LOVELACE'26

SMURFING HUNTER AI
GRAPH-BASED BLOCKCHAIN FORENSICS SYSTEM

1. Introduction

With the rapid growth of cryptocurrencies, blockchain networks have become attractive platforms for financial crimes such as money laundering, fraud, and illicit fund transfers. Although blockchains are transparent, the sheer volume and complexity of transactions make manual investigation extremely difficult. Criminals exploit this by using techniques like **smurfing (layering)** and **peeling chains** to hide the origin and destination of funds.

This project, **Smurfing Hunter AI**, proposes a hybrid forensic system that combines **graph theory**, **machine learning**, and **rule-based topology detection** to identify suspicious transaction behavior in blockchain networks. The system not only predicts risk but also explains *why* a transaction is suspicious using structural patterns and visual evidence.

2. Problem Statement

Traditional fraud detection systems focus mainly on transaction-level features such as amount, frequency, or timestamps. However, money laundering is fundamentally a **network problem**:

- Funds are split, merged, and routed through multiple intermediaries
- Suspicious behavior emerges from **how transactions are connected**, not in isolation
- Many transactions are unlabeled or marked as unknown

Hence, there is a need for a system that:

1. Understands transaction relationships

2. Detects known laundering structures
3. Provides explainable and visual results for investigators

3. Objectives of the Project

The main objectives of Smurfing Hunter AI are:

- To model blockchain transactions as a directed graph
- To enrich transaction data with graph-based features
- To assign a risk score to each transaction using machine learning
- To detect common money-laundering patterns such as smurfing and peeling chains
- To provide an interactive dashboard for forensic investigation
- To ensure explainability and reproducibility of results

4. Dataset Description

The project uses the **Elliptic Bitcoin Transaction Dataset**, which contains:

- Transaction features (166 attributes per transaction)
- Transaction classes:
    - Class 1: Illicit
    - Class 2: Licit
    - Unknown: Unlabeled transactions
- Transaction edges representing fund flow between transactions

This dataset is well-suited for graph-based analysis and is widely used in academic research on blockchain forensics.

5. System Architecture

The system follows a multi-stage pipeline:

1. Data loading and preprocessing
2. Graph construction

3. Graph feature extraction
4. Machine learning model training
5. Topology-based pattern detection
6. Interactive visualization and reporting

Each stage contributes to building an explainable and robust forensic system.

6. Graph Construction and Feature Engineering

6.1 Graph Modeling

The blockchain is modeled as a **directed graph**:

- Nodes represent transactions
- Directed edges represent fund flow between transactions

This structure allows the system to analyze money movement paths and transaction dependencies.

6.2 Graph Features

Two important graph features are calculated:

- **PageRank**: Measures how influential a transaction is within the network. Transactions that lie on important fund flow paths receive higher scores.
- **Degree Centrality**: Measures how many connections a transaction has with others.

These features help the model understand transaction importance beyond raw transaction attributes.

7. Machine Learning Risk Scoring

7.1 Model Selection

A **Random Forest Classifier** is used due to:

- Its ability to handle high-dimensional data
- Robustness to noise
- Interpretability compared to deep learning models

7.2 Training Process

- Only transactions with known labels (licit or illicit) are used for training
- Graph features are merged with transaction features
- The model learns patterns associated with illicit behavior

7.3 Risk Prediction

After training, the model predicts the probability of being illicit for all transactions, including unknown ones. This probability is converted into a **risk score (0–100%)**.

---

8. Topology-Based Pattern Detection

Machine learning alone cannot guarantee explainability. Therefore, rule-based detectors are added.

8.1 Smurfing / Layering Detection

This detector identifies:

- **Fan-Out patterns**: One transaction sending funds to many others
- **Fan-In patterns**: Many transactions sending funds to one transaction
- **Diamond structures**: Multiple paths connecting the same source and destination

These patterns are commonly used to hide the source of funds.

8.2 Peeling Chain Detection

This detector identifies long transaction chains where:

- Funds move step-by-step through multiple transactions
- Small side outputs are generated at each step

Such patterns are typical in advanced money laundering schemes.

9. Dashboard and Visualization

An interactive dashboard is built using Streamlit and PyVis.

9.1 Dashboard Features

- Transaction search by ID
- Adjustable neighborhood depth
- Risk score and centrality metrics
- Automatic pattern labeling

9.2 Graph Visualization

- Nodes are color-coded based on risk level
- Suspicious paths and pattern participants are highlighted
- Investigators can visually trace money flow

This visual approach significantly improves interpretability and trust.

10. Testing and Validation

Unit tests are implemented to validate:

- Fan-in detection
- Fan-out detection
- Smurfing diamond patterns
- Peeling chain detection

This ensures that forensic logic behaves correctly and consistently.

11. Results and Output

The system produces:

- Risk score for every transaction
- Detected laundering patterns with confidence
- Interactive subgraph visualization
- Exportable CSV investigation reports

These outputs support both automated screening and manual investigation.

12. Advantages of the Proposed System

- Combines AI with domain knowledge
- High explainability
- Scalable to large blockchain networks
- Investigator-friendly interface
- Suitable for academic and real-world forensic use

13. Limitations

- Does not use transaction amounts (structural analysis only)
- Pattern detectors rely on heuristics
- Performance depends on graph completeness

These limitations open opportunities for future improvements.

14. Future Enhancements

- Incorporating transaction amounts and timestamps
- Using Graph Neural Networks (GNNs)

- Cross-chain analysis
- Real-time transaction monitoring

15. Conclusion

Smurfing Hunter AI demonstrates how graph-based machine learning combined with explainable topology detection can effectively identify money laundering behavior in blockchain networks. By focusing on structure, flow, and visualization, the system bridges the gap between automated detection and human investigation, making it a powerful tool for blockchain forensics.