BlockScan: Detecting Anomalies in Blockchain Transactions

Author: Jiahao Yu*, Xian Wu*, Hao Liu, Wenbo Guo, Xinyu Xing
Presenter: Jiahao Yu

Northwestern

NYU

Meta

The Problem: DeFi is Under Attack

Attacks against DeFi protocols (front-running, re-entrancy, etc.) cause billions in losses. We need to detect anomalous transactions *before* they do irreversible damage.

Goal: Identify transactions that exploit vulnerabilities, characterized by irregular method calls, abnormal parameters, and unusual operations.

Why Existing Methods Fail

Rule-Based & Traditional ML

Rule-Based: Cannot generalize to new, unseen attack patterns.

Traditional ML (GMM, LSTM): Fail to capture the complex, high-dimensional, and sequential nature of transaction data.

Vanilla LLMs (e.g., GPT-4/5)

Wrong Tool for the Job: Off-the-shelf models don't understand the unique *data structure* of a transaction.

Tokenization Failure: They treat critical hashes (like `0x9fa0bc94`) and large numbers as generic text, losing all semantic meaning.

Our Solution: BlockScan

A Customized Transformer for Transactions

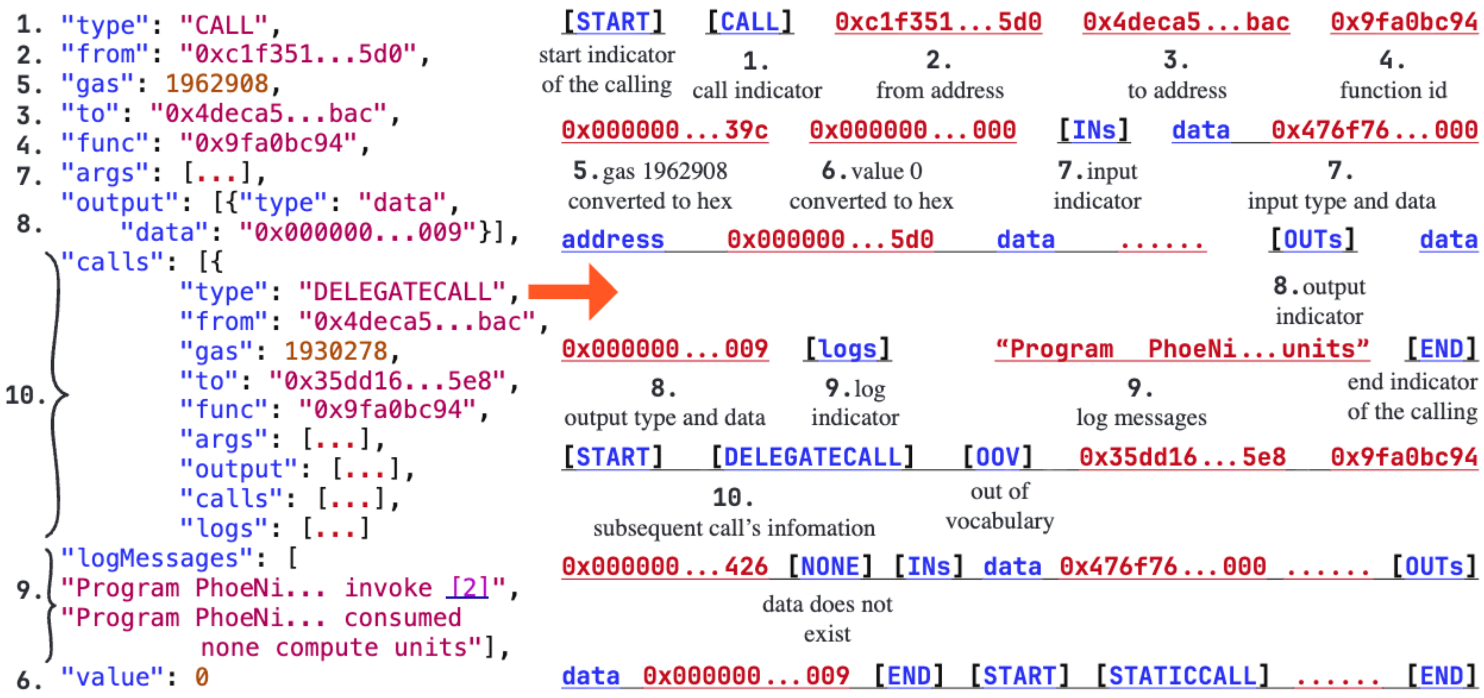
A BERT-style (RoBERTa) model, pretrained with Masked Language Modeling (MLM), to learn the "normal" patterns of benign transactions. Anomalies are transactions the model fails to reconstruct.

Core Challenge: Data is Multi-Modal

A blockchain transaction isn't just text. It's a complex, multi-modal structure.

- **Hashes/Addresses:** e.g., `0x4deca5...`
- **Large Numbers:** e.g., `1962908` (gas)
- **Text Logs:** e.g., "Program... consumed..."

Key Design 1: Multi-Modal Tokenizer



Key Design 2:Model & Detection

1. Pre-training (MLM)
We train a RoBERTa-style model on millions of *benign* transactions.

2. Detection
Anomalies = High Reconstruction Error.

Experimental Results

Method	k=10			k=15			k=20		
	FPR	**Recall**	**Precision**	**FPR**	**Recall**	**Precision**	**FPR**	**Recall**	**Precision**
BlockGPT	0.47%	16.67%	30%	0.73%	22.22%	26.67%	1%	27.78%	25%
Doc2Vec	0.67%	0%	0%	1%	0%	0%	1.13%	0%	0%
GPT-4o	0.67%	0%	0%	1%	0%	0%	1.13%	0%	0%
Heuristic	0.67%	0%	0%	1%	0%	0%	1.13%	0%	0%
`BlockScan`	**0.13%**	**44.44%**	**80%**	**0.2%**	**66.67%**	**80%**	**0.47%**	**72.22%**	**65%**

Table 1: **Performance comparison with different k values for Solana.**

Method	k=5			k=10			k=15		
	FPR	**Recall**	**Precision**	**FPR**	**Recall**	**Precision**	**FPR**	**Recall**	**Precision**
BlockGPT	0.14%	40%	80%	0.42%	70%	70%	0.99%	80%	53.33%
Doc2Vec	0.56%	10%	20%	1.12%	20%	20%	1.83%	20%	13.33%
GPT-4o	0.28%	30%	60%	0.98%	30%	30%	1.55%	40%	26.67%
Heuristic	0.14%	40%	80%	0.42%	70%	70%	1.13%	70%	46.67%
`BlockScan`	**0%**	**50%**	**100%**	**0.28%**	**80%**	**80%**	**0.97%**	**80%**	**53.33%**

Table 2: **Performance comparison with different k values for Ethereum.**

Thank you!

Code: https://github.com/nuwuxian/tx_fm