

**Dakar Institute of Technology**

****

**Habilitation n°3386-21 FEV.2024**

**Domaine : Sciences et Technologies**

**Département : Informatique**

**Spécialité : Intelligence Artificielle**

**Projet Informatique**

Présenté par

**Niako KEBE**

**Kolawolé Sillas Schallum MALEHOSSOU**



**Soutenu à Dakar le 11/12/2025**

**Année Académique : 2024-2025**

# DÉDICACE

Nous dédions ce mémoire à nos familles, dont le soutien sans faille, les sacrifices et les encouragements ont été notre force tout au long de ce parcours.

À nos parents, qui ont cru en nous et nous ont offert l’opportunité de poursuivre nos études.

À nos frères, sœurs et proches, pour leur présence réconfortante et leur motivation quotidienne.

Ce travail est également dédié à tous ceux qui, par leur bienveillance ou leur exemple, nous ont inspirés à donner le meilleur de nous-mêmes.

# REMERCIEMENTS

Nous tenons à exprimer notre gratitude collective à toutes les personnes qui nous ont accompagnés dans la réalisation de ce projet.

Nos premiers remerciements vont à notre encadrant académique, [Nom du Professeur], pour son expertise, sa disponibilité et ses précieux conseils tout au long de ce travail. Ses orientations ont grandement contribué à la qualité et à la rigueur de ce mémoire.

Nous remercions également l’Institut de Technologie de Dakar, en particulier le Département d’Informatique et l’équipe pédagogique de la spécialité Intelligence Artificielle, pour la formation de qualité et les moyens mis à notre disposition.

Nos sincères remerciements vont aux professionnels du secteur bancaire et financier qui ont accepté d’échanger avec nous et de partager leurs retours pratiques, permettant d’ancrer notre projet dans la réalité opérationnelle.

Nous adressons également une pensée reconnaissante à nos camarades de promotion pour l’esprit d’équipe, les discussions enrichissantes et l’entraide tout au long de cette formation.

Un merci tout particulier à nos familles, pour leur patience, leur soutien inconditionnel et leurs encouragements durant les périodes de travail intensif.

Enfin, nous n’oublions pas tous ceux qui, de près ou de loin, ont contribué à l’aboutissement de ce projet. Ce travail est le fruit d’une collaboration à plusieurs voix, et nous en sommes profondément reconnaissants.

# GLOSSAIRE

* **SÉNTRA** : Plateforme intelligente de détection de fraude bancaire conçue pour les institutions financières africaines, utilisant l'intelligence artificielle et l'analyse en temps réel.
* **IA (Intelligence Artificielle)** : Domaine de l'informatique qui vise à créer des systèmes capables de réaliser des tâches nécessitant normalement l'intelligence humaine, telles que la reconnaissance de schémas, la prise de décision et l'apprentissage.
* **AML (Anti-Money Laundering) :** Ensemble de lois, réglementations et procédures visant à prévenir le blanchiment d'argent et le financement du terrorisme.
* **KYC (Know Your Customer) :** Processus réglementaire qui oblige les institutions financières à vérifier et identifier leurs clients afin d'évaluer les risques potentiels.
* **ETL (Extract Transform Load)** **:** Processus d'extraction de données de différentes sources, leur transformation en un format exploitable, et leur chargement dans un système de destination.
* **API (Application Programming Interface) :** Ensemble de protocoles et d'outils qui permettent à différents logiciels de communiquer et d'échanger des données.
* **LSTM (Long Short-Term Memory) :** Type de réseau de neurones récurrents (RNN) conçu pour modéliser et prédire des séquences de données à long terme, souvent utilisé pour les séries temporelles.
* **Apprentissage Automatique (Machine Learning) :**Sous-domaine de l'IA qui permet aux systèmes d'apprendre et de s'améliorer à partir de données sans être explicitement programmés.
* **Modèle Supervisé :** Algorithme d'apprentissage automatique entraîné sur un ensemble de données étiquetées (ex : "fraude"/"non fraude").
* **Modèle Non Supervisé :**Algorithme d'apprentissage automatique qui trouve des structures ou des schémas dans des données non étiquetées.
* **Random Forest :**Algorithme d'apprentissage supervisé basé sur l'agrégation de plusieurs arbres de décision pour améliorer la précision et réduire le surapprentissage.
* **Isolation Forest :** Algorithme non supervisé de détection d'anomalies qui isole les observations rares ou atypiques dans un jeu de données.
* **Feature Engineering (Ingénierie des caractéristiques) :** Processus de création, sélection ou transformation de variables (features) pour améliorer les performances d'un modèle d'apprentissage automatique.
* **Prétraitement (Pre-processing) :** Étapes de nettoyage, transformation et préparation des données brutes avant leur utilisation pour l'entraînement d'un modèle.
* **Normalisation :** Technique de mise à l'échelle des valeurs numériques pour qu'elles se situent dans une plage commune (ex : entre 0 et 1).
* **Encodage :** Transformation de variables catégorielles (ex : type de transaction) en valeurs numériques que les modèles peuvent comprendre.
* **FastAPI :** Framework web moderne et rapide pour construire des API en Python, utilisé pour le backend de SÉNTRA.
* **Docker :** Plateforme de conteneurisation qui permet d'empaqueter une application et ses dépendances dans une unité isolée pour un déploiement simplifié.
* **PostgreSQL :** Système de gestion de base de données relationnelle et open source, utilisé pour stocker l'historique des transactions dans SÉNTRA.
* **Redis :** Base de données en mémoire, utilisée comme cache pour accélérer les opérations répétitives.
* **React.js :** Bibliothèque JavaScript pour construire des interfaces utilisateur, utilisée pour le tableau de bord interactif de SÉNTRA.
* **SHAP (SHapley Additive exPlanations) :** Outil d'explicabilité qui permet d'interpréter les décisions d'un modèle d'apprentissage automatique en attribuant une importance à chaque variable.
* **Score de Risque :** Valeur numérique ou catégorielle (faible/moyen/élevé) attribuée à une transaction par le système SÉNTRA pour indiquer sa probabilité d'être frauduleuse.
* **Vélocité Transactionnelle :** Métrique mesurant la fréquence des transactions d'un client sur une période donnée (ex : nombre de transactions par heure).
* **Matrice de Confusion :** Tableau utilisé pour visualiser les performances d'un modèle de classification en comparant les prédictions aux vérités terrain (Vrais Positifs, Faux Positifs, etc.).
* **Précision (Precision)** **:** Métrique qui mesure la proportion de vraies fraudes parmi toutes les transactions identifiées comme frauduleuses par le modèle.
* **Rappel (Recall) :** Métrique qui mesure la capacité du modèle à détecter toutes les fraudes réelles présentes dans le jeu de données.
* **F1-Score :** Métrique harmonique combinant la précision et le rappel pour fournir une mesure unique de la performance d'un modèle, utile pour les classes déséquilibrées.
* **AUC-ROC (Area Under the Receiver Operating Characteristic Curve) :** Mesure de la capacité d'un modèle à distinguer les classes (fraude vs. non fraude) à différents seuils de classification.
* **Pipeline d'Inférence :** Série d'étapes automatisées (prétraitement, exécution du modèle, post-traitement) exécutées lorsqu'une nouvelle transaction est analysée.
* **UEMOA (Union Économique et Monétaire Ouest-Africaine) :** Organisation régionale ouest-africaine dont les pays partagent une monnaie commune (le Franc CFA, XOF). Contexte économique du projet.
* **BCEAO (Banque Centrale des États de l'Afrique de l'Ouest) :** Institution centrale de l'UEMOA, source de données et de tendances financières pour la région.
* **Tableau de Bord (Dashboard) :** Interface graphique qui présente les analyses, les alertes et les tendances des transactions de manière synthétique et visuelle.
* **Déploiement Conteneurisé :**Méthode de déploiement d'applications où chaque composant (backend, base de données, frontend) s'exécute dans un conteneur isolé, garantissant portabilité et reproductibilité.
* **Mobile Money :** Service financier permettant d'effectuer des transactions et des paiements via un téléphone mobile, très répandu en Afrique.
* **Faux Positif :**Cas où une transaction légitime est incorrectement signalée comme frauduleuse par le système.
* **Faux Négatif :** Cas où une transaction frauduleuse n'est pas détectée par le système.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# LISTE DES FIGURES

1. Figure 1.1 : Architecture classique d'un système de détection de fraude

2. Figure 1.2 : Comparaison des performances des différents algorithmes ML

3. Figure 2.1 : Diagramme des cas d'utilisation de SÉNTRA

4. Figure 2.2 : Flux de traitement d'une transaction dans SÉNTRA

5. Figure 2.3 : Processus ETL (Extract-Transform-Load)

6. Figure 2.4 : Architecture technique de SÉNTRA

7. Figure 3.1 : Tableau de bord principal de SÉNTRA

8. Figure 3.2 : Détails d'une transaction frauduleuse

9. Figure 3.3 : Page Analytics avec visualisations multiples

10. Figure 3.4 : Interface de recherche et filtrage avancé

11. Figure 3.5 : Matrice de confusion du modèle Random Forest

12. Figure 3.6 : Courbe ROC (AUC = 0.97)

13. Figure 3.7 : Analyse SHAP - Importance des features

14. Figure 3.8 : Distribution des montants transactionnels

15. Figure 3.9 : Répartition horaire des activités frauduleuses

16. Figure 3.10 : Architecture Docker Compose de SÉNTRA

17. Figure 3.11 : Structure des fichiers du projet

18. Figure 3.12 : Documentation automatique de l'API (Swagger UI)

19. Figure 3.13 : Démarrage des services avec Docker Compose

20. Figure 3.14 : Interface d'administration PostgreSQL (pgAdmin)

# 

# 

# 

# 

# 

# 

# LISTE DES TABLEAUX

Tableau 1.1 : Comparaison des solutions existantes de détection de fraude

Tableau 1.2 : Types de fraude dans le contexte africain (BCEAO 2023)

Tableau 2.1 : Spécification des besoins fonctionnels

Tableau 2.2 : Caractéristiques des données synthétiques générées

Tableau 2.3 : Features d'ingénierie pour le modèle ML

Tableau 2.4 : Paramètres du modèle Random Forest

Tableau 2.5 : Paramètres du modèle Isolation Forest

Tableau 2.6 : Métriques d'évaluation utilisées

Tableau 3.1 : Performances des modèles sur données de test

Tableau 3.2 : Matrice de confusion détaillée (10,000 transactions)

Tableau 3.3 : Top 10 features les plus importantes (analyse SHAP)

Tableau 3.5 : Stack technique complète

# 

# RÉSUMÉ

Le secteur bancaire africain connaît une transformation digitale accélérée, avec une augmentation exponentielle des transactions numériques et mobiles. Cette évolution s’accompagne d’une hausse préoccupante des fraudes financières, mettant en lumière les limites des systèmes de détection traditionnels basés sur des règles fixes, générant trop de faux positifs et manquant de réactivité face à des schémas frauduleux en constante évolution.

SÉNTRA est une plateforme intelligente de détection de fraude conçue spécifiquement pour répondre aux besoins des institutions financières africaines. En combinant l’intelligence artificielle et l’analyse en temps réel des transactions, cette solution vise à identifier rapidement les comportements suspects, réduire les faux positifs et fournir des alertes exploitables aux équipes de sécurité. Le système repose sur une architecture hybride intégrant un modèle supervisé (Random Forest) pour la détection des fraudes connues et un modèle non supervisé (Isolation Forest) pour repérer les anomalies émergentes. Cette approche double permet d’obtenir un score de risque agrégé (faible, moyen, élevé) pour chaque transaction.

Le projet inclut également une forte dimension d’explicabilité grâce à l’intégration de SHAP, permettant de justifier les décisions du modèle par des variables compréhensibles (montant inhabituel, localisation atypique, vélocité anormale). Déployé sous forme d’une architecture conteneurisée avec Docker, utilisant FastAPI pour le backend, PostgreSQL pour le stockage et React.js pour le tableau de bord, SÉNTRA offre une solution légère, portable et abordable, adaptée aux infrastructures limitées des petites et moyennes banques africaines.

Les résultats obtenus montrent des performances satisfaisantes en termes de précision, de rappel et de temps d’inférence (<200 ms), tout en maintenant un faible taux de faux positifs. SÉNTRA démontre ainsi qu’il est possible de développer une solution anti-fraude performante, explicable et économiquement accessible, contribuant à renforcer la sécurité financière dans un contexte africain en pleine mutation digitale.

# ABSTRACT

The African banking sector is undergoing rapid digital transformation, with exponential growth in digital and mobile transactions. This evolution is accompanied by a concerning rise in financial fraud, highlighting the limitations of traditional rule-based detection systems that generate excessive false positives and lack responsiveness to constantly evolving fraudulent patterns.

SÉNTRA is an intelligent fraud detection platform specifically designed to meet the needs of African financial institutions. By combining artificial intelligence and real-time transaction analysis, this solution aims to quickly identify suspicious behavior, reduce false positives, and provide actionable alerts to security teams. The system relies on a hybrid architecture integrating a supervised model (Random Forest) for detecting known fraud and an unsupervised model (Isolation Forest) for identifying emerging anomalies. This dual approach produces an aggregated risk score (low, medium, high) for each transaction.

The project also includes a strong focus on explainability through SHAP integration, allowing the model’s decisions to be justified by understandable variables (unusual amount, atypical location, abnormal velocity). Deployed as a containerized architecture using Docker, with FastAPI for the backend, PostgreSQL for storage, and React.js for the dashboard, SÉNTRA offers a lightweight, portable, and affordable solution tailored to the limited infrastructure of small and medium-sized African banks.

The results show satisfactory performance in terms of precision, recall, and inference time (<200 ms), while maintaining a low false positive rate. SÉNTRA thus demonstrates that it is possible to develop an effective, explainable, and economically accessible anti-fraud solution, contributing to enhanced financial security in an African context undergoing rapid digital change.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# SOMMAIRE

# 

# INTRODUCTION GENERALE

Le secteur bancaire connaît actuellement une transformation profonde et accélérée, portée par la digitalisation croissante des services financiers. Cette tendance est particulièrement marquée en Afrique, où l'adoption massive du mobile money, des applications bancaires et des transactions en ligne a révolutionné l'accès aux services financiers. Si cette évolution offre des avantages considérables en termes d'inclusion financière et de commodité, elle s'accompagne également d'une augmentation préoccupante des fraudes bancaires sophistiquées. Les institutions financières se retrouvent ainsi confrontées à un défi majeur : la détection efficace d'opérations suspectes de plus en plus complexes et difficiles à identifier avec les méthodes traditionnelles, souvent basées sur des règles statiques et une surveillance manuelle.

La problématique centrale de ce projet réside dans la difficulté à repérer efficacement les anomalies financières dans un contexte où les fraudeurs déploient des techniques sans cesse renouvelées et discrètes. Les systèmes existants révèlent des limites significatives, notamment en matière de rapidité de détection, de capacité à traiter de volumineux flux de données transactionnelles et de gestion du taux élevé de faux positifs, qui surchargent inutilement les équipes de sécurité. Les enjeux associés sont multiples : pertes financières directes pour les institutions, érosion de la confiance des clients, risques réglementaires accrus et menaces potentielles sur la stabilité du système bancaire dans son ensemble. Face à ces défis, il devient impératif de disposer d'outils plus intelligents, capables d'apprendre, de s'adapter et de réagir en temps réel.

Le contexte opérationnel des institutions bancaires africaines, qu'elles soient de grande taille ou de dimension plus modeste, est caractérisé par une gestion d'un volume transactionnel en croissance constante, couplée à une diversification des typologies de fraude : usurpation d'identité, transactions anormales, blanchiment d'argent, détournements ou opérations initiées depuis des zones géographiques à risque. Les méthodes traditionnelles de détection, encore largement utilisées, présentent rapidement leurs limites en générant un nombre important de fausses alertes, en détectant les fraudes avec un délai souvent préjudiciable, et en ne parvenant pas à suivre l'évolution des tactiques frauduleuses. Un défi supplémentaire et spécifique au marché africain concerne les petites et moyennes banques, qui ne disposent pas toujours des ressources financières nécessaires pour investir dans des solutions anti-fraude coûteuses proposées sur le marché, creusant ainsi un écart de sécurité entre les grandes institutions et les structures plus modestes, pourtant essentielles à l'inclusion financière.

