Detecção de Ataques de Front-Running na Blockchain Ethereum aplicando técnicas de Mineração de Dados e Aprendizado de Máquina

by Marcelo Corni Alves

Compreendendo os Ataques de Front-Running

1 Definição

Ataques de front-running ocorrem quando um agente mal-intencionado antecipa transações legítimas, pagando taxas mais altas para priorizar suas transações na Blockchain.

2 Tipos de Ataques

▶️ Deslocamento: A transação do atacante substitui a transação da vítima.

3 Tipos de Ataques

▶️ Inserção: A transação do atacante é inserida antes da transação da vítima.

4 Tipos de Ataques

▶️ Supressão: A transação do atacante impede que a transação da vítima seja confirmada.

Importância

Detectar esses ataques é essencial para manter a integridade da Blockchain Ethereum e abrir campo para que sejam implentadas soluções de mitigação.

Revisão dos artigos e motivação

FRAD: Front-Running Attacks Detection on Ethereum Using Ternary Classification Model

Contribuição

Modelo de classificação ternária que categoriza ataques em deslocamento, inserção e supressão, utilizando dados do Frontrunner Jones and the Raiders of the Dark Forest.

Motivação

Permite uma detecção mais granular e precisa dos diferentes tipos de ataques.

Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain

Contribuição

Análise empírica detalhada de mais de 11 milhões de blocos, identificando quase 200 mil ataques.

Motivação

Insights sobre as características dos ataques e padrões de comportamento dos bots.

Desafios na Mineração de Dados na Blockchain Ethereum

Volume de dados

A Ethereum gera um volume massivo de dados, tornando o processamento em tempo real desafiador.

Complexidade dos dados

As transações podem envolver múltiplas interações com contratos inteligentes.

Exigências de tempo real

A detecção eficaz de ataques requer análise quase instantânea.

Pipeline Proposto

1

Coleta de Dados

Início com dados das transações a partir do primeiro bloco da era POS (The Merge).

2

Pré-processamento

Extração e normalização de características como taxa de transação e uso de gas.

3

Autoencoder

▶️ Autoencoder: Reduz a dimensionalidade dos dados, capturando as características mais relevantes.

4

Isolation Forest

▶️ Isolation Forest: Detecta outliers eficientemente, essencial para identificar transações suspeitas.

5

BayesianOptimization

▶️Seleção dos melhores hiperparâmetros para otimizar o desempenho dos modelos.

6

Pós-processamento

▶️Codificação de um classificador para detectar os 3 tipos de Front-Running.

Por que Autoencoder + Isolation Forest?

▶️ Autoencoder: Capaz de aprender representações complexas e reduzir a dimensionalidade dos dados.

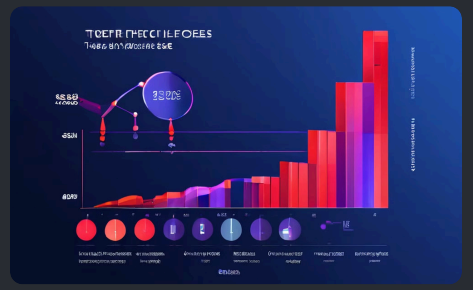
▶️ Isolation Forest: Especializado em isolar rapidamente as anomalias (outliers).

▶️ Combinação das técnicas: Detecção em conjuntos de dados grandes e complexos.

▶️ BayesianOptimization: Garantir a escolha dos melhors hiper parâmetros para os modelos.

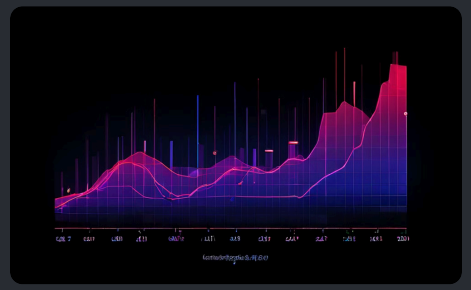
Sistema de Inferência Nebulosa (SIN)

▶️ Exemplos de variáveis extraídas



Taxas de transação

Valores elevados podem indicar tentativas de front-running.

