

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Marcelo Corni Alves

**Detecção de Anomalias Transacionais na Ethereum com Aprendizado Não
Supervisionado e Validação de Padrões de Front-Running**

Juiz de Fora

2026

Marcelo Corni Alves

Detecção de Anomalias Transacionais na Ethereum com Aprendizado Não Supervisionado e Validação de Padrões de Front-Running

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Saulo Moraes Villela

Coorientadores: Profs. Drs. Alex Borges Vieira e Heder Soares Bernardino

Juiz de Fora

2026

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Corni Alves, Marcelo.

Detecção de Anomalias Transacionais na Ethereum com Aprendizado
Não Supervisionado e Validação de Padrões de Front-Running / Marcelo
Corni Alves. – 2026.

113 f. : il.

Orientador: Saulo Moraes Villela

Coorientadores: Alex Borges Vieira e Heder Soares Bernardino

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade
de Ciência da Computação. Pós-Graduação em Ciência da Computação,
2026.

Marcelo Corni Alves

Detecção de Anomalias Transacionais na Ethereum com Aprendizado Não Supervisionado e Validação de Padrões de Front-Running

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. Saulo Moraes Villela - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Alex Borges Vieira - Coorientador
Universidade Federal de Juiz de Fora

Prof. Dr. Heder Soares Bernardino - Coorientador
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

A realização deste trabalho representa não apenas a conclusão de uma etapa acadêmica, mas o resultado de um esforço coletivo, sustentado por apoio, compreensão e generosidade de pessoas fundamentais ao longo dessa jornada.

Agradeço, em primeiro lugar, à minha esposa, Stéphanie, cujo apoio foi absolutamente essencial para que este trabalho se tornasse possível. Ao longo de todo o período desta pesquisa, foram inúmeras noites em que precisou fazer as crianças dormirem sozinha, muitas vezes enquanto eu permanecia em aulas online, reuniões acadêmicas ou atividades de pesquisa. Além disso, assumiu de forma integral diversas rotinas domésticas que antes eram compartilhadas, sustentando o cotidiano da família para que eu pudesse me dedicar a este projeto. Sua paciência, força e compreensão diante das ausências, do cansaço e dos desafios diários foram decisivas. Nada do que foi realizado neste trabalho teria sido possível sem o seu esforço constante, visível e profundamente significativo.

Ao meu orientador, Saulo Moraes Villela, deixo meu sincero agradecimento pelo apoio incondicional ao longo de todo o percurso. Sua compreensão diante dos desafios impostos pela rotina profissional, pelas questões de saúde e pelos momentos emocionalmente difíceis foi decisiva para que eu não desistisse. Seus ensinamentos em redes neurais ampliaram de forma significativa minha visão acadêmica e permitiram aprofundar a utilização do Autoencoder neste trabalho, elevando a qualidade e o alcance da pesquisa desenvolvida.

Agradeço ao coorientador Heder Soares Bernardino pelos valiosos ensinamentos que possibilitaram uma coleta de dados conduzida com rigor e excelência. Sua contribuição foi fundamental para a aplicação adequada dos algoritmos, em especial do Isolation Forest, fornecendo a base técnica necessária para a construção de uma análise consistente e confiável.

Ao coorientador Alex Borges Vieira, expresso minha gratidão pela excelência nos ensinamentos relacionados aos métodos quantitativos. Seu apoio foi essencial para a representação dos dados de forma mais clara, objetiva e legível, contribuindo diretamente para a compreensão dos resultados e para a solidez das análises apresentadas.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho. Cada incentivo, cada palavra de apoio e cada gesto de compreensão foram fundamentais para transformar desafios em aprendizado e esforço em conquista.

RESUMO

A Ethereum constitui um ambiente transacional de grande volume, heterogeneidade e evolução contínua, no qual a ausência de rótulos abrangentes dificulta distinguir comportamentos incomuns de alterações temporais na distribuição das transações. Esta dissertação investiga a detecção de comportamentos transacionais anômalos utilizando exclusivamente atributos observáveis na *blockchain* Ethereum. Foi desenvolvido um pipeline reprodutível de detecção não supervisionada com dados públicos da rede principal referentes a 2024 e 2025. Uma amostra determinística de aproximadamente 5% dos blocos, preservando todas as transações dos blocos selecionados e totalizando 47.545.805 transações, foi organizada em sete janelas cronológicas de treinamento, validação, teste, transição e estresse associadas às atualizações Dencun, Pectra e Fusaka. Autoencoder e Isolation Forest foram avaliados em cinco configurações de atributos *on-chain*, incluindo um estudo de ablação das taxas monetárias absolutas. As configurações completas apresentaram aumento das taxas de sinalização nos períodos posteriores, enquanto a remoção das taxas absolutas reduziu essa variação em determinadas configurações. A concordância reduzida entre os conjuntos sinalizados indicou que os modelos priorizam conjuntos parcialmente distintos de transações. Em uma avaliação externa independente do treinamento, os escores foram comparados com rótulos hipotéticos derivados de endereços classificados como bots de MEV no Label Cloud. No teste final, o Isolation Forest alcançou ROC-AUC de até 0,9538 e o Autoencoder, 0,8526. Esses rótulos constituem evidência auxiliar: sua presença não comprova ataque, nem sua ausência, normalidade. Regras estruturais adaptadas da literatura foram aplicadas para gerar candidatos aos padrões de inserção, deslocamento e supressão. Os candidatos foram enriquecidos por RPC com *calldata*, recibos, logs e metadados de tokens e submetidos à validação semântica e à adjudicação automatizadas. Considerando somente eventos confirmados na política estrita, as configurações selecionadas na validação alcançaram ROC-AUC de 0,6779 para o Autoencoder e 0,7503 para o Isolation Forest no teste final. O atacante recebeu escore superior ao da vítima em 71,26% dos pares no Autoencoder e em 78,50% no Isolation Forest. Essas métricas se referem à população de atacantes e vítimas contextuais dos eventos identificados, sem estimar desempenho sobre a totalidade da rede. Os resultados favoráveis foram concentrados no padrão de inserção; os casos de deslocamento apresentaram ordenação inversa dos escores, e não houve eventos de supressão confirmados disponíveis para avaliação. Conclui-se que atributos exclusivamente *on-chain* produzem sinais úteis para priorizar investigações, sobretudo de inserção, mas sua interpretação exige evidências estruturais e semânticas e consideração das mudanças temporais e da cobertura dos rótulos.

Palavras-chave: Ethereum; detecção de anomalias; aprendizado não supervisionado; MEV; *front-running*.

ABSTRACT

Ethereum is a high-volume, heterogeneous, and continuously evolving transactional environment in which the lack of comprehensive labels makes it difficult to distinguish unusual behavior from temporal shifts in transaction distributions. This dissertation investigates the detection of anomalous transactional behavior using exclusively attributes observable on the Ethereum *blockchain*. A reproducible unsupervised detection pipeline was developed using public mainnet data from 2024 and 2025. A deterministic sample of approximately 5% of blocks, retaining all transactions within the selected blocks and comprising 47,545,805 transactions, was organized into seven chronological training, validation, testing, transition, and stress windows associated with the Dencun, Pectra, and Fusaka upgrades. Autoencoder and Isolation Forest were evaluated across five configurations of *on-chain* features, including an ablation study of absolute monetary fee attributes. The full configurations exhibited increased flagging rates in later periods, whereas removing absolute fee attributes reduced this variation in certain configurations. Limited agreement between the flagged sets indicated that the models prioritize partially distinct subsets of transactions. In an external evaluation independent of training, scores were compared against hypothetical labels derived from addresses classified as MEV bots in Label Cloud. On the final test set, Isolation Forest achieved a ROC-AUC of up to 0.9538 and Autoencoder, 0.8526. These labels provide auxiliary evidence: their presence does not establish an attack, nor does their absence establish normality. Structural rules adapted from the literature were applied to identify candidates for insertion, displacement, and suppression patterns. Candidates were enriched through RPC with *calldata*, receipts, logs, and token metadata, then subjected to automated semantic validation and adjudication. Under the strict policy, which includes only confirmed events, the configurations selected on the validation set achieved final-test ROC-AUC values of 0.6779 for Autoencoder and 0.7503 for Isolation Forest. Attackers received higher scores than victims in 71.26% of pairs for Autoencoder and 78.50% for Isolation Forest. These metrics concern attackers and contextual victims within the identified events and do not estimate network-wide performance. Favorable results were concentrated in insertion events; displacement cases exhibited reversed score ordering, and no confirmed suppression events were available for evaluation. The findings indicate that exclusively *on-chain* attributes yield useful signals for prioritizing investigations, particularly of insertion events. Their interpretation nevertheless requires structural and semantic evidence, together with consideration of temporal shifts and label coverage.

Keywords: Ethereum; anomaly detection; unsupervised learning; MEV; *front-running*.

LISTA DE FIGURAS

Figura 1 – Fluxo de uma transação Ethereum à obtenção de atributos <i>on-chain</i> . .	18
Figura 2 – Obtenção do escore de anomalia pelo erro de reconstrução do Autoencoder.	24
Figura 3 – Obtenção do escore de anomalia pelo princípio de isolamento do Isolation Forest.	26
Figura 4 – Padrões estruturais de inserção, deslocamento e supressão em <i>front-running</i> . 30	
Figura 5 – Pipeline metodológico do trabalho.	38
Figura 6 – Correlação de Pearson entre os atributos no período de treinamento. .	43
Figura 7 – Correlação de Spearman entre os atributos no período de treinamento. 44	
Figura 8 – Perdas de treinamento e validação do Autoencoder.	87
Figura 9 – Sinalização do Autoencoder por configuração e janela no corte $Q_{99,5}$. 88	
Figura 10 – Quantis dos escores do Isolation Forest. Recortes: 1 – Pré-Dencun; 2 – Transição Dencun; 3 – Treinamento Pós-Dencun; 4 – Validação Pré-Pectra; 5 – Transição Pectra; 6 – Teste final; 7 – Fusaka.	90
Figura 11 – Curvas de sobrevivência dos escores do Isolation Forest. A numeração dos recortes corresponde à apresentada na Figura 10.	90
Figura 12 – Sinalização do Isolation Forest por configuração e janela no corte $Q_{99,5}$. 91	
Figura 13 – Sinalização do Autoencoder e Isolation Forest no limiar $Q_{99,5}$	92
Figura 14 – Curvas ROC e <i>Precision-Recall</i> da avaliação com o Label Cloud no teste final.	94
Figura 15 – <i>Lift</i> nas frações superiores dos escores no teste final.	95
Figura 16 – Curvas ROC e <i>Precision-Recall</i> da política estrita no teste final. . . .	98
Figura 17 – Superioridade dos escores dos atacantes sobre as vítimas na política estrita.	99

LISTA DE TABELAS

Tabela 1 – Comparação com trabalhos sobre anomalias e fraudes em <i>blockchain</i>	32
Tabela 2 – Comparação com trabalhos sobre <i>front-running</i>	34
Tabela 3 – Grupos de informações disponíveis nos dados de origem.	39
Tabela 4 – Recortes temporais do desenho experimental na amostra de aproximadamente 5% dos blocos.	40
Tabela 5 – Atributos das matrizes experimentais.	45
Tabela 6 – Configurações de atributos dos modelos.	49
Tabela 7 – Adaptação das regras de identificação estrutural.	69
Tabela 8 – Informações recuperadas no enriquecimento RPC.	73
Tabela 9 – Situações da validação semântica.	77
Tabela 10 – Política de adjudicação dos eventos.	78
Tabela 11 – Referências semânticas consolidadas.	79
Tabela 12 – Transações por janela temporal do experimento.	85
Tabela 13 – Resumo do treinamento do Autoencoder.	86
Tabela 14 – Percentual de transações sinalizadas pelo Autoencoder no limiar $Q_{99,5}$	88
Tabela 15 – Percentual de transações sinalizadas pelo Isolation Forest no limiar $Q_{99,5}$	91
Tabela 16 – Jaccard entre Autoencoder e Isolation Forest no limiar $Q_{99,5}$	92
Tabela 17 – Avaliação com o Label Cloud no teste final.	94
Tabela 18 – Cobertura dos rótulos semânticos na validação e no teste final.	96
Tabela 19 – Configurações selecionadas e resultados no teste final.	97
Tabela 20 – Síntese das evidências do protocolo experimental.	101

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Definição do Problema	12
1.2	Objetivos	13
1.2.1	Objetivo Geral	13
1.2.2	Objetivos Específicos	14
1.3	Contribuições	14
1.4	Organização do Trabalho	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Blockchain e a Rede Ethereum	17
2.2	Comportamento Transacional e Segurança Econômica na Ethereum . .	18
2.3	Deteção de Anomalias	20
2.3.1	Conceito de Anomalia	20
2.3.2	Deteção Supervisionada e Não Supervisionada	20
2.3.3	Escore de Anomalia e Limiar de Decisão	21
2.3.4	Avaliação na Ausência de Ground Truth	21
2.4	Autoencoders	22
2.4.1	Autoencoders para Deteção de Anomalias	23
2.4.2	Representação Latente e Capacidade de Reconstrução	24
2.5	Isolation Forest	25
2.5.1	Escore de Anomalia	26
2.5.2	Relação com o Autoencoder	27
2.6	Front-Running e Padrões Relacionados na Ethereum	27
2.6.1	Front-Running	28
2.6.2	Padrão de Inserção	28
2.6.3	Padrão de Deslocamento	29
2.6.4	Padrão de Supressão	29
2.6.5	Da Estrutura Candidata à Validação Semântica	30
2.7	Trabalhos Relacionados	31
2.7.1	Deteção de Anomalias e Fraudes em Blockchain	31
2.7.2	Deteção de Front-Running e MEV	33
2.7.3	Posicionamento desta Dissertação	34
2.8	Considerações Finais	35
3	METODOLOGIA	36
3.1	Visão Geral da Metodologia	36
3.1.1	Fonte e Estrutura dos Dados	38
3.1.2	Período de Análise	39
3.2	Construção da Representação Transacional	41

3.2.1	Análise de Correlação entre Atributos	41
3.2.2	Atributos Observáveis e Atributos Derivados	45
3.2.3	Validação dos Atributos	46
3.2.4	Pré-processamento e Escalonamento	46
3.3	Configurações de Atributos e Estudo de Ablação	47
3.3.1	Configurações Completas	47
3.3.2	Remoção das Taxas Absolutas	48
3.4	Organização Temporal dos Experimentos	49
3.4.1	Separação Cronológica entre Treinamento, Validação e Teste	49
3.4.2	Recortes Temporais e Atualizações da Rede	50
3.4.3	Análise de Mudança de Distribuição	50
3.5	Treinamento dos Modelos	52
3.5.1	Autoencoder	52
3.5.2	Isolation Forest	53
3.6	Produção dos Escores e Definição dos Limiares	54
3.6.1	Escores de Anomalia	54
3.6.2	Calibração dos Limiares na Validação	54
3.6.3	Crerios Q_{99} , $Q_{99,5}$ e $Q_{99,9}$	55
3.7	Protocolo Experimental e Reprodutibilidade	56
3.7.1	Disponibilidade do Código e dos Artefatos	57
3.8	Considerações Finais	58
4	AVALIAÇÃO E VALIDAÇÃO	59
4.1	Estratégia Geral de Avaliação	59
4.2	Avaliação Intrínseca dos Escores de Anomalia	60
4.2.1	Distribuição dos Escores	60
4.2.2	Taxas de Sinalização	61
4.2.3	Estabilidade entre Recortes Temporais	61
4.3	Concordância entre Autoencoder e Isolation Forest	62
4.3.1	Comparação dos Conjuntos e Índice de Jaccard	63
4.3.2	Concordância entre Configurações	63
4.4	Avaliação Externa com o Label Cloud	64
4.4.1	Fonte e Associação dos Rótulos	64
4.4.2	Avaliação dos Escores Contínuos	65
4.4.3	Avaliação dos Limiares e do Topo do Ranqueamento	66
4.5	Validação Estrutural e Semântica dos Candidatos de Front-Running	67
4.5.1	Adaptação do Método de Identificação Estrutural	68
4.5.2	Identificação de Candidatos ao Padrão de Deslocamento	69
4.5.3	Identificação de Candidatos ao Padrão de Inserção	70
4.5.4	Identificação de Candidatos ao Padrão de Supressão	71

4.5.5	Enriquecimento dos Candidatos por RPC	72
4.5.6	Validação Semântica dos Eventos	74
4.5.6.1	Validação dos Candidatos de Inserção	74
4.5.6.2	Validação dos Candidatos de Deslocamento	75
4.5.6.3	Situações produzidas pela validação	76
4.5.7	Adjudicação e Verificação das Evidências	77
4.5.8	Consolidação dos Rótulos Semânticos	78
4.5.9	Protocolo de Avaliação com os Rótulos Semânticos	80
4.5.10	Limitações da Validação Semântica	82
4.6	Considerações Finais	83
5	RESULTADOS E DISCUSSÃO	85
5.1	Visão Geral dos Resultados Experimentais	85
5.2	Resultados do Autoencoder	86
5.2.1	Convergência do Treinamento	86
5.2.2	Taxas de Sinalização ao Longo do Tempo	87
5.2.3	Efeito da Remoção das Taxas Absolutas	88
5.3	Resultados do Isolation Forest	89
5.3.1	Ajuste dos Modelos	89
5.3.2	Distribuição Temporal dos Escores	89
5.3.3	Taxas de Sinalização	89
5.3.4	Efeito da Remoção das Taxas Absolutas	91
5.4	Comparação e Concordância entre Autoencoder e Isolation Forest	92
5.4.1	Comparação das Taxas de Sinalização	92
5.4.2	Concordância entre os Modelos	92
5.4.3	Concordância entre Configurações de Atributos	93
5.4.4	Diferenças entre os Mecanismos de Detecção	93
5.5	Resultados da Avaliação Externa com o Label Cloud	93
5.5.1	Capacidade de Ordenação dos Escores	94
5.5.2	Resultados nos Limiares Predefinidos	95
5.5.3	Enriquecimento no Topo dos Escores	95
5.5.4	Interpretação da Evidência Externa	96
5.6	Resultados da Validação Estrutural e Semântica de Front-Running	96
5.6.1	Cobertura dos Rótulos Semânticos	96
5.6.2	Seleção das Configurações na Validação	97
5.6.3	Discriminação no Teste Final	97
5.6.4	Desempenho dos Limiares de Anomalia	98
5.6.5	Concentração de Atacantes no Topo	98
5.6.6	Comparação Pareada entre Atacantes e Vítimas	99
5.6.7	Resultados por Tipo de Ataque	100

5.6.8	Síntese da Avaliação Semântica	100
5.7	Discussão Integrada dos Resultados	100
5.7.1	Resposta à Questão de Pesquisa	101
5.7.2	Estabilidade Temporal e Escolha dos Atributos	102
5.7.3	Diferenças entre Autoencoder e Isolation Forest	102
5.7.4	Convergência entre as Avaliações Externas	102
5.7.5	Ordenação em Comparação com Classificação por Limiar	103
5.7.6	Heterogeneidade entre os Padrões de Front-Running	103
5.7.7	Implicações para o Uso do Pipeline	103
5.8	Considerações Finais	103
6	CONCLUSÕES	105
6.1	Síntese da Pesquisa e Resposta à Questão Central	105
6.2	Contribuições da Pesquisa	106
6.2.1	Contribuições Metodológicas	106
6.2.2	Contribuições Experimentais	106
6.2.3	Contribuições Técnicas	107
6.3	Limitações da Pesquisa	107
6.3.1	Dados, Amostragem e Representação	107
6.3.2	Modelos e Referências Externas	107
6.3.3	Identificação e Validação de Front-Running	108
6.3.4	Reprodutibilidade e Validade Externa	108
	REFERÊNCIAS	110

1 INTRODUÇÃO

1.1 Definição do Problema

A rede Ethereum constitui um ambiente transacional de elevada escala e complexidade, no qual diferentes tipos de operações são registrados continuamente e permanecem publicamente observáveis na *blockchain*. Transferências de valor, execução de contratos inteligentes, consumo de gás, taxas, posicionamento das transações nos blocos e outras características da atividade da rede produzem um conjunto amplo de informações que pode ser utilizado para descrever o comportamento transacional sem recorrer, necessariamente, a fontes externas.

A identificação de comportamentos incomuns nesse ambiente, entretanto, apresenta desafios importantes. Diferentemente de problemas tradicionais de classificação, nos quais exemplos previamente rotulados podem ser utilizados para ensinar um modelo a reconhecer classes conhecidas, não existe na *blockchain* um rótulo nativo que estabeleça se uma transação deve ser considerada normal, anômala ou potencialmente maliciosa. Bases externas de conhecimento podem fornecer informações sobre determinados endereços ou atividades já identificadas, mas sua cobertura é limitada e a ausência de um rótulo não constitui evidência suficiente de normalidade. Dessa forma, a construção de uma verdade fundamental (*ground truth*) abrangente e confiável para esse tipo de problema é particularmente difícil.

Essa característica torna relevante a investigação de técnicas de aprendizado não supervisionado, capazes de modelar propriedades da distribuição transacional sem depender de exemplos previamente classificados. Nesse contexto, uma anomalia corresponde inicialmente a uma observação que apresenta comportamento distinto daquele aprendido ou esperado pelo modelo. Essa definição é estatística e não implica, por si só, a ocorrência de fraude, ataque ou qualquer outra atividade maliciosa. Uma transação legítima pode apresentar características raras, enquanto determinadas estratégias maliciosas podem produzir transações individualmente semelhantes às operações normais da rede.