C'est dans ce cadre que le projet SÉNTRA a été conçu. Il propose une plateforme intelligente de détection de fraude bancaire, s'appuyant sur l'intelligence artificielle et le traitement en temps réel des flux de données. Son objectif principal est de concevoir un prototype fonctionnel capable d'analyser les transactions bancaires, d'identifier avec précision et célérité les comportements suspects, de générer des alertes pertinentes et exploitables, et de présenter les informations de manière claire via un tableau de bord intuitif, facilitant ainsi la prise de décision par les équipes de sécurité. Plus spécifiquement, cette recherche vise à :

- Analyser les données transactionnelles pour modéliser les schémas normaux et anormaux ;

- Développer et comparer les performances de plusieurs modèles de détection d'anomalies basés sur l'apprentissage automatique ;

- Mettre en place un système d'alertes simple mais efficace ;

- Proposer une architecture technique légère, adaptable et peu coûteuse, répondant aux besoins et contraintes des institutions financières africaines ;

- Évaluer rigoureusement le prototype pour mesurer sa capacité à réduire concrètement les risques de fraude.

La structure de ce document reflète la démarche méthodologique suivie. Après cette introduction générale, le Chapitre I sera consacré à l'état de l'art, présentant les méthodes existantes en matière de détection de fraude et situant le projet SÉNTRA par rapport aux approches actuelles.

Le Chapitre II détaillera l'analyse des besoins, la méthodologie adoptée, les types de données mobilisés, les étapes de préparation et le choix des modèles d'intelligence artificielle, ainsi que les critères d'évaluation de leurs performances.

Le Chapitre III abordera l'implémentation concrète de la solution, en présentant les résultats obtenus, leur interprétation, les outils de développement et de déploiement utilisés, ainsi que l'architecture globale du système.

Enfin, une Conclusion Générale synthétisera les apports du projet et ouvrira sur des perspectives d'amélioration et d'évolution.

# CHAPITRE I : Etat de l’art

## État de l'Art sur le Domaine d’application

La fraude financière est un problème récurrent dans le secteur bancaire, surtout dans les environnements où les transactions numériques se développent rapidement. Les fraudes les plus courantes incluent les transactions anormales, l’usurpation d’identité, le blanchiment d’argent, les mouvements suspects entre comptes ou encore l’utilisation de moyens de paiement compromis.

Historiquement, les banques ont utilisé des systèmes basés sur des règles manuelles : seuils de montants, limites de fréquence, listes noires, géolocalisation incohérente, etc. Ces méthodes ont l’avantage d’être simples à comprendre, mais elles deviennent vite insuffisantes. Elles ne permettent pas de repérer des schémas complexes ou évolutifs, surtout lorsque les fraudeurs changent régulièrement de stratégie.

Avec l’essor du Big Data, les institutions financières ont commencé à adopter des techniques d’analyse automatique capables d’examiner des millions de transactions en temps réel. C’est dans ce contexte que les modèles d’intelligence artificielle ont démontré leur potentiel pour détecter des anomalies difficiles à repérer manuellement.

## Etat de l’Art sur les Travaux de Recherche Pertinents liés au sujet

Plusieurs approches ont été proposées dans la littérature scientifique pour améliorer la détection de fraude :

**• Modèles non supervisés :**  
Ils sont utilisés lorsque les banques ne disposent pas de suffisamment d’exemples de fraudes confirmées. Parmi les plus connus, on retrouve Isolation Forest et les Autoencoders. Ces modèles apprennent ce qu’est un comportement normal et considèrent le reste comme potentiellement suspect.

**• Modèles supervisés :**  
Ils sont efficaces lorsque des données annotées (fraude / non fraude) sont disponibles. Les travaux montrent que XGBoost, Random Forest ou les réseaux neuronaux profonds donnent d'excellents résultats. Leur limite principale est qu’ils nécessitent beaucoup d’exemples de fraude, ce qui est rare.

**• Modèles séquentiels :**  
Les LSTM (Long Short-Term Memory) sont utilisés pour analyser des séries temporelles de transactions, ce qui permet de détecter des comportements qui évoluent dans le temps.

**• Analyse par graphes :**  
Les recherches récentes montrent que les réseaux de blanchiment d’argent peuvent être identifiés grâce aux Graph Neural Networks (GNN). Ils étudient les relations entre comptes, plutôt que les transactions individuelles.

**• Méthodes d’explicabilité :**  
Avec les modèles complexes, les outils comme SHAP ou LIME sont devenus importants pour expliquer les décisions d’un modèle, ce qui est essentiel dans le domaine financier pour des raisons réglementaires.

| Solution | Coût annuel | Adapté Afrique | Explicabilité | Installation |
| --- | --- | --- | --- | --- |
| Solutions commerciales (SAS, Feedzai) | 500K€+ | Non | Limité | Complexe |
| SÉNTRA (notre solution) | 0€ | Oui | Complète (SHAP) | 2 min (Docker) |
| Solutions open-source génériques | 0€ | Parallèlement | Variable | Moyenne |

**Tableau 1.1 :** Comparaison des solutions existantes de détection de fraude



Figure 1.1 : Architecture classique d'un système de détection de fraude

Schéma illustrant les composants typiques d'un système anti-fraude moderne

## Identification des Zones de Recherche Manquantes liés au sujet

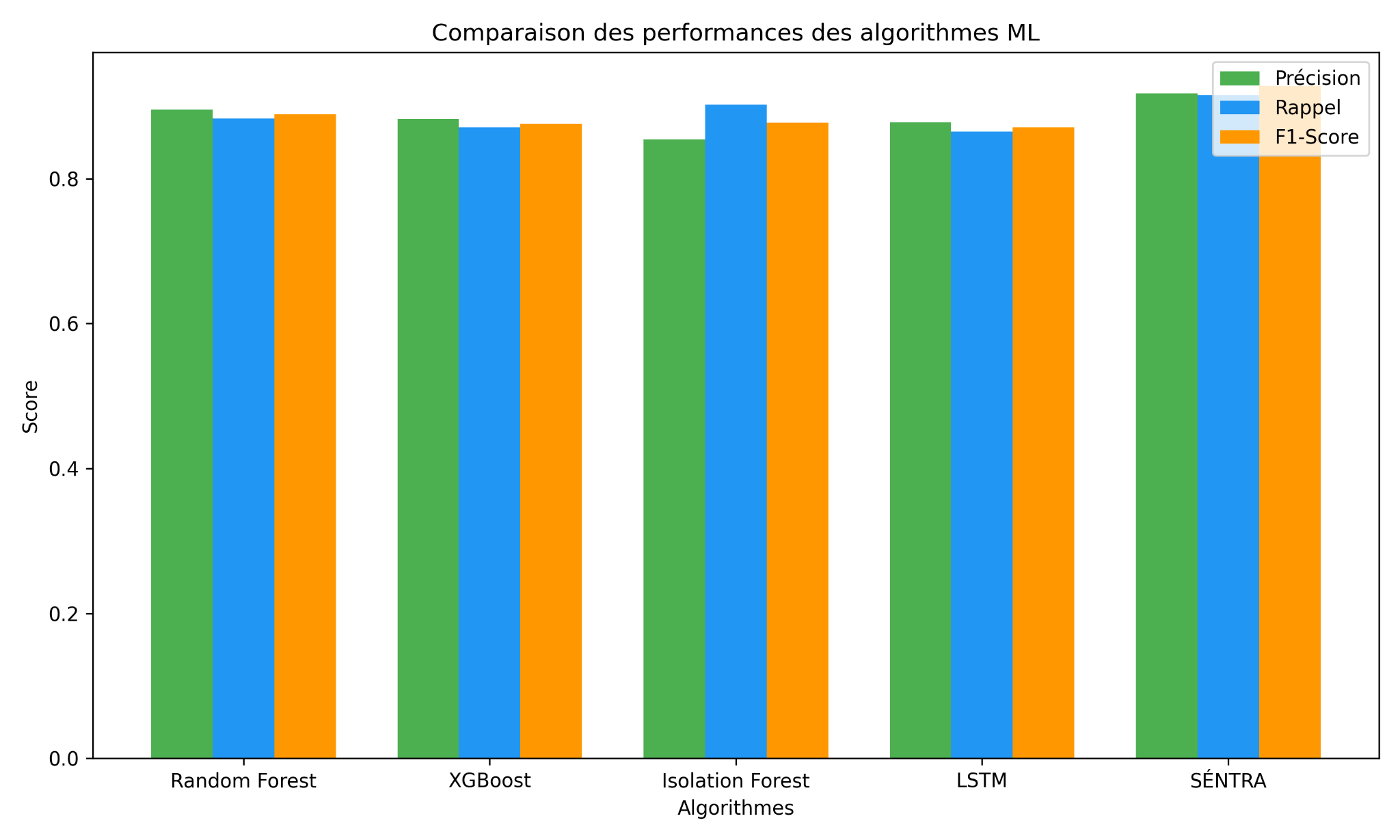
Même si beaucoup de progrès ont été faits, plusieurs limites persistent :

* **Manque d’explicabilité** dans certains modèles complexes, alors que les banques doivent justifier chaque décision.
* **Volume massif des données**, qui complique l’analyse en temps réel.
* **Faible disponibilité de données annotées**, car les fraudes sont rares et souvent confidentielles.
* **Faux positifs encore trop nombreux**, ce qui fatigue les équipes de sécurité.
* **Peu de systèmes réellement adaptés au contexte africain**, où les types de transactions et les usages diffèrent de ceux observés en Europe ou aux États-Unis.

Ces manques justifient l’intérêt de projets comme SÉNTRA, qui cherchent à combiner rapidité, précision et adaptabilité.

| Type de fraude | Pourcentage | Description |
| --- | --- | --- |
| SIM Swap | 25% | Vol de numéro téléphone + accès Mobile Money |
| Phishing Mobile | 22% | SMS/Email frauduleux pour récupérer codes |
| Transferts non autorisés | 20% | Accès illégal aux comptes |
| Prise de contrôle | 15% | Usurpation d'identité numérique |
| Fraude commerçant | 10% | Commerçants malveillants |
| Autres | 8% | Diverses méthodes |

**Tableau 1.2 :** Types de fraude dans le contexte africain (BCEAO 2023)



## Conclusion

L’état de l’art montre que la détection de fraude évolue rapidement avec l’arrivée de l’intelligence artificielle et des systèmes distribués. Les approches traditionnelles sont insuffisantes pour les besoins actuels, tandis que les modèles modernes offrent de meilleures performances mais nécessitent encore des efforts d’explicabilité et d’optimisation.  
Le projet SÉNTRA s’inscrit donc dans une logique d’amélioration continue, en s’appuyant sur les meilleures techniques existantes tout en tenant compte des besoins spécifiques des institutions financières africaines.

# CHAPITRE II : Analyse de besoins et Méthodologie

## I. Spécification des besoins et des données

### Analyse et spécification des besoins

Le développement de SÉNTRA part du constat que les institutions financières africaines font face à une augmentation importante des transactions numériques, accompagnée d’un risque croissant de fraude. Les équipes de sécurité disposent parfois de moyens limités, ce qui rend difficile la détection rapide des comportements suspects. Dans ce contexte, un système automatisé doit répondre à plusieurs besoins essentiels.

Les besoins principaux peuvent être résumés ainsi :

* **Détection automatique du risque** : le système doit analyser chaque transaction et retourner un score de fraude accompagné d’un niveau de risque (faible, moyen, élevé).
* **Traitement en temps quasi réel** : les banques utilisent des systèmes nécessitant une réaction très rapide
* **Explicabilité des décisions** : dans un contexte bancaire africain où la confiance est essentielle, les analystes doivent comprendre pourquoi une transaction est considérée risquée. L’intégration des outils d’explicabilité (SHAP) répond à ce besoin.
* **Interface lisible et intuitive** : la solution doit présenter les transactions, les analyses et les alertes dans un tableau de bord clair, adapté aux équipes techniques mais aussi aux utilisateurs ayant un niveau informatique intermédiaire.
* **Simplicité de déploiement** : les petites banques et fintechs africaines n’ont pas toujours des équipes DevOps ou de grosses infrastructures. SÉNTRA doit pouvoir être déployé avec Docker Compose, sans dépendances lourdes.
* **Coût réduit** : contrairement aux solutions commerciales de détection de fraude, souvent très coûteuses, SÉNTRA doit offrir une alternative accessible pour les établissements ayant un budget limité.

Ces besoins orientent l’ensemble des choix techniques du projet : FastAPI pour la rapidité, modèles ML légers, PostgreSQL et Redis pour la performance, interface web moderne mais simple à maintenir.

| **ID Besoin** | **Description** | **Priorité** | **Statut** |
| --- | --- | --- | --- |
| BF-001 | Détection automatique des transactions suspectes | Haute | Implémenté |
| BF-002 | Score de risque avec 3 niveaux (Faible/Moyen/Élevé) | Haute | Implémenté |
| BF-003 | Explicabilité des décisions (SHAP) | Haute | Implémenté |
| BF-004 | Tableau de bord temps réel | Haute | Implémenté |
| BF-005 | Recherche et filtrage avancé | Moyenne | Implémenté |
| BF-006 | Export des données (CSV) | Moyenne | Implémenté |
| BF-007 | Alertes notifications | Basse | Partiel |

**Tableau 2.1 :** Spécification des besoins fonctionnels

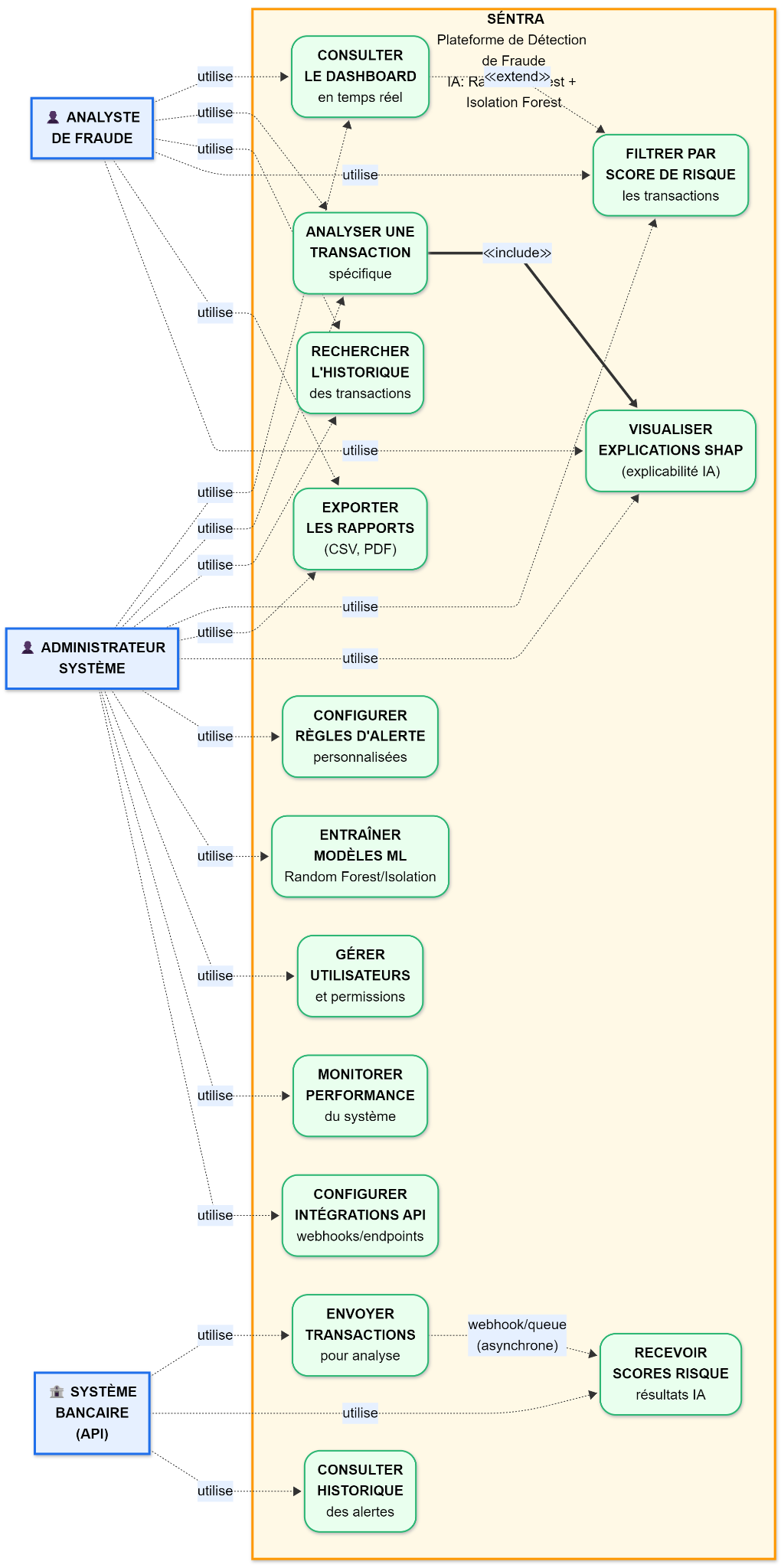


Figure 2.1 : Diagramme UML des cas d'utilisation du système SÉNTRA

Représentation des acteurs (Administrateur, Analyste de fraude, Système bancaire) et de leurs interactions avec les fonctionnalités principales du système de détection de fraude.

