

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

Marcelo Corni Alves  
Universidade Federal de Juiz de Fora  
marcelo.corni@estudante.ufjf.br



Figure 1: Front-runner. (Fonte: Bing Image Creator,2024)

## RESUMO

Neste trabalho, foi proposta uma pipeline para a detecção de ataques de Front-Running, utilizando dados da Blockchain Ethereum, com foco nas transações realizadas entre 01/01/2024 e 07/01/2024. A pipeline envolve as etapas de pré-processamento de dados, processamento com técnicas de mineração de dados e aprendizado de máquina, como Autoencoder e Isolation Forest, e avaliação dos resultados através de gráficos e métricas. A detecção dos ataques do tipo supressão, deslocamento e inserção foi identificada como um desafio de pós-processamento futuro, sendo o foco de adaptação de algoritmos presentes na literatura, como o código em Python de Frontrunner Jones and the Raiders of the Dark Forest[3]. Os resultados iniciais mostram um potencial significativo no uso de técnicas automatizadas para detecção de anomalias na Blockchain.

## PALAVRAS-CHAVE

Blockchain, Ethereum, Front-Running, Anomalias, Autoencoder, Isolation Forest

## Referência:

Marcelo Corni Alves. 2024. Detecção de Ataques de Front-Running na Blockchain Ethereum aplicando técnicas de Mineração de Dados e Aprendizado de Máquina. Universidade Federal de Juiz de Fora, 2024, 5 páginas. <https://github.com/marcelocorni/tec-report-dm-ml/>

## 1 INTRODUÇÃO

A Blockchain Ethereum, amplamente utilizada para transações financeiras, contratos inteligentes e aplicações descentralizadas, tem atraído crescente atenção devido à sua versatilidade e inovação. No entanto, essa mesma popularidade trouxe à tona vulnerabilidades exploradas por agentes maliciosos, sendo os ataques de Front-Running uma das principais ameaças. Esses ataques consistem em manipular a ordem das transações na Blockchain, visando obter vantagens financeiras ao prever e se antecipar a transações de outros usuários. A literatura destaca três tipos principais de ataques: supressão, deslocamento e inserção[1].

A prática do Front-Running foi inicialmente identificada em mercados financeiros tradicionais, mas tem evoluído de maneira alarmante no ecossistema das Finanças Descentralizadas (DeFi) devido à natureza pública e imutável da Blockchain. O artigo *SoK: Transparent Dishonesty: Front-Running Attacks on Blockchain*[2] demonstra

como a transparência das transações no Ethereum permite a exploração por agentes maliciosos, especialmente em momentos de alta volatilidade e congestão da rede. Esse cenário é ainda mais agravado pelo uso de bots automatizados que detectam e agem rapidamente sobre transações, como explorado no artigo *Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability*[1], que descreve um dos ataques mais conhecidos na prática de Front-Running, o sandwich attack.

Embora várias abordagens de mitigação tenham sido propostas, como o uso de contratos inteligentes otimizados e soluções de segunda camada, o monitoramento de transações em tempo real e a detecção de padrões maliciosos permanecem desafios críticos. O artigo *Frontrunner Jones and the Raiders of the Dark Forest: An Empirical Study of Frontrunning on the Ethereum Blockchain*[4] introduz uma série de técnicas para combater esses ataques, utilizando aprendizado de máquina e análise de dados para identificar comportamentos atípicos. A proposta de algoritmos como Frontrunner Jones[3] é um exemplo de como métodos automatizados podem ser aplicados para detectar ataques de supressão, deslocamento e inserção.

Com o aumento da utilização de redes descentralizadas, combinado com a complexidade crescente das transações, a necessidade de ferramentas eficazes para o monitoramento e detecção de ataques se torna ainda mais evidente. Este trabalho visa contribuir para essa lacuna, desenvolvendo uma pipeline automatizada para identificar possíveis comportamentos maliciosos e otimizar a segurança na rede Ethereum. A abordagem combina técnicas de aprendizado de máquina, como Autoencoder e Isolation Forest, com foco na detecção de anomalias em grandes volumes de transações, conforme sugerido em estudos anteriores sobre detecção de anomalias na Blockchain.

2 METODOLOGIA E RESULTADOS

A pipeline desenvolvida segue quatro etapas principais: Pré-processamento, Processamento, Avaliação e Pós-Processamento.