Outro aspecto relevante é o caráter evolutivo da própria Ethereum. Durante o período considerado neste trabalho ocorreram atualizações relevantes, como Dencun, Pectra e Fusaka, que constituem marcos temporais para a análise da evolução da rede. Um modelo treinado em determinado regime da rede pode atribuir escores elevados a observações pertencentes a períodos posteriores simplesmente porque a distribuição dos atributos se modificou. Avaliar como alterações temporais da distribuição, ou *drift*, afetam os escores de anomalia e as taxas de sinalização constitui, portanto, parte do problema investigado.

Além disso, mesmo quando uma anomalia estatística é identificada, permanece o problema de sua interpretação. Comportamentos relacionados à extração de valor máximo

(*Maximal Extractable Value* – MEV), por exemplo, podem envolver relações entre múltiplas transações, contratos, eventos e posições dentro de um bloco que não são necessariamente representadas por um único escore de anomalia. Entre esses comportamentos encontram-se estratégias de *front-running* e padrões relacionados, cuja identificação exige considerar não apenas características estatísticas, mas também a ordem e a semântica das operações envolvidas.

Diante desse cenário, esta dissertação separa a detecção estatística de anomalias da construção de evidências para sua interpretação. Os modelos de aprendizado não supervisionado utilizam exclusivamente atributos derivados de informações observáveis na *blockchain*, sem rótulos externos durante o treinamento. Posteriormente, seus escores são avaliados com conhecimento externo e com uma referência semântica construída a partir de regras estruturais de *front-running*. Essas regras são aplicadas aos blocos amostrados independentemente da sinalização dos modelos, permitindo comparar os escores de atacantes e vítimas contextuais nos eventos identificados. Essa organização permite investigar a utilidade dos sinais estatísticos sem pressupor que toda transação sinalizada corresponda a um ataque.

A partir dessas considerações, a questão central que orienta esta pesquisa é:

É possível detectar comportamentos transacionais anômalos utilizando exclusivamente informações observáveis na blockchain Ethereum?

Responder a essa questão envolve avaliar a estabilidade temporal dos escores, a sensibilidade dos modelos às mudanças na distribuição dos atributos, a concordância entre diferentes técnicas e sua capacidade de priorizar transações associadas às evidências disponíveis. A avaliação com rótulos semânticos é condicionada aos eventos identificados pelas regras estruturais e não representa uma avaliação exaustiva da rede. O objetivo é, portanto, investigar em que medida atributos exclusivamente *on-chain* produzem sinais úteis para investigação, reconhecendo que uma anomalia estatística não constitui, isoladamente, comprovação de ataque.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver e avaliar um pipeline reproduzível de aprendizado não supervisionado para detectar e priorizar comportamentos transacionais anômalos na Ethereum, utilizando exclusivamente atributos *on-chain* no treinamento e investigando a estabilidade temporal dos escores, sua associação com conhecimento externo e sua capacidade de distinguir atacantes de vítimas contextuais nos eventos de *front-running* identificados e validados semanticamente.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- construir uma base experimental de transações da rede principal Ethereum mediante seleção determinística de blocos completos e separação cronológica entre treinamento, validação e teste;
- preparar e caracterizar atributos exclusivamente *on-chain*, contemplando diferentes representações do comportamento transacional;
- implementar e avaliar modelos de detecção de anomalias baseados em Autoencoder e Isolation Forest;
- analisar a distribuição dos escores e as taxas de sinalização ao longo do tempo, incluindo janelas associadas às atualizações Dencun, Pectra e Fusaka;
- investigar, por meio de ablação, o efeito da remoção das taxas monetárias absolutas sobre a estabilidade temporal e a capacidade de priorização dos modelos;
- comparar os conjuntos de transações sinalizados pelo Autoencoder e pelo Isolation Forest, avaliando sua concordância;
- avaliar a associação dos escores com rótulos hipotéticos derivados de endereços classificados como bots de MEV no Label Cloud, explicitando as limitações dessa referência externa;
- aplicar regras estruturais aos blocos amostrados para identificar candidatos aos padrões de inserção, deslocamento e supressão, independentemente da sinalização dos modelos;
- enriquecer os candidatos com dados obtidos por RPC e realizar sua validação semântica e adjudicação automatizadas;
- avaliar a discriminação e a priorização de atacantes em relação às vítimas contextuais, considerando as políticas estrita e de sensibilidade e as diferenças entre os tipos de padrão;
- disponibilizar o código, a configuração e os artefatos necessários à reprodução e analisar as limitações metodológicas e computacionais do experimento.

1.3 Contribuições

As contribuições desta dissertação concentram-se na construção e na avaliação de um pipeline de detecção não supervisionada de anomalias transacionais na Ethereum,

considerando a evolução temporal dos dados e a ausência de uma referência abrangente de ataques. Destacam-se:

- a implementação de um pipeline reproduzível com amostragem determinística de blocos completos, configuração centralizada e resultados isolados por execução, permitindo processar diferentes percentuais sem substituir os artefatos de experimentos anteriores;
- a construção de um desenho experimental com separação cronológica entre treinamento, validação e teste e janelas associadas às atualizações Dencun, Pectra e Fusaka, preservando o ajuste do pré-processamento no treinamento e a seleção de configurações e limiares na validação;
- a avaliação de Autoencoder e Isolation Forest em cinco configurações de atributos *on-chain*, incluindo a ablação das taxas monetárias absolutas, para investigar estabilidade temporal, discriminação e concordância entre os conjuntos sinalizados;
- a avaliação externa dos escores com rótulos hipotéticos derivados do Label Cloud, mantendo essas informações fora do treinamento e explicitando que a ausência de classificação de um endereço não comprova comportamento normal;
- a adaptação de regras estruturais da literatura para identificar candidatos aos padrões de inserção, deslocamento e supressão independentemente dos escores dos modelos, distinguindo a identificação de candidatos de sua confirmação semântica;
- a implementação de enriquecimento RPC retomável, com cache compartilhado entre execuções e distribuição concorrente de janelas entre provedores, integrado à recuperação de evidências, à geração de dossiês e à adjudicação automatizada;
- a construção de uma referência semântica com papéis transacionais de atacante e vítima contextual, acompanhada de políticas estrita e de sensibilidade e de avaliação agregada, por tipo de padrão e por comparação pareada dentro dos eventos;
- a disponibilização do código, dos manifestos, dos resultados e de um painel de consulta, permitindo examinar as evidências e os limites das métricas produzidas pelo experimento.

Essas contribuições permitem avaliar a utilidade dos escores de anomalia sem equipará-los à comprovação de ataques. A interpretação dos resultados considera tanto as mudanças temporais na distribuição dos atributos quanto a cobertura da referência semântica, cujas métricas são condicionadas aos eventos identificados pelas regras estruturais.

1.4 Organização do Trabalho

Esta dissertação está organizada em seis capítulos, que apresentam o problema de pesquisa, a fundamentação teórica, a metodologia, os procedimentos de avaliação, os resultados e as conclusões.

O Capítulo 1 apresenta o problema de pesquisa, os objetivos e as principais contribuições do trabalho.

O Capítulo 2 aborda os conceitos de *blockchain*, Ethereum, detecção de anomalias e aprendizado não supervisionado, com ênfase no Autoencoder e no Isolation Forest. Também apresenta MEV, os padrões de inserção, deslocamento e supressão e os trabalhos relacionados à pesquisa.

O Capítulo 3 descreve a obtenção dos dados, a seleção determinística de blocos completos, a construção dos atributos, o pré-processamento e as configurações de ablação. Apresenta a separação cronológica dos conjuntos, o treinamento dos modelos e a calibração dos limiares, além da configuração das execuções e dos mecanismos de reprodutibilidade.

O Capítulo 4 descreve a análise dos escores, das taxas de sinalização e da concordância entre os modelos, bem como a avaliação exploratória com o Label Cloud. Detalha as regras estruturais, o enriquecimento RPC com cache compartilhado e processamento concorrente, a validação semântica e a adjudicação automatizada. Define ainda as políticas estrita e de sensibilidade e os procedimentos de avaliação dos escores com os papéis transacionais identificados.

O Capítulo 5 apresenta os resultados da execução com aproximadamente 5% dos blocos, incluindo estabilidade temporal, ablação, concordância entre os modelos e avaliações externa e semântica. Discute as diferenças entre os tipos de padrão e o alcance das evidências obtidas.

Por fim, o Capítulo 6 retoma a questão de pesquisa, sintetiza as contribuições e discute as limitações metodológicas e computacionais e as possibilidades de continuidade da pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os fundamentos necessários à compreensão da abordagem proposta. Inicialmente, são discutidas características da *blockchain* Ethereum relevantes à representação e à ordenação das transações, bem como aspectos de segurança econômica associados a esse ambiente. Em seguida, são apresentados os conceitos de detecção de anomalias e os métodos não supervisionados utilizados nesta pesquisa, com ênfase no Autoencoder e no Isolation Forest. Por fim, são discutidos MEV, *front-running* e seus principais padrões, além dos trabalhos relacionados à identificação desses comportamentos.

2.1 Blockchain e a Rede Ethereum

Blockchain é uma estrutura de registro distribuído na qual as informações são organizadas em blocos encadeados e compartilhadas entre os participantes da rede. Mecanismos criptográficos e regras de consenso permitem verificar a integridade dos registros e manter um histórico auditável das operações realizadas.

A Ethereum amplia esse modelo ao permitir a execução de contratos inteligentes (*smart contracts*) por meio da Ethereum Virtual Machine (EVM). Além da transferência do ativo nativo Ether (ETH), uma transação pode invocar funções de contratos e modificar o estado da rede, possibilitando a construção de aplicações descentralizadas e protocolos financeiros (19, 18).

Cada transação contém informações como remetente, destinatário, valor, número sequencial da conta (*nonce*), limite de *gas*, parâmetros de taxa e, quando aplicável, dados destinados à execução de contratos. O *gas* representa o custo computacional das operações executadas pela EVM e participa da determinação do custo financeiro da transação. Após a execução, tornam-se disponíveis informações adicionais, como o consumo efetivo de *gas*, o resultado da operação, o recibo e os eventos (*logs*) emitidos pelos contratos envolvidos (18).

A Ethereum utiliza atualmente um mecanismo de consenso baseado em *Proof of Stake* (PoS)(20), no qual validadores participam da proposição e da validação dos blocos. Antes da inclusão, transações podem permanecer em conjuntos locais de operações pendentes, usualmente denominados *mempools*. A seleção e a ordenação dessas transações não são determinadas exclusivamente pela taxa oferecida pelo remetente, podendo também ser influenciadas pelos mecanismos contemporâneos de construção de blocos e por fluxos privados (19, 11).

Para esta pesquisa, uma propriedade particularmente relevante da Ethereum é a observabilidade dos dados produzidos por esse processo. Após a inclusão das transações, podem ser recuperadas diretamente da *blockchain* informações relacionadas a valores, taxas, consumo de *gas*, resultado da execução, posição no bloco e endereços envolvidos. A

partir delas também podem ser construídos atributos derivados, como medidas relativas ao contexto do bloco, características temporais e indicadores de atividade dos participantes.

A Figura 1 apresenta conceitualmente o percurso entre a submissão de uma transação, sua inclusão em um bloco e a obtenção das informações utilizadas para representar seu comportamento. O conjunto exato de atributos e as transformações aplicadas nesta pesquisa são apresentados no Capítulo 3.

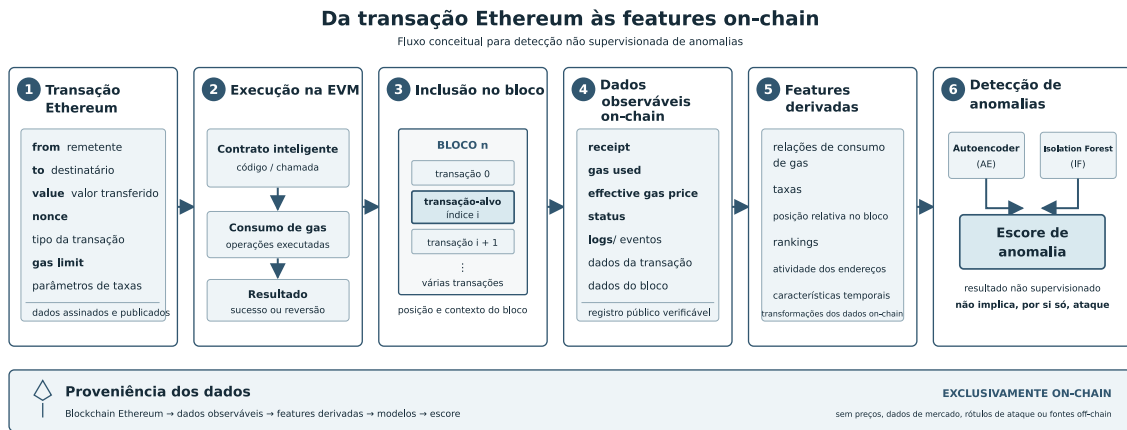


Figura 1 – Fluxo de uma transação Ethereum à obtenção de atributos *on-chain*.

Fonte: Elaboração própria.

Embora parte dos atributos empregados pelos modelos seja derivada, sua construção utiliza exclusivamente informações observáveis *on-chain*. Preços externos de ativos, rótulos de endereços e outras fontes externas não participam da representação utilizada para o treinamento dos modelos.

Essa observabilidade, entretanto, não determina o significado de uma transação. Combinações incomuns de taxas, consumo de *gas*, posição no bloco ou atividade de endereços podem decorrer tanto de comportamentos economicamente estratégicos quanto de operações legítimas pouco frequentes ou de mudanças na utilização da rede. Os dados *on-chain* permitem, portanto, representar e quantificar diferenças de comportamento, mas não estabelecer isoladamente sua causa.

Essa distinção fundamenta a abordagem adotada nesta dissertação: as transações são inicialmente representadas exclusivamente por informações observáveis na *blockchain*, permitindo investigar se métodos não supervisionados identificam desvios dos padrões predominantes sem receber previamente rótulos de ataques ou endereços conhecidos.

2.2 Comportamento Transacional e Segurança Econômica na Ethereum

As transações da Ethereum refletem atividades heterogêneas, incluindo transferências de ativos, interações com contratos inteligentes e operações em aplicações des-

centralizadas. Consequentemente, comportamentos bastante distintos podem coexistir em um mesmo período ou bloco, e uma operação estatisticamente incomum não deve ser interpretada automaticamente como maliciosa.

Além dessa diversidade, a Ethereum constitui um ambiente no qual a inclusão e a ordenação das transações podem produzir consequências econômicas. Quando diferentes operações interagem com um estado compartilhado, informações sobre transações pendentes e sua posição relativa podem representar oportunidades para participantes capazes de influenciar ou explorar a ordem de execução.

Esse contexto está relacionado ao conceito de *Maximal Extractable Value* (MEV), que designa o valor potencialmente obtido por estratégias envolvendo a inclusão, exclusão ou ordenação de transações durante a construção dos blocos. A literatura anterior à transição para *Proof of Stake* empregava originalmente a expressão *Miner Extractable Value* (4).

A existência de MEV não implica necessariamente comportamento malicioso. Estratégias de extração de valor podem decorrer da dinâmica econômica dos mercados descentralizados e produzir efeitos distintos sobre usuários e protocolos. Algumas estratégias, entretanto, exploram informações ou relações de ordenação entre transações para obter vantagem econômica sobre outras operações (4, 5).

Essa característica estabelece uma distinção fundamental para esta pesquisa:

$$\text{anomalia estatística} \neq \text{comportamento malicioso}. \quad (2.1)$$

Uma estratégia economicamente relevante pode ocorrer com frequência suficiente para não produzir um desvio estatístico pronunciado, enquanto uma operação legítima e rara pode receber elevado escore de anomalia. A detecção estatística e a interpretação econômica constituem, portanto, problemas relacionados, mas não equivalentes.

A interpretação também deve considerar a evolução temporal da rede. Mudanças em aplicações, mecanismos de taxas, tipos de transação, construção de blocos e comportamento dos usuários podem alterar a distribuição dos atributos ao longo do tempo. Um comportamento incomum em relação a determinado período de referência pode tornar-se frequente posteriormente, caracterizando um problema de deriva na distribuição dos dados.

Por essas razões, a abordagem desta dissertação separa a identificação de anomalias de sua interpretação posterior. Os modelos não supervisionados produzem inicialmente escores sem conhecimento de ataques ou endereços específicos. Somente em etapas posteriores esses resultados são confrontados com fontes externas e com padrões estruturais e semânticos de interesse.

2.3 Detecção de Anomalias

A detecção de anomalias busca identificar observações que se desviam dos padrões predominantes em um conjunto de dados. Esses desvios podem decorrer de eventos raros, mudanças no processo gerador dos dados, erros ou comportamentos específicos e, portanto, não possuem necessariamente origem maliciosa. Em dados transacionais, os escores de anomalia devem ser interpretados principalmente como mecanismos de identificação e priorização de observações para investigação posterior.

2.3.1 Conceito de Anomalia

Uma anomalia pode ser definida como uma observação ou padrão que não está em conformidade com uma noção estabelecida de comportamento esperado (14). Essa definição depende do contexto, da representação dos dados e do objetivo da análise.

As anomalias são frequentemente classificadas como pontuais, contextuais ou coletivas. Uma anomalia pontual corresponde a uma observação individual incomum em relação às demais; uma anomalia contextual depende das condições em que a observação ocorre; e uma anomalia coletiva decorre da relação entre múltiplas observações que, isoladamente, podem não apresentar comportamento incomum (14).

Essa distinção é relevante para a Ethereum. Os modelos empregados nesta pesquisa operam inicialmente sobre representações individuais das transações, compostas por atributos observáveis ou derivados de dados *on-chain*. Entretanto, determinados comportamentos, como padrões de *front-running*, dependem da relação entre múltiplas transações e são analisados posteriormente por procedimentos estruturais e semânticos.

De forma geral, o resultado de um detector pode ser representado por:

$$s(x) \in \mathbb{R}, \quad (2.2)$$

em que x representa o vetor de atributos de uma observação e $s(x)$ seu escore de anomalia. A interpretação e a escala desse escore dependem do algoritmo utilizado, razão pela qual valores absolutos produzidos por métodos distintos não são necessariamente comparáveis.

2.3.2 Detecção Supervisionada e Não Supervisionada

Métodos supervisionados utilizam observações acompanhadas de rótulos durante o treinamento, permitindo aprender uma função associada às classes de interesse. Em problemas de segurança, entretanto, rótulos confiáveis podem ser escassos, incompletos ou pouco representativos, especialmente para comportamentos raros ou que se modificam ao longo do tempo (14).

Métodos não supervisionados investigam a estrutura dos dados sem exigir esses rótulos durante o treinamento. Essa característica é compatível com a questão investigada nesta dissertação, na qual não se assume a existência de uma referência abrangente capaz de classificar previamente as transações da Ethereum como normais ou anômalas.

Nesse cenário, o modelo não aprende diretamente a distinguir transações legítimas de maliciosas. Ele produz uma medida de desvio segundo o mecanismo de cada algoritmo. A associação posterior entre esses escores e comportamentos externamente identificados constitui uma etapa de avaliação e interpretação, mantida separada do treinamento.

2.3.3 Escores de Anomalia e Limiar de Decisão

Métodos de detecção de anomalias frequentemente produzem um escore contínuo que representa o grau de desvio atribuído a cada observação. Para uma transação x_i , seja:

$$s_i = s(x_i). \quad (2.3)$$

Adotando a convenção de que valores maiores indicam maior grau de anormalidade, uma decisão binária pode ser obtida a partir de um limiar τ :

$$\hat{y}_i = \begin{cases} 1, & \text{se } s_i \geq \tau, \\ 0, & \text{caso contrário.} \end{cases} \quad (2.4)$$

Nesse contexto, $\hat{y}_i = 1$ significa apenas que a observação atingiu ou ultrapassou o critério de anomalia estabelecido pelo modelo; não constitui um rótulo de ataque.

A escolha de τ controla a proporção de observações sinalizadas. Na ausência de rótulos durante a calibração, o limiar pode ser definido a partir da própria distribuição dos escores em um conjunto de referência, por exemplo, por meio de quantis superiores. Essa estratégia preserva o caráter não supervisionado da detecção.

O escore contínuo também deve ser preservado para análises de ordenação. Um limiar constitui uma política operacional de seleção, enquanto o escore contém informação sobre a posição relativa das observações e permite avaliar se transações de interesse tendem a se concentrar nas regiões superiores do ranqueamento.

2.3.4 Avaliação na Ausência de Ground Truth

A avaliação de métodos não supervisionados é particularmente desafiadora quando não existe uma referência completa e confiável (*ground truth*) para as observações analisadas. Nesse cenário, diferentes perspectivas podem ser combinadas para caracterizar os resultados sem assumir que todas as observações possuem uma classe verdadeira conhecida.

Uma primeira perspectiva consiste na avaliação intrínseca dos modelos e de seus escores, considerando propriedades como distribuição, proporção de observações sinalizadas e estabilidade entre diferentes conjuntos ou períodos. Quando múltiplos métodos são aplicados aos mesmos dados, também é possível examinar a concordância entre as observações destacadas. Essa comparação não determina qual método está correto, mas permite verificar se mecanismos distintos identificam regiões semelhantes do espaço de atributos.