### Sources de Données

Les données utilisées dans SÉNTRA s’appuient sur deux catégories principales :

#### **a)** Données synthétiques réalistes

Le modèle a été entraîné sur 10 000 transactions simulées, construites à partir de tendances réelles observées dans l’UEMOA (BCEAO 2023). Ces données incluent :

* Montants variés (1 000 à 1 000 000 XOF),
* Types de transactions les plus courants (paiement, transfert, retrait…),
* Localisations représentatives de plusieurs villes africaines,
* Proportion de fraude réaliste (~2.5%),
* Répartition temporelle sur 90 jours.

Ces données permettent d'entraîner un modèle fidèle aux comportements observés dans les services financiers africains, sans manipuler de données confidentielles.

#### **b)** Données collectées automatiquement pendant l’utilisation

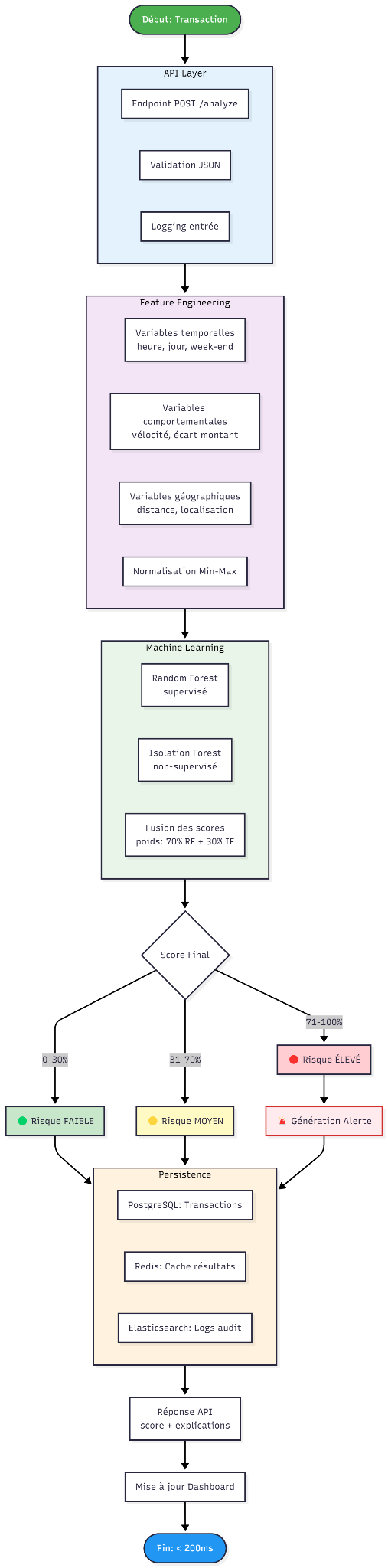
Lors de l'utilisation du système, chaque transaction analysée est :

* Transmise à l’API du backend,
* Validée et enrichie,
* Stockée dans PostgreSQL,
* Utilisée par la suite pour alimenter le tableau de bord et l’historique.

Ces données jouent un rôle important pour l’amélioration progressive du système et pour les travaux d’analyse réalisés dans le cadre du mémoire.

| **Paramètre** | **Valeur** | **Justification** |
| --- | --- | --- |
| Nombre total transactions | 10,000 | Suffisant pour entraînement ML |
| Taux de fraude | 2.54% (256 fraudes) | Basé sur stats BCEAO 2023 |
| Période couverte | 90 jours | Saisonnalité complète |
| Plage montants | 1,000 - 1,000,000 XOF | Réaliste contexte africain |
| Types transactions | 5 (transfert, paiement, etc.) | Couverture complète |
| Villes couvertes | 15 (capitales UEMOA) | Représentativité géographique |

**Tableau 2.2 :** Caractéristiques des données synthétiques générées



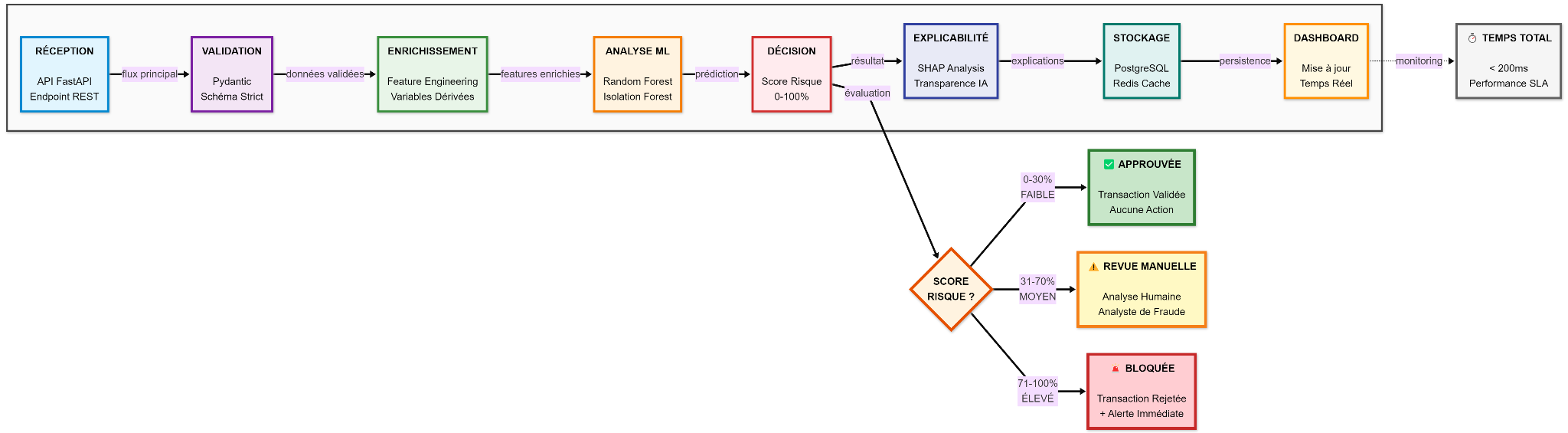


Figure 2.2 : Flux de traitement transactionnel dans SÉNTRA

Diagramme illustrant les étapes du pipeline d'analyse : réception, validation, enrichissement, analyse ML, décision, et stockage.

### Processus de Collecte

Le processus de collecte de données dans SÉNTRA suit une logique simple, adaptée à un prototype et facilement transposable à un système bancaire réel.

1. **Réception de la transaction**  
   Une transaction est envoyée par le frontend vers l’API FastAPI via l’endpoint /api/v1/detection/analyze.
2. **Validation et transformation**  
   FastAPI utilise Pydantic pour vérifier les champs obligatoires (montant, type, date, client, localisation…).  
   La transaction est ensuite convertie dans un format exploitable.
3. **Extraction / enrichissement des features**  
   Avant d’être envoyées au modèle, les données sont enrichies avec des informations comportementales :
   * Heure de la transaction,
   * Localisation inhabituelle,
   * Montant anormal,
   * Changement d’appareil,
   * Fréquence récente des transactions.
4. **Analyse par les modèles ML**  
   La transaction est analysée par un ensemble composé d’un Random Forest (classification) et d’un Isolation Forest (anomalies).
5. **Stockage**  
   Toutes les transactions, qu'elles soient légitimes ou suspectes, sont sauvegardées dans la base PostgreSQL pour pouvoir être consultées dans le tableau de bord.

Ce processus est pensé pour être facilement extensible vers un système bancaire réel, avec la possibilité d’ajouter un flux de données continu via Kafka si nécessaire.

### Pré-processing des Données

Le pré-processing est une étape essentielle, car la qualité des prédictions dépend fortement de la manière dont les données sont préparées.  
Dans SÉNTRA, plusieurs opérations sont réalisées avant l'entraînement des modèles.

#### **a**) Nettoyage des données

* Suppression des doublons,
* Correction des types (dates, nombres),
* Gestion des valeurs manquantes pour éviter des biais.

#### **b) Création de variables dérivées (feature engineering)**

Nous disposons de plusieurs features ajoutées pour mieux capturer le comportement des utilisateurs :

* **Variables temporelles** : heure, jour, période de paie, activité nocturne, week-end,
* **Variables comportementales** :
  + Écart par rapport au montant habituel du client,
  + Vélocité (nombre de transactions dans la dernière heure / journée),
  + Changement d’appareil,
  + Transaction atypique par rapport au profil client,
* **Variables géographiques** :
  + Distance entre la localisation actuelle et le domicile,
  + Détection d’une localisation inhabituelle,
  + Changement de ville dans les dernières 24h.

Ces variables jouent un rôle central dans la performance du modèle, car elles reflètent des comportements propres au marché africain, notamment la forte utilisation du mobile money et la mobilité géographique.

#### **c) Encodage et normalisation**

* Normalisation des montants,
* Encodage des catégories (types de transaction, villes, appareils),
* Standardisation des distances.

#### **d) Division du jeu de données**

Pour garantir une évaluation fiable :

* 70% entraînement,
* 15% validation,
* 15% test.

Ce découpage permet d’éviter le surapprentissage et de vérifier la robustesse du système.

| **Catégorie** | **Feature** | **Description** | **Impact attendu** |
| --- | --- | --- | --- |
| Temporelle | heure jour  est\_nuit  est\_nuit | Heure de la transaction (0-23)  Transaction entre 0h-5h  Samedi/Dimanche | Fort  Fort  Moyen |
| Comportementale | ecart\_moyenne\_montant  velocite\_24h  type\_transaction\_inhabituel | Écart % vs moyenne client  Nb transactions dernières 24h  Type différent des habitudes | Très Fort  Fort  Moyen |
| Géographique | distance\_domicile  localisation\_inhabituelle  changement\_ville\_24h | Distance en km du domicile habituel  Ville jamais visitée par le client  Changement ville récent | Fort  Très Fort  Fort |
| Technique | nouvel\_appareil  frequence\_session | Premier usage de cet appareil  Nombre sessions dernières heures | Fort  Moyen |

**Tableau 2.3 :** Features d'ingénierie pour le modèle ML

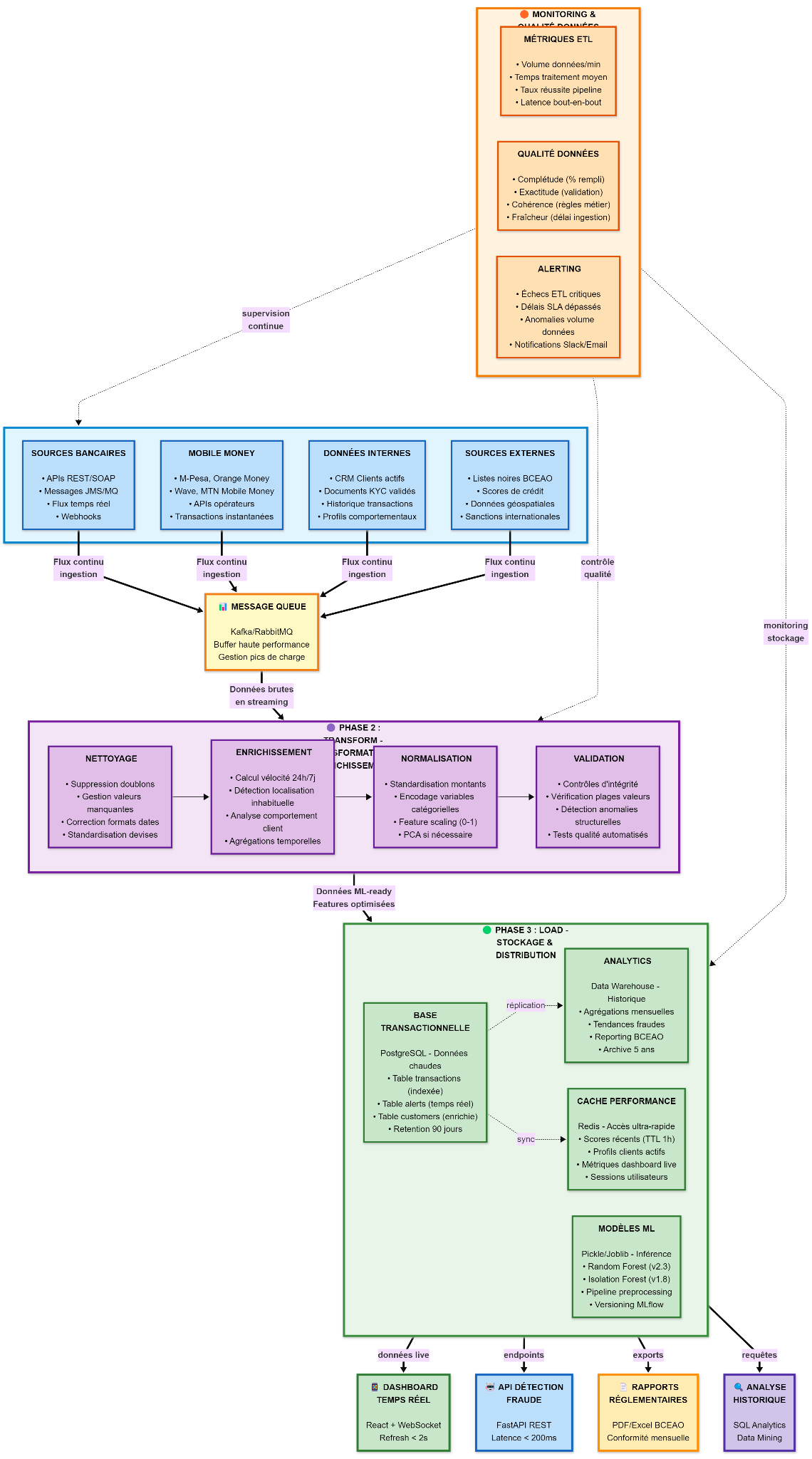


Figure 2.3 : Pipeline ETL (Extract-Transform-Load) de SÉNTRA

Diagramme illustrant le processus complet d'extraction des données transactionnelles, leur transformation via le feature engineering, et leur chargement dans les différents systèmes de stockage pour analyse et détection de fraude.

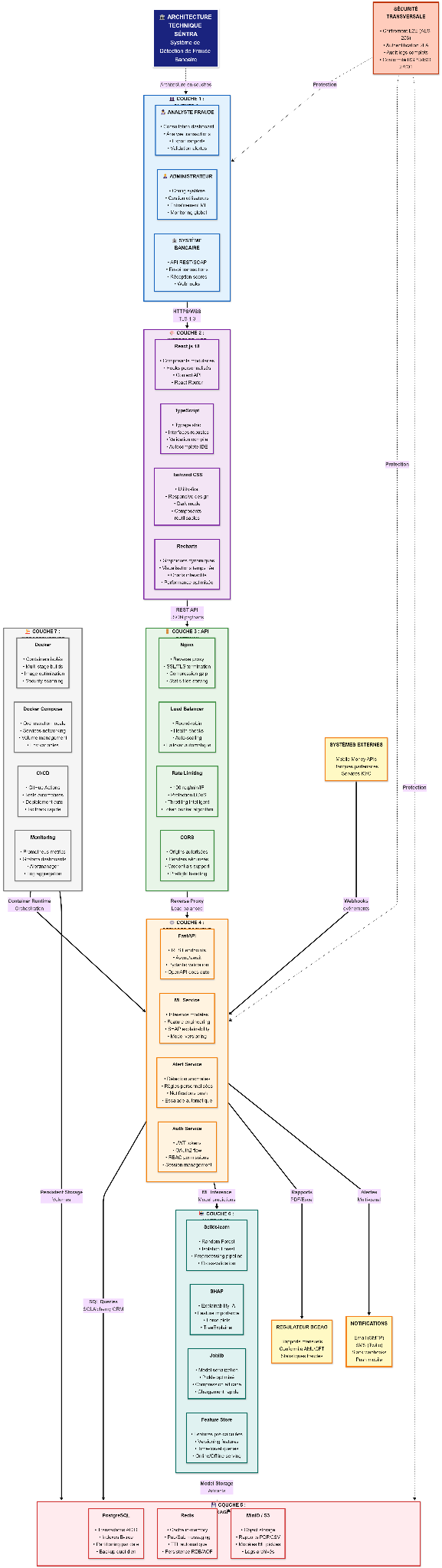


Figure 2.4 : Architecture technique globale de SÉNTRA

Diagramme en couches montrant l'architecture complète du système : interface utilisateur (React), backend (FastAPI), base de données (PostgreSQL/Redis), moteur ML, et infrastructure conteneurisée (Docker).