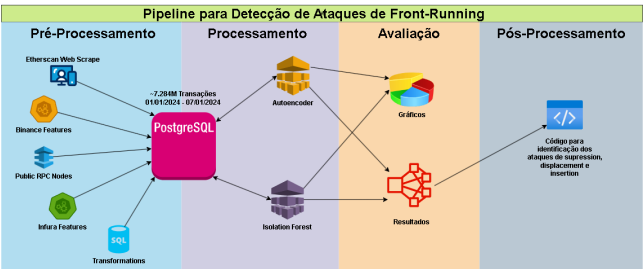


Figure 2: Pipeline para Detecção de Ataques de Front-Running

2.1 Pré-Processamento

As fontes de dados utilizadas incluem o Etherscan Web Scrape, características da Binance, e APIs de nós públicos de RPC e Infura. Esses dados fornecem informações sobre mais de 7.28 milhões de transações, realizadas entre 01/01/2024 e 07/01/2024, armazenadas em uma base de dados PostgreSQL. Nesse estágio, foram realizadas transformações para unificar os diferentes dados e integrá-los ao

banco de dados para as etapas subsequentes. Abaixo são apresentadas as distribuições dos dados e de algumas features.

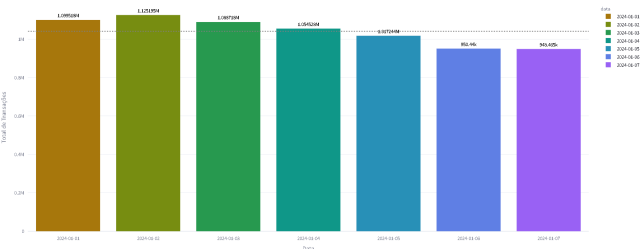


Figure 3: Quantidade de Transações por Dia

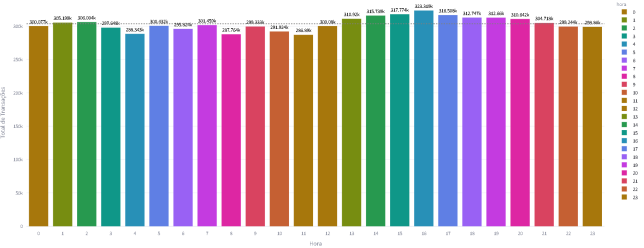


Figure 4: Quantidade de Transações Agrupadas por Hora do Dia

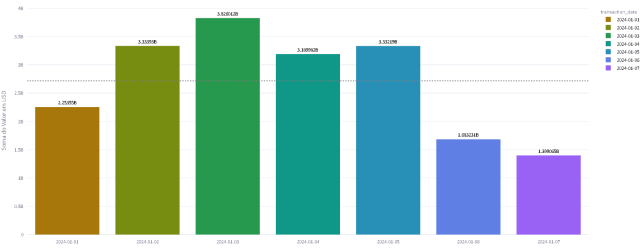


Figure 5: Valor Total em USD por Dia

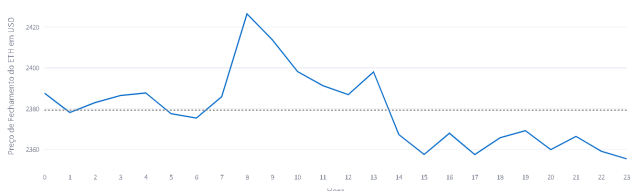


Figure 6: Variação Máxima do Preço de Fechamento do Ethereum por Hora do Dia

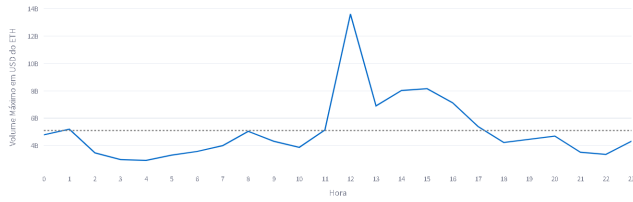


Figure 7: Volume Máximo em USD do Ethereum por Hora do Dia

## 2.2 Processamento

As técnicas de aprendizado de máquina escolhidas foram Autoencoder e Isolation Forest, ambas adequadas para detectar padrões anômalos. O Autoencoder foi utilizado para reduzir a dimensionalidade e extrair características relevantes, bem como para detecção de anomalias baseadas no erro de reconstrução. O Isolation Forest focou na detecção de anomalias. Ambas as técnicas apresentam resultados onde podem existir possíveis ataques de Front-Running. Cada modelo foi treinado com os dados pré-processados para identificar comportamentos anômalos nas transações.

Para o processamento foram selecionadas 12 features que tiveram melhores correlações para uma melhor qualidade nos resultados de detecção de padrões anômalos.