Informações externas que não participaram do treinamento podem fornecer uma segunda perspectiva. Quando disponíveis, rótulos independentes permitem investigar se determinadas classes de interesse tendem a receber escores mais elevados, sem que essa referência seja necessariamente considerada uma verdade fundamental completa.

A capacidade de ordenação dos escores pode ser avaliada por métricas como ROC-AUC e por medidas baseadas na curva *Precision-Recall*. Essas perspectivas são complementares e devem ser interpretadas considerando a composição da população avaliada (21, 22). A Average Precision sintetiza a relação entre *precision* e *recall* para a classe definida como positiva, que não é necessariamente a menos frequente. Medidas de enriquecimento, como o *lift*, permitem comparar a concentração dessa classe no topo do ranqueamento com sua prevalência na população de referência.

O uso de rótulos na avaliação não altera o caráter não supervisionado do treinamento quando essas informações permanecem ausentes do ajuste dos modelos. Entretanto, se os rótulos orientam a seleção de configurações ou critérios de decisão na validação, essa escolha passa a depender da referência disponível e deve ser distinguida da avaliação no teste. A associação estatística com uma fonte externa também não estabelece, isoladamente, a natureza do comportamento observado, podendo exigir evidências contextuais ou semânticas adicionais.

Esses princípios fundamentam a estratégia de avaliação adotada nesta dissertação, detalhada no Capítulo 4. As seções seguintes apresentam os dois métodos não supervisionados utilizados para produzir perspectivas complementares sobre o comportamento transacional: Autoencoder e Isolation Forest.

2.4 Autoencoders

Autoencoders são redes neurais treinadas para reconstruir suas próprias entradas a partir de uma representação intermediária. Sua arquitetura é composta, essencialmente, por um codificador (*encoder*), que transforma a entrada em uma representação latente, e um decodificador (*decoder*), responsável por reconstruí-la (15).

Para uma observação $\mathbf{x} \in \mathbb{R}^d$, o codificador produz:

$$\mathbf{z} = f_{\theta}(\mathbf{x}), \quad (2.5)$$

em que f_{θ} representa a função de codificação parametrizada por θ . O decodificador produz então:

$$\hat{\mathbf{x}} = g_{\phi}(\mathbf{z}), \quad (2.6)$$

em que g_{ϕ} representa a função de decodificação parametrizada por ϕ . O processo pode ser representado por:

$$\mathbf{x} \xrightarrow{f_{\theta}} \mathbf{z} \xrightarrow{g_{\phi}} \hat{\mathbf{x}}. \quad (2.7)$$

Durante o treinamento, os parâmetros são ajustados para minimizar uma função de perda que representa a diferença entre a entrada e sua reconstrução. Utilizando o erro quadrático médio (*Mean Squared Error* – MSE), por exemplo:

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{d} \sum_{j=1}^d (x_j - \hat{x}_j)^2. \quad (2.8)$$

A minimização do erro de reconstrução permite que o modelo represente regularidades presentes nos dados de treinamento. A qualidade dessa representação depende, entre outros fatores, da arquitetura da rede, da representação dos atributos e do processo de otimização.

2.4.1 Autoencoders para Detecção de Anomalias

A aplicação de Autoencoders à detecção de anomalias baseia-se na hipótese de que observações compatíveis com as estruturas aprendidas durante o treinamento tendem a ser reconstruídas com menor erro, enquanto observações que se afastam dessas estruturas podem apresentar maior dificuldade de reconstrução (17).

O erro pode, portanto, ser utilizado como escore de anomalia. Para uma observação $\mathbf{x}_i$:

$$s_{\text{AE}}(\mathbf{x}_i) = d(\mathbf{x}_i, \hat{\mathbf{x}}_i), \quad (2.9)$$

em que $d(\cdot)$ representa uma medida de diferença entre a entrada e sua reconstrução. Quando utilizado o erro quadrático médio:

$$s_{\text{AE}}(\mathbf{x}_i) = \frac{1}{d} \sum_{j=1}^d (x_{ij} - \hat{x}_{ij})^2. \quad (2.10)$$

Nessa convenção, escores maiores correspondem a observações reconstruídas com maior erro e, portanto, consideradas mais anômalas segundo o critério do modelo.

Um erro elevado, entretanto, não determina a causa do desvio. O Autoencoder não possui conhecimento semântico sobre fraude, MEV ou *front-running*. Seu escore quantifica o erro de reconstrução da observação segundo as estruturas aprendidas no treinamento. Da mesma forma, um erro baixo não exclui a ocorrência de ataque, pois uma transação envolvida em um padrão malicioso pode apresentar atributos semelhantes aos das demais transações.

A Figura 2 apresenta conceitualmente esse processo, desde a entrada do vetor de atributos até sua reconstrução e a obtenção do escore de anomalia.

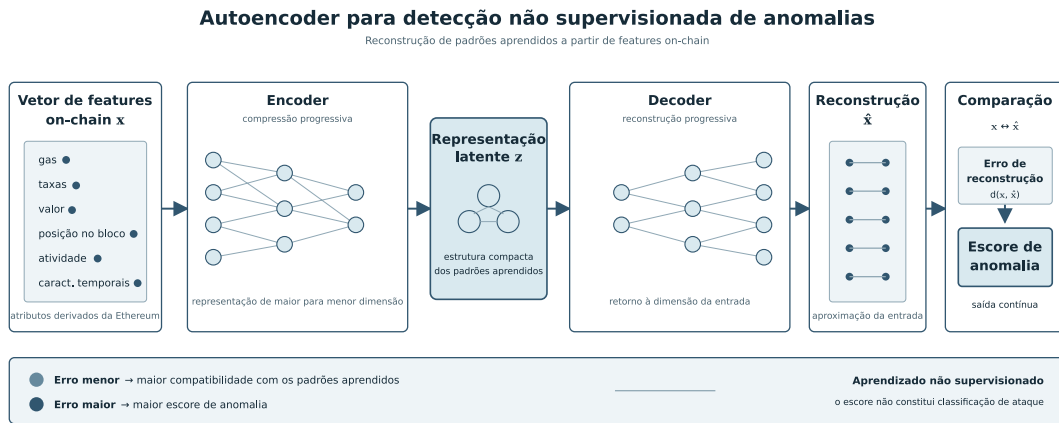


Figura 2 – Obtenção do escore de anomalia pelo erro de reconstrução do Autoencoder.

Fonte: Elaboração própria.

2.4.2 Representação Latente e Capacidade de Reconstrução

A representação latente pode permitir ao Autoencoder capturar relações entre os atributos e estruturas recorrentes no conjunto de treinamento. Dependendo da arquitetura e das restrições empregadas, essa representação pode favorecer a extração de estruturas não lineares presentes nos dados multivariados (15).

A capacidade de reconstrução, contudo, não implica separação perfeita entre observações normais e anômalas. Um modelo com capacidade excessiva pode reconstruir adequadamente observações incomuns, enquanto capacidade insuficiente pode produzir erros elevados mesmo para padrões representativos do treinamento. A arquitetura e o processo de otimização influenciam, portanto, diretamente a distribuição dos escores.

Mudanças temporais na distribuição dos dados também podem elevar o erro de reconstrução. Um modelo treinado em determinado regime pode reconstruir com maior dificuldade observações produzidas posteriormente, mesmo quando a mudança decorre de

uma transformação legítima da rede. Por essa razão, nesta pesquisa a distribuição dos escores é examinada ao longo de diferentes recortes temporais.

O Autoencoder fornece, assim, uma perspectiva de anomalia baseada na dificuldade de reconstrução. A transformação do escore contínuo em sinalização segue o procedimento geral apresentado na Seção **2.3.3**, enquanto sua definição experimental é detalhada no Capítulo 3.

2.5 Isolation Forest

O Isolation Forest é um método não supervisionado de detecção de anomalias proposto por Liu, Ting e Zhou (16). Diferentemente de abordagens que modelam explicitamente o comportamento predominante, o método explora a facilidade com que determinadas observações podem ser isoladas no espaço de atributos.

O algoritmo constrói um conjunto de árvores de isolamento (*Isolation Trees* – iTrees). Em cada árvore, um atributo e um ponto de corte são selecionados aleatoriamente para particionar recursivamente as observações. Para um conjunto X , uma divisão pelo atributo j e pelo ponto p pode ser representada por:

$$X_{\text{esq}} = \{\mathbf{x} \in X : x_j < p\}, \quad X_{\text{dir}} = \{\mathbf{x} \in X : x_j \geq p\}. \quad (2.11)$$

O número de arestas percorridas até o isolamento de uma observação define seu comprimento de caminho (*path length*). Como observações incomuns tendem a ocupar regiões menos densas ou mais facilmente separáveis, elas geralmente necessitam de menos particionamentos para serem isoladas(16). Assim:

$$\text{menor comprimento médio do caminho} \implies \text{maior grau de anormalidade}. \quad (2.12)$$

Como uma única árvore depende dos particionamentos realizados durante sua construção, o método utiliza um conjunto de árvores. Para uma observação $\mathbf{x}$, seu comportamento pode ser resumido pelo comprimento esperado $E[h(\mathbf{x})]$, em que $h(\mathbf{x})$ representa o caminho necessário para isolá-la em uma árvore.

A Figura 3 ilustra esse princípio: observações em regiões densamente ocupadas tendem a permanecer agrupadas por mais particionamentos, enquanto pontos em regiões esparsas podem ser isolados por caminhos mais curtos.

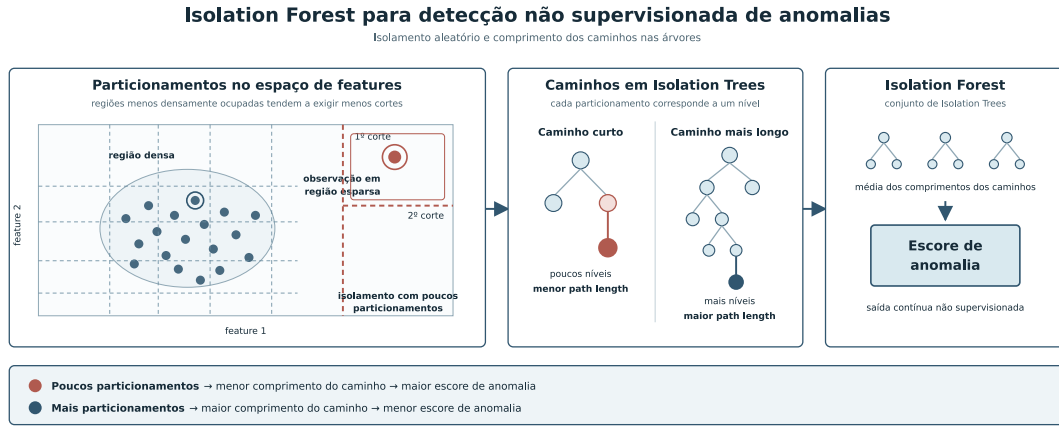


Figura 3 – Obtenção do escore de anomalia pelo princípio de isolamento do Isolation Forest.

Fonte: Elaboração própria.

2.5.1 Escore de Anomalia

O comprimento médio do caminho pode ser convertido em um escore normalizado. Na formulação original do Isolation Forest(16):

$$s(\mathbf{x}, n) = 2^{-\frac{E[h(\mathbf{x})]}{c(n)}}, \quad (2.13)$$

em que n representa o número de observações da subamostra utilizada na construção de cada árvore, e não o tamanho total da base experimental. Para $n > 2$, o fator de normalização $c(n)$ é dado por:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}, \quad (2.14)$$

sendo $H(i)$ o número harmônico, aproximado por:

$$H(i) \approx \ln(i) + \gamma, \quad (2.15)$$

em que γ representa a constante de Euler–Mascheroni.

Nessa formulação, caminhos médios menores produzem escores maiores, indicando maior facilidade de isolamento e, conseqüentemente, maior grau de anormalidade segundo o método.

Na implementação utilizada nesta pesquisa, o escore retornado por `score_samples` tem seu sinal invertido. Dessa forma, valores maiores do escore registrado correspondem a maior anormalidade, preservando a convenção adotada para interpretar os resultados dos dois modelos.

Assim como no Autoencoder, o escore é uma medida contínua. Sua conversão em uma sinalização binária utiliza o procedimento geral apresentado na Seção **2.3.3**, e não representa, por si só, uma classificação semântica da transação.

2.5.2 Relação com o Autoencoder

Autoencoder e Isolation Forest produzem escores de anomalia por princípios distintos. No primeiro, o desvio está associado à dificuldade de reconstrução da observação. No segundo, à facilidade de seu isolamento no espaço de atributos. De forma conceitual:

Autoencoder: dificuldade de reconstrução $\rightarrow$ anomalia,

Isolation Forest: facilidade de isolamento $\rightarrow$ anomalia.

Não se espera, portanto, que os métodos produzam ordenações idênticas. Uma transação pode apresentar elevado erro de reconstrução sem ser particularmente fácil de isolar, ou ocorrer a situação inversa.

Nesta dissertação, os dois algoritmos são utilizados independentemente sobre representações equivalentes das transações. A comparação entre suas sinalizações permite investigar em que medida mecanismos distintos priorizam as mesmas observações ou produzem conjuntos diferentes de candidatos, sem que a concordância entre eles seja interpretada isoladamente como evidência de correção.

2.6 Front-Running e Padrões Relacionados na Ethereum

A execução de contratos inteligentes pode produzir resultados dependentes da ordem em que as transações são processadas. Quando diferentes operações interagem com um mesmo estado, alterações em sua sequência podem modificar preços, saldos, oportunidades disponíveis ou outros resultados econômicos. A posição de uma transação dentro de um bloco pode, portanto, possuir consequências relevantes.

Essa propriedade está relacionada ao conceito de *Maximal Extractable Value* (MEV), introduzido anteriormente. Entre as estratégias associadas à ordenação encontra-se o *front-running*, no qual um participante procura obter vantagem posicionando uma ou mais operações de forma favorável em relação a outra transação(4, 11).

Historicamente, a observação de transações pendentes no *mempool* público constituiu um mecanismo importante para identificar essas oportunidades(4, 10). Entretanto, o processo contemporâneo de construção de blocos também envolve participantes especializados e fluxos privados, de modo que o *mempool* público não representa necessariamente toda a informação utilizada na ordenação(11).

Esta dissertação adota como referência a taxonomia proposta por Eskandari, Moosavi e Clark (5), que distingue três padrões de *front-running*: deslocamento (*displacement*),

inserção (*insertion*) e supressão (*suppression*). Essa classificação também fundamenta a investigação empírica de Torres, Camino e State (10), utilizada posteriormente como referência para a construção das regras estruturais deste trabalho.

2.6.1 Front-Running

O *front-running* pode ser compreendido como uma estratégia na qual um participante utiliza conhecimento ou antecipação de uma operação para posicionar uma ou mais transações de maneira economicamente favorável em relação a ela(10). Em contratos inteligentes, essa oportunidade pode surgir quando o resultado de uma operação depende do estado modificado por outra transação.

Considere uma transação de referência T_v e uma transação T_a posicionada antes dela. Uma relação básica de precedência pode ser representada por:

$$\text{pos}(T_a) < \text{pos}(T_v). \quad (2.16)$$

Essa condição representa apenas uma relação estrutural. A precedência, por si só, não caracteriza *front-running*. São necessárias evidências de que as operações estejam relacionadas e de que sua ordenação seja compatível com a estratégia considerada.

Essa distinção fundamenta o procedimento utilizado nesta dissertação. Relações estruturais geram candidatos, cuja interpretação depende de evidências semânticas adicionais. A confirmação de um evento corresponde ao atendimento dos critérios operacionais definidos pelo pipeline. Ela não comprova, isoladamente, o acesso prévio à transação da vítima nem a intenção do participante.

2.6.2 Padrão de Inserção

No padrão de inserção (*insertion*), a operação original permanece relevante, mas é executada após uma transação que modifica o estado utilizado em sua execução. Na taxonomia de Eskandari, Moosavi e Clark (5), essa alteração permite ao agente explorar o efeito produzido pela operação intermediária.

Uma manifestação conhecida desse padrão é o *sandwich*, no qual duas transações relacionadas são posicionadas ao redor de uma transação de referência, configuração também analisada empiricamente por Torres, Camino e State (10). De forma simplificada:

$$T_{a1} \rightarrow T_v \rightarrow T_{a2}, \quad (2.17)$$

em que T_{a1} representa a operação anterior, T_v a transação intermediária e T_{a2} uma operação posterior relacionada ao mesmo agente ou estratégia.

Em um cenário típico, a primeira operação modifica temporariamente uma condição relevante para a execução de T_v , enquanto a operação posterior procura capturar o efeito resultante ou reverter a posição inicialmente assumida.

A sequência estrutural, entretanto, não comprova isoladamente a ocorrência de um *sandwich*. A confirmação pode exigir evidências sobre os contratos, funções executadas, ativos movimentados, eventos emitidos e direção das operações. Por essa razão, estruturas desse tipo são inicialmente tratadas como candidatos.

2.6.3 Padrão de Deslocamento

No padrão de deslocamento (*displacement*), uma operação procura ser executada antes de outra transação para obter a vantagem associada à oportunidade que esta pretendia explorar. Diferentemente da inserção, a execução posterior da transação original não é necessária para que a operação precedente alcance seu objetivo (5, 10).

A estrutura básica pode ser representada por:

$$T_a \rightarrow T_v. \quad (2.18)$$

O elemento relevante não é apenas a precedência entre T_a e T_v , mas a existência de uma relação funcional entre as operações. Essa relação pode ser investigada por meio da similaridade das chamadas, dos contratos envolvidos, dos resultados de execução e de outras evidências disponíveis.

Assim, proximidade e ordenação constituem evidências para geração de candidatos, enquanto a confirmação exige elementos adicionais que sustentem a relação semântica entre as transações.

2.6.4 Padrão de Supressão

No padrão de supressão (*suppression*), o objetivo consiste em atrasar ou impedir temporariamente a execução de uma operação de referência. Na taxonomia de Eskandari, Moosavi e Clark (5), após produzir o atraso pretendido, torna-se indiferente para o agente que a operação original venha ou não a ser posteriormente executada.

Torres, Camino e State (10) investigam uma manifestação desse padrão baseada no consumo intensivo de *gas* e no preenchimento da capacidade dos blocos, estratégia também denominada *block stuffing*.

Esse padrão apresenta uma dificuldade particular para análises baseadas exclusivamente em transações confirmadas. Quando uma operação deixa de ser incluída em determinado bloco, sua ausência não permite determinar, por si só, se ela estava disponível para inclusão ou por qual razão foi postergada.

A confirmação de uma hipótese de supressão pode, portanto, depender de evidências históricas do *mempool*, como a presença e o momento de propagação da transação enquanto ainda estava pendente. Essas informações não podem ser reconstruídas integralmente a partir da sequência final de blocos.

Por essa razão, nesta dissertação uma estrutura identificada exclusivamente a partir de transações confirmadas pode ser considerada compatível com supressão, mas não é tratada como evento semanticamente confirmado quando sua validação depende de evidência histórica do *mempool*.

A Figura 4 sintetiza as relações estruturais dos três padrões considerados nesta dissertação. As configurações representam evidências para a geração de candidatos e não, isoladamente, a confirmação de *front-running*.

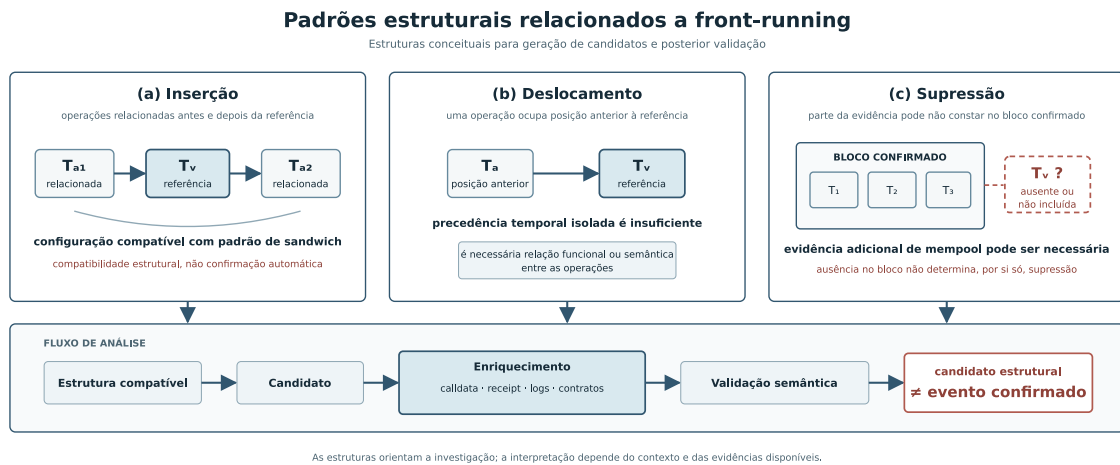


Figura 4 – Padrões estruturais de inserção, deslocamento e supressão em *front-running*.

Fonte: Elaboração própria, com base em (5, 10).

2.6.5 Da Estrutura Candidata à Validação Semântica

A taxonomia apresentada permite identificar relações entre transações estruturalmente compatíveis com diferentes padrões de *front-running*. Entretanto, reconhecer uma configuração nos dados não equivale a demonstrar seu significado econômico.