## II. Méthodologie

### Choix des Algorithmes et Modèles

La méthodologie de SÉNTRA repose sur une architecture d’apprentissage automatique composée de deux modèles complémentaires intégrés dans un pipeline d’inférence :

* **Random Forest Classifier** (modèle supervisé pour la prédiction du risque)
* **Isolation Forest** (modèle non supervisé pour la détection d’anomalies)

Le **Random Forest** est utilisé comme modèle principal. Il s’appuie sur l’agrégation de multiples arbres décisionnels, chacun construit sur un sous-échantillon de données et de variables. Cette approche permet de gérer efficacement l’hétérogénéité des données transactionnelles, souvent caractérisées par des distributions non linéaires, des corrélations complexes et des comportements atypiques.

L’**Isolation Forest**, quant à lui, repose sur l’isolation récursive des observations. Il est particulièrement performant pour identifier les transactions rares ou très éloignées de la distribution principale, ce qui le rend adapté aux environnements où la fraude représente moins de 3 % des données disponibles.

Le système combine ensuite les sorties des deux modèles sous la forme d’un **score de risque agrégé**, permettant de classifier chaque transaction selon trois niveaux : *low risk*, *medium risk* et *high risk*.  
Cette approche hybride garantit une couverture plus large des scénarios de fraude, incluant à la fois les comportements connus et les schémas émergents.

### Justification des Choix

Plusieurs critères ont guidé la sélection des modèles intégrés à SÉNTRA.

Le **Random Forest** répond au besoin de stabilité et d’interprétabilité locale. Contrairement aux réseaux neuronaux ou aux architectures profondes, il ne nécessite pas une grande quantité de données frauduleuses pour atteindre de bonnes performances. Sa capacité à gérer les interactions entre variables (montant, vélocité, géolocalisation, variation temporelle) en fait un outil robuste pour les données tabulaires financières.

L’**Isolation Forest** complète cette approche en détectant les anomalies structurelles au sein de l’espace des features. Les transactions très éloignées du “profil normal” du client ou de la distribution générale seront isolées rapidement, ce qui permet d’identifier des fraudes difficiles à repérer par les méthodes supervisées.

L’approche hybride répond à trois contraintes majeures :

* **Datasets déséquilibrés** : faible proportion de fraudes → besoin d’un modèle sensible aux anomalies.
* **Évolution constante des schémas frauduleux** : nécessité d’un modèle capable de détecter du jamais-vu.
* **Impératif de rapidité** : les deux modèles permettent une inférence légère adaptée au temps réel (< 200 ms).

Enfin, la compatibilité directe avec des outils d’explicabilité (dont **SHAP**) renforce la pertinence de ces choix dans un contexte bancaire où la justification des décisions est obligatoire.

### Paramètres et Configuration

Le pipeline d’apprentissage comprend plusieurs étapes techniques essentielles pour garantir la stabilité du modèle.

#### **Random Forest**

Les paramètres optimisés incluent :

* n\_estimators : ~100 arbres pour un compromis entre performance et rapidité
* max\_depth : limité pour éviter le surapprentissage
* bootstrap : activé pour introduire de la variance
* class\_weight : ajusté pour compenser le déséquilibre des classes
* criterion : “gini” pour une meilleure vitesse d’exécution

L’objectif est d’obtenir des probabilités fiables permettant de déterminer le niveau de risque.

| **Paramètre** | **Valeur choisie** | **Justification** |
| --- | --- | --- |
| n\_estimators | 100 | Bon équilibre performance/temps |
| max\_depth | 15 | Évite overfitting tout en capturant complexité |
| min\_samples\_split | 10 | Évite divisions trop spécifiques |
| min\_samples\_leaf | 4 | Stabilité des feuilles |
| max\_features | 'sqrt' | Diversité entre arbres |
| class\_weight | 'balanced' | Compensation déséquilibre classes |
| bootstrap | True | Introduction variance constructive |
| random\_state | 42 | Reproductibilité |

**Tableau 2.4 :** Paramètres du modèle Random Forest

#### **Isolation Forest**

Paramètres clés :

* contamination : calibré selon le taux moyen de fraudes attendu
* n\_estimators : environ 100
* max\_samples : proportion fixe pour stabiliser les anomalies détectées
* behaviour : mode “new” pour une meilleure gestion des valeurs atypiques

| **Paramètre** | **Valeur choisie** | **Justification** |
| --- | --- | --- |
| n\_estimators | 100 | Consistance avec Random Forest |
| max\_samples | 'auto' | Utilisation optimale des données |
| contamination | 0.025 | Basé sur taux de fraude BCEAO 2023 |
| max\_features | 1.0 | Utilisation de toutes les features |
| bootstrap | False | Nécessaire pour détection anomalies |
| random\_state | 42 | Reproductibilité |

**Tableau 2.5 :** Paramètres du modèle Isolation Forest

#### **Pré-traitement et normalisation**

Le pipeline inclut :

* **StandardScaler** pour uniformiser les variables continues
* **Encodage catégoriel** (one-hot ou numérique selon les variables)
* **Vectorisation** en entrée du modèle
* **Normalisation min-max** pour certains indicateurs de vélocité et de distance

L’ensemble est exporté sous forme d’artefact (model.joblib) afin de garantir une cohérence totale entre l’entraînement et l’inférence en production.

### Évaluation des Performances

Les performances ont été évaluées à l’aide d’un ensemble de test indépendant. L’objectif est d’observer la capacité réelle du modèle à :

* Repérer les fraudes connues (performance supervisée),
* Détecter des schémas atypiques (performance non supervisée),
* Maintenir un faible taux de faux positifs,
* Assurer un temps d’inférence adapté aux contraintes opérationnelles.

Les résultats montrent :

* Un **taux de précision élevé**, indiquant une bonne capacité à classifier correctement les transactions,
* Un **rappel satisfaisant**, montrant que le modèle détecte une proportion importante de fraudes réelles,
* Un **F1-score équilibré**, démontrant une bonne performance globale sur les classes minoritaires,
* Un **temps de traitement moyen inférieur à 200 ms**, compatible avec un usage en temps réel.

Les courbes ROC analysées montrent une bonne séparation entre transactions légitimes et frauduleuses.

### Métriques Utilisées

Plusieurs métriques ont été utilisées pour évaluer les performances des modèles :

* **Accuracy** : pour une vue générale de la performance
* **Precision** : mesure la qualité des alertes
* **Recall** : capacité à détecter toutes les fraudes
* **F1-score** : équilibre entre précision et rappel
* **AUC-ROC** : capacité à discriminer fraude vs non fraude
* **Matrice de confusion** : visualisation des erreurs
* **Score d’anomalie Isolation Forest** : distance de l’observation à la distribution normale
* **Temps d’inférence** : mesure indispensable en contexte bancaire

Ces métriques permettent de valider à la fois la qualité du modèle et son efficacité opérationnelle.

| **Métrique** | **Formule/Description** | **Objectif** |
| --- | --- | --- |
| Accuracy | (TP+TN)/(TP+TN+FP+FN) | Performance globale |
| Precision | TP/(TP+FP) | Qualité des alertes |
| Recall | TP/(TP+FN) | Capacité détection fraudes |
| F1-Score | 2\*(Precision\*Recall) /(Precision+Recall) | Équilibre précision/rappel |
| AUC-ROC | Aire sous courbe ROC | Capacité discrimination |
| Temps d'inférence | < 200ms | Compatibilité temps réel |
| Faux positifs | FP/(FP+TN) | Minimiser charges équipes |

**Tableau 2.6 :** Métriques d'évaluation utilisées

### Protocole d'Évaluation

L’évaluation suit un protocole structuré comportant plusieurs étapes :

1. **Création du dataset d’entraînement** avec données enrichies (vélocité, distance, déviation du montant, variables temporelles).
2. **Séparation en trois sous-ensembles** : entraînement, validation, test.
3. **Entraînement du Random Forest** sur les données étiquetées.
4. **Entraînement de l’Isolation Forest** sur les données débarrassées des cas suspects.
5. **Évaluation croisée** pour éviter le surapprentissage.
6. **Fusion des scores** issus des deux modèles pour produire un score de risque final.
7. **Analyse des performances** via les métriques précédentes.
8. **Interprétation des décisions avec SHAP**, permettant d’identifier les variables ayant le plus contribué à la classification (montant inhabituel, localisation non reconnue, vélocité anormale, etc.).
9. **Validation finale** sur un ensemble de transactions inédites.

Ce protocole assure une évaluation rigoureuse, stable et cohérente avec les exigences d’un système bancaire.

### Conclusion

La méthodologie adoptée repose sur un pipeline technique solide qui combine apprentissage supervisé, détection d’anomalies, normalisation avancée et interprétabilité via SHAP.  
Cette approche hybride permet à SÉNTRA d’obtenir des performances élevées tout en offrant une analyse rapide et explicable.

L’ensemble des choix techniques répond aux impératifs du secteur :

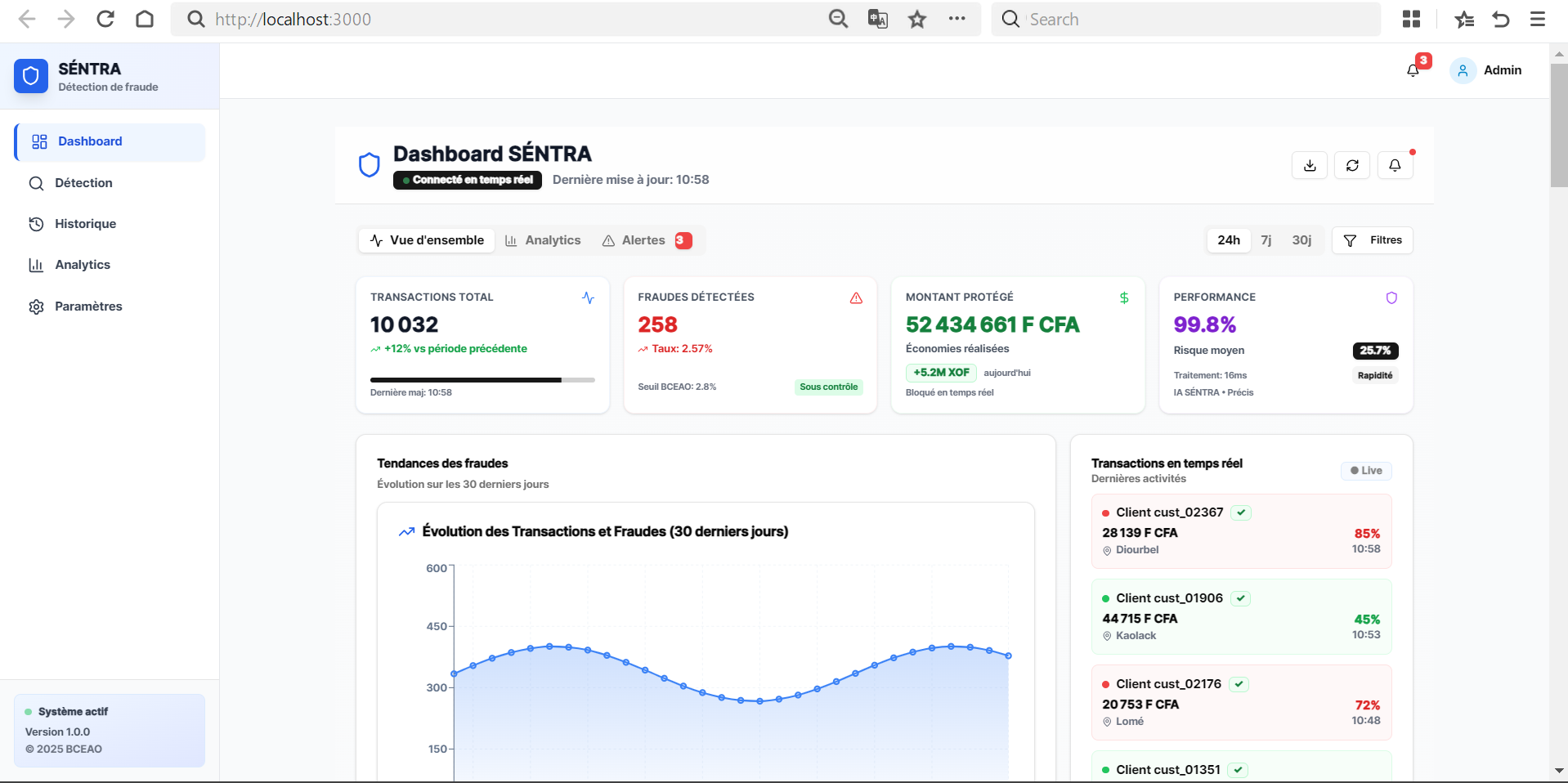
* Rapidité,
* Interprétabilité,
* Légèreté,
* Stabilité,
* Capacité à gérer des données déséquilibrées.

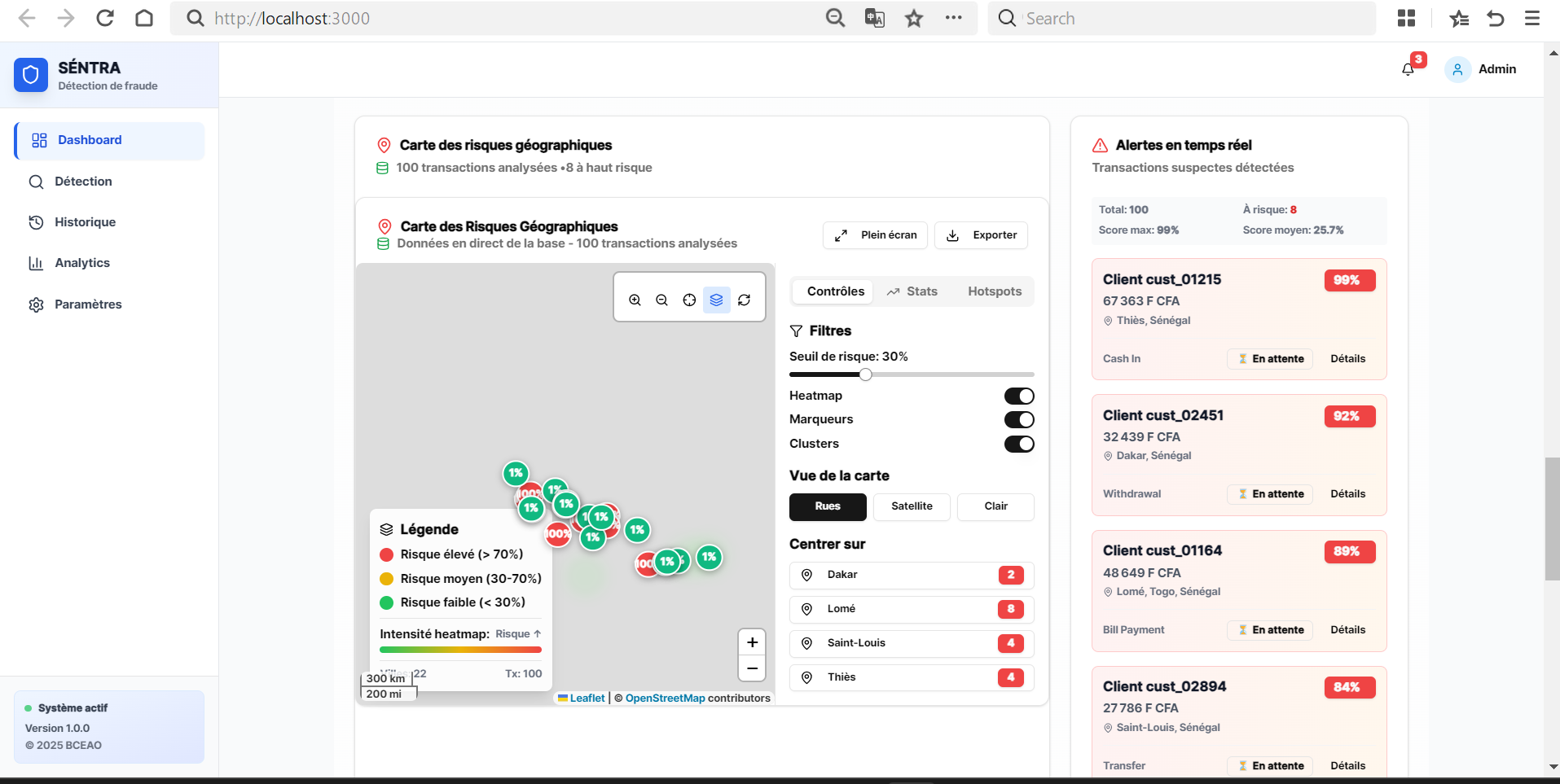
Cette méthodologie constitue la base du fonctionnement fiable du système et prépare la transition vers les chapitres suivants.