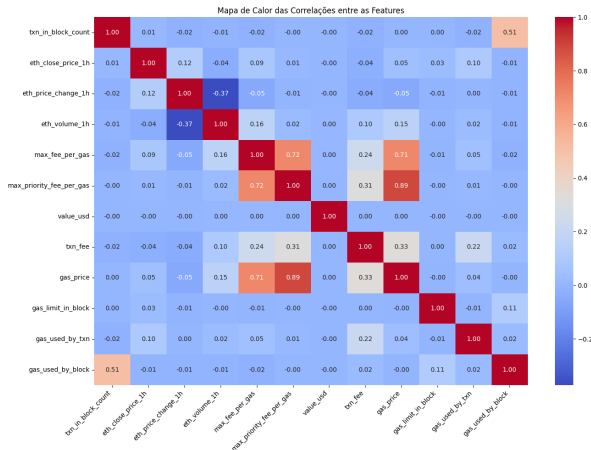


Figure 8: Mapa de Calor da Correlação entre as Features

O Isolation Forest foi parametrizado com *max\_samples* igual 10% do total dos dados de transações, *estimators* igual à 233 e *contamination* igual 0.001. O tempo total para o processamento foi de 11 minutos e 37 segundos para 7.14 milhões de transações com status igual a *Success*.

O Autoencoder foi parametrizado com *input\_dim* igual à 12 features de entrada, *latent\_space\_dim* igual à 38, *learning\_rate* igual à 0.007, *batch\_size* igual à 32 e *epochs* igual à 95. O tempo total para

o processamento foi de 3 horas, 56 minutos e 40 segundos para 7.14 milhões de transações com status igual à *Success*.

Ambos os algoritmos tiveram separação de dados para treino, teste/validação na razão 70/30.

## 2.3 Avaliação

A avaliação dos resultados foi realizada com base em gráficos e relatórios gerados. As transações identificadas como anômalas foram marcadas para posterior análise. Além disso, foram gerados gráficos que mostram a distribuição das anomalias por feature, oferecendo insights visuais sobre os padrões detectados.