Nesta dissertação, essa diferença é tratada pela geração de candidatos e pela análise posterior de suas evidências. Os candidatos são identificados a partir de relações observáveis entre transações, como posição, participantes e propriedades de execução. O enriquecimento acrescenta *calldata*, recibos e eventos emitidos pelos contratos. Essas informações são submetidas à validação semântica e à adjudicação automatizada, que consolidam a decisão sobre cada candidato segundo os critérios definidos no pipeline.

O processo pode ser sintetizado por:

$$\begin{aligned} \text{estrutura compatível} &\rightarrow \text{candidato} \rightarrow \text{enriquecimento} \\ &\rightarrow \text{validação semântica} \rightarrow \text{adjudicação.} \end{aligned} \quad (2.19)$$

A mesma separação deve ser mantida em relação aos modelos de anomalia. Uma transação pode apresentar elevado escore sem participar de uma estrutura compatível com *front-running*, enquanto um evento estruturalmente compatível pode envolver transações que não apresentem escores extremos.

Consequentemente, os níveis de evidência considerados neste trabalho são conceitualmente distintos:

$$\text{anomalia estatística} \neq \text{candidato estrutural} \neq \text{evento semanticamente validado.} \quad (2.20)$$

Essa separação permite investigar as relações entre os diferentes níveis sem assumir previamente que uma transação estatisticamente incomum representa um comportamento específico.

2.7 Trabalhos Relacionados

A literatura sobre detecção de comportamentos anômalos em *blockchain* compreende problemas com diferentes unidades de análise, níveis de supervisão e objetivos. Há estudos voltados à classificação de contas ou contratos, à detecção de anomalias em transações ou séries temporais e à identificação de padrões específicos relacionados a MEV e *front-running*. Essas diferenças devem ser consideradas ao comparar resultados, pois métodos destinados a objetos e rótulos distintos não resolvem necessariamente o mesmo problema.

Li et al. (7) organizam essa literatura em métodos gerais de detecção de anomalias e abordagens direcionadas a comportamentos específicos, destacando desafios como disponibilidade de rótulos, representação dos dados e interpretabilidade dos modelos. Essas limitações motivam a utilização, nesta dissertação, de treinamento não supervisionado e a separação entre produção dos escores e utilização posterior de evidências externas.

2.7.1 Detecção de Anomalias e Fraudes em Blockchain

Sayadi, Rejeb e Choukair (3) combinam One-Class SVM e *k-means* para detectar anomalias em séries temporais agregadas da rede Bitcoin, utilizando anomalias introduzidas para avaliação. Embora compartilhe o caráter não supervisionado, a unidade de análise difere das transações individuais consideradas nesta dissertação.

Podgorelec et al. (8) propõem um mecanismo de detecção de anomalias para sistemas de assinatura automatizada de transações Ethereum, avaliando novas operações em relação ao comportamento habitual de usuários ou contas. Seu objetivo é a proteção

operacional de carteiras, enquanto esta pesquisa examina transações de forma global, sem construir previamente um perfil individual para cada endereço.

Outros estudos formulam o problema de maneira supervisionada. Ostapowicz e Zbikowski (2) utilizam atributos agregados de contas Ethereum e comparam Random Forest, SVM e XGBoost, enquanto Farrugia, Ellul e Azzopardi (6) empregam XGBoost para distinguir contas previamente rotuladas. Jung et al. (1), por sua vez, combinam atributos de código e histórico transacional para identificar contratos associados a esquemas Ponzi.

Al-E'mari et al. (9) abordam a disponibilidade de referências rotuladas ao propor um conjunto de transações Ethereum contendo categorias de comportamento ilícito. Bases desse tipo são relevantes para avaliações controladas, mas seus rótulos permanecem condicionados às categorias e períodos cobertos.

A Tabela 1 sintetiza as principais diferenças entre esses estudos e a abordagem adotada nesta dissertação.

Tabela 1 – Comparação com trabalhos sobre anomalias e fraudes em *blockchain*.

Trabalho	Unidade analisada	Abordagem	Diferença em relação a esta dissertação
Sayadi et al. (3)	Séries temporais da rede Bitcoin	<i>One-Class SVM</i> e <i>k-means</i>	Analisa indicadores agregados e utiliza anomalias introduzidas para avaliação.
Podgorelec et al. (8)	Comportamento de usuário ou conta Ethereum	Modelo personalizado de detecção de anomalias	Tem como objetivo decidir sobre a assinatura automática de novas transações a partir do perfil individual do usuário.
Ostapowicz e Zbikowski (2)	Contas Ethereum	Classificação supervisionada com <i>Random Forest</i> , SVM e XGBoost	Utiliza rótulos de fraude e atributos agregados no nível da conta.
Jung et al. (1)	Contratos inteligentes Ethereum	Classificação supervisionada com atributos de código e transações	Procura especificamente contratos associados a esquemas Ponzi.
Farrugia et al. (6)	Contas Ethereum	Classificação supervisionada com XGBoost	Depende de contas previamente rotuladas como ilícitas ou regulares.
Al-E'mari et al. (9)	Transações Ethereum	Construção de conjunto rotulado para sistemas de detecção	Os rótulos representam categorias específicas e não constituem uma descrição exhaustiva do comportamento da rede.
Esta dissertação	Transações individuais Ethereum	Autoencoder e Isolation Forest não supervisionados	Produz escores transacionais sem utilizar rótulos no treinamento e realiza avaliação externa e semântica em etapas posteriores.

Fonte: Elaboração própria.

Em conjunto, esses trabalhos evidenciam diferenças importantes de unidade de análise e supervisão. Resultados obtidos na classificação de contas, contratos ou cate-

gorias previamente rotuladas não são diretamente equivalentes aos de um detector não supervisionado que produz escores para transações individuais.

2.7.2 Detecção de Front-Running e MEV

Daian et al. (4) documentam agentes de arbitragem em *exchanges* descentralizadas e as chamadas *priority gas auctions*, demonstrando a relevância econômica da competição pela ordenação das transações. O trabalho contribui para a fundamentação do fenômeno de MEV, mas não propõe um detector geral de anomalias transacionais.

Eskandari, Moosavi e Clark (5) sistematizam o *front-running* em aplicações *blockchain* e propõem a classificação em deslocamento, inserção e supressão adotada nesta dissertação.

Torres, Camino e State (10) operacionalizam essa taxonomia em uma investigação empírica de larga escala da Ethereum, desenvolvendo procedimentos para localizar estruturas compatíveis com os três padrões. Esse trabalho constitui a principal referência para o componente estrutural adaptado nesta pesquisa. As regras correspondentes são utilizadas para gerar candidatos independentemente dos escores produzidos pelo Autoencoder e pelo Isolation Forest.

Zhang et al. (12) propõem o FRAD, abordagem supervisionada para classificar ocorrências previamente identificadas de *front-running* entre deslocamento, inserção e supressão. O problema difere do investigado nesta dissertação porque parte de transações já associadas aos ataques e procura distinguir suas categorias.

Li et al. (13) desenvolvem um detector de ataques *sandwich* integrado ao cliente Geth e voltado ao processamento em tempo próximo ao real. Seu escopo concentra-se no padrão de inserção do tipo *sandwich*, enquanto esta pesquisa considera também candidatos aos padrões de deslocamento e supressão.

A Tabela 2 sintetiza a relação entre esses trabalhos e a abordagem proposta.

Tabela 2 – Comparação com trabalhos sobre *front-running*

Trabalho	Objetivo principal	Método ou evidência	Relação com esta dissertação
Daian et al. (4)	Caracterizar arbitragem, competição por ordenação e MEV	Observação empírica e modelagem econômica de leilões de taxas	Fundamenta a relação entre ordenação, incentivos econômicos e MEV.
Eskandari et al. (5)	Sistematizar formas de <i>front-running</i>	Estudos de caso e taxonomia conceitual	Fornece as categorias de deslocamento, inserção e supressão.
Torres et al. (10)	Medir historicamente os três tipos de <i>front-running</i>	Regras estruturais e análise de larga escala da <i>blockchain</i>	Fundamenta o detector estrutural adaptado para geração independente de candidatos.
Zhang et al. (12)	Classificar ataques entre três categorias	Classificadores supervisionados sobre transações previamente selecionadas	Resolve uma classificação ternária, enquanto esta dissertação aplica modelos não supervisionados às transações dos blocos amostrados.
Li et al. (13)	Detectar ataques <i>sandwich</i> em tempo próximo ao real	Detector integrado ao cliente Geth	Concentra-se em inserção do tipo <i>sandwich</i> , enquanto esta pesquisa considera também candidatos a deslocamento e supressão.
Esta dissertação	Investigar a relação entre anomalias transacionais e padrões de <i>front-running</i>	Autoencoder, Isolation Forest, regras estruturais, enriquecimento RPC, validação semântica e adjudicação automatizada	Distingue anomalia estatística, candidato estrutural e evento confirmado segundo critérios operacionais.

Fonte: Elaboração própria.

2.7.3 Posicionamento desta Dissertação

Os trabalhos analisados podem ser organizados em três linhas: classificação supervisionada de objetos previamente rotulados, detecção de anomalias sem conhecimento completo das classes e identificação de padrões específicos por conhecimento do domínio.

Esta dissertação combina elementos das duas últimas perspectivas. Autoencoder e Isolation Forest são treinados sem rótulos e produzem escores para transações individuais dos blocos amostrados. O Label Cloud fornece uma referência externa exploratória. Paralelamente, regras adaptadas da literatura de *front-running* geram candidatos independentemente da sinalização dos modelos. Esses candidatos são enriquecidos e submetidos à validação semântica e à adjudicação automatizada.

A investigação examina em que medida escores produzidos exclusivamente a partir de atributos *on-chain* priorizam transações associadas às evidências disponíveis. A referência semântica permite comparar atacantes e vítimas contextuais nos eventos identificados,

mas sua cobertura depende das regras estruturais e dos critérios de confirmação. Portanto, ela não constitui uma descrição exaustiva dos ataques na rede.

O trabalho distingue três objetos. A anomalia estatística é definida em relação ao modelo e à distribuição dos dados. O candidato estrutural é identificado por relações observáveis entre transações. O evento confirmado atende aos critérios operacionais de validação e adjudicação, sustentados por evidências adicionais de execução. Essa distinção orienta a interpretação dos resultados e o protocolo experimental apresentado nos capítulos seguintes.

2.8 Considerações Finais

Este capítulo apresentou os fundamentos necessários à detecção não supervisionada de anomalias transacionais na Ethereum e à interpretação posterior de padrões relacionados a *front-running*. A observabilidade da *blockchain* permite construir representações quantitativas das transações, mas não determina isoladamente o significado econômico ou semântico dos comportamentos observados.

Autoencoder e Isolation Forest fornecem perspectivas complementares de anomalia, baseadas, respectivamente, na dificuldade de reconstrução e na facilidade de isolamento. Seus escores representam desvios estatísticos e permanecem conceitualmente distintos das evidências externas ou estruturais utilizadas posteriormente para interpretá-los.

A taxonomia de *front-running* apresentada estabelece os padrões de inserção, deslocamento e supressão utilizados na geração independente de candidatos. A fundamentação sustenta, assim, a distinção entre anomalia estatística, candidato estrutural e evento confirmado segundo critérios operacionais de validação e adjudicação. O Capítulo 3 apresenta o desenho experimental utilizado para aplicar esses conceitos aos dados transacionais da Ethereum.

3 METODOLOGIA

Este capítulo apresenta a metodologia utilizada para investigar a detecção de comportamentos transacionais anômalos na Ethereum por meio de aprendizado não supervisionado. O experimento utiliza dados públicos da rede principal referentes a 2024 e 2025. A seleção determinística de aproximadamente 5% dos blocos preserva todas as transações dos blocos escolhidos e reúne 47.545.805 transações. As representações fornecidas aos modelos são construídas exclusivamente a partir de atributos *on-chain*.

Autoencoder (AE) e Isolation Forest (IF) são avaliados em cinco configurações de atributos. O desenho experimental preserva a ordem cronológica entre treinamento, validação e teste e inclui janelas associadas às atualizações Dencun, Pectra e Fusaka, permitindo examinar o comportamento dos modelos em diferentes períodos.

Os modelos são ajustados sem rótulos e produzem escores contínuos de anomalia. Os valores dos limiares são obtidos da distribuição dos escores na validação. As referências externas e semânticas são utilizadas posteriormente na avaliação e, quando aplicável, na seleção de configurações e limiares entre as alternativas predefinidas. Esses procedimentos são descritos no Capítulo 4, mantendo o teste final fora das decisões de seleção.

3.1 Visão Geral da Metodologia

A metodologia foi organizada como um *pipeline* que transforma dados transacionais da Ethereum em representações quantitativas, produz escores de anomalia por métodos não supervisionados e, posteriormente, submete esses resultados a diferentes formas de avaliação.

Inicialmente, os dados são amostrados e preparados preservando blocos completos e a ordem cronológica das observações. A partir dos campos disponíveis na *blockchain* são construídos atributos relacionados à execução, taxas, valor, atividade dos participantes, contexto do bloco e tempo.

Sobre essa representação são definidas cinco configurações experimentais:

- a_completa;
- b_completa;
- c_completa;
- b_sem_taxas_absolutas;
- c_sem_taxas_absolutas.

As duas últimas constituem variantes de ablação destinadas a avaliar a influência dos atributos associados às magnitudes absolutas das taxas. A composição de cada configuração é apresentada na Seção 3.3.

Os dados são organizados em sete recortes cronológicos:

1. `estresse_pre_dencun_2024`;
2. `transicao_dencun_2024`;
3. `treino_2024_pos_dencun`;
4. `validacao_2025_pre_pectra`;
5. `transicao_pectra_2025`;
6. `teste_final_2025`;
7. `estresse_fusaka_2025`.

O recorte `treino_2024_pos_dencun` é utilizado para ajustar os modelos e os parâmetros do pré-processamento. O recorte `validacao_2025_pre_pectra` fornece a distribuição dos escores para calibrar os limiares e, na avaliação semântica, orienta a seleção de configurações e limiares entre as alternativas predefinidas. O teste final permanece fora dessas decisões. As demais janelas permitem examinar o comportamento dos modelos em períodos distintos. Datas, quantidades de observações e funções experimentais são apresentadas na Seção 3.1.2.

Cada configuração é processada independentemente pelo Autoencoder e pelo Isolation Forest, resultando em dez modelos. Para cada modelo, os escores obtidos na validação determinam três limiares:

$$\tau_m^{(q)} = Q_q(S_{m,\text{val}}), \quad q \in \{0,99, 0,995, 0,999\}, \quad (3.1)$$

em que $S_{m,\text{val}}$ representa os escores produzidos pelo modelo m no conjunto de validação. Uma observação é sinalizada quando:

$$\hat{y}_{i,m}^{(q)} = \mathbb{I} \left[s_m(x_i) \geq \tau_m^{(q)} \right]. \quad (3.2)$$

Uma vez calibrados, os limiares permanecem fixos nos demais recortes. Dessa forma, alterações nas taxas de sinalização refletem mudanças na relação entre os escores e a referência estabelecida na validação, em vez de recalibrações específicas para cada período.

A Figura 5 apresenta a organização geral do *pipeline*. O núcleo não supervisionado compreende a preparação dos dados, a construção das representações, o treinamento

dos modelos e a produção dos escores. A concordância entre métodos e a comparação com o Label Cloud constituem perspectivas de avaliação. As regras estruturais são aplicadas independentemente da sinalização dos modelos e geram candidatos submetidos ao enriquecimento RPC, à validação semântica e à adjudicação automatizada. Os rótulos resultantes permitem avaliar os escores dentro da população de eventos identificados, conforme detalhado no Capítulo 4.

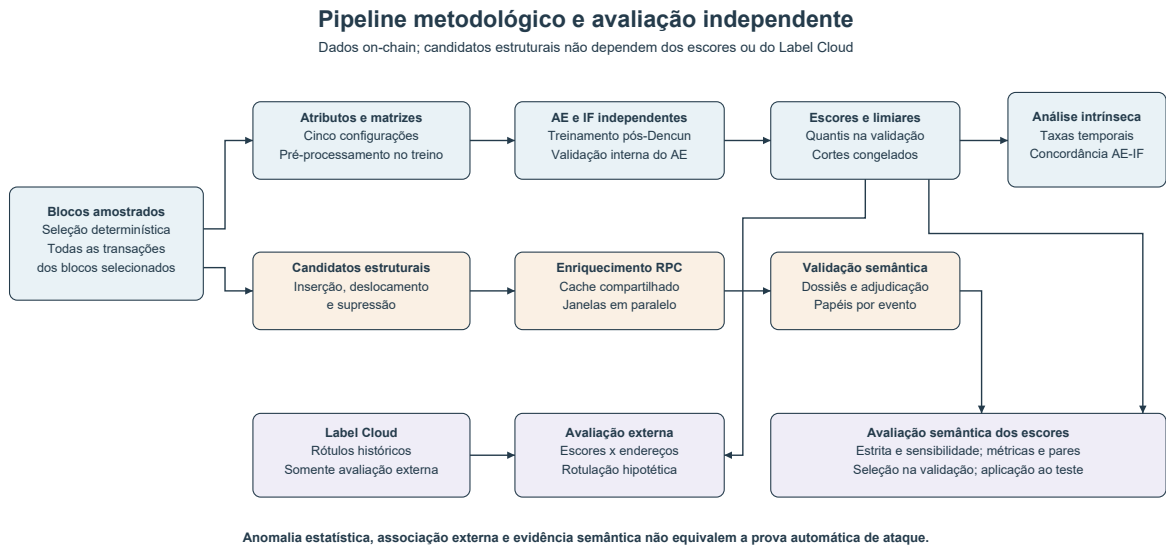


Figura 5 – Pipeline metodológico do trabalho.

Fonte: Elaboração própria.

3.1.1 Fonte e Estrutura dos Dados

Os dados foram obtidos do AWS Public Blockchain Dataset para a Ethereum (33), disponibilizado em arquivos Parquet com partições diárias. A fonte contém informações transacionais e de execução utilizadas na construção dos atributos.

Entre os campos disponíveis encontram-se hash e posição da transação, número e hash do bloco, endereços de origem e destino, valor, *nonce*, limite e preço de *gas*, consumo efetivo de *gas*, estado do recibo, tipo da transação, parâmetros de taxas e instante de inclusão no bloco. A Tabela 3 sintetiza os principais grupos de informações utilizados.

Para reduzir o volume armazenado, a coluna `input` foi removida das bases locais. Essa coluna não participa das representações fornecidas aos modelos. Quando necessária à interpretação dos candidatos estruturais, a `calldata` é recuperada posteriormente por RPC, conforme descrito no Capítulo 4.

No período de treinamento, entre 1º de abril e 31 de dezembro de 2024, a fonte contém 275 arquivos Parquet e 322.293.750 registros. O experimento adota uma amostra determinística de aproximadamente 5% dos blocos. A seleção segue a regra:

$$\text{block_number} \bmod N = 0, \quad N = 20. \quad (3.3)$$

Quando um bloco satisfaz o critério, todas as suas transações disponíveis na fonte são preservadas. O bloco constitui, portanto, a unidade de amostragem. Essa escolha mantém as relações intrabloco necessárias à construção de atributos contextuais e à análise estrutural, incluindo posição relativa, quantidade de transações, frequência de endereços e ordenação das taxas.

O percentual é configurável e deve satisfazer $p = 100/N$, com N inteiro positivo e p expresso em porcentagem. Entre os percentuais inteiros de 1% a 10%, são permitidos 1%, 2%, 4%, 5% e 10%. Outros valores são admitidos quando atendem à mesma regra. Essa restrição preserva a seleção por módulo inteiro e permite reproduzir os mesmos blocos para uma fonte e uma configuração equivalentes. O percentual de blocos não determina exatamente o percentual de transações, pois a quantidade de operações varia entre blocos.

No treinamento, a seleção resultou em 16.080.647 transações distribuídas em 98.369 blocos, sem hashes duplicados. A mesma regra foi aplicada às demais janelas. As sete janelas totalizam 47.545.805 transações em 261.206 blocos.

Tabela 3 – Grupos de informações disponíveis nos dados de origem.

Grupo	Campos de origem
Identificação	hash, block_number, block_hash, transaction_index
Participantes	from_address, to_address
Valor e sequência	value, nonce
Gas e taxas	gas, gas_price, max_fee_per_gas, max_priority_fee_per_gas
Execução e recibo	receipt_gas_used, receipt_cumulative_gas_used, receipt_effective_gas_price, receipt_status, receipt_contract_address
Tipo	transaction_type
Tempo	block_timestamp, date

Fonte: Elaboração própria a partir do esquema dos dados utilizados.

3.1.2 Período de Análise

O período experimental compreende janeiro de 2024 a dezembro de 2025 e é dividido em sete recortes cronológicos não sobrepostos. A Tabela 4 apresenta os intervalos, o número de transações amostradas e a função de cada recorte no protocolo.

A definição desses recortes utiliza como marcos cronológicos as atualizações Dencun, Pectra e Fusaka da rede Ethereum. Dencun introduziu, entre outras modificações, as transações com *blobs* por meio da EIP-4844; Pectra incorporou alterações relacionadas a contas, validadores e capacidade de *blobs*; e Fusaka introduziu o *Peer Data Availability Sampling* (PeerDAS), além de outras modificações relacionadas à escalabilidade e à

execução da rede (34, 35, 36). Essas atualizações constituem, portanto, marcos externos e objetivamente datados para a organização temporal do experimento. Sua utilização não pressupõe que eventuais alterações observadas nas distribuições dos atributos ou nos escores dos modelos tenham sido causadas pelas respectivas atualizações.