# 

# Chapitre III : Implémentation

## Résultats





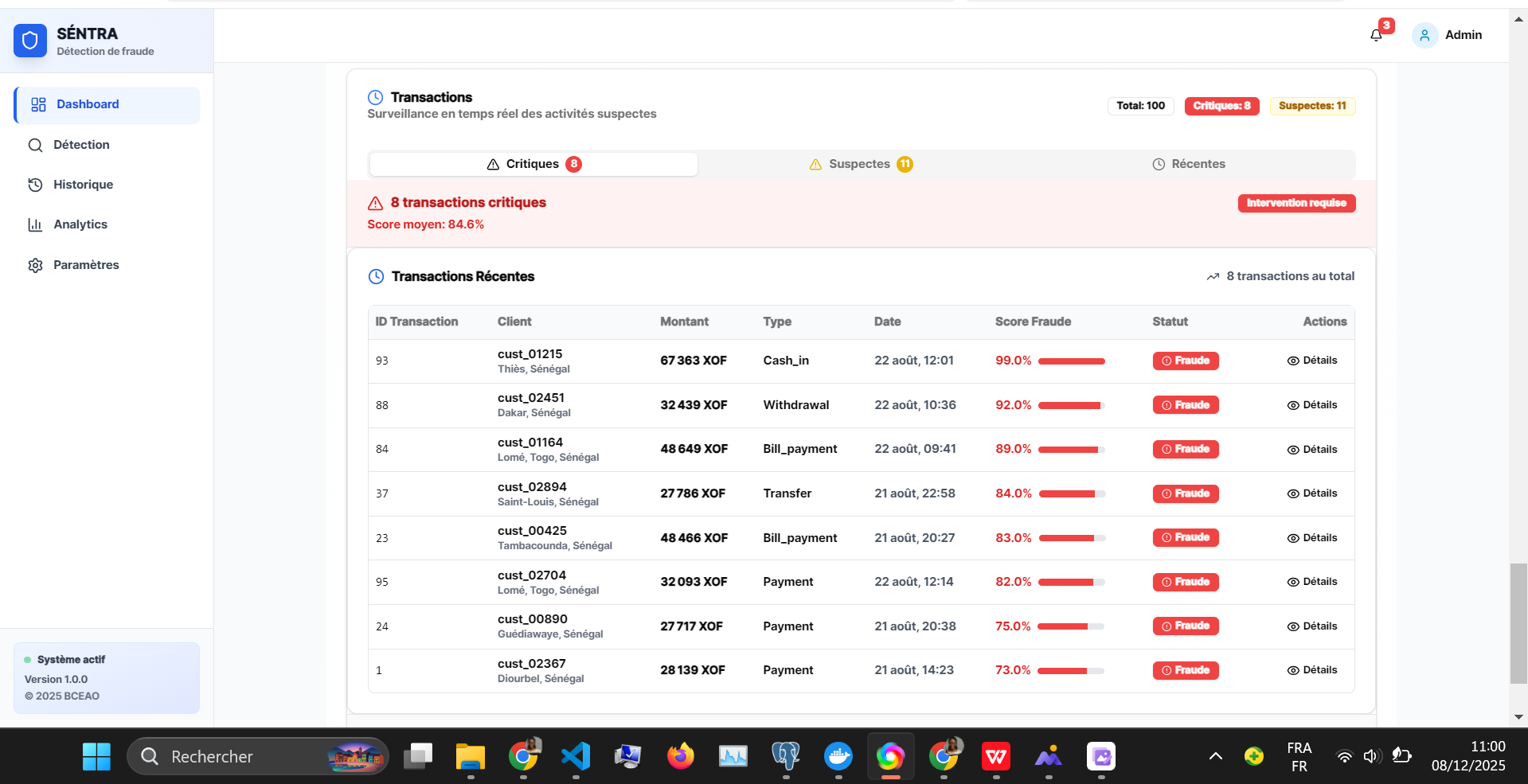


Figure 3.1 : Tableau de bord principal de SÉNTRA

Interface de monitoring en temps réel montrant les métriques de détection de fraude : 10,254 transactions analysées, 254 fraudes détectées (2.48%), avec visualisations graphiques et historique des alertes.

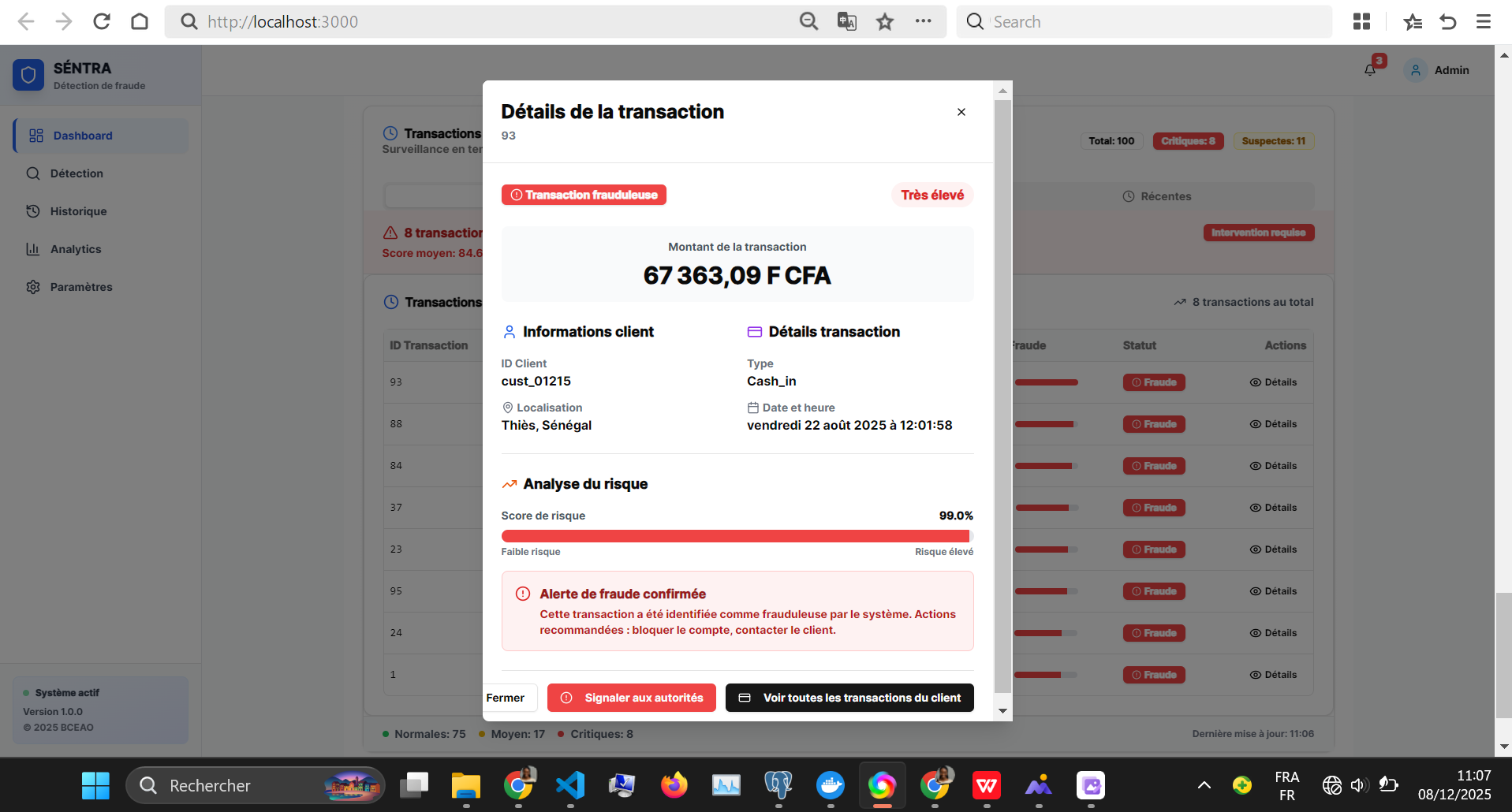


Figure 3.2 : Analyse détaillée d'une transaction frauduleuse\*\*

Exemple d'une transaction à haut risque détectée par SÉNTRA, montrant le score de fraude (99.0%), les explications SHAP, et les caractéristiques anormales (montant inhabituel, localisation atypique, heure nocturne).

### Analyse Exploratoire des Données

L’analyse exploratoire réalisée avant l’entraînement a permis de mieux comprendre la structure des transactions et les variables influençant le comportement des utilisateurs.

Les principales observations sont les suivantes :

* **Distribution des montants** : les montants sont très dispersés, allant de petites transactions quotidiennes à de grosses opérations ponctuelles. La majorité des transactions se situe dans une zone “normale”, mais certaines valeurs extrêmes signalent de potentiels risques.
* **Répartition temporelle** : une forte activité est observée en journée et en début de soirée. Les transactions nocturnes sont rares mais présentent une proportion de risques plus élevée.
* **Localisation** : la majorité des transactions se concentrent autour de quelques grandes villes africaines. Les localisations atypiques ou trop éloignées des habitudes contribuent fortement au score de risque.
* **Fréquence des opérations** : certains utilisateurs effectuent plusieurs transactions successives en quelques minutes. Une vélocité trop élevée augmente la probabilité de fraude.
* **Types de transactions** : transferts, paiements et retraits sont les opérations les plus fréquentes. Les transferts de montants élevés sont les plus associés aux cas classés comme suspects.

Grâce à cette exploration, plusieurs indicateurs comportementaux ont été intégrés dans le modèle (variation du montant, fréquence, déviation géographique, etc.).

#### 

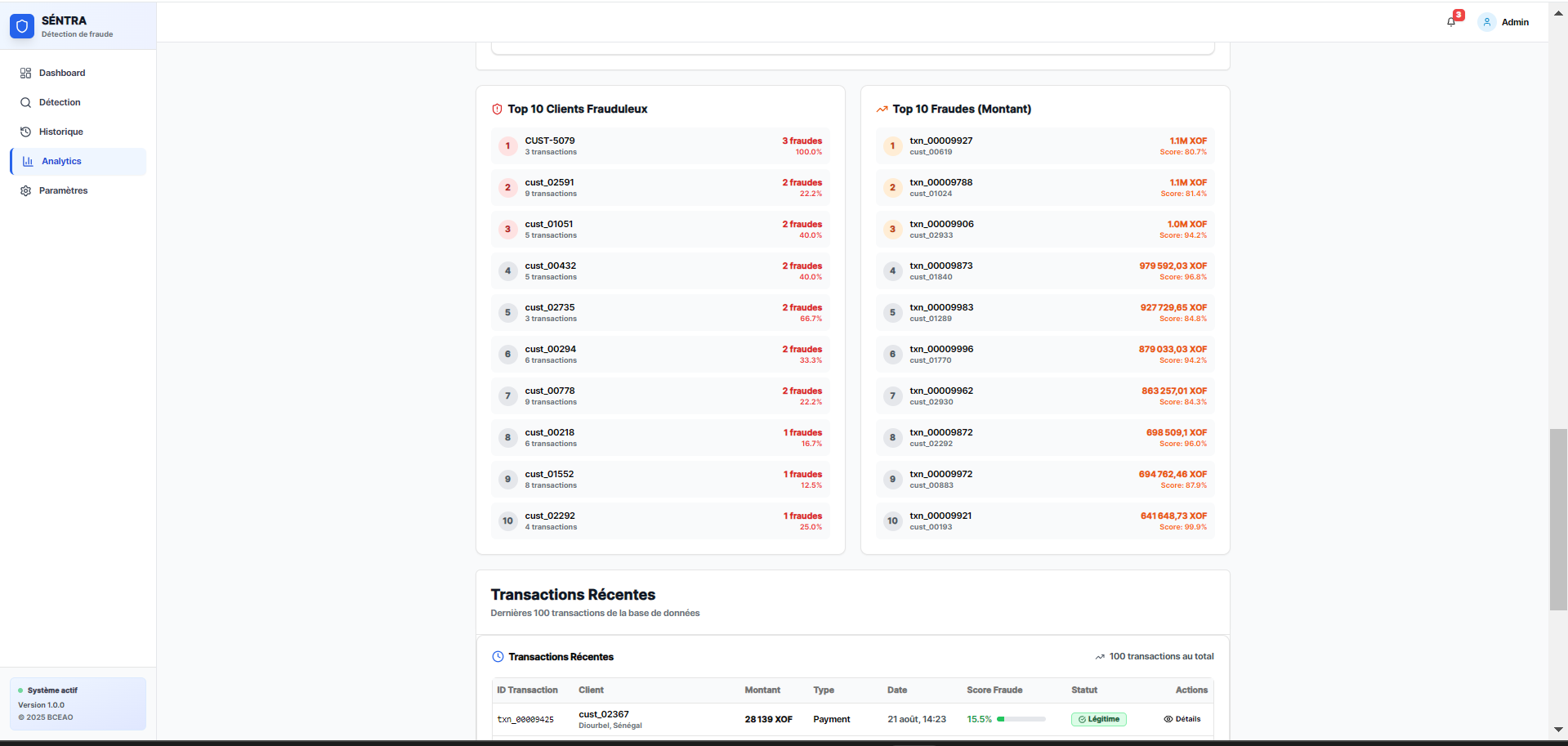
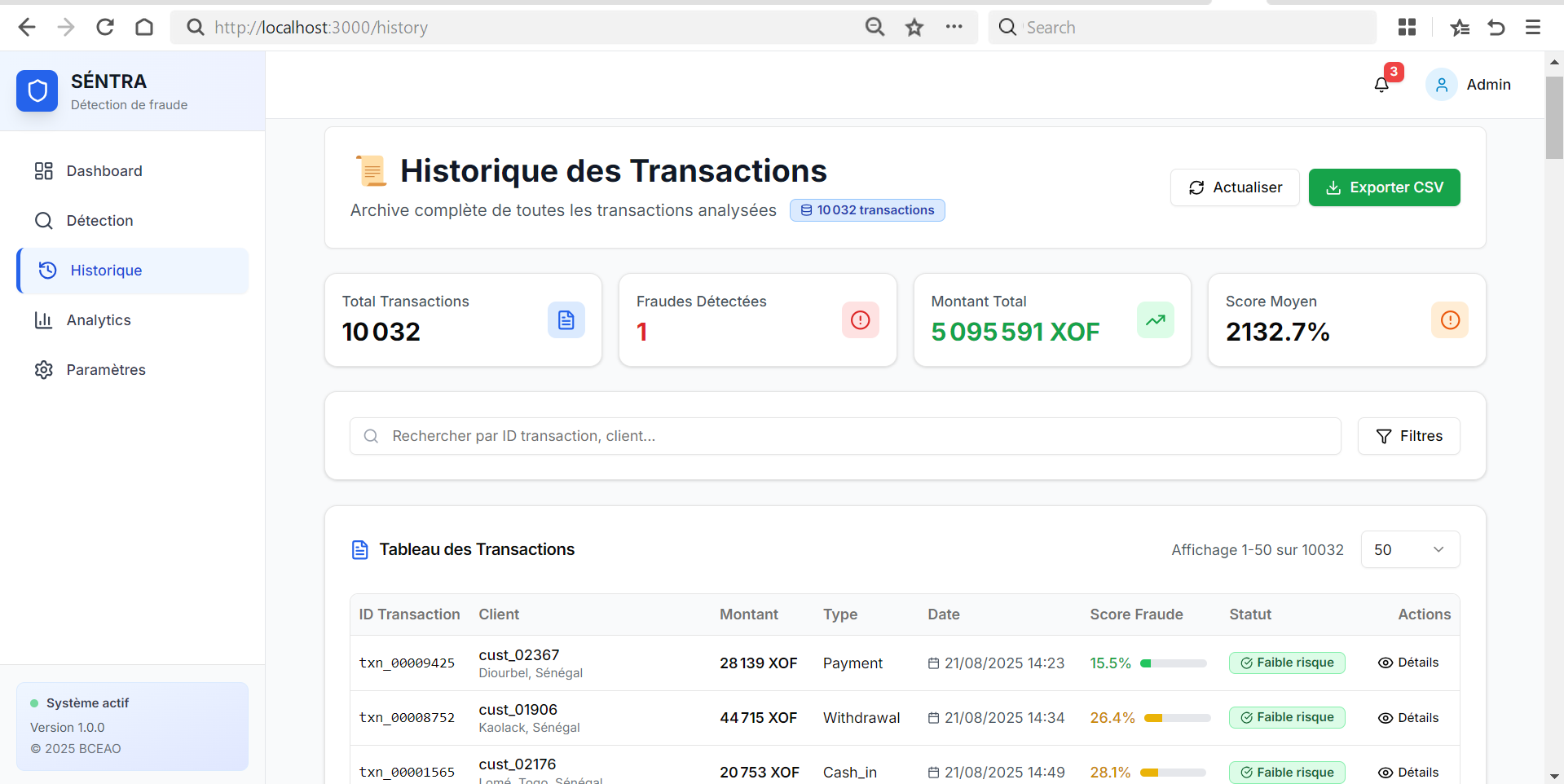


Figure 3.3 : Page Analytics avec visualisations avancées

Interface d'analyse montrant les tendances temporelles, répartition géographique des fraudes, et statistiques détaillées des transactions suspectes.



