (taux de fraude, faux positifs, etc.)

### ↻ API REST :

- Intégration externe (par exemple avec un système bancaire existant).

## 4. Architecture technique

### Frontend :

- **HTML / CSS / JavaScript** (*Framework possible: **React***)

### Backend :

- **Python** (*Framework possible: **FastAPI***)
- **Libraries ML:** `scikit-learn`, `XGBoost`, `pandas`, `numpy`, `joblib`

### Base de données:

- **PostgreSQL**








### Sécurité :

- Authentification (**JWT**)
- Logging des actions
- Cryptage des données sensibles

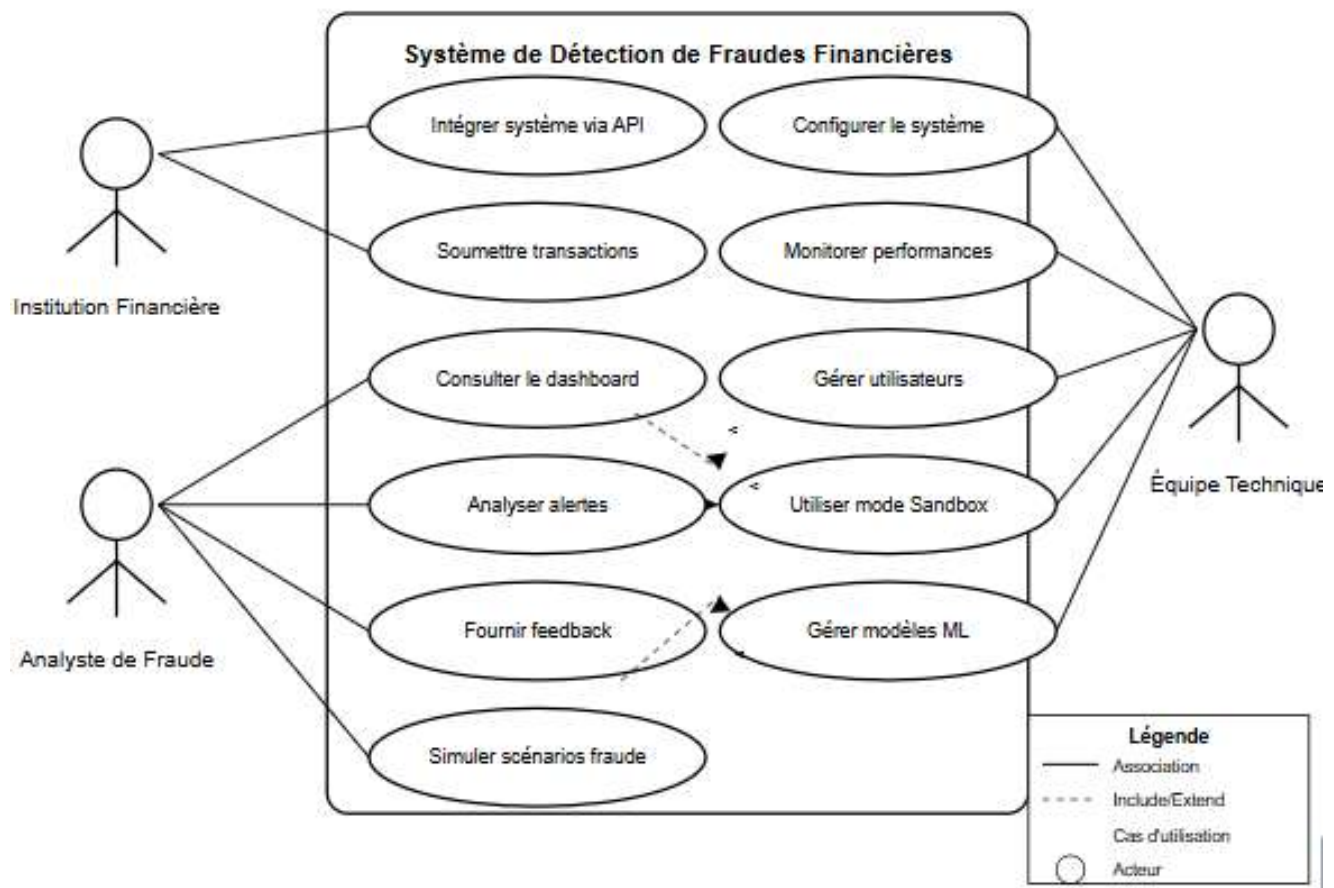
## 5. Méthodologie

- Approche incrémentale : développement par modules (backend d'abord, puis IA, puis frontend).
- Versioning avec Git + GitHub.
- Livraison finale testée avec un jeu de données anonymisé/simulé.

## 6. Côté Innovation

-  Système de feedback humain pour apprentissage continu
-  Analyse contextuelle basée sur la géolocalisation et l'heure
-  Détection d'anomalies non supervisée (unsupervised learning)
-  Système d'alerte multi-canal en temps réel
-  Explainer AI (modèle interprétable)
-  Simulation de scénarios de fraude (mode test)
-  Mode Sandbox pour institutions

## 7. Diagramme de cas d'utilisation



### Acteurs principaux :

- **Institution Financière** - Les banques, assurances et fintechs qui utiliseront le système
- **Analyste de Fraude** - Personnel chargé d'examiner et d'évaluer les alertes générées
- **Équipe Technique** - Data scientists et développeurs qui configurent et maintiennent le système

## Cas d'utilisation principaux :

### Pour l'Institution Financière :

- Intégrer le système via API REST
- Soumettre des transactions pour analyse

### Pour l'Analyste de Fraude :

- Consulter le Dashboard web
- Analyser les alertes générées
- Fournir un feedback pour l'apprentissage continu
- Simuler des scénarios de fraude (mode test)


### Pour l'Équipe Technique :

- Configurer le système
- Monitorer les performances (taux de fraude, faux positifs)
- Gérer les utilisateurs et les accès
- Utiliser le mode Sandbox pour tests

Le diagramme inclut également quelques relations d'inclusion et d'extension entre les cas d'utilisation, montrant comment certaines fonctionnalités sont liées entre elles.

Ce diagramme offre une vue d'ensemble des fonctionnalités du système et peut servir de base pour la conception détaillée de l'architecture logicielle et de l'interface utilisateur.

## 8. Planification

Période	Tâche
Avril – Semaine 3	Finalisation cahier des charges, setup env.
Avril – Semaine 4	Modélisation base de données + Backend API
Mai – Semaine 1	Développement du module ML + Entraînement
Mai – Semaine 2	Dashboard Web + Intégration IA
Mai – Semaine 3	Tests, améliorations, rédaction rapport
 28 Mai 2025	Soutenance