Tabela 4 – Recortes temporais do desenho experimental na amostra de aproximadamente 5% dos blocos.

Recorte	Início	Fim	Transações	Função
Pré-Dencun	01/01/2024	12/03/2024	4.153.783	Estresse histórico
Transição Dencun	13/03/2024	31/03/2024	1.232.305	Transição de protocolo
Pós-Dencun	01/04/2024	31/12/2024	16.080.647	Treinamento
Pré-Pectra	01/01/2025	30/04/2025	7.248.601	Validação
Transição Pectra	01/05/2025	31/05/2025	2.108.033	Transição de protocolo
Pós-Pectra	01/06/2025	30/11/2025	14.127.430	Teste final
Fusaka	01/12/2025	31/12/2025	2.595.006	Estresse temporal posterior
Total			47.545.805	

Fonte: Elaboração própria, com base no manifesto dos recortes temporais da execução de 5%.

O período Pós-Dencun, correspondente a `treino_2024_pos_dencun`, constitui o único recorte utilizado para ajuste dos modelos e dos parâmetros de pré-processamento. O período Pré-Pectra, `validacao_2025_pre_pectra`, é utilizado para calibração dos limiares de sinalização, sem reajuste dos modelos, e para seleção das configurações e dos limiares na avaliação semântica. O período Pós-Pectra, `teste_final_2025`, constitui o conjunto principal de teste e permanece fora das decisões de ajuste, calibração e seleção.

Os demais recortes são utilizados para análise temporal: Pré-Dencun representa um período histórico anterior ao treinamento; Dencun e Pectra representam janelas de transição; e Fusaka constitui uma janela posterior de estresse. A proximidade com atualizações do protocolo é utilizada para organizar a análise e não implica atribuição causal de eventuais mudanças observadas.

A organização cronológica pode ser representada por:

$$\begin{aligned}
 \text{Pré-Dencun} \rightarrow \text{Dencun} \rightarrow \underbrace{\text{Pós-Dencun}}_{\text{treinamento}} \rightarrow \underbrace{\text{Pré-Pectra}}_{\text{validação}} \\
 \rightarrow \text{Pectra} \rightarrow \underbrace{\text{Pós-Pectra}}_{\text{teste final}} \rightarrow \text{Fusaka}.
 \end{aligned} \tag{3.4}$$

Os parâmetros dependentes dos dados são ajustados exclusivamente no conjunto de treinamento, e os limiares são calibrados exclusivamente na validação. Essa separação temporal evita que informações dos períodos posteriores sejam utilizadas no ajuste dos modelos.

3.2 Construção da Representação Transacional

Os campos de origem são transformados em uma representação quantitativa composta por atributos relacionados à execução, taxas, valor, conta, tipo da transação, contexto do bloco e tempo. Todos os atributos são construídos exclusivamente a partir de informações *on-chain*.

Parte das variáveis resulta de transformações diretas dos campos originais, enquanto outras dependem do contexto do bloco. Antes da definição das configurações experimentais, os atributos candidatos foram submetidos a uma análise diagnóstica de correlação para caracterizar possíveis redundâncias.

3.2.1 Análise de Correlação entre Atributos

A análise de correlação foi utilizada como procedimento diagnóstico para caracterizar relações entre os atributos candidatos, e não como critério automático de seleção. O cálculo foi realizado exclusivamente sobre o período de treinamento, utilizando uma subamostra de 500.000 observações das 16.080.647 transações selecionadas nesse recorte. A subamostra foi obtida por amostragem de reservatório, com semente pseudoaleatória fixa igual a 20240916, permitindo reproduzir o procedimento nas mesmas condições de execução.

O conjunto analisado antecede a definição final das matrizes A, B e C e inclui variáveis posteriormente mantidas ou descartadas, como `is_zero_value`, `tx_type_2` e `value_rank_block`. As matrizes devem, portanto, ser interpretadas como um diagnóstico da etapa de engenharia de atributos.

Foram utilizados os coeficientes de Pearson e Spearman. Para duas variáveis X e Y , o coeficiente de Pearson é dado por:

$$r_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}. \quad (3.5)$$

A correlação de Spearman, baseada nos postos das observações, permite caracterizar associações monotônicas que não precisam apresentar comportamento estritamente linear (25). O coeficiente pode ser representado como a correlação de Pearson aplicada aos postos das variáveis:

$$\rho_{X,Y} = r_{\text{rank}(X), \text{rank}(Y)}. \quad (3.6)$$

As Figuras 6 e 7 apresentam as matrizes obtidas. Como referência diagnóstica, foram destacados pares cuja magnitude do coeficiente satisfaz:

$$|c| \geq 0,90, \quad (3.7)$$

Entre os pares com elevada magnitude de correlação, destacaram-se os atributos relacionados ao valor da transação. A associação entre `transaction_value_log` e `value_rank_block` apresentou coeficientes de aproximadamente 0,975 em Pearson e 0,988 em Spearman. Já a relação entre `transaction_value_log` e `is_zero_value` apresentou coeficientes negativos de aproximadamente $-0,978$ e $-0,933$, respectivamente. Também foram observadas associações elevadas entre indicadores do tipo da transação e atributos de taxas, como `max_fee_per_gas_log`. Esses resultados evidenciam dependências entre variáveis que representam aspectos relacionados do comportamento transacional, sem determinar, isoladamente, sua exclusão das configurações experimentais.

Essas associações não determinam automaticamente a exclusão de atributos. A decisão considera também o significado operacional das variáveis e sua função nas configurações experimentais, evitando utilizar um único limiar de correlação como critério de seleção.

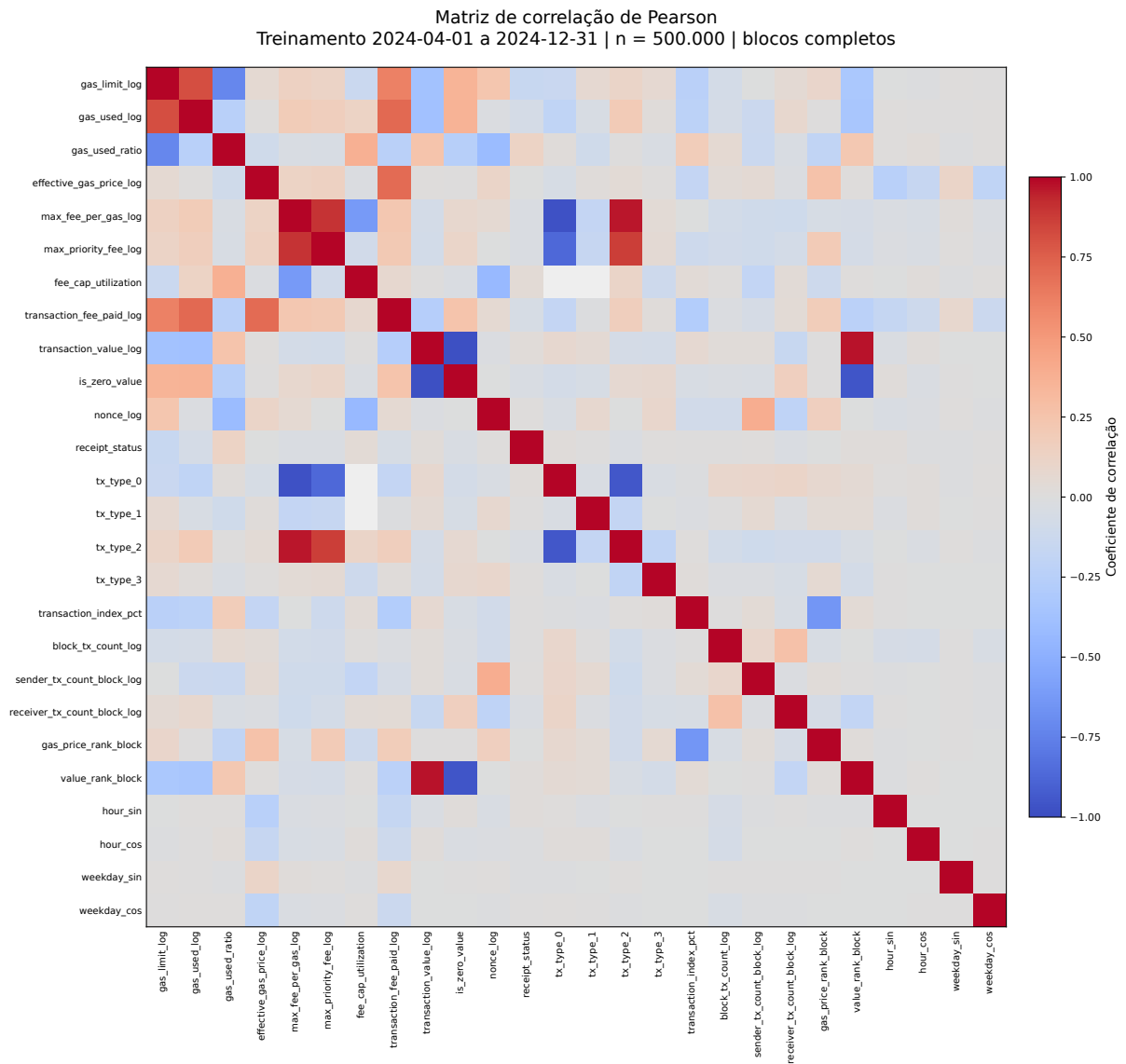


Figura 6 – Correlação de Pearson entre os atributos no período de treinamento.

Fonte: Elaboração própria a partir dos dados *on-chain* do Ethereum.

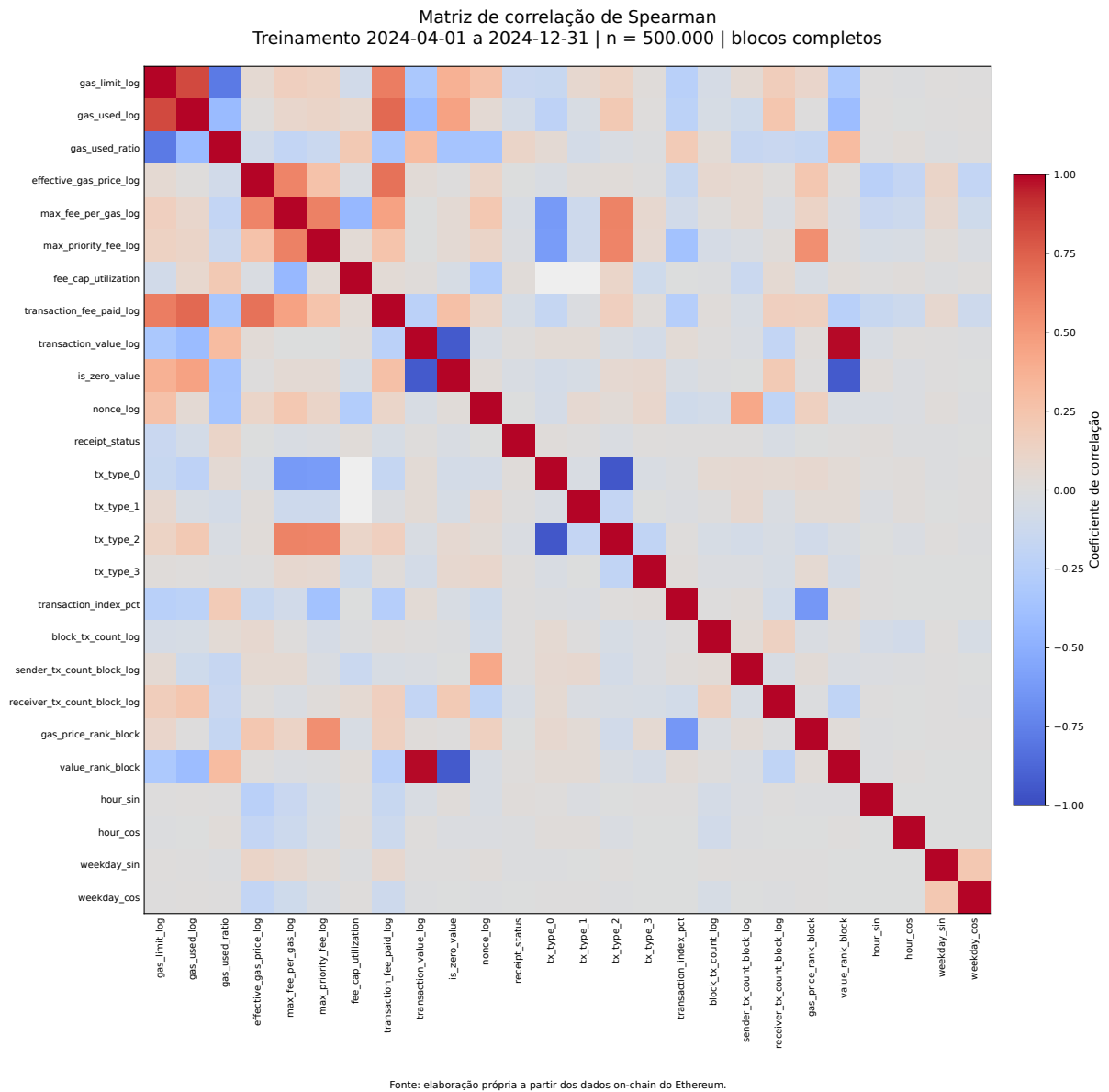


Figura 7 – Correlação de Spearman entre os atributos no período de treinamento.

Fonte: Elaboração própria a partir dos dados *on-chain* do Ethereum.

3.2.2 Atributos Observáveis e Atributos Derivados

A representação final contém 23 atributos organizados em sete grupos conceituais: execução, taxas, valor, conta, tipo de transação, contexto do bloco e contexto temporal. A Tabela 5 apresenta suas definições operacionais.

Tabela 5 – Atributos das matrizes experimentais.

Atributo	Grupo	Definição
gas_used_log	Execução	Transformação $\log(1 + x)$ do gás efetivamente consumido.
gas_used_ratio	Execução	Razão entre o gás consumido e o limite de gás informado pela transação.
effective_gas_price_log	Taxa	Transformação $\log(1 + x)$ do preço efetivo do gás, utilizando <code>gas_price</code> como valor alternativo quando necessário.
max_fee_per_gas_log	Taxa	Transformação $\log(1 + x)$ do teto máximo de taxa, assumindo zero quando o campo não é aplicável.
max_priority_fee_log	Taxa	Transformação $\log(1 + x)$ da taxa máxima de prioridade, assumindo zero quando não aplicável.
fee_cap_utilization	Taxa	Razão entre o preço efetivo do gás e o teto máximo de taxa, assumindo zero quando a relação não é aplicável.
transaction_value_log	Valor	Transformação $\log(1 + x)$ do valor nativo transferido.
nonce_log	Conta	Transformação $\log(1 + x)$ do <i>nonce</i> da conta remetente.
receipt_status	Execução	Indicador binário do resultado da execução, com valor 1 para sucesso e 0 para falha.
tx_type_0	Tipo	Indicador de transação Ethereum do tipo 0.
tx_type_1	Tipo	Indicador de transação Ethereum do tipo 1.
tx_type_3	Tipo	Indicador de transação do tipo 3, associada a <i>blob transactions</i> .
transaction_index_pct	Bloco	Índice da transação dividido pela quantidade de transações do bloco menos um, assumindo zero em blocos com uma única transação.
block_tx_count_log	Bloco	Transformação $\log(1 + x)$ da quantidade de transações presentes no bloco.
sender_tx_count_block_log	Bloco	Transformação $\log(1 + x)$ da frequência do remetente no mesmo bloco.
receiver_tx_count_block_log	Bloco	Transformação $\log(1 + x)$ da frequência do destinatário no mesmo bloco.
gas_price_rank_block	Bloco	Posição percentilica do preço efetivo do gás da transação em relação às demais transações do mesmo bloco.
hour_sin	Tempo	Componente seno da hora UTC de inclusão da transação.
hour_cos	Tempo	Componente cosseno da hora UTC de inclusão da transação.
weekday_sin	Tempo	Componente seno do dia da semana em UTC.
weekday_cos	Tempo	Componente cosseno do dia da semana em UTC.
gas_limit_log	Execução	Transformação $\log(1 + x)$ do limite de gás informado pela transação.
transaction_fee_paid_log	Taxa	Transformação $\log(1 + x)$ do produto entre o gás efetivamente consumido e o preço efetivo do gás.

Fonte: Elaboração própria.

A codificação do tipo da transação utiliza indicadores para os tipos 0, 1 e 3. No treinamento, o tipo 2 constitui a categoria de referência, representada por valores iguais a zero nesses indicadores. A mesma codificação é mantida nos períodos posteriores, sem

inclusão de novos atributos. Consequentemente, as transações do tipo 4 presentes nesses recortes também recebem zero nos três indicadores, não sendo explicitamente diferenciadas das transações do tipo 2 por esse grupo de atributos. Essa característica constitui uma limitação da representação diante da evolução do protocolo.

Variáveis com ampla faixa de valores são transformadas por $\log(1 + x)$, permitindo representar valores iguais a zero e reduzir a assimetria das distribuições. Essa transformação é aplicada, entre outros casos, ao consumo e limite de gás, taxas, valor transferido, *nonce* e contagens intrabloco.

Os atributos `gas_used_ratio`, `fee_cap_utilization` e `transaction_index_pct` representam relações relativas. Os atributos `block_tx_count_log`, `sender_tx_count_block_log`, `receiver_tx_count_block_log` e `gas_price_rank_block` incorporam o contexto do bloco, razão pela qual a amostragem preserva todas as transações dos blocos selecionados.

As variáveis temporais utilizam representações cíclicas da hora e do dia da semana, evitando tratar os limites dos respectivos ciclos como valores numericamente distantes. As definições e transformações apresentadas nesta seção são aplicadas de forma consistente aos diferentes recortes temporais.

3.2.3 Validação dos Atributos

Antes da composição das matrizes experimentais, os atributos foram submetidos a verificações de integridade no conjunto de treinamento, incluindo presença de valores ausentes ou não finitos, cardinalidade e ocorrência de variáveis constantes.

Os 23 atributos finais apresentaram valores válidos para as 16.080.647 transações do conjunto de treinamento, sem valores ausentes ou não finitos e sem atributos constantes. Essas verificações documentam a integridade numérica da representação utilizada nos experimentos, mas não asseguram, isoladamente, sua capacidade de distinguir comportamentos anômalos.

3.2.4 Pré-processamento e Escalonamento

Após a construção e validação dos atributos, as variáveis contínuas selecionadas são submetidas a escalonamento robusto. Todos os parâmetros da transformação são estimados exclusivamente no recorte `treino_2024_pos_dencun` e permanecem congelados nos demais períodos.

Para um atributo x , a transformação é definida por:

$$x' = \frac{x - \tilde{x}_{\text{treino}}}{\text{IQR}_{\text{treino}}}, \quad (3.8)$$

em que $\tilde{x}_{\text{treino}}$ representa a mediana e:

$$\text{IQR} = Q_3 - Q_1. \quad (3.9)$$

O uso da mediana e do intervalo interquartil reduz a influência de valores extremos sobre os parâmetros de escalonamento. Quando $\text{IQR}_{\text{treino}} = 0$, utiliza-se escala unitária após a centralização, evitando divisão por zero sem recorrer a informações de outros recortes.

Os atributos `max_fee_per_gas_log` e `max_priority_fee_log` recebem tratamento específico. Seus parâmetros são estimados somente sobre observações em que os campos são aplicáveis; para transações dos tipos 0 e 1, os valores permanecem iguais a zero. Dessa forma, preserva-se a distinção entre ausência de aplicabilidade e um valor de taxa efetivamente observado.

Razões, indicadores binários, posições percentílicas e componentes cíclicos são mantidos sem escalonamento adicional. Após o pré-processamento, as matrizes A, B e C preservam as 16.080.647 transações do conjunto de treinamento e possuem, respectivamente, 13, 21 e 23 atributos. As verificações registraram ausência de valores inválidos nas três matrizes, tanto no treinamento quanto nos demais recortes temporais.

A reutilização dos parâmetros estimados no treinamento mantém uma referência comum de transformação entre os períodos e evita o uso de informações dos demais recortes no ajuste do escalonamento. Esse procedimento não elimina mudanças temporais nas distribuições, que podem se refletir nos valores transformados e nos escores dos modelos.

3.3 Configurações de Atributos e Estudo de Ablação

A partir das matrizes A, B e C são definidas cinco configurações experimentais. As matrizes correspondem às representações produzidas pela engenharia e pelo pré-processamento, enquanto as configurações determinam os subconjuntos efetivamente fornecidos aos algoritmos.

As mesmas cinco configurações são utilizadas pelo Autoencoder e pelo Isolation Forest. Cada algoritmo produz, portanto, cinco modelos independentes, permitindo comparar os métodos sob representações equivalentes.

3.3.1 Configurações Completas

As configurações `a_completa`, `b_completa` e `c_completa` preservam integralmente os atributos das matrizes A, B e C, respectivamente.

A configuração `a_completa` contém 13 atributos e representa a configuração de menor dimensionalidade. A configuração `b_completa` contém 21 atributos, acrescentando informações de contexto intrabloco e temporal. A configuração `c_completa` contém 23 atributos e acrescenta à matriz B `gas_limit_log` e `transaction_fee_paid_log`.