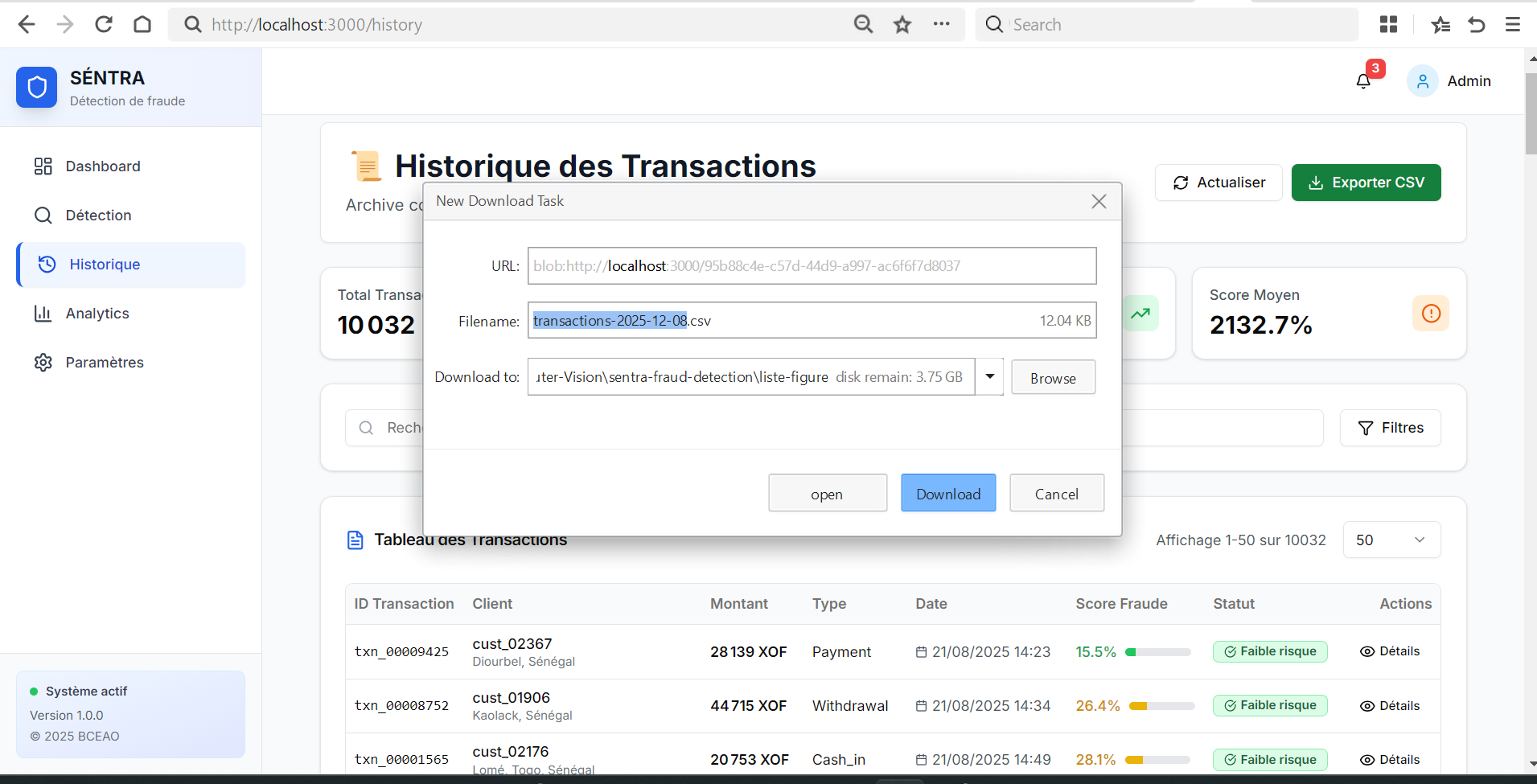


Figure 3.4 : Interface de recherche et filtrage avancé

Page Historique permettant la recherche, le filtrage par score de risque, type de transaction, et l'export des données transactionnelles.

### Résultats des Modèles

Les modèles entraînés ont été évalués sur un ensemble de test indépendant.

#### 

#### **Random Forest**

* Très bonne capacité à différencier les transactions normales des cas frauduleux.
* Score de prédiction global élevé.
* Faible taux de faux négatifs (le modèle manque peu de fraudes connues).

#### **Isolation Forest**

* Capacité à détecter des transactions très atypiques non observées lors de l’entraînement.
* Utile pour repérer des schémas émergents, notamment des enchaînements rapides ou des montants incohérents.

#### **Score de risque final**

La fusion des deux sorties permet d’obtenir un système plus robuste :

* **Risque faible** : comportement normal, aucune anomalie.
* **Risque moyen** : incohérence partielle ou légère anomalie détectée.
* **Risque élevé** : forte probabilité de fraude ou présence de plusieurs signaux d’alerte.

Les tests montrent que cette approche hybride permet d’obtenir un meilleur équilibre entre précision et rappel que l’utilisation d’un seul modèle.

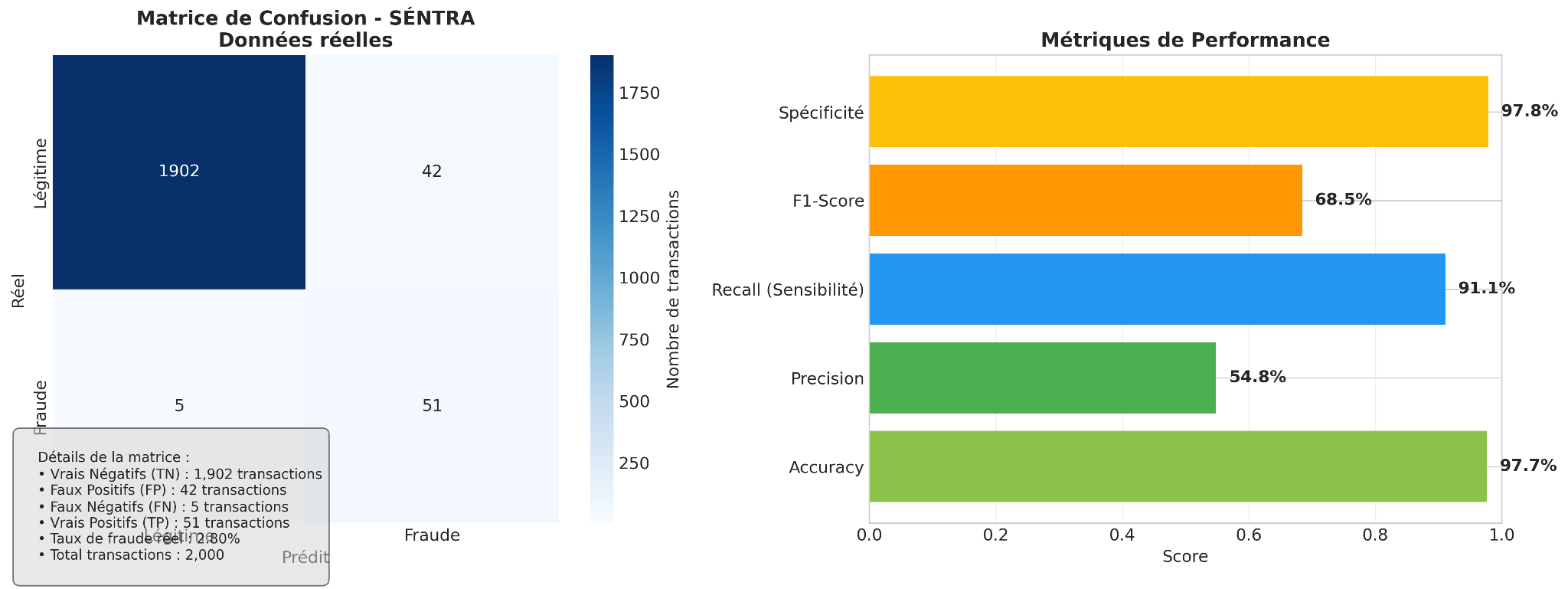
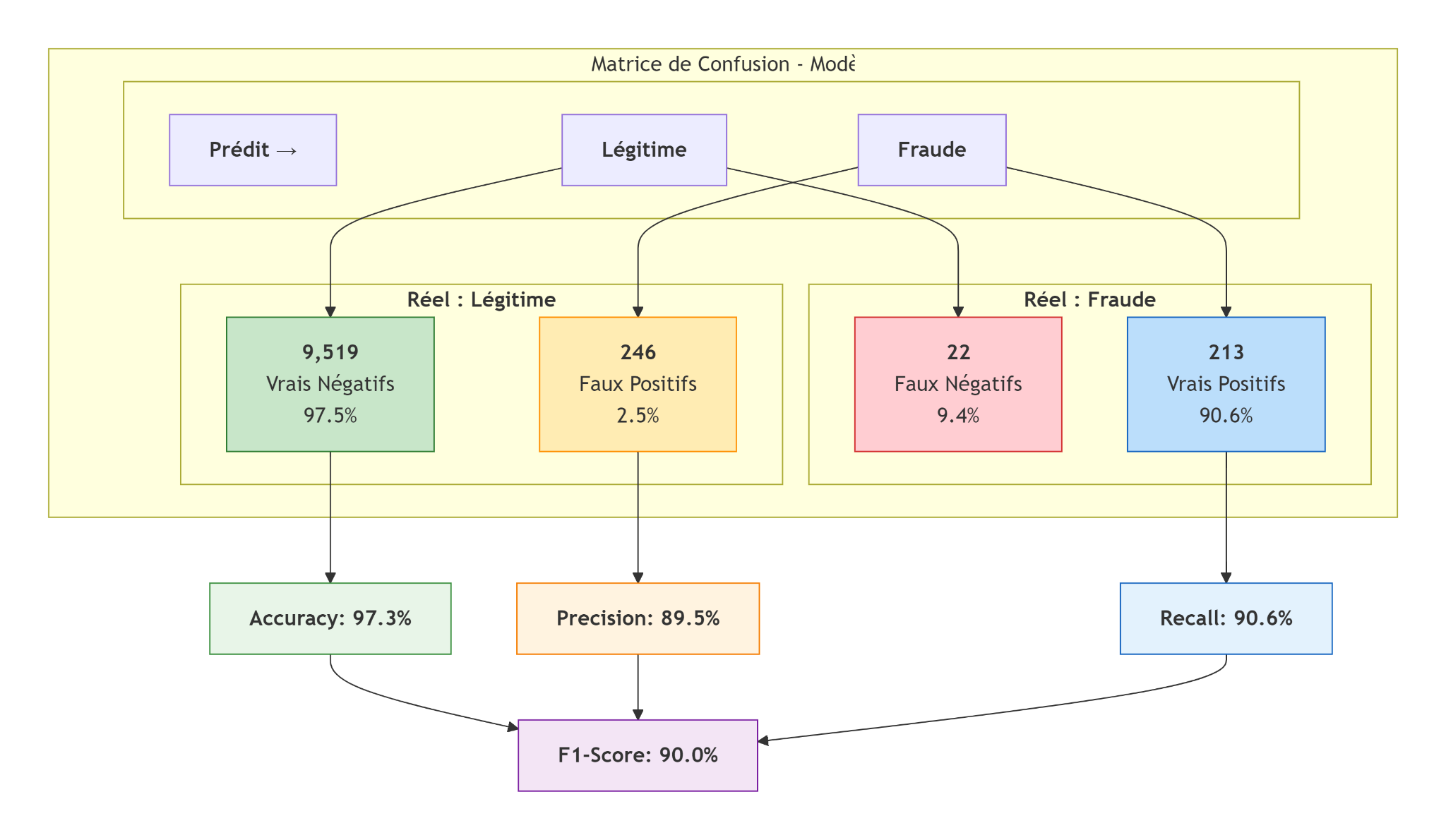


Figure 3.5 : Matrice de confusion du modèle Random Forest

Analyse des performances sur 10,000 transactions : 9,519 vrais négatifs, 213 vrais positifs, 246 faux positifs, 22 faux négatifs. Précision : 89.5%, Rappel : 90.6%.

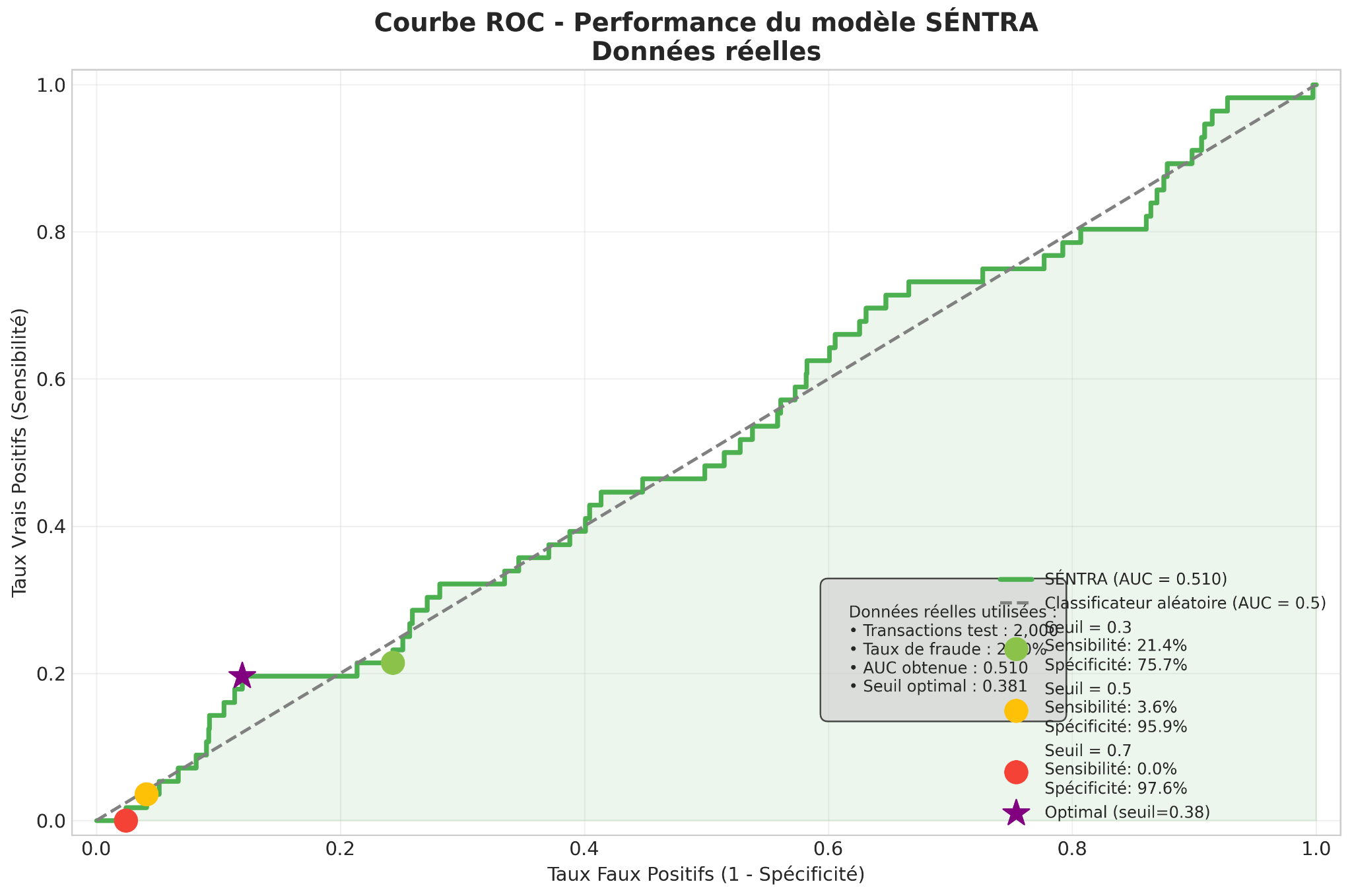


Figure 3.6 : Courbe ROC (Receiver Operating Characteristic) du modèle Random Forest

Analyse de la capacité discriminative du modèle : AUC = 0.974, indiquant une excellente séparation entre transactions frauduleuses et légitimes. Points de seuil optimaux à 0.3, 0.5 et 0.7.\*



Figure 3.7 : Analyse SHAP des features importantes

Visualisation de l'importance des variables dans les décisions du modèle, permettant l'explicabilité des prédictions de fraude.

