Problem Statement

I need a dataset of 10,000 to 50,000 crypto transactions and related off-chain data to build an AI-driven fraud detection model. Your task is to collect this data from specified public sources, ensuring it includes both on-chain transaction details and off-chain contextual insights. The data must be structured with the column names provided below, sourced from the listed locations, and delivered in a usable format (e.g., CSV). The goal is to capture transaction patterns, wallet behaviors, and scam reports to train a predictive model.

Detailed Instructions

On-Chain Data (10,000–50,000 Transactions)

- **Objective**: Collect transaction and wallet data from public blockchains (e.g., Ethereum).
- **Sources**:
  - **Etherscan** (etherscan.io): Use free API (5 req/sec) to scrape transaction histories and wallet interactions.
  - **Glassnode** (glassnode.com): Free tier for on-chain metrics like trade volume.
  - **The Graph** (thegraph.com): Query subgraphs via studio.thegraph.com for smart contract data.
- **Method**: Write a script (e.g., Python with etherscan-python) to pull 10K–50K transactions. Filter by recent blocks or known scam-related wallets for relevance.
- **Column Names**:
  - Sender (wallet address)
  - Receiver (wallet address)
  - Amount (transaction value in ETH/stablecoins)
  - Time (timestamp)
  - Trade_Frequency (count of trades over time)
  - Withdrawal_Speed (time between deposits/withdrawals)
  - Trade_Volume (total value traded)
  - Contract_Address (smart contract involved)
  - Interaction_Type (e.g., lending, NFT purchase)
  - Connected_Wallet (linked wallet address)
  - Scam_Flag (binary: known scam or not)
  - Interaction_Count (number of interactions)

Off-Chain Data (Supporting Contextual Insights)

- **Objective**: Gather user behavior and scam reports from public platforms to complement on-chain data.
- **Sources**:
  - **Twitter (X)** (x.com): Scrape posts mentioning crypto scams using Tweepy or similar (search terms: "crypto scam," "rug pull").

    - **Telegram** (telegram.org): Join public crypto groups (e.g., scam report channels) and scrape via bot API.
    - **Reddit** (reddit.com): Use Reddit API on r/CryptoCurrency for scam-related posts/comments.
    - **Bitcointalk** (bitcointalk.org): Scrape forum threads on scams via web scraping (e.g., BeautifulSoup).
    - **CryptoScamDB** (cryptoscamdb.org): Pull scam reports and addresses.
    - **BadBitcoin.org**: Extract listed scam details.
- **Method**: Use APIs (Twitter, Reddit) or scraping tools (Python: requests, BeautifulSoup) to collect 10K–50K entries. Link off-chain scam mentions to on-chain addresses where possible (e.g., via reported wallet IDs).
- **Column Names**:
    - IP_Address (if available from public metadata, otherwise omit)
    - Timestamp (post/login time)
    - User_ID (unique user/poster identifier)
    - Withdrawal_Amount (if mentioned in reports, otherwise omit)
    - Login_Time (if available, otherwise omit)
    - API_Call_Count (omit unless exchange data is sourced)
    - Post_ID (unique post identifier)
    - Platform (e.g., Twitter, Telegram)
    - Text (content of post/report)
    - Fraud_Signal (binary: scam mention detected via keywords/NLP)

Deliverables

- A dataset (CSV) with 10,000–50,000 rows combining on-chain and off-chain data.
- All specified columns populated where data is available; mark unavailable fields (e.g., IP_Address) as "N/A."
- Source documentation (e.g., which rows came from Etherscan, Twitter, etc.).
- Ensure data is clean, deduplicated, and timestamp-aligned where applicable.