Os conjuntos de atributos das configurações completas possuem relação incremental:

$$A \subset B \subset C, \quad (3.10)$$

com:

$$|A| = 13, \quad |B| = 21, \quad |C| = 23. \quad (3.11)$$

Nessa notação, A, B e C representam os conjuntos de atributos das respectivas matrizes. Esse desenho permite examinar o efeito da incorporação progressiva de informações contextuais e de atributos relacionados à execução e ao custo da transação, sem pressupor que o aumento da dimensionalidade resulte em melhor desempenho.

3.3.2 Remoção das Taxas Absolutas

Além das configurações completas, são consideradas duas variantes de ablação, `b_sem_taxas_absolutas` e `c_sem_taxas_absolutas`, destinadas a avaliar a influência das magnitudes absolutas relacionadas às taxas.

Define-se:

$$F_{\text{abs}} = \left\{ \begin{array}{l} \text{effective_gas_price_log,} \\ \text{max_fee_per_gas_log,} \\ \text{max_priority_fee_log,} \\ \text{transaction_fee_paid_log} \end{array} \right\}. \quad (3.12)$$

As configurações de ablação são obtidas por:

$$B_{\text{sem taxas absolutas}} = B \setminus F_{\text{abs}}, \quad C_{\text{sem taxas absolutas}} = C \setminus F_{\text{abs}}. \quad (3.13)$$

A operação considera somente os atributos efetivamente presentes na matriz de origem. Como `transaction_fee_paid_log` pertence apenas a C, `b_sem_taxas_absolutas` possui 18 atributos e `c_sem_taxas_absolutas`, 19.

A Tabela 6 resume as cinco configurações utilizadas.

A ablação consiste na remoção dos atributos pertencentes a F_{abs} e no treinamento de novos modelos com os atributos restantes. Não se trata, portanto, de retirar variáveis apenas durante a aplicação de um modelo previamente ajustado. São preservadas informações relativas associadas às taxas, particularmente `fee_cap_utilization` e `gas_price_rank_block`, além dos demais atributos transacionais e contextuais. A comparação com as configurações completas permite examinar como a retirada dessas magnitudes afeta os escores, as taxas de sinalização e o desempenho nas avaliações. Essa

Tabela 6 – Configurações de atributos dos modelos.

Configuração	Base	Atributos	Caracterização
a_completa	A	13	Representação transacional de menor dimensionalidade.
b_completa	B	21	Matriz A acrescida de contexto intrabloco e temporal.
c_completa	C	23	Matriz B acrescida do limite de gás e do custo total da transação.
b_sem_taxas_absolutas	B	18	Matriz B sem as três magnitudes absolutas de taxas nela presentes.
c_sem_taxas_absolutas	C	19	Matriz C sem os quatro atributos definidos como taxas absolutas.

Fonte: Elaboração própria.

análise não implica eliminar toda informação econômica, pois o valor nativo transferido e as relações relativas de taxas permanecem na representação.

As cinco configurações são definidas antes da avaliação dos conjuntos posteriores e permanecem inalteradas durante os experimentos.

3.4 Organização Temporal dos Experimentos

A organização temporal separa os dados destinados ao ajuste dos modelos e do pré-processamento daqueles utilizados para calibração e avaliação. Os parâmetros de pré-processamento e os modelos são ajustados exclusivamente no treinamento. A validação temporal é utilizada para calibrar os limiares e selecionar as configurações na avaliação semântica, enquanto o teste final permanece reservado à avaliação. Os sete recortes e seus intervalos foram apresentados na Seção 3.1.2. Nesta seção são descritas suas funções no protocolo experimental e as análises de mudança de distribuição.

3.4.1 Separação Cronológica entre Treinamento, Validação e Teste

O ajuste dos modelos e dos parâmetros de pré-processamento utiliza exclusivamente `treino_2024_pos_dencun`. O recorte `validacao_2025_pre_pectra`, temporalmente posterior, calibra os limiares Q_{99} , $Q_{99,5}$ e $Q_{99,9}$ e seleciona configurações e cortes na avaliação semântica. O `teste_final_2025` permanece fora dessas decisões.

O fluxo principal é, portanto:

$$D_{\text{treino}} \longrightarrow D_{\text{validação}} \longrightarrow D_{\text{teste}}. \quad (3.14)$$

No Autoencoder existe adicionalmente uma validação interna ao próprio recorte de treinamento. A divisão é determinística: observações cujo `row_id` satisfaz

$$\text{row_id} \bmod 10 = 0 \quad (3.15)$$

são reservadas para acompanhamento da perda, seleção do estado do modelo e aplicação do critério de *early stopping*. As demais participam da atualização dos pesos. Na execução de 5%, essa divisão destinou 14.472.583 observações ao ajuste dos pesos e 1.608.064 à validação interna, correspondendo a aproximadamente 90% e 10% do recorte de treinamento, respectivamente.

Essa divisão não deve ser confundida com a validação temporal de 2025:

$$\begin{aligned} V_{\text{interna}}^{\text{AE}} &\rightarrow \text{seleção do estado do modelo e } \textit{early stopping}, \\ D_{\text{validação}} &\rightarrow \text{calibração e seleção na avaliação semântica.} \end{aligned} \quad (3.16)$$

O Isolation Forest não utiliza validação interna para *early stopping*. Seu ajuste permanece restrito a uma amostra determinística do conjunto de treinamento, descrita na Seção 3.5.2.

3.4.2 Recortes Temporais e Atualizações da Rede

Além de treinamento, validação e teste, o desenho experimental inclui recortes associados aos períodos Pré-Dencun, Transição Dencun, Transição Pectra e Fusaka. Essas janelas permitem examinar os modelos em dados anteriores ao treinamento, em períodos próximos a mudanças do protocolo e em um período posterior ao teste principal.

As atualizações Dencun, Pectra e Fusaka são utilizadas como marcos cronológicos, e não como variáveis explicativas ou evidências causais. Eventuais alterações nas distribuições dos atributos, nos escores ou nas taxas de sinalização são, portanto, descritas em relação aos períodos observados sem atribuí-las automaticamente às atualizações.

O recorte Pré-Dencun constitui uma análise retrospectiva de estresse, pois é temporalmente anterior ao treinamento. Fusaka, por sua vez, fornece uma janela posterior de estresse na qual modelos, pré-processamento e limiares permanecem congelados.

3.4.3 Análise de Mudança de Distribuição

A estabilidade dos atributos é examinada por meio de uma análise diagnóstica de mudança de distribuição (*distribution drift*). O recorte `treino_2024_pos_dencun` é utilizado como referência fixa, e todos os parâmetros necessários às comparações são determinados exclusivamente a partir dele.

A análise é aplicada aos 23 atributos da matriz C. Para cada atributo e recorte são calculadas medidas complementares de mudança de distribuição.

A primeira medida é o *Population Stability Index* (PSI). Os limites dos intervalos são definidos pelos decis da distribuição de treinamento. Limites repetidos são removidos,

de modo que o número efetivo de intervalos K pode ser inferior a dez, particularmente em atributos discretos ou com elevada concentração de valores. Se p_i representa a proporção de observações da referência e q_i a proporção do recorte comparado no intervalo i , então:

$$\text{PSI} = \sum_{i=1}^K (\tilde{q}_i - \tilde{p}_i) \ln \left(\frac{\tilde{q}_i}{\tilde{p}_i} \right), \quad (3.17)$$

em que $\tilde{p}_i = \max(p_i, \varepsilon)$ e $\tilde{q}_i = \max(q_i, \varepsilon)$, com $\varepsilon = 10^{-6}$. Esse tratamento evita divisões por zero e logaritmos indefinidos em intervalos sem observações.

Como referência descritiva, valores inferiores a 0,10 são tratados como baixa mudança, valores entre 0,10 e 0,25 como mudança moderada e valores iguais ou superiores a 0,25 como mudança elevada. Essas faixas são heurísticas e não representam níveis de significância estatística(26).

Como medida complementar, utiliza-se uma aproximação da estatística de Kolmogorov–Smirnov(27):

$$D = \sup_x |F_{\text{treino}}(x) - F_{\text{recorte}}(x)|. \quad (3.18)$$

Na implementação, a estatística é aproximada sobre uma grade de 101 quantis determinada pela distribuição de treinamento e não são calculados valores- p . Valores inferiores a 0,10 são tratados descritivamente como baixa mudança, entre 0,10 e 0,20 como moderada e iguais ou superiores a 0,20 como elevada.

Também são registradas a mediana, o intervalo interquartil,

$$\text{IQR} = Q_{0,75} - Q_{0,25}, \quad (3.19)$$

a proporção de valores iguais a zero e medidas de extrapolação em relação à distribuição de treinamento. Estas últimas quantificam observações situadas fora do intervalo mínimo–máximo da referência e de suas cercas robustas.

As transações do tipo 4 são monitoradas separadamente. Como essa categoria não estava presente no período de treinamento, ela não possui indicador específico na representação fornecida aos modelos. Sua identificação é preservada apenas nos metadados para permitir análises estratificadas posteriores.

As medidas de *drift* possuem finalidade diagnóstica. Elas não são utilizadas para reajustar os modelos, modificar atributos ou recalibrar os limiares, nem constituem evidência de comportamento malicioso. Sua função é fornecer contexto para a interpretação de alterações temporais nos escores e nas taxas de sinalização.

3.5 Treinamento dos Modelos

Autoencoder e Isolation Forest são treinados independentemente para cada uma das cinco configurações de atributos, resultando em dez modelos. Todo o ajuste é restrito ao recorte `treino_2024_pos_dencun`, e os parâmetros de pré-processamento e as configurações experimentais permanecem congelados nos recortes posteriores.

Nos procedimentos estocásticos é utilizada semente pseudoaleatória igual a 42. As subseções seguintes apresentam os parâmetros específicos de cada algoritmo.

3.5.1 Autoencoder

O Autoencoder utilizado é uma rede neural *feedforward* simétrica. Para uma entrada $\mathbf{x} \in \mathbb{R}^d$, o modelo aprende uma reconstrução:

$$\hat{\mathbf{x}} = f_{\theta}(\mathbf{x}), \quad (3.20)$$

em que θ representa os parâmetros aprendidos.

A arquitetura utilizada é:

$$d \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow d. \quad (3.21)$$

A dimensão d varia conforme a configuração experimental, assumindo 13, 21, 23, 18 ou 19 atributos. A representação latente possui dimensão 4 em todos os modelos.

As camadas intermediárias utilizam LeakyReLU com coeficiente negativo 0,1, e a saída é linear. A função objetivo é o erro quadrático médio:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{d} \sum_{j=1}^d (x_j - \hat{x}_j)^2. \quad (3.22)$$

O treinamento utiliza AdamW(28) com *learning rate* de 10^{-3} , *weight decay* de 10^{-5} , lotes de 65.536 observações e limite máximo de 200 épocas.

A validação interna descrita na Seção 3.4.1 é utilizada para acompanhar a perda e selecionar o estado do modelo. Uma melhoria é reconhecida quando a perda diminui mais de 10^{-5} em relação ao melhor valor anteriormente aceito. O critério de *early stopping* prevê a interrupção após 12 épocas consecutivas sem melhoria suficiente, com restauração dos pesos do melhor estado registrado segundo esse critério. Na execução de 5%, as cinco configurações atingiram o limite de 200 épocas, sem acionamento da interrupção antecipada.

Na execução de 5%, os cinco modelos foram treinados em GPU, utilizando precisão numérica mista automática (*Automatic Mixed Precision* – AMP). As sementes do Python,

NumPy e PyTorch são fixadas em 42, e o *backend* cuDNN é configurado para favorecer execução determinística, com desativação da seleção automática de algoritmos por *benchmark*. Essas medidas favorecem a reprodutibilidade, mas não asseguram identidade numérica entre diferentes plataformas ou versões das bibliotecas. Para cada configuração são preservados o estado selecionado do modelo, o histórico de treinamento e os parâmetros do experimento.

3.5.2 Isolation Forest

Para cada configuração de atributos é ajustado um Isolation Forest, método proposto por Liu, Ting e Zhou (16), composto por 200 árvores. Cada árvore utiliza no máximo 8.192 observações, todos os atributos permanecem disponíveis (`max_features=1.0`) e a amostragem é realizada sem reposição.

Devido ao volume do conjunto de treinamento, o ajuste utiliza uma subamostra determinística de `treino_2024_pos_dencun`, com tamanho alvo de 500.000 transações. Se N representa o número de observações disponíveis, define-se:

$$n_{\text{alvo}} = \min(N, 500000). \quad (3.23)$$

Quando $N > n_{\text{alvo}}$, uma observação é selecionada se:

$$\text{hash}(\text{row_id}) \bmod N < n_{\text{alvo}}. \quad (3.24)$$

Esse procedimento distribui a seleção pelo conjunto de treinamento sem exigir sua ordenação pelo valor do *hash*. O tamanho efetivo pode diferir do alvo, pois o critério não impõe uma cardinalidade exata. Na execução de 5%, foram selecionadas 501.045 das 16.080.647 transações do recorte, correspondendo a aproximadamente 3,12%. A mesma subamostra é utilizada nas cinco configurações de atributos. Quando o conjunto possui até 500.000 observações, todas são utilizadas.

A seleção é distribuída pelo conjunto por meio de uma função de *hash* aplicada ao `row_id`, permitindo reproduzir a mesma amostra e evitando selecionar simplesmente as primeiras observações do período.

O estimador utiliza `contamination=auto`, mas esse parâmetro não define a sinalização final. Os limiares empregados nos experimentos são calibrados posteriormente no conjunto de validação temporal, conforme a Seção 3.6.

A semente pseudoaleatória é fixada em 42. Para cada configuração são preservados o estimador ajustado, a lista de atributos e as informações de pré-processamento necessárias à reprodução.

Ao final, o desenho experimental produz:

$$5 \text{ configurações} \times 2 \text{ algoritmos} = 10 \text{ modelos.} \quad (3.25)$$

3.6 Produção dos Escores e Definição dos Limiares

Após o treinamento, cada um dos dez modelos produz um escore contínuo para cada transação. Como Autoencoder e Isolation Forest utilizam mecanismos distintos, seus escores possuem escalas próprias e não são comparados diretamente em magnitude.

Os limiares são, portanto, calibrados separadamente para cada combinação entre algoritmo e configuração de atributos, utilizando exclusivamente o recorte `validacao_2025_pre_pectra`. Uma vez determinados, permanecem congelados nos demais recortes temporais.

3.6.1 Escores de Anomalia

Para o Autoencoder, o escore corresponde ao erro quadrático médio de reconstrução. Para uma observação $\mathbf{x}_i \in \mathbb{R}^d$ e sua reconstrução $\hat{\mathbf{x}}_i$:

$$s_{\text{AE}}(\mathbf{x}_i) = \frac{1}{d} \sum_{j=1}^d (x_{ij} - \hat{x}_{ij})^2. \quad (3.26)$$

A divisão por d expressa o erro médio por atributo, evitando que o escore corresponda simplesmente à soma dos erros de reconstrução. Entretanto, essa normalização não torna os escores diretamente comparáveis entre configurações, pois os conjuntos de atributos e suas distribuições diferem. Cada modelo possui, portanto, calibração própria de seus limiares.

No Isolation Forest, utiliza-se o negativo do valor retornado por `score_samples`:

$$s_{\text{IF}}(\mathbf{x}_i) = -\text{score_samples}(\mathbf{x}_i). \quad (3.27)$$

Essa transformação estabelece a mesma convenção interpretativa nos dois métodos:

$$s(\mathbf{x}_i) \uparrow \implies \text{maior grau de anomalia atribuído pelo modelo.} \quad (3.28)$$

A convenção não torna as escalas numericamente equivalentes. Comparações entre os algoritmos são realizadas posteriormente a partir das sinalizações e de medidas de concordância.

3.6.2 Calibração dos Limiares na Validação

Para cada modelo m , seja

$$S_{m,\text{val}} = \{s_m(\mathbf{x}_1), s_m(\mathbf{x}_2), \dots, s_m(\mathbf{x}_n)\} \quad (3.29)$$

o conjunto de escores produzido no recorte de validação temporal. O limiar associado ao quantil q é definido por:

$$\tau_m^{(q)} = Q_q(S_{m,\text{val}}). \quad (3.30)$$

O procedimento é realizado independentemente para os cinco Autoencoders e os cinco Isolation Forests, de modo que cada combinação entre algoritmo e configuração possui seus próprios limiares.

Nenhum conjunto de teste, transição ou estresse participa dessa calibração. Após a determinação de $\tau_m^{(q)}$, uma observação de qualquer outro recorte é sinalizada segundo:

$$\hat{y}_{i,m}^{(q)} = \mathbb{I}[s_m(\mathbf{x}_i) \geq \tau_m^{(q)}]. \quad (3.31)$$

A sinalização representa exclusivamente uma decisão estatística produzida pelo modelo e pelo limiar considerados. Em particular,

$$\hat{y}_{i,m}^{(q)} = 1 \not\Rightarrow \text{fraude, ataque, MEV ou front-running}. \quad (3.32)$$

3.6.3 Critérios $Q99$, $Q99,5$ e $Q99,9$

São utilizadas três políticas de sinalização:

$$q \in \{0,99, 0,995, 0,999\}, \quad (3.33)$$

correspondentes a $Q99$, $Q99,5$ e $Q99,9$. No próprio conjunto de validação, esses limiares delimitam aproximadamente os 1%, 0,5% e 0,1% superiores da distribuição dos escores, respectivamente.

$Q99,5$ constitui a referência principal das análises de sinalização temporal e de concordância entre os modelos. $Q99$ e $Q99,9$ permitem examinar a sensibilidade dos resultados a critérios menos e mais seletivos. Na avaliação semântica, a seleção entre os três limiares é realizada pelo maior F_1 na validação, após a seleção da configuração de atributos. Na execução de 5%, esse procedimento selecionou $Q99$ para ambos os algoritmos, tanto na política estrita quanto na de sensibilidade, sem utilizar os resultados do teste final.

Nos demais recortes, os limiares não são recalibrados. Para um modelo m , recorte D_t e quantil q , a taxa de sinalização é:

$$R_{m,t}^{(q)} = \frac{1}{|D_t|} \sum_{\mathbf{x}_i \in D_t} \mathbb{I} \left[s_m(\mathbf{x}_i) \geq \tau_m^{(q)} \right]. \quad (3.34)$$

Consequentemente, as taxas observadas fora da validação não são forçadas a permanecer próximas de 1%, 0,5% ou 0,1%. Sua variação constitui parte da resposta dos modelos às distribuições encontradas nos diferentes períodos.

Os valores numéricos dos três limiares são calculados exclusivamente a partir da distribuição dos escores na validação temporal, sem rótulos externos. A escolha entre esses limiares na avaliação semântica utiliza os rótulos derivados da adjudicação no mesmo recorte, sem alterar os valores previamente calibrados. O parâmetro `contamination=auto` do Isolation Forest não determina as sinalizações finais, que seguem os quantis calibrados na validação temporal.

3.7 Protocolo Experimental e Reprodutibilidade

O protocolo experimental preserva a rastreabilidade entre pré-processamento, configurações, modelos e resultados. Os parâmetros das transformações são estimados exclusivamente no treinamento e persistidos em um artefato identificado por uma assinatura de pré-processamento compartilhada pelos experimentos com Autoencoder e Isolation Forest.

Cada execução de treinamento possui adicionalmente uma assinatura experimental construída a partir das informações relevantes à sua reprodução, incluindo algoritmo, recortes utilizados, configuração de atributos, hiperparâmetros e assinatura do pré-processamento. Essas informações são registradas nos manifestos associados aos artefatos produzidos.

O pipeline permite organizar múltiplas execuções por meio do arquivo `pipeline.json`, que define o identificador da execução, o percentual de blocos e os diretórios utilizados. Cada execução mantém seus modelos, resultados e manifestos em um diretório próprio, sob `execucoes/<identificador>`, permitindo preservar os artefatos de diferentes configurações de amostragem. Nesta dissertação, são analisados exclusivamente os resultados da execução `amostra-05pct`.

Os dados obtidos por RPC e os metadados de tokens são armazenados em um cache compartilhado entre execuções. A reutilização desses dados reduz consultas repetidas aos provedores, sem implicar reutilização dos modelos treinados ou dos resultados analíticos, que permanecem específicos de cada execução.

As operações pseudoaleatórias controladas pelo protocolo utilizam as sementes definidas nas seções anteriores. As divisões internas e as amostras de treinamento também seguem critérios determinísticos, permitindo reproduzir os subconjuntos empregados quando preservadas as mesmas entradas e configurações.

O treinamento do Autoencoder foi executado em uma NVIDIA GeForce RTX 3060 com CUDA e precisão numérica mista automática. O ambiente registrado utilizou Python 3.13.14, PyTorch 2.14.0 com CUDA 13.0, NumPy 2.5.3, PyArrow 23.0.1, DuckDB 1.5.5 e Matplotlib 3.11.2.

O experimento com Isolation Forest utilizou Python 3.13.14, NumPy 2.5.3, PyArrow 23.0.1, DuckDB 1.5.5, scikit-learn 1.9.1 e Joblib 1.6.0.

O protocolo mantém a sequência:

$$\text{treinamento} \rightarrow \text{validação, calibração e seleção} \rightarrow \text{teste}, \quad (3.35)$$

sem utilizar resultados do teste para reajustar arquitetura, atributos, pré-processamento ou limiares. A seleção na avaliação semântica utiliza exclusivamente o recorte de validação, conforme o procedimento descrito nas seções correspondentes.