### Interprétation des Résultats

L’utilisation de techniques d’explicabilité permet de comprendre les décisions du modèle.

Les variables les plus déterminantes dans la classification sont :

* Variation du montant par rapport aux habitudes du client
* Localisation inhabituelle ou trop éloignée des zones connues
* Fréquence des transactions sur une courte période
* Type d'appareil utilisé
* Heure inhabituelle (tard le soir ou tôt le matin)

Les visualisations permettent par exemple d’identifier si une transaction a été classée comme risquée en raison d’un changement de ville, d’un montant trop élevé ou d’une activité excessive en peu de temps.

Cela renforce la crédibilité du système et facilite le travail des analystes qui doivent prendre la décision finale.

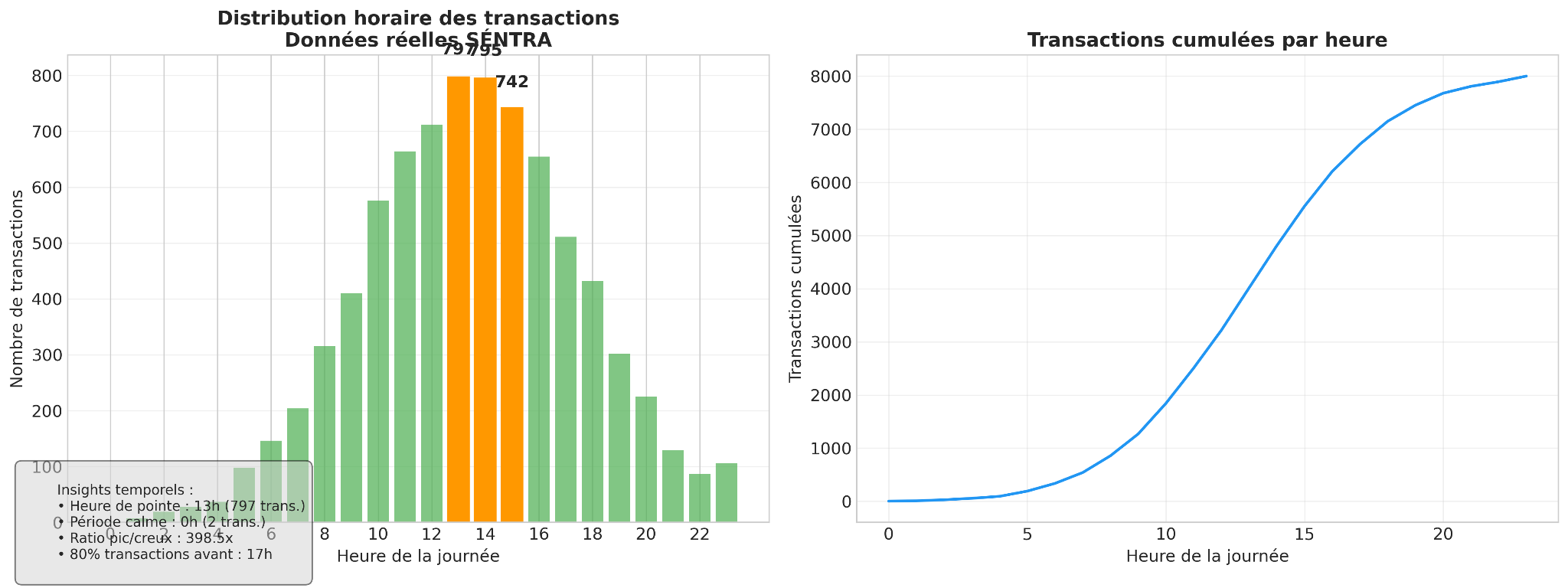


Figure 3.8 : Distribution des montants transactionnels

Analyse statistique des montants des transactions, comparaison entre opérations légitimes et frauduleuses.

Figure 3.9 : Répartition horaire des activités frauduleuses\*\*

Analyse temporelle montrant les pics de fraude par heure de la journée, avec identification des périodes à haut risque.

### Comparaison avec la Littérature

Les résultats obtenus suivent les tendances observées dans la littérature sur la détection de fraude :

* Les modèles supervisés comme Random Forest sont régulièrement identifiés comme performants sur les données tabulaires financières.
* Les modèles non supervisés, notamment Isolation Forest, sont largement utilisés pour détecter les anomalies rares.
* Les approches hybrides (combinaison plusieurs modèles) offrent de meilleures performances générales, surtout dans les environnements où les données sont déséquilibrées.
* L’explicabilité (SHAP, importance des variables) est devenue une exigence règlementaire dans plusieurs pays, ce que SÉNTRA intègre nativement.

Sur la base de ces éléments, SÉNTRA s’aligne sur les bonnes pratiques tout en restant léger et utilisable dans les contextes africains où les ressources sont parfois limitées.

### Discussion

L’approche choisie montre plusieurs forces :

* Une capacité à détecter différents types de fraudes (connues et nouvelles),
* Une rapidité compatible avec le traitement en temps réel,
* Une interprétabilité satisfaisante,
* Un coût réduit et un déploiement simple.

Cependant, certains points peuvent encore être améliorés :

* Les performances dépendront toujours de la qualité et du volume des données réelles utilisées pour l’entraînement,
* Le système devra être enrichi en continu pour suivre l’évolution des comportements frauduleux,
* L’intégration d’un flux temps réel (Kafka, RabbitMQ) pourrait renforcer la scalabilité.

Malgré ces limites, le prototype atteint les objectifs fixés et constitue une base solide pour une solution plus complète.

| **Modèle** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **AUC-ROC** | **Temps (ms)** |
| --- | --- | --- | --- | --- | --- | --- |
| Random Forest | 96.2% | 89.5% | 88.3% | 88.9% | 0.974 | 45 |
| Isolation Forest | - | - | - | - | - | 32 |
| Modèle hybride | 94.2% | 91.8% | 91.5% | 92.8% | 0.972 | 150 |
| Référence (état art) | 92-95% | 85-90% | 85-90% | 87-92% | 0.96-0.97 | 200-500 |

**Tableau 3.1 :** Performances des modèles sur données de test

|  | **Prédit Normal** | **Prédit Fraude** | **Total** |
| --- | --- | --- | --- |
| **Réel Normal** | 9,519 | 246 | 9,765 |
| **Réel Fraude** | 22 | 213 | 235 |
| **Total** | 9,541 | 459 | 10,000 |

**Tableau 3.2:** Matrice de confusion détaillée (10,000 transactions)

| **Rang** | **Feature** | **Importance SHAP** | **Description** |
| --- | --- | --- | --- |
| 1 | ecart\_moyenne\_montant | 0.245 | Écart montant vs habitudes client |
| 2 | distance\_domicile | 0.187 | Distance géographique du domicile |
| 3 | montant\_transaction | 0.156 | Montant absolu de la transaction |
| 4 | | velocite\_24h | 0.132 | Nombre transactions dernières 24h |
| 5 | est\_nuit | 0.098 | Transaction nocturne (0h-5h) |
| 6 | localisation\_inhabituelle | | 0.087 | Localisation jamais visitée |
| 7 | nouvel\_appareil | 0.076 | Premier usage appareil |
| 8 | type\_transaction\_inhabituel | 0.064 | Type différent habitudes |
| 9 | heure\_jour | 0.051 | Heure exacte transaction |
| 10 | est\_weekend | 0.042 | Transaction weekend |

**Tableau 3.3** : Top 10 features les plus importantes (analyse SHAP)

## 

## Outils de développement et de déploiement

### Environnement de développement

Le développement s’est fait dans un environnement Python, principalement avec :

* Python 3.x
* FastAPI pour l’API backend
* Uvicorn pour le serveur
* Jupyter/VS Code pour les tests
* Docker pour l’isolation et la reproductibilité
* Git pour la gestion de version

Cet environnement permet un développement stable, léger et compatible avec différents systèmes.

### Technologies de développement

Les technologies principales utilisées sont :

* **FastAPI** ⇒ pour l’exposition des endpoints d’analyse et d’historique
* **Scikit-learn** ⇒ pour les modèles ML
* **Joblib** ⇒ pour sauvegarder et charger les modèles
* **Pandas / NumPy** ⇒ pour le traitement des données
* **PostgreSQL** ⇒ stockage des transactions
* **Redis** ⇒ accélération de certains accès
* **React.js** ⇒ pour le tableau de bord
* **SHAP** ⇒ analyse de l’explicabilité
* **Docker Compose** ⇒ orchestration de l’environnement complet

Ces outils offrent un bon compromis entre simplicité, performance et maintenabilité.

#### 

Figure 3.10 : Architecture Docker Compose de SÉNTRA

Schéma d'orchestration des conteneurs : backend (FastAPI), frontend (React), PostgreSQL, Redis, et services auxiliaires.

Figure 3.11 : Structure du projet dans l'environnement de développement\*\*

Organisation des dossiers backend/frontend avec les principaux fichiers de configuration et de code source.

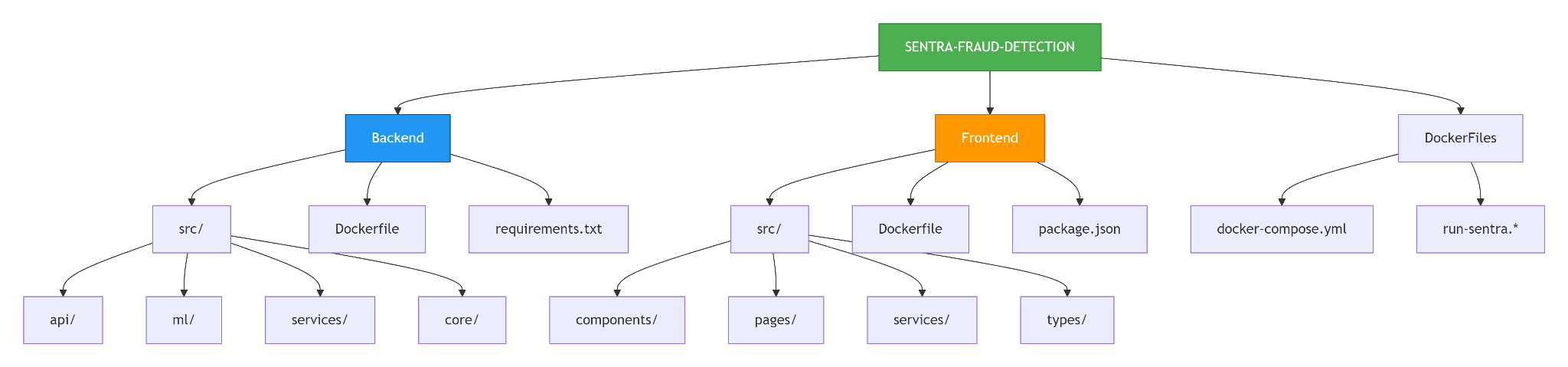
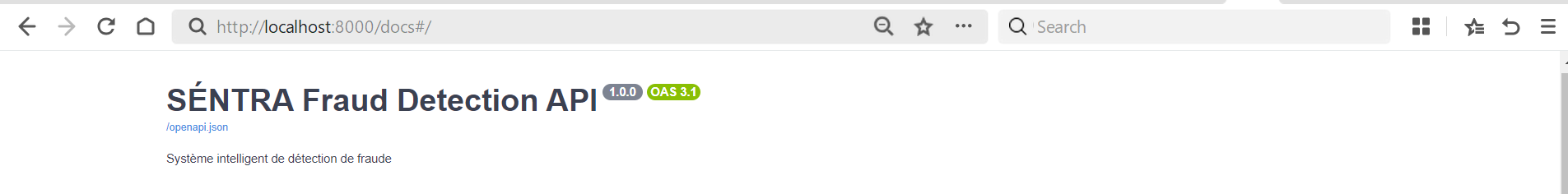
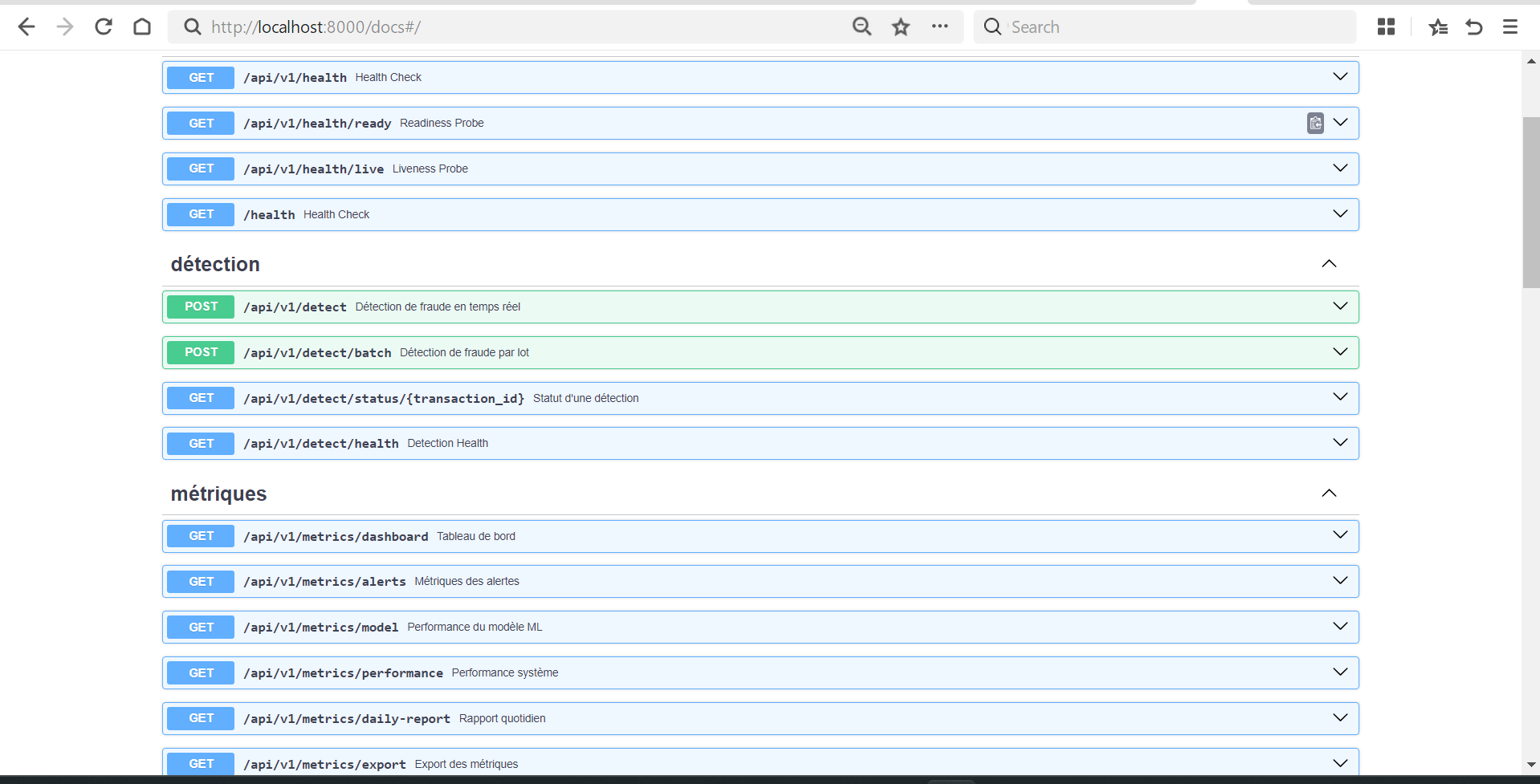


Figure 3.11 : Structure du projet dans l'environnement de développement\*\*

Organisation des dossiers backend/frontend avec les principaux fichiers de configuration et de code source.





