Project Synopsis: Crypto-Hunter

Project Title: Automated Anti-Money Laundering (AML) Ring Detection System using Graph Neural Networks & Generative AI

Domain: Financial Forensics, Big Data, Deep Learning, GenAI

1. Introduction

Financial crime is evolving. Modern money launderers use complex networks of transactions to hide the origin of illicit funds, a process known as "Structuring" or "Smurfing." Traditional banking systems, which analyze transactions in isolation (row-by-row), often fail to detect these sophisticated patterns.

"Crypto-Hunter" is a next-generation forensic tool designed to detect these hidden laundering "rings" by treating financial data as a **Graph Network** (A web of connections) rather than a simple spreadsheet. It leverages **Apache Spark** for big data processing, **Graph Neural Networks (GNNs)** for pattern recognition, and **Generative AI (RAG)** to automate the investigation reporting process.

2. Problem Statement

- **The Blind Spot:** Standard Rule-Based Systems (e.g., "Flag if amount > $10,000") are easily bypassed by criminals who break large sums into small, seemingly innocent amounts.
- **Complexity:** Laundering involves "Loops" (A $\to$ B $\to$ C $\to$ A) and "Fan-out/Fan-in" patterns that are computationally expensive to find in massive datasets using standard SQL.
- **Operational Bottleneck:** Even when a system flags a suspect, a human analyst must manually investigate the transaction history and map it to legal statutes (e.g., PMLA, IPC) to write a report, which is slow and error-prone.

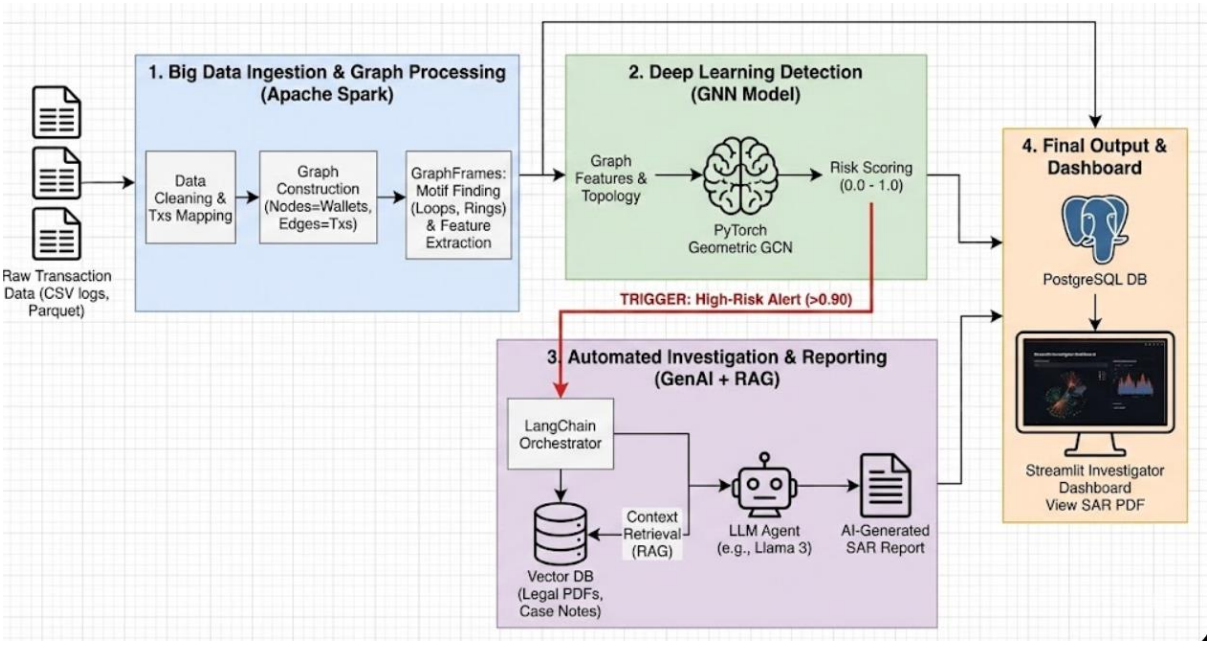
3. Proposed Solution

We propose a **Graph-First Architecture** that moves beyond checking *how much* money is moved, and instead analyzes the *shape* of how it moves.

Key Innovations:

1. **Graph Analytics (The "Eye"):** We use **Apache Spark (GraphFrames)** to build a transaction graph and identify specific "Motifs" (circular loops and structural anomalies) that indicate laundering rings.

2. **Deep Learning (The "Brain"):** We implement a **Graph Convolutional Network (GCN)**. This model uses "Guilt by Association"—if a normal user frequently interacts with a high-risk cluster, their risk score increases.
3. **Automated Reporting (The "Voice"):** We integrate **Generative AI (RAG)**. When a ring is detected, the system retrieves relevant legal context (e.g., Banking Regulations) and automatically drafts a detailed "Suspicious Activity Report" (SAR) for the investigator.



4. Input & Output Specifications

A. Input (Data Ingestion)

The system ingests raw transaction logs (e.g., The Elliptic Bitcoin Dataset).

- **Format:** CSV / Parquet
- **Volume:** High-volume transaction data (200k+ nodes).
- **Attributes:** Sender_ID, Receiver_ID, Timestamp, Amount, Currency_Type.

B. Output (Deliverables)

The system produces three distinct outputs for the end-user (Bank/Investigator):

1. **Risk Alerts:** A structured list of Wallet IDs flagged as "High Risk" (Risk Score > 0.90).
2. **Visual Evidence:** A subgraph visualization showing the specific "Ring" or "Loop" involving the suspect.

3. **AI-Generated Forensic Report:** A PDF document containing:
    - Summary of the incident.
    - Visual proof of the laundering loop.
    - Citation of relevant legal acts (e.g., PMLA Section 12) explaining the violation.

5. Technology Stack & Justification

Component	Technology	Role in Project
Big Data Processing	**Apache Spark (PySpark)**	Required to handle the massive computation of building graphs from millions of transaction rows.
Graph Analytics	**GraphFrames**	Used for "Motif Finding" (identifying circular patterns like A-B-C-A).
Deep Learning	**PyTorch Geometric**	Implements the Graph Neural Network (GCN) to classify nodes based on topology.
Generative AI	**LangChain + Ollama**	Orchestrates the "AI Investigator" to write reports using local LLMs (Llama 3).
Database	**PostgreSQL**	Stores the final results (Flagged Wallets, Risk Scores) for the dashboard.
Visualization	**Streamlit / NetworkX**	Provides the user interface for investigators.

---

6. Project Workflow (Step-by-Step)

1. **Ingest:** Raw data is loaded into Spark.
2. **Process:** Spark converts data into Vertices (Users) and Edges (Transactions).

3. **Detect:**
    - Spark looks for specific **shapes** (Loops).
    - GNN Model analyses **relationships** to assign a probability score.
4. **Act:** If a user is flagged (Score > 90%), the GenAI module is triggered.
5. **Report:** The AI retrieves legal knowledge and drafts the explanation.

7. Conclusion

"Crypto-Hunter" is not just a detection script; it is an end-to-end **Financial Intelligence Platform**. By combining the raw power of Big Data (Spark) with the reasoning capabilities of AI (GNN & GenAI), it solves the "Needle in a Haystack" problem while simultaneously solving the "Alert Fatigue" problem for human investigators.