São preservados, como artefatos independentes, os modelos treinados, escores por recorte, limiares calibrados, estatísticas descritivas, taxas de sinalização e manifestos experimentais. Para o Autoencoder também são mantidos o histórico de treinamento e as curvas de perda.

Esses mecanismos permitem relacionar os resultados às condições sob as quais foram produzidos. Não se assume, entretanto, equivalência *bit a bit* entre diferentes plataformas de hardware ou versões de bibliotecas, especialmente para operações executadas em GPU. A reprodutibilidade considerada neste trabalho corresponde à preservação documentada dos dados, transformações, configurações e decisões necessárias à reexecução do protocolo.

3.7.1 Disponibilidade do Código e dos Artefatos

O código-fonte, as configurações e os artefatos destinados à inspeção dos resultados estão disponíveis em repositório público no GitHub. A versão utilizada para a execução de 5% encontra-se na branch **feature/multiplas-execucoes**:

<<https://github.com/marcelocorni/dissertacao/tree/feature/multiplas-execucoes>>

Os resultados analisados nesta dissertação estão organizados no diretório **execucoes/amostra-05pct**. Os artefatos publicados incluem manifestos, configurações experimentais, tabelas de resultados, relatórios e figuras, permitindo inspecionar as condições de execução e os resultados das etapas do protocolo.

Os arquivos Parquet de dados e matrizes intermediárias não são incluídos no repositório devido ao seu volume. Também permanecem excluídos o cache compartilhado, os ambientes locais e os diretórios de compilação **bin** e **obj**. Os dados de origem podem ser obtidos no AWS Public Blockchain Dataset, conforme as instruções de coleta disponibilizadas no repositório.

Como recurso opcional para reprodução, o cache compartilhado acumulado nos experimentos até a execução de 5% é disponibilizado no arquivo `cache-compartilhado.7z`. O README do repositório informa o endereço para download, as instruções de extração e o resumo criptográfico SHA-256 para verificação da integridade do arquivo. Esse cache permite reutilizar respostas RPC e metadados de tokens já obtidos, mas não substitui os dados de origem nem assegura que uma nova execução dispense consultas adicionais.

A reprodução deve preservar a versão do código, as configurações, os dados de entrada e o ambiente documentado nos manifestos. O cache constitui um recurso de redução de consultas repetidas, não uma condição obrigatória para executar o protocolo.

3.8 Considerações Finais

Este capítulo apresentou o protocolo experimental utilizado para produzir escores não supervisionados de anomalia a partir de informações observáveis na Ethereum. Os dados foram amostrados deterministicamente por blocos completos e organizados em recortes cronológicos, preservando a separação entre treinamento, validação, teste e análises temporais.

A execução analisada utiliza uma amostra determinística de aproximadamente 5% dos blocos, preservando todas as transações dos blocos selecionados. A representação transacional foi organizada em cinco configurações de atributos, incluindo duas variantes de ablação das taxas absolutas, aplicadas independentemente ao Autoencoder e ao Isolation Forest. O pré-processamento e os modelos são ajustados exclusivamente a partir do período de treinamento. Os limiares $Q99$, $Q99,5$ e $Q99,9$ são calibrados na validação e mantidos congelados nos demais períodos. A seleção das configurações e dos limiares na avaliação semântica também utiliza exclusivamente a validação, preservando o teste final para avaliação. Medidas de mudança de distribuição fornecem contexto adicional para a análise temporal sem modificar retrospectivamente o protocolo.

O resultado dessa metodologia é um conjunto de escores e sinalizações de natureza estatística. Essas sinalizações não constituem, isoladamente, evidência de fraude, MEV ou *front-running*. O Capítulo 4 apresenta as camadas independentes de avaliação utilizadas para caracterizar os escores e relacioná-los a evidências externas, estruturais e semânticas.

4 AVALIAÇÃO E VALIDAÇÃO

4.1 Estratégia Geral de Avaliação

A avaliação dos modelos foi estruturada considerando a natureza não supervisionada do problema investigado. Como o Autoencoder e o Isolation Forest foram treinados sem rótulos de normalidade, ataque ou comportamento malicioso, um escore elevado representa inicialmente apenas um desvio em relação ao comportamento aprendido pelo modelo. Dessa forma, uma transação sinalizada como anômala não é automaticamente interpretada como evidência de fraude, MEV ou *front-running*.

Para separar a detecção estatística de sua interpretação posterior, a avaliação foi organizada em quatro camadas complementares. A primeira caracteriza as distribuições dos escores, as taxas de sinalização e sua estabilidade temporal. A segunda examina a concordância entre Autoencoder e Isolation Forest, permitindo identificar sinalizações compartilhadas e específicas de cada método. A terceira introduz o Label Cloud como fonte externa e independente de informação. Por fim, a quarta relaciona os escores aos eventos derivados de candidatos estruturais de *front-running*, gerados independentemente das sinalizações dos modelos, enriquecidos por RPC e submetidos à validação semântica e à adjudicação automatizadas.

Essa estratégia pode ser representada conceitualmente por:

$$\begin{aligned}
 \text{escore de anomalia} &\rightarrow \text{avaliação intrínseca} \\
 &\rightarrow \text{concordância entre modelos} \\
 &\rightarrow \text{evidência externa} \\
 &\rightarrow \text{validação estrutural e semântica.}
 \end{aligned} \tag{4.1}$$

As camadas acrescentam progressivamente informações à interpretação dos escores, mas não constituem etapas de realimentação dos modelos. Os resultados obtidos durante a avaliação não são utilizados para modificar os atributos, retreinar os algoritmos ou recalibrar os limiares com base no conjunto de teste. Essa separação preserva a independência entre a produção das sinalizações e as evidências utilizadas posteriormente para avaliá-las.

Do ponto de vista interpretativo, distinguem-se três níveis de evidência: anomalia estatística, associação com uma fonte externa e evidência estrutural e semântica de um comportamento transacional específico. Esses níveis não são equivalentes. Uma transação pode ser estatisticamente anômala sem possuir correspondência no Label Cloud, assim como a associação de um endereço a uma categoria externa não comprova, isoladamente, a ocorrência de um evento de *front-running*.

A ausência de correspondência nas fontes externas também não é interpretada como evidência de normalidade. Cada referência possui cobertura e limitações próprias,

de modo que as métricas construídas a partir desses dados devem ser interpretadas no contexto da população efetivamente coberta.

A avaliação considera os dez modelos definidos no protocolo, correspondentes às cinco configurações experimentais aplicadas separadamente ao Autoencoder e ao Isolation Forest. Quando aplicável, são utilizados os limiares $Q99$, $Q99,5$ e $Q99,9$. $Q99,5$ constitui a referência principal das análises temporais e de concordância, enquanto os demais permitem examinar a sensibilidade das sinalizações. Na avaliação semântica, a configuração de cada algoritmo é selecionada pelo maior PR-AUC na validação. Em seguida, o limiar é selecionado pelo maior F_1 nesse mesmo recorte. Na execução de 5%, foram selecionados o Autoencoder com `c_sem_taxas_absolutas` e o Isolation Forest com `b_completa`, ambos com $Q99$, nas políticas estrita e de sensibilidade. O teste final não participa dessas decisões.

As seções seguintes apresentam os procedimentos utilizados para caracterizar os escores, comparar as sinalizações dos dois algoritmos, realizar a avaliação externa e investigar os padrões estruturais e semânticos relacionados a *front-running*.

4.2 Avaliação Intrínseca dos Escores de Anomalia

A avaliação intrínseca caracteriza o comportamento dos escores e das sinalizações sem recorrer a rótulos externos. A análise é realizada separadamente para cada combinação entre algoritmo e configuração de atributos, considerando os cinco modelos do Autoencoder e os cinco modelos do Isolation Forest ao longo dos sete recortes temporais definidos no protocolo experimental.

O objetivo é examinar a distribuição, a frequência e a estabilidade temporal das sinalizações, sem atribuir significado semântico às observações destacadas. Em particular, uma taxa de sinalização maior ou menor não é interpretada isoladamente como evidência de melhor ou pior desempenho.

4.2.1 Distribuição dos Escores

Para cada modelo e recorte temporal são calculadas estatísticas descritivas da distribuição dos escores de anomalia, incluindo mínimo, média, mediana, desvio padrão, máximo e os quantis $Q25$, $Q75$, $Q95$, $Q99$, $Q99,5$ e $Q99,9$.

Seja $S_{m,t}$ o conjunto de escores produzidos pelo modelo m no recorte temporal t :

$$S_{m,t} = \{s_m(x_1), s_m(x_2), \dots, s_m(x_{n_t})\}. \quad (4.2)$$

A mediana e os quartis caracterizam a região central da distribuição, enquanto os quantis superiores permitem examinar progressivamente sua cauda. Os quantis calculados em cada recorte possuem finalidade descritiva e não substituem os limiares calibrados na

validação, que permanecem fixos. Média e desvio padrão são mantidos como medidas complementares.

As distribuições são analisadas separadamente para Autoencoder e Isolation Forest. Como os algoritmos produzem escores por mecanismos distintos e em escalas próprias, seus valores absolutos não são comparados diretamente. A comparação entre os métodos é realizada posteriormente a partir das observações sinalizadas e das respectivas medidas de concordância.

4.2.2 Taxas de Sinalização

Para cada modelo, recorte temporal e limiar, é calculada a quantidade de observações cujo escore é maior ou igual ao valor previamente definido na validação.

Para um modelo m , recorte t e quantil q , a quantidade de observações sinalizadas é dada por:

$$A_{m,t}^{(q)} = \sum_{x_i \in D_t} \mathbb{I} \left[s_m(x_i) \geq \tau_m^{(q)} \right]. \quad (4.3)$$

em que $\tau_m^{(q)}$ corresponde ao limiar do modelo m associado ao quantil q . A taxa de sinalização é definida por:

$$R_{m,t}^{(q)} = \frac{A_{m,t}^{(q)}}{|D_t|}. \quad (4.4)$$

São considerados os limiares $Q99$, $Q99,5$ e $Q99,9$. Como esses valores são calibrados no recorte `validacao_2025_pre_pectra`, as taxas nesse conjunto correspondem aproximadamente a 1%, 0,5% e 0,1%, respectivamente. Essas proporções podem apresentar pequenas diferenças em decorrência de empates nos escores e da inclusão de observações iguais ao limiar.

Nos demais recortes, os valores dos limiares permanecem congelados. Dessa forma, as taxas observadas podem variar conforme a distribuição dos escores em cada período. Recalcular os quantis separadamente em cada recorte manteria artificialmente as proporções próximas aos valores de referência e ocultaria parte das alterações temporais que se pretende observar.

As taxas são, portanto, medidas descritivas da resposta dos modelos aos diferentes períodos. Elas não representam taxas de fraude, ataque ou *front-running*, nem constituem isoladamente medidas de erro ou acerto.

4.2.3 Estabilidade entre Recortes Temporais

A estabilidade temporal é examinada comparando-se as distribuições dos escores e as taxas de sinalização de cada modelo entre os diferentes recortes. O modelo, a representação

dos atributos, o pré-processamento e os limiares permanecem fixos, permitindo observar sua resposta quando aplicados a distribuições temporalmente distintas.

Para cada configuração são considerados os períodos Pré-Dencun, Transição Dencun, Pós-Dencun, Pré-Pectra, Transição Pectra, Pós-Pectra e Fusaka. A variação da taxa de sinalização em relação à referência de validação é expressa por:

$$\Delta R_{m,t}^{(q)} = R_{m,t}^{(q)} - R_{m,\text{val}}^{(q)}. \quad (4.5)$$

Valores positivos indicam que uma proporção maior de observações atingiu ou excedeu o limiar congelado naquele período, enquanto valores negativos indicam uma proporção menor. Essa medida é utilizada de forma descritiva e não estabelece, isoladamente, relação causal entre mudanças da rede e alterações na resposta dos modelos.

A interpretação da estabilidade considera também as mudanças observadas nas distribuições dos atributos. As configurações completas e suas variantes de ablação são submetidas ao mesmo protocolo, permitindo investigar se a remoção das taxas monetárias absolutas está associada a diferenças na estabilidade temporal das sinalizações.

Autoencoder e Isolation Forest são avaliados separadamente nessa etapa, uma vez que caracterizam anomalias segundo mecanismos distintos. A comparação entre eles é realizada posteriormente, evitando interpretar diferenças entre seus escores ou taxas como evidência de erro de um dos métodos.

A avaliação intrínseca fornece, assim, a caracterização inicial dos dez modelos e permite identificar alterações nos escores e nas frequências de sinalização antes da introdução de qualquer informação externa ou interpretação semântica.

4.3 Concordância entre Autoencoder e Isolation Forest

Além da avaliação individual dos escores e das taxas de sinalização, foi analisada a concordância entre Autoencoder e Isolation Forest. O objetivo é investigar em que medida métodos não supervisionados baseados em mecanismos distintos identificam as mesmas transações como anômalas quando recebem representações equivalentes dos dados.

A comparação é realizada entre modelos correspondentes à mesma configuração de atributos, evitando combinar simultaneamente diferenças de algoritmo e de representação. Para cada configuração e recorte temporal, são comparados os conjuntos de transações sinalizadas pelos dois métodos.

A análise principal utiliza o limiar $Q_{99,5}$, calibrado separadamente para cada modelo no recorte de validação e mantido congelado nos demais períodos. Como a comparação é realizada a partir dos conjuntos sinalizados, não é necessário que os escores produzidos pelos algoritmos estejam na mesma escala.

4.3.1 Comparação dos Conjuntos e Índice de Jaccard

Seja $A_{c,t}$ o conjunto de observações sinalizadas pelo Autoencoder para a configuração c no recorte temporal t , e $I_{c,t}$ o conjunto correspondente produzido pelo Isolation Forest:

$$A_{c,t} = \left\{ i : x_i \in D_t, s_{\text{AE},c}(x_i) \geq \tau_{\text{AE},c}^{(0,995)} \right\}, \quad (4.6)$$

$$I_{c,t} = \left\{ i : x_i \in D_t, s_{\text{IF},c}(x_i) \geq \tau_{\text{IF},c}^{(0,995)} \right\}. \quad (4.7)$$

Para cada par são registrados o número de observações sinalizadas por cada algoritmo, a interseção e a união dos conjuntos. A interseção representa as transações que atingem ou excedem simultaneamente os respectivos limiares, enquanto a união reúne todas aquelas sinalizadas por pelo menos um dos métodos.

A sobreposição é quantificada pelo índice de Jaccard(29):

$$J_{c,t} = \frac{|A_{c,t} \cap I_{c,t}|}{|A_{c,t} \cup I_{c,t}|}. \quad (4.8)$$

Valores mais elevados indicam maior sobreposição relativa entre os conjuntos, enquanto valores próximos de zero indicam pequena concordância. O índice é utilizado exclusivamente como medida descritiva e não como métrica de desempenho. Uma concordância elevada não implica maior capacidade de identificar eventos de interesse, assim como uma concordância reduzida indica apenas que os algoritmos priorizam conjuntos parcialmente distintos de observações.

A comparação é realizada nos sete recortes temporais, permitindo observar se a concordância se modifica ao longo do período analisado. Uma transação sinalizada simultaneamente pelos dois métodos apresenta evidência estatística convergente, mas essa convergência não constitui, isoladamente, confirmação de comportamento malicioso.

4.3.2 Concordância entre Configurações

Complementarmente, é analisada a concordância entre as diferentes configurações de atributos dentro de cada algoritmo no recorte de validação. Os cinco conjuntos produzidos pelo Autoencoder são comparados aos pares e o mesmo procedimento é aplicado separadamente ao Isolation Forest, resultando em dez comparações internas por algoritmo.

Essa análise permite avaliar a sensibilidade das sinalizações à representação de atributos mantendo fixo o mecanismo de detecção. Em particular, ela fornece elementos para examinar posteriormente o efeito das configurações experimentais e da remoção das taxas monetárias absolutas.

A concordância constitui, assim, uma perspectiva complementar à avaliação intrínseca. Enquanto as distribuições e taxas caracterizam a resposta individual dos modelos, o índice de Jaccard permite examinar em que medida diferentes algoritmos e representações convergem sobre as mesmas observações, sem atribuir significado semântico a essa convergência.

4.4 Avaliação Externa com o Label Cloud

A terceira camada de avaliação utiliza uma fonte externa de informação para investigar a relação entre os escores produzidos pelos modelos e endereços previamente associados a atividades relacionadas a MEV. Essa comparação é realizada somente após o treinamento, a produção dos escores e a calibração dos limiares, preservando sua independência em relação ao núcleo não supervisionado.

A referência utilizada consiste em um registro histórico derivado do sistema de *Name Tags* e *Labels* do Etherscan, posteriormente normalizado para utilização no protocolo experimental. Esses rótulos não são tratados como *ground truth*: sua cobertura é limitada aos endereços conhecidos e catalogados pela fonte, e a ausência de correspondência não constitui evidência de comportamento normal.

4.4.1 Fonte e Associação dos Rótulos

Após normalização e remoção de duplicidades, a referência externa, derivada do sistema de Name Tags e Labels do Etherscan(30), contém 3.750 endereços únicos distribuídos entre as categorias `mev_builder`, `mev_bot`, `mev_relay` e `mev_protection`. Essas categorias possuem significados distintos: `mev_builder` e `mev_relay`, por exemplo, podem representar componentes de infraestrutura, enquanto `mev_protection` está associada a mecanismos de proteção. Por essa razão, apenas `mev_bot`, contendo 3.748 endereços únicos na referência normalizada, é utilizada como evidência positiva na avaliação binária.

A data de coleta da referência externa não está documentada no arquivo fornecido. Por esse motivo, o registro é tratado como uma referência histórica fixa, sem assumir que suas classificações estavam disponíveis nas datas das transações analisadas. A associação possui finalidade retrospectiva e não representa uma avaliação prospectiva com rótulos conhecidos em cada período.

Mesmo nesse caso, o rótulo representa associação histórica do endereço a uma categoria externa, e não confirmação de que cada transação originada por ele corresponda a uma estratégia específica de MEV ou a um ataque.

A associação é realizada pelo endereço de origem da transação. Seja a_i o endereço de origem da transação i e L_{MEVBot} o conjunto de endereços classificados como `mev_bot`. Defina-se:

$$y_i^{LC} = \begin{cases} 1, & \text{se } \text{lower}(a_i) \in L_{\text{MEVBot}}, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.9)$$

A classe $y_i^{LC} = 1$ representa, portanto, uma correspondência observada com a referência externa. Por outro lado,

$$y_i^{LC} = 0 \not\Rightarrow \text{ausência de MEV ou comportamento anômalo.} \quad (4.10)$$

Essa assimetria é fundamental para a interpretação das métricas. As denominações “positiva” e “negativa” são utilizadas operacionalmente para permitir o cálculo das métricas binárias, mas a classe zero representa ausência de correspondência na referência disponível, e não normalidade confirmada.

O procedimento foi aplicado aos sete recortes temporais, totalizando 47.545.805 transações associadas ao indicador externo. A prevalência da classe positiva varia entre os períodos em função da atividade dos endereços catalogados. No teste final, 250.616 das 14.127.430 transações apresentam correspondência com a categoria `mev_bot`, representando aproximadamente 1,77% desse recorte. Em todos os casos, a atribuição dos rótulos independe dos escores e das sinalizações produzidas pelos modelos.

4.4.2 Avaliação dos Escores Contínuos

A associação entre os escores contínuos e o indicador externo é avaliada por meio de ROC-AUC(21) e Average Precision (AP). As duas métricas são calculadas para cada combinação entre algoritmo, configuração de atributos e recorte temporal.

A ROC-AUC avalia a capacidade de ordenação dos escores ao longo de diferentes limiares e pode ser interpretada, neste contexto, como a tendência de uma transação pertencente à classe positiva operacional receber escore superior ao de uma observação da classe zero. Por utilizar os escores contínuos, seu cálculo independe dos limiares Q_{99} , $Q_{99,5}$ e $Q_{99,9}$.

Como a classe positiva é rara e a classe zero não representa um conjunto confirmado de transações normais, a ROC-AUC é analisada em conjunto com a curva Precision–Recall(22, 23). Para um determinado ponto de corte, *precision* e *recall* são definidas por:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}. \quad (4.11)$$

A *Average Precision* resume o comportamento da relação entre *precision* e *recall* ao longo da ordenação. Sua interpretação considera a prevalência da classe positiva em

cada recorte, uma vez que, em populações fortemente desbalanceadas, a magnitude da métrica deve ser analisada em relação à frequência de base da classe de interesse.

ROC-AUC e *Average Precision* são, portanto, utilizadas como medidas complementares da associação entre a ordenação produzida pelos modelos e a referência externa. Nenhuma delas é utilizada para reajustar os modelos, escolher atributos com base no conjunto de teste ou recalibrar os limiares de anomalia.

4.4.3 Avaliação dos Limiares e do Topo do Ranqueamento

Além da avaliação dos escores contínuos, os limiares $Q99$, $Q99,5$ e $Q99,9$ são confrontados com o indicador externo. Para cada limiar são obtidos os elementos da matriz de confusão e as métricas derivadas correspondentes. $Q99,5$ permanece como referência principal, enquanto $Q99$ e $Q99,9$ são utilizados na análise de sensibilidade.

Esses limiares não são otimizados em função do Label Cloud. Eles foram definidos anteriormente a partir da distribuição dos escores no protocolo não supervisionado, de modo que a avaliação externa mede o comportamento de políticas de sinalização estabelecidas antes da introdução dos rótulos.