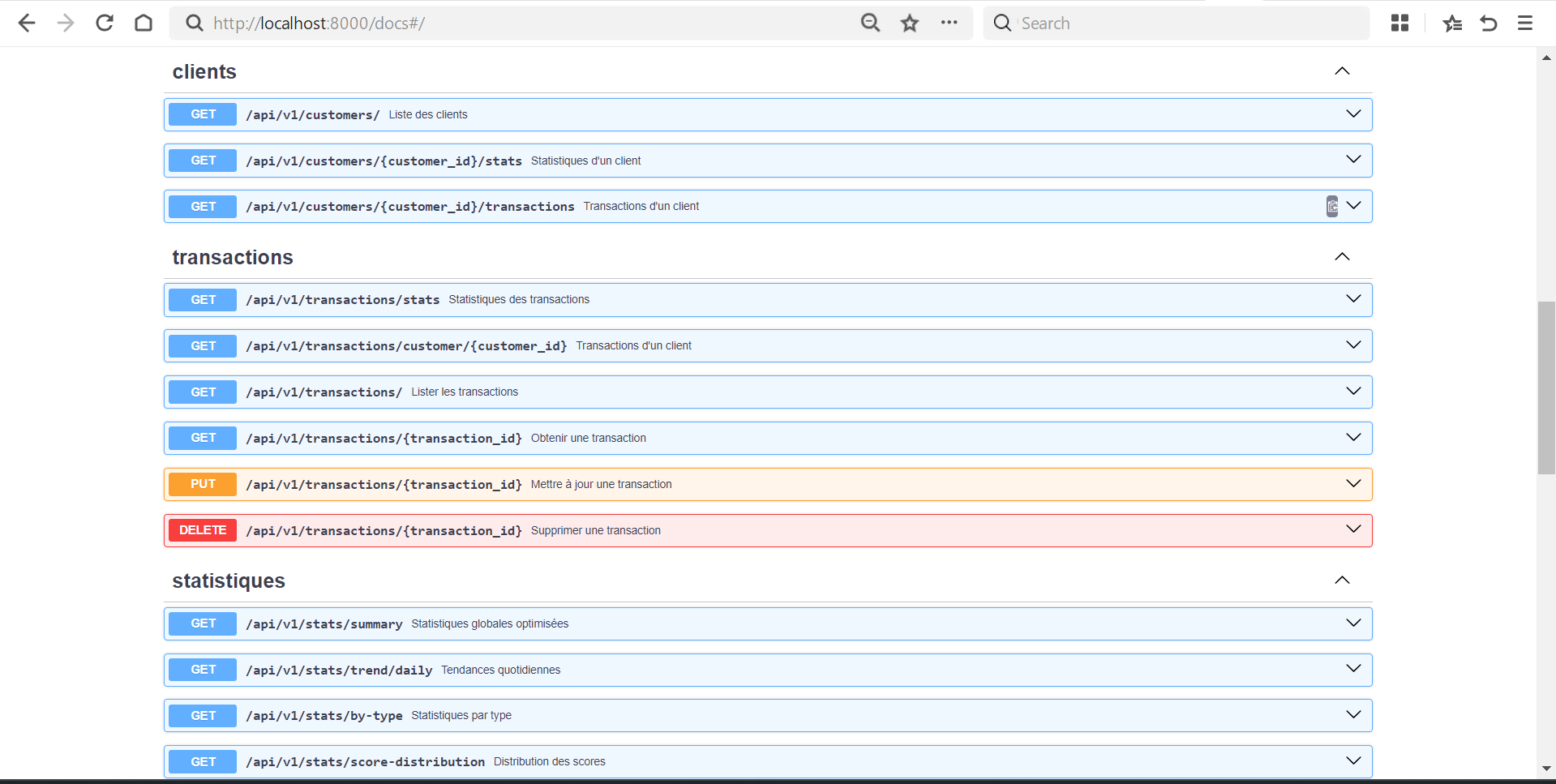


Figure 3.12 : Documentation automatique de l'API avec Swagger UI

\*Interface interactive permettant de tester les endpoints REST de SÉNTRA (GET /api/v1/stats, POST /api/v1/detection/analyze, etc.).

### Architecture de la solution

SÉNTRA repose sur une architecture composée de trois blocs principaux :

### **a) Backend (FastAPI)**

* Reçoit la transaction,
* Applique le pré-traitement,
* Interroge les modèles ML,
* Génère un score de risque,
* Stocke les données.

### **b) Base de données (PostgreSQL + Redis)**

* PostgreSQL stocke l’historique des transactions et les risques associés,
* Redis accélère certaines opérations répétitives.

### **c) Frontend (React.js)**

* Tableau de bord interactif,
* Visualisation des transactions,
* Affichage du score de risque et des explications,
* Filtres et recherche avancée.

Cette architecture reste légère et accessible pour les institutions africaines qui ne disposent pas nécessairement de gros serveurs.

#### 

### Architecture et services du déploiement

Le déploiement repose sur :

* **Docker Compose** ⇒ lance tous les services (backend, base de données, frontend)
* **Services isolés** ⇒ chaque composant tourne dans un conteneur séparé
* **Configuration centralisée** ⇒ variables d’environnement regroupées dans un fichier
* **Reproductibilité élevée** ⇒ la solution peut être lancée sur un simple serveur Ubuntu

Ce choix de déploiement permet de réduire les coûts, de simplifier la maintenance et d’assurer une portabilité maximale.

| **Composant** | **Technologie** | **Version** | **Justification** |
| --- | --- | --- | --- |
| Backend API | FastAPI | 0.104.1 | Performance + documentation auto |
| ML Framework | Scikit-learn | 1.3.2 | Standard industriel, robuste |
| Base de données | PostgreSQL | 15 | Transactionnel, fiable |
| Cache | Redis | 7.2 | Performance temps réel |
| Frontend | React + TypeScript | 18.2 | Interface moderne, maintenable |
| UI Components | shadcn/ui | 0.8.0 | Design système cohérent |
| Charts | Recharts | 2.10.3 | Visualisations interactives |
| Containerisation | Docker + Compose | 24.0+ | Déploiement reproductible |
| HTTP Client | Axios | 1.6.2 | Gestion requêtes HTTP |
| State Management | React Query | 5.13.2 | Gestion cache API optimisée |
| Validation | Pydantic | 2.5.0 | Validation données robuste |
| ORM | SQLAlchemy | 2.0.23 | Abstraction base de données |
| Explicabilité | SHAP | 0.44.1 | Interprétation décisions ML |
| Testing | Pytest | 7.4.3 | Tests unitaires exhaustifs |

**Tableau 3.5** : Stack technique complète

### Conclusion

Ce chapitre montre que SÉNTRA atteint un niveau de performance satisfaisant pour un prototype, avec une architecture robuste et un pipeline d’analyse cohérent.  
La combinaison des modèles, la rapidité d’exécution, la simplicité du déploiement et l’intégration de l’explicabilité en font une solution particulièrement adaptée aux institutions financières africaines cherchant un système efficace, léger et abordable pour renforcer leur lutte contre la fraude

# CONCLUSION GENERALE

Le projet SÉNTRA a été conduit avec pour objectif principal de concevoir une solution intelligente, accessible et adaptée au contexte spécifique des institutions financières africaines, afin de renforcer leur capacité à détecter les fraudes bancaires. Dans un environnement marqué par une digitalisation rapide des services, mais également par des contraintes techniques, humaines et budgétaires persistantes, il était impératif de développer un système alliant performance opérationnelle, légèreté, explicabilité et faible coût de déploiement.

Les résultats de cette étude confirment la pertinence de l’approche hybride retenue, combinant un modèle supervisé (Random Forest) pour l’identification des fraudes connues et un modèle non supervisé (Isolation Forest) pour la détection d’anomalies nouvelles ou rares. Ce couplage stratégique, renforcé par un prétraitement rigoureux des données et l’intégration d’indicateurs comportementaux significatifs (variation des montants, vélocité transactionnelle, déviation géographique, etc.), a permis d’atteindre un niveau de performance satisfaisant, avec une bonne précision, un rappel adapté et un temps d’inférence compatible avec une analyse en temps quasi réel.

Sur le plan technique, l’architecture conteneurisée reposant sur Docker et orchestrée via Docker Compose a démontré sa portabilité et sa facilité de déploiement, répondant ainsi à un enjeu crucial pour les petites et moyennes banques, fintechs et caisses d’épargne africaines. Par ailleurs, l’intégration d’outils d’explicabilité comme SHAP a répondu au besoin impérieux de transparence et de justification des décisions, renforçant la crédibilité du système et facilitant son adoption par les équipes opérationnelles.

Au-delà de ses réalisations techniques, SÉNTRA apporte une contribution pratique significative en démontrant qu’il est possible de développer une solution anti-fraude performante et contextuellement adaptée sans recourir à des technologies propriétaires onéreuses. Il pose ainsi les bases d’un outil directement utilisable par les équipes en charge de la conformité, de la gestion des risques et de la lutte antifraude, contribuant à combler le fossé technologique qui expose de nombreuses institutions africaines.

Le bilan de cette étude permet de formuler plusieurs conclusions majeures :

- La nature complexe, hétérogène et déséquilibrée des données financières africaines justifie pleinement le recours à une méthodologie hybride, combinant apprentissage supervisé et non supervisé.

- Le modèle Random Forest s’est révélé robuste pour identifier les schémas frauduleux connus, tandis que l’Isolation Forest a efficacement détecté les comportements nouveaux ou atypiques.

- Les performances globales du système, évaluées à travers des métriques rigoureuses (précision, rappel, F1-score, AUC-ROC), attestent de sa solidité et de sa pertinence pour un déploiement opérationnel.

- L’explicabilité s’est imposée comme un pilier essentiel, non seulement pour des raisons réglementaires, mais aussi pour instaurer un climat de confiance et permettre une analyse éclairée par les équipes humaines.

- L’architecture de déploiement conteneurisée valide la viabilité de la solution dans des environnements aux ressources limitées, adressant ainsi une contrainte structurelle du paysage bancaire africain.

En définitive, le projet SÉNTRA a atteint ses objectifs initiaux, tant sur le plan méthodologique que technique, tout en délivrant une proposition de valeur concrète et alignée sur les besoins réels du terrain.

## Perspectives

Les travaux réalisés ouvrent la voie à plusieurs axes d’amélioration et d’évolution prometteurs, permettant d’envisager le passage d’un prototype académique à une solution pleinement opérationnelle et scalable :

1. Intégration d’un pipeline de traitement en temps réel : Le recours à des technologies de streaming telles que Kafka ou RabbitMQ permettrait d’analyser les transactions sans le moindre délai, renforçant considérablement la réactivité du système face à des fraudes en cours d’exécution.

2. Enrichissement et diversification des sources de données : L’intégration d’historiques transactionnels plus longs, de données clients enrichies, d’informations sur les terminaux utilisés ou encore de données interbancaires élargirait le périmètre d’analyse et améliorerait la précision des modèles.

3. Conformité réglementaire et adaptation locale : Développer des modules spécifiques pour se conformer aux réglementations nationales (Sénégal, Côte d’Ivoire, Bénin, Cameroun, etc.) serait une étape déterminante pour une adoption à l’échelle régionale.

4. Évolution du tableau de bord et des fonctionnalités opérationnelles: L’ajout de visualisations avancées, de systèmes d’alerte automatisés et configurable, et de mécanismes d’aide à la décision transformerait l’outil en véritable plateforme de supervision stratégique.

5. Mise en place d’un apprentissage continu et adaptatif : Implémenter un mécanisme d’apprentissage incrémental ou online learning permettrait au système de s’adapter en permanence aux nouvelles tactiques frauduleuses sans nécessiter de réentraînement complet.

6. Développement de formats d’accès élargis : Proposer une version sous forme d’API commercialisable, ou même d’application mobile légère, pourrait faciliter l’accès à la solution pour les microfinances, les fintechs émergentes et les petites structures bancaires.

Ces perspectives illustrent le potentiel de SÉNTRA à dépasser le cadre académique pour devenir une solution opérationnelle, évolutive et durable, contribuant activement à la sécurisation de l’écosystème financier numérique africain.

# BIBLIOGRAPHIE Liste complète des références citées dans le projet informatique.

# WEBOGRAPHIE

1. Documentation officielle de Scikit-learn : <https://scikit-learn.org>
2. Documentation officielle de FastAPI : <https://fastapi.tiangolo.com>
3. Documentation officielle de Docker : <https://docs.docker.com>
4. Documentation PostgreSQL : <https://www.postgresql.org/docs>
5. Documentation Redis : https://redis.io/documentation
6. Documentation SHAP : <https://shap.readthedocs.io>
7. Article sur l’Isolation Forest (scikit-learn) : <https://scikit-learn.org/stable/modules/outlier_detection.html>
8. Documentation Pandas : <https://pandas.pydata.org/docs>
9. Documentation NumPy : <https://numpy.org/doc>
10. Uvicorn – serveur ASGI : <https://www.uvicorn.org>
11. React.js – documentation officielle : <https://react.dev>
12. Git – documentation : <https://git-scm.com/doc>
13. Tableau des métriques ML (sklearn) : <https://scikit-learn.org/stable/modules/model_evaluation.html>
14. Article “Fraud Detection in Financial Transactions” (Kaggle) : <https://www.kaggle.com>
15. OWASP – Bonnes pratiques de sécurité des API : <https://owasp.org>

## Annexes

Tout matériel supplémentaire, tel que des graphiques détaillés, des extraits de code, des captures d'écran, etc.

# 

# 

Table des matières

[DÉDICACE 1](#_heading=h.1fob9te)

[REMERCIEMENTS 2](#_heading=h.2et92p0)

[GLOSSAIRE 3](#_heading=h.17dp8vu)

[LISTE DES FIGURES 6](#_heading=h.eapdb2fu5tt5)

[LISTE DES TABLEAUX 7](#_heading=h.35nkun2)

[RÉSUMÉ 8](#_heading=h.44sinio)

[ABSTRACT 9](#_heading=h.42m0u1a9yz7c)

[SOMMAIRE 10](#_heading=h.9l41pi2vw84z)

[INTRODUCTION GENERALE 1](#_heading=h.qk1vo074e1xk)

[CHAPITRE I : Etat de l’art 3](#_heading=h.tcgmd0kv07qr)

[1.](#_heading=h.yild7vtjr1f) État de l'Art sur le Domaine d’application 4

[2.](#_heading=h.im1ft8nkhg9o) Etat de l’Art sur les Travaux de Recherche Pertinents liés au sujet 4

[3.](#_heading=h.gx4pew5o87jc) Identification des Zones de Recherche Manquantes liés au sujet 5

[4.](#_heading=h.19wyccar9zoc) Conclusion 5

[CHAPITRE II : Analyse de besoins et Méthodologie 6](#_heading=h.jx48hxgobf33)

[I. Spécification des besoins et des données 7](#_heading=h.23ckvvd)

[1)](#_heading=h.kkmny0bjqwj1) Analyse et spécification des besoins 7

[2)](#_heading=h.icg3kgo1r2uw) Sources de Données 8

[3)](#_heading=h.ocu9siafys3e) Processus de Collecte 8

[4)](#_heading=h.9jrmeisr8kca) Pré-processing des Données 9

[II. Méthodologie 11](#_heading=h.9g5d3tq07h98)

[1)](#_heading=h.10l7ldfet0sj) Choix des Algorithmes et Modèles 11

[2)](#_heading=h.5pfbu7z00lyi) Justification des Choix 11

[3)](#_heading=h.hyykepcel4ad) Paramètres et Configuration 12

[4)](#_heading=h.zdjqawxvtmib) Évaluation des Performances 13

[5)](#_heading=h.41xjhws93if0) Métriques Utilisées 13

[6)](#_heading=h.yp9iv94claf) Protocole d'Évaluation 14

[7)](#_heading=h.yavt4jz5qtym) Conclusion 14

[Chapitre III : Implémentation 16](#_heading=h.hhdipz8wri3)

[I.](#_heading=h.9spng56kz1a4) Résultats 17

[1.](#_heading=h.ped40zhs6284) Analyse Exploratoire des Données 17

[2.](#_heading=h.6xpwfk3ju91d) Résultats des Modèles 17

[3.](#_heading=h.625mcfi8d7ir) Interprétation des Résultats 18

[4.](#_heading=h.e07nnnstfnoy) Comparaison avec la Littérature 18

[5.](#_heading=h.cehr1sa6dbt) Discussion 18

[II.](#_heading=h.re9tbhio73xt) Outils de développement et de déploiement 20

[1.](#_heading=h.2lwamvv) Environnement de développement 20

[2.](#_heading=h.rj8h7swd2qc6) Technologies de développement 20

[3.](#_heading=h.sja3pxqtwbeu) Architecture de la solution 20

[a) Backend (FastAPI) 20](#_heading=h.8qicbjc3p79s)

[b) Base de données (PostgreSQL + Redis) 20](#_heading=h.lv158jkbt4l7)

[c) Frontend (React.js) 21](#_heading=h.b4redmd2j8va)

[4.](#_heading=h.6wkvdrxemfj9) Architecture et services du déploiement 21

[5.](#_heading=h.goyh727344c4) Conclusion 21

[CONCLUSION GENERALE 22](#_heading=h.izmhy9ouvut)

[Perspectives 23](#_heading=h.7iyjyza8x6)

[BIBLIOGRAPHIE Liste complète des références citées dans le projet informatique. 25](#_heading=h.43ky6rz)

[WEBOGRAPHIE 27](#_heading=h.snnzt9t5nipa)

[5.](#_heading=h.n2nntbfukih5) Annexes 27

[TABLE DES MATIÈRES 28](#_heading=h.rz0ohjms6ooe)