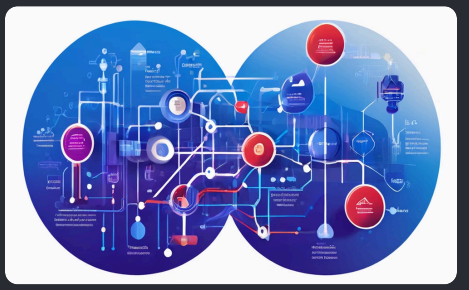


Padrões de uso de gas

Picos anormais no uso de gas podem sinalizar manipulação.

Tempo entre transações

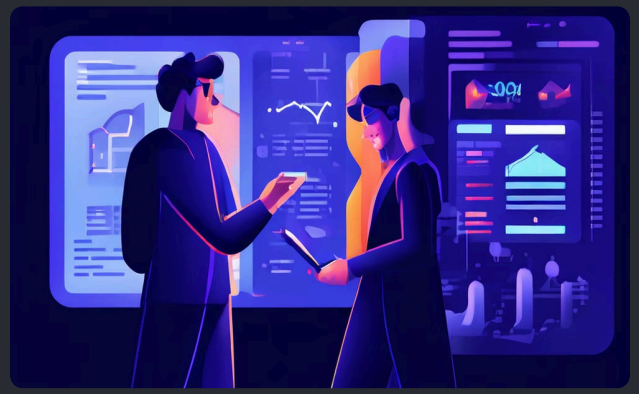
Intervalos curtos podem indicar coordenação entre transações.



Complexidade da transação

Transações complexas são mais suscetíveis a ataques.

▶️ Exemplos de regras para o SIN



▶️ Regra 1

Uma taxa de transação alta e um uso elevado de gas podem indicar um risco de front-running.

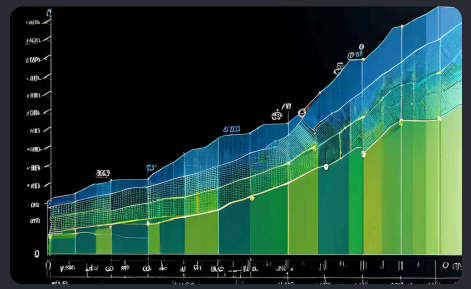
▶️ Regra 2

Um tempo curto entre transações consecutivas pode indicar uma possível inserção maliciosa.

▶️ Regra 3

Um contrato complexo com uma taxa de transação baixa pode indicar um risco de supressão.

Resultados esperados e trabalhos futuros



▶️ Melhoria na Detecção

Detecção mais precisa e em tempo real de ataques de front-running.

▶️ Redução de Falsos Positivos

Redução significativa de falsos positivos com Autoencoder e Isolation Forest.

▶️ Criar base para um Sistema de Inferência Nebulosa

Criação de um SIN eficiente e adaptável.

▶️ Expansão do Dataset

Continuar a coleta de dados para incluir novas variantes de ataques.

▶️ Testes em outras Blockchains

Aplicar o pipeline em outras Blockchains para validar sua eficácia.

▶️ Integração com Plataformas DeFi

Prevenção proativa de ataques em plataformas DeFi.

▶️ Resumo Da Abordagem

▶️ Utilização de um pipeline para a detecção de ataques de front-running na Blockchain Ethereum.

▶️ A combinação de Autoencoder com Isolation Forest foi proposta para melhorar a detecção de anomalias.

▶️BayesianOptimization irá garantir aseleção dos melhores hiper parâmetros para os modelos.

▶️ Impacto Potencial

▶️ O uso dessas técnicas avançadas pode aumentar significativamente a segurança da Ethereum, prevenindo ataques de front-running em tempo real.

▶️ A proposta de um Sistema de Inferência Nebulosa (SIN) oferece uma abordagem flexível e adaptável para a detecção e possível mitigação de anomalias.

Referências

[1] Christof Torres. Frontrunner-Jones. https: //github.com/christoftorres/Frontrunner-Jones.

[2] Christof Ferreira Torres, Ramiro Camino, et al. "Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain". In: 30th USENIX Security Symposium (USENIX Security 21). 2021, pp. 1343–1359.

[3] Yuheng Zhang et al. "FRAD: Front-Running Attacks Detection on Ethereum Using Ternary Classification Model".

In: *International Conference on Ubiquitous Security*. Springer. 2023, pp. 63–75.