### 2.3.1 Isolation Forest.

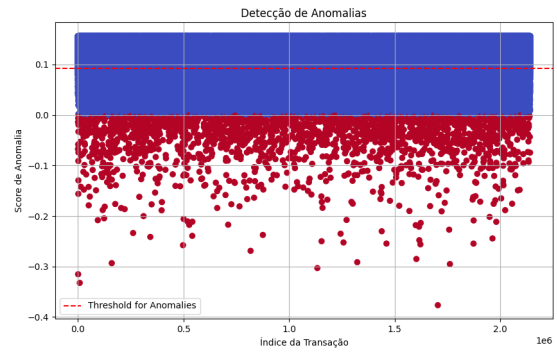


Figure 9: Detecção de anomalias

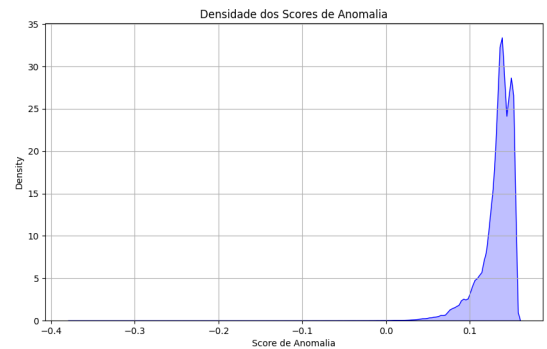


Figure 10: Densidade dos Scores de Anomalia

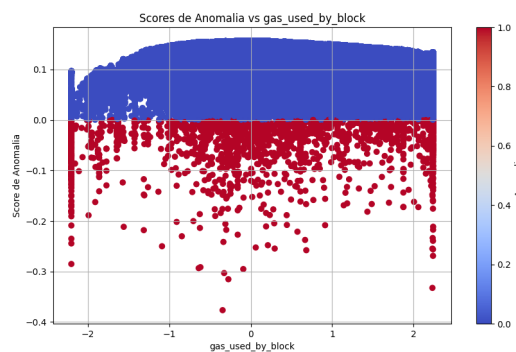


Figure 11: Scores de Anomalia vs gas\_used\_by\_block

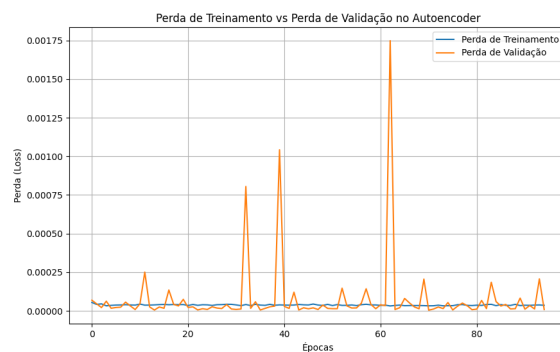


Figure 14: Perda de Treinamento vs Perda de Validação

### 2.3.2 Autoencoder.

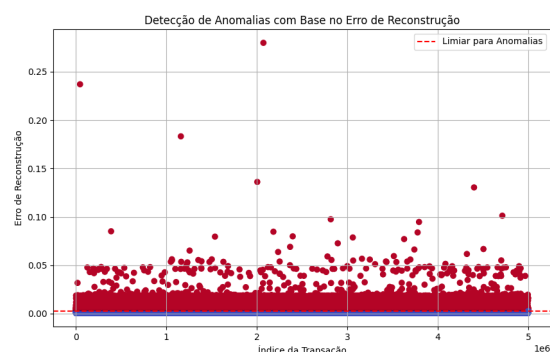


Figure 12: Detecção de anomalias por Erro de Reconstrução

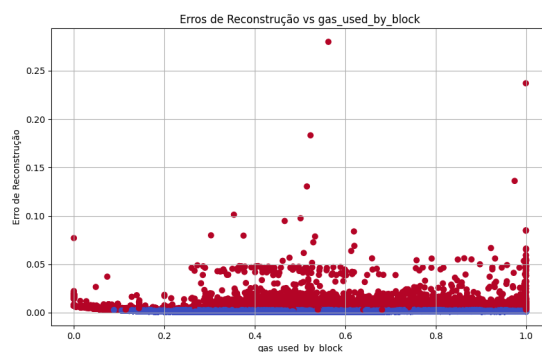


Figure 15: Erros de Reconstrução vs gas\_used\_by\_block

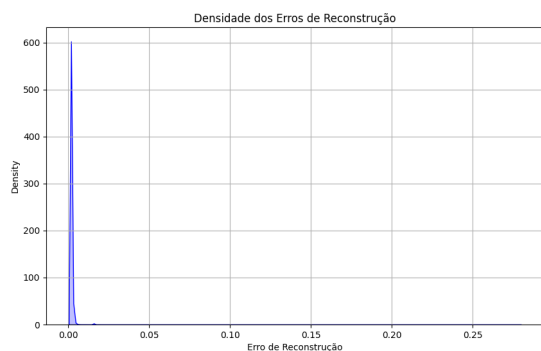


Figure 13: Densidade dos Erros de Reconstrução

## 3 CONCLUSÃO

Neste trabalho, foi apresentada uma pipeline para a detecção de ataques de Front-Running na Blockchain Ethereum, utilizando técnicas de mineração de dados e aprendizado de máquina. O pré-processamento dos dados, provenientes de fontes como Etherscan e APIs de nós públicos e principalmente da API do Infura, que foi parte fundamental para o preenchimento das features, permitiu a aplicação de técnicas de detecção de anomalias em transações realizadas no período de análise. As análises geraram gráficos detalhados sobre a distribuição das anomalias e padrões detectados.

Os resultados demonstraram a eficácia das técnicas aplicadas na detecção de comportamentos anômalos, como possível Front-Running, em um volume significativo de transações (7,14 milhões). O uso de Autoencoder mostrou-se eficiente na redução de dimensionalidade e no reconhecimento de anomalias com base em erros de reconstrução, enquanto o Isolation Forest se destacou na identificação de transações isoladas com características suspeitas.

Uma das principais frentes para trabalhos futuros está na expansão do dataset utilizado. Embora o presente estudo tenha abordado

transações realizadas em um período restrito, a análise de um intervalo temporal mais extenso ou de diferentes contextos econômicos e de rede (como altas variações no preço do Ethereum ou momentos de congestão da rede) pode trazer considerações adicionais. Isso permitiria um melhor entendimento sobre os padrões de ataques de Front-Running e a evolução de técnicas de ataque.

Além disso, outro aspecto fundamental para o aprimoramento deste trabalho é a codificação de classificadores, que corresponde à etapa de Pós-Processamento da pipeline, baseados na lógica descrita no repositório de código de *Frontrunner Jones and the Raiders of the Dark Forest*[4]. A adaptação e integração desses classificadores, que se concentram em ataques de supressão, deslocamento e inserção, podem tornar o sistema de detecção mais especializado. Implementar tais abordagens deverá permitir que o sistema detecte tipos específicos de ataques com maior precisão e menor taxa de falsos positivos, complementando as técnicas de aprendizado de máquina já aplicadas.

Essa combinação entre a expansão do dataset e o uso de classificadores mais sofisticados, baseados em heurísticas pré-existentes sobre a lógica dos ataques de Front-Running, poderá agregar maior aplicabilidade prática ao sistema desenvolvido(16).

## REFERENCES

- [1] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2020. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE symposium on security and privacy (SP)*. IEEE, 910–927.

- [2] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. 2020. Sok: Transparent dishonesty: front-running attacks on blockchain. In *Financial Cryptography and Data Security: FC 2019 International Workshops, VOTING and WTSC, St. Kitts, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*. Springer, 170–189.
- [3] Christof Torres. [n. d.]. Frontrunner-Jones. <https://github.com/christoftorres/Frontrunner-Jones>.
- [4] Christof Ferreira Torres, Ramiro Camino, et al. 2021. 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)*. 1343–1359.

## APÊNDICE

### Protótipo do Sistema de Classificação



Figure 16: Classificador de Ataques de Front-Running