Nesse contexto, um falso positivo corresponde operacionalmente a uma transação sinalizada cujo endereço de origem não pertence ao conjunto histórico `mev_bot`. Essa denominação não implica que a transação seja efetivamente normal ou que não esteja relacionada a alguma atividade de MEV.

Também é avaliada a concentração da classe positiva nas regiões superiores do ranqueamento, abordagem particularmente útil para examinar os itens de maior prioridade(24). Seja T_p o conjunto correspondente à fração p das transações com maiores escores e π a prevalência da classe positiva no recorte considerado. O enriquecimento no topo pode ser expresso pelo *lift*:

$$\text{Lift}@p = \frac{\frac{\sum_{i \in T_p} y_i^{LC}}{|T_p|}}{\pi}. \quad (4.12)$$

Valores superiores a 1 indicam que a classe positiva está mais concentrada na região selecionada do que na população completa. A análise considera as frações superiores de 0,1%, 0,5%, 1% e 5% das transações de cada recorte. Diferentemente da sinalização por limiares congelados, essa seleção utiliza uma quantidade previamente definida de observações com os maiores escores no próprio recorte, arredondada para cima. O procedimento permite avaliar a concentração da referência externa sob diferentes volumes de transações priorizadas, sem recalibrar os limiares de anomalia.

Essa avaliação complementa ROC-AUC e Average Precision porque enfatiza a região do ranqueamento mais relevante em um cenário de triagem. Um modelo pode

produzir uma ordenação global informativa sem que um limiar extremo recupere grande parte da classe positiva; da mesma forma, a concentração no topo pode revelar utilidade operacional mesmo quando a classe de interesse possui prevalência reduzida.

Em conjunto, as métricas contínuas, as políticas de limiar e as medidas de enriquecimento permitem avaliar diferentes propriedades da associação entre os escores e o Label Cloud. Os resultados permanecem condicionados à cobertura e à natureza histórica dessa referência e não são interpretados como estimativas da prevalência de MEV na rede.

4.5 Validação Estrutural e Semântica dos Candidatos de Front-Running

A quarta camada de avaliação investiga se os escores produzidos pelos modelos apresentam relação com padrões transacionais de *front-running*. Como Autoencoder e Isolation Forest identificam desvios estatísticos sem atribuir significado às observações, foi implementado um processo independente de geração, enriquecimento e validação de candidatos estruturais.

As regras utilizadas foram adaptadas da taxonomia de Eskandari, Moosavi e Clark (5) e da abordagem empírica de Torres, Camino e State (10). A identificação é executada sobre todas as transações pertencentes aos blocos selecionados pela amostragem determinística, independentemente de seus escores ou de terem sido sinalizadas por qualquer modelo. Essa separação evita que a população utilizada na avaliação semântica seja previamente condicionada pelas próprias sinalizações que serão posteriormente avaliadas.

Dentro de cada bloco, as transações são ordenadas pelo campo `transaction_index`. As regras estruturais utilizam informações disponíveis nos dados locais, como posição, remetente, destinatário, valor e preço de gás, para produzir candidatos aos padrões de deslocamento, inserção e supressão. Esses candidatos representam configurações compatíveis com os padrões investigados, mas não constituem, nessa etapa, rótulos positivos.

O processo completo pode ser sintetizado por:

$$\begin{aligned}
 \text{blocos amostrados} &\rightarrow \text{regras estruturais} \rightarrow \text{eventos candidatos} \\
 &\rightarrow \text{enriquecimento por RPC} \rightarrow \text{validação semântica} \\
 &\rightarrow \text{adjudicação automatizada} \rightarrow \text{rótulos consolidados.}
 \end{aligned} \tag{4.13}$$

Informações adicionais, como *calldata*, recibos e logs, são recuperadas por RPC para os blocos que contêm transações requisitadas pela validação dos candidatos. Na execução de 5%, o enriquecimento utiliza consultas por bloco, que retornam as transações completas e os respectivos recibos. Dessas respostas são extraídas as evidências necessárias às transações da fila de enriquecimento. Essa estratégia mantém a geração dos candidatos

independente dos modelos e evita consultar indiscriminadamente todos os blocos da amostra. As respostas obtidas são preservadas em cache para reutilização.

4.5.1 Adaptação do Método de Identificação Estrutural

A implementação toma como referência a taxonomia de Eskandari, Moosavi e Clark (5) e a metodologia empírica apresentada por Torres, Camino e State (10), mas não constitui uma reprodução literal do sistema *Frontrunner Jones*. As regras foram adaptadas ao período de 2024–2025, à estrutura dos dados disponíveis e ao objetivo de produzir candidatos para posterior validação semântica.

O método de referência foi desenvolvido sobre blocos produzidos entre 2015 e 2020, em um contexto anterior à transição da Ethereum para *proof-of-stake*, à EIP-1559 e à consolidação dos mecanismos contemporâneos de construção de blocos. No período analisado nesta dissertação, a posição de uma transação não pode ser explicada exclusivamente pelo preço de gás informado pelo remetente. Construtores de blocos e fluxos privados, entre outros mecanismos, também influenciam a ordenação(11). Assim, diferenças de taxa e relações de precedência são tratadas como evidências estruturais, e não como confirmação de causalidade.

Outra diferença está na finalidade da identificação. Neste trabalho, as regras operam segundo uma arquitetura orientada a candidatos (*candidate-first*): condições observáveis nos blocos são utilizadas para reduzir o espaço de busca e atribuir papéis preliminares, enquanto a confirmação depende das evidências recuperadas posteriormente.

A Tabela 7 sintetiza as diferenças entre o método de referência e a adaptação utilizada neste trabalho

A adaptação separa, portanto, a identificação em duas etapas. Na primeira, são utilizados atributos de baixo custo computacional para localizar relações estruturais potencialmente relevantes. Na segunda, os candidatos são enriquecidos com informações de execução necessárias à avaliação semântica.

Essa separação é especialmente importante para supressão. Como a *blockchain* confirmada não preserva integralmente transações que permaneceram no mempool, foram substituídas ou não chegaram a ser incluídas, uma configuração estrutural compatível com esse padrão não é suficiente para produzir um rótulo confirmado.

As regras são aplicadas separadamente aos sete recortes temporais. Os eventos resultantes preservam os hashes, os papéis candidatos, o bloco, as posições intrabloco e a regra responsável por sua geração. Candidatos equivalentes são deduplicados antes da etapa de enriquecimento.

Tabela 7 – Adaptação das regras de identificação estrutural.

Padrão	Método de referência	Adaptação deste trabalho
Deslocamento	Comparação de conteúdos de chamadas por sequências de bytes, identificação de contratos intermediários e validação por reexecução das transações em ordens distintas.	Triagem inicial por adjacência no bloco, destinatário comum, remetentes distintos, valor compatível e taxa superior; conteúdo das chamadas e efeitos da execução são verificados posteriormente.
Inserção	Identificação de três eventos de transferência em aplicações descentralizadas, considerando ativos, quantidades, ordenação e taxas das transações.	Identificação inicial de três transações consecutivas, com duas transações externas atribuídas ao mesmo remetente e uma transação intermediária atribuída à vítima; transferências e direção econômica são verificadas após o enriquecimento.
Supressão	Agrupamento de transações com alto consumo de gás, análise de blocos adjacentes e inspeção de traços de execução para reconhecer estratégias de preenchimento de blocos.	Geração de uma evidência estrutural preliminar quando múltiplas transações competitivas precedem uma âncora; nenhuma ocorrência é confirmada sem evidência adicional de <i>mempool</i> ou de atraso e exclusão.

Fonte: Elaborada pelo autor a partir de Eskandari, Moosavi e Clark (5), Torres, Camino e State (10) e da implementação desenvolvida neste trabalho.

4.5.2 Identificação de Candidatos ao Padrão de Deslocamento

No padrão de deslocamento, uma transação concorrente é posicionada antes da operação da vítima com o objetivo de explorar uma oportunidade revelada por ela(5, 10). No método de referência, a identificação utiliza a semelhança entre os conteúdos das chamadas. Como o campo `input` não é mantido nos Parquets locais utilizados na triagem, neste trabalho aplica-se inicialmente uma regra estrutural de menor custo, seguida de enriquecimento dos pares selecionados.

Sejam t_A e t_V duas transações do mesmo bloco, interpretadas, respectivamente, como atacante candidato e vítima candidata. Para uma transação t , sejam $b(t)$ o bloco, $i(t)$ a posição intrabloco, $s(t)$ o remetente, $d(t)$ o destinatário, $v(t)$ o valor nativo transferido e $g(t)$ o preço de gás. O par ordenado (t_A, t_V) é selecionado quando:

$$\begin{aligned}
 b(t_A) &= b(t_V), \\
 i(t_A) + 1 &= i(t_V), \\
 s(t_A) &\neq s(t_V), \\
 d(t_A) &= d(t_V), \\
 g(t_A) &> g(t_V), \\
 |v(t_A) - v(t_V)| &\leq 0,01 \max(|v(t_V)|, 1).
 \end{aligned} \tag{4.14}$$

As condições exigem adjacência no mesmo bloco, remetentes distintos, destinatário comum, preço de gás superior para o atacante candidato e valores nativos compatíveis

dentro da tolerância estabelecida. A adjacência reduz o espaço de busca, enquanto o destinatário comum e a proximidade dos valores selecionam operações potencialmente relacionadas.

O preço de gás superior é utilizado apenas como evidência de prioridade econômica. No período analisado, a posição efetiva no bloco também pode ser influenciada por construtores e fluxos privados; portanto, essa condição não é interpretada como prova de que a diferença de taxa causou a precedência observada.

Os pares equivalentes encontrados a partir de diferentes âncoras são deduplicados. Cada ocorrência recebe papéis preliminares de atacante e vítima e permanece pendente de validação semântica. A confirmação exige posteriormente evidências de compatibilidade entre as chamadas e os efeitos de execução, pois adjacência, destinatário comum e valores semelhantes não demonstram, por si só, a existência de deslocamento.

4.5.3 Identificação de Candidatos ao Padrão de Inserção

O padrão de inserção envolve uma transação anterior e outra posterior à operação da vítima. Uma manifestação conhecida é o *sandwich*, no qual o atacante altera temporariamente o estado em que a transação intermediária é executada e posteriormente encerra sua posição(5, 10).

Na adaptação utilizada, a primeira etapa identifica uma configuração estrutural de três transações consecutivas. Sejam t_F , t_V e t_B , respectivamente, a perna anterior do atacante, a vítima candidata e a perna posterior do atacante(10). Uma tripla (t_F, t_V, t_B) é selecionada quando:

$$\begin{aligned}
 b(t_F) &= b(t_V) = b(t_B), \\
 i(t_F) + 1 &= i(t_V), \\
 i(t_V) + 1 &= i(t_B), \\
 s(t_F) &= s(t_B), \\
 s(t_F) &\neq s(t_V), \\
 d(t_F) &= d(t_B), \\
 v(t_V) &= 0, \\
 |v(t_F) - v(t_B)| &\leq 10^{-12} \max(|v(t_F)|, 1).
 \end{aligned} \tag{4.15}$$

A regra exige que as três operações sejam consecutivas no mesmo bloco e que as transações externas pertençam ao mesmo remetente, distinto da vítima. As duas pernas externas também devem possuir o mesmo destinatário e valores nativos compatíveis segundo a tolerância definida.

Diferentemente do detector de deslocamento, o preço de gás não é utilizado como condição determinante nessa triagem. A posição efetivamente observada no bloco é preser-

vada como evidência estrutural, uma vez que a ordenação contemporânea da Ethereum também pode ser influenciada por construtores de blocos e fluxos privados(11).

Quando as condições são satisfeitas, são atribuídos os papéis preliminares `attacker_front_candidate`, `victim_candidate` e `attacker_back_candidate`. A confirmação posterior exige evidências de execução compatíveis com negociações no mesmo pool.

Para os eventos reconhecidos nos formatos de Uniswap V2 e V3, seja $\delta(t, p)$ a direção da operação produzida pela transação t no pool p . A estrutura é semanticamente compatível com um *sandwich* quando existe um pool comum às três transações tal que:

$$\begin{aligned} p &\in P(t_F) \cap P(t_V) \cap P(t_B), \\ \delta(t_F, p) &= \delta(t_V, p), \\ \delta(t_B, p) &\neq \delta(t_F, p). \end{aligned} \tag{4.16}$$

Assim, a perna anterior e a vítima operam na mesma direção, enquanto a perna posterior realiza a operação inversa no mesmo pool. Essa confirmação caracteriza a estrutura e a direção das negociações compatíveis com um *sandwich*, mas não demonstra, isoladamente, lucro líquido do atacante.

A validação por eventos de Uniswap V2 e V3 também delimita a cobertura do procedimento. Eventos realizados em outros protocolos ou sem logs compatíveis podem permanecer sem confirmação, de modo que a ausência de um evento validado não implica ausência do padrão na população analisada.

4.5.4 Identificação de Candidatos ao Padrão de Supressão

No padrão de supressão, o atacante procura atrasar ou impedir a inclusão da transação de uma vítima, podendo utilizar transações de alta prioridade ou consumo de gás para ocupar capacidade dos blocos(5, 10). Diferentemente de inserção e deslocamento, sua confirmação depende de informações sobre operações que podem não ter sido incluídas na *blockchain*.

Como o conjunto utilizado nesta pesquisa não contém um histórico completo do mempool nem traços de execução para toda a população, a regra implementada possui caráter exploratório e identifica apenas configurações competitivas que justificam investigação posterior.

Seja t_C uma transação utilizada como âncora. Define-se a janela das cinco posições imediatamente anteriores no mesmo bloco por:

$$W_5(t_C) = \{t \mid b(t) = b(t_C), \max(i(t_C) - 5, 0) \leq i(t) < i(t_C)\}. \tag{4.17}$$

Dentro dessa janela, considera-se o subconjunto:

$$Q(t_C) = \left\{ t \in W_5(t_C) \left| \begin{array}{l} s(t) \neq s(t_C), \\ d(t) = d(t_C), \\ v(t) > 0, \\ g(t) > g(t_C) \end{array} \right. \right\}. \quad (4.18)$$

Uma ocorrência estrutural preliminar é registrada quando:

$$|Q(t_C)| \geq 2. \quad (4.19)$$

A regra seleciona, portanto, situações em que ao menos duas transações precedem a âncora, possuem remetentes diferentes do remetente dessa âncora, interagem com o mesmo destinatário, transferem valor nativo e apresentam preço de gás superior. Não se exige que os remetentes dessas transações sejam distintos entre si. Entre as qualificadas, são preservadas as duas posições mais próximas da transação de referência.

Essas condições não demonstram que as operações tenham consumido parcela significativa do limite de gás do bloco, pertençam ao mesmo agente ou tenham provocado atraso ou exclusão de outra transação. Além disso, a própria presença de t_C no bloco demonstra que a transação utilizada como âncora foi eventualmente incluída.

Por essa razão, os candidatos recebem a situação `requires_mempool_evidence`. Eles não originam papéis positivos de atacante ou vítima e não são incorporados às referências semânticas utilizadas nas avaliações estrita e de sensibilidade.

A confirmação desse padrão exigiria informações históricas sobre transações pendentes, substituídas ou não incluídas e, idealmente, sobre sua propagação e permanência no mempool. Assim, a regra de supressão é mantida como mecanismo exploratório de localização de configurações competitivas, preservando a distinção entre indício estrutural e ataque confirmado.

4.5.5 Enriquecimento dos Candidatos por RPC

As regras estruturais utilizam atributos disponíveis nos arquivos Parquet locais, suficientes para examinar posição, participantes, valores e parâmetros de taxa, mas insuficientes para interpretar semanticamente os candidatos. Em particular, o conteúdo integral da chamada, os recibos de execução e os logs não são mantidos na representação utilizada na etapa de triagem.

Para recuperar essas informações sem materializá-las para toda a população, foi implementado um enriquecimento seletivo por meio da interface JSON-RPC da Ethereum(31). Somente as transações pertencentes aos candidatos de inserção e deslocamento compõem a fila de enriquecimento. Os candidatos de supressão não são incluídos, pois informações

recuperadas de blocos confirmados não permitem reconstruir o estado histórico do mempool necessário à confirmação desse padrão.

A entrada do processo é a fila produzida pelo gerador de candidatos. Para cada transação são preservados o hash, o bloco, os detectores associados e os papéis preliminares. Quando um mesmo hash participa de múltiplos eventos, essas informações são agregadas, evitando consultas redundantes.

As consultas são organizadas por bloco. Por meio dos métodos `eth_getBlockByNumber` e `eth_getBlockReceipts`, são recuperados os objetos completos das transações e seus respectivos recibos. Embora a consulta seja realizada no contexto do bloco, somente os registros correspondentes aos candidatos são preservados no conjunto enriquecido.

A Tabela 8 apresenta os principais grupos de informações recuperados.

Tabela 8 – Informações recuperadas no enriquecimento RPC.

Grupo	Informações recuperadas
Identificação e posição	Hash da transação, hash e número do bloco e posição da transação no bloco.
Participantes	Endereços de origem e destino e endereço de contrato criado, quando aplicável.
Valor e execução	Valor nativo, limite de gás, situação do recibo e gás efetivamente consumido.
Taxas	Preço de gás, taxa máxima, prioridade máxima e preço efetivo de gás registrado no recibo.
Chamada	Conteúdo integral do campo <i>input</i> , incluindo seletor de função e parâmetros codificados.
Eventos	Endereço emissor, tópicos e dados dos <i>logs</i> presentes no recibo.
Proveniência	Identificador do provedor, instante da consulta, bloco consultado e hashes solicitados.

Fonte: Elaborada pelo autor.

O conteúdo recuperado é normalizado e armazenado por recorte temporal. O processo verifica a correspondência entre os hashes solicitados, as transações e seus recibos, registrando a cobertura da coleta em manifestos. As evidências obtidas são preservadas em um cache compartilhado entre execuções, permitindo reutilizar registros já coletados e reduzir consultas repetidas. A retomada verifica a compatibilidade do conteúdo armazenado com a fila atual, consultando novamente os blocos quando o cache não atende aos hashes requisitados.

O processamento das janelas pode ser executado concorrentemente por múltiplos endpoints RPC. Um escalonador distribui as janelas pendentes entre os provedores habilitados e aprovados nos testes dos métodos necessários. A alocação considera a quantidade de blocos ainda sem cache e estimativas de velocidade dos provedores, atualizadas a partir das execuções. Quando um trabalhador termina, o endpoint pode receber outra janela

pendente. O paralelismo ocorre entre janelas, sem dividir simultaneamente uma mesma janela entre múltiplos trabalhadores.

A capacidade efetiva depende da disponibilidade dos métodos, da latência, dos limites de requisições e das cotas de cada provedor. Assim, aumentar o número de endpoints pode reduzir o tempo de coleta, mas não assegura ganho proporcional. Os registros em cache permitem retomar o enriquecimento após interrupções sem repetir todas as consultas já concluídas.

A validação principal não depende da disponibilidade de ABIs completas dos contratos. Para a interpretação das chamadas são utilizados os seletores presentes na calldata e, para os eventos, assinaturas conhecidas registradas nos logs. Quando necessário à interpretação dos fluxos, metadados padronizados de tokens ERC-20, como `decimals()`, `symbol()` e `name()`, são recuperados adicionalmente no bloco de referência.

Essa estratégia mantém as consultas externas restritas ao subconjunto necessário à validação semântica. A triagem de larga escala permanece baseada nos dados locais, enquanto informações de maior custo são recuperadas somente para os candidatos selecionados.

4.5.6 Validação Semântica dos Eventos

Após o enriquecimento, os candidatos são submetidos a regras semânticas específicas para cada padrão. Essa etapa verifica se a relação estrutural observada no bloco é acompanhada por evidências compatíveis no conteúdo das chamadas, nos recibos e nos logs.

A validação é realizada no nível do evento, preservando os hashes e os papéis atribuídos durante a geração estrutural. Eventos para os quais alguma transação necessária não esteja disponível recebem a situação `pending_enrichment` e não originam papéis confirmados.

Os candidatos de inserção são avaliados principalmente a partir das operações de troca registradas nos logs. Os candidatos de deslocamento são avaliados pela compatibilidade das chamadas e pelos resultados de execução. Os candidatos de supressão permanecem dependentes de evidência histórica de mempool e, portanto, não são promovidos a eventos confirmados por essa etapa.

4.5.6.1 Validação dos Candidatos de Inserção

Para cada candidato são recuperadas as três transações que compõem o evento: a perna anterior do atacante t_F , a vítima t_V e a perna posterior t_B . As três operações devem ter sido executadas com sucesso, e as pernas externas devem preservar o mesmo remetente identificado durante a geração estrutural.

Os logs são examinados em busca de eventos de troca compatíveis com Uniswap V2 ou Uniswap V3. Para cada pool p , seja

$$\Delta(t, p) = (\Delta_0(t, p), \Delta_1(t, p)) \quad (4.20)$$

o par de variações dos ativos registrado pela transação t . A direção $\delta(t, p)$ é definida quando uma das componentes é positiva e a outra negativa.

Um candidato é confirmado quando existe ao menos um pool comum às três transações para o qual:

$$\begin{aligned} p &\in P(t_F) \cap P(t_V) \cap P(t_B), \\ \delta(t_F, p) &\neq \emptyset, \\ \delta(t_V, p) &= \delta(t_F, p), \\ \delta(t_B, p) &\neq \emptyset, \\ \delta(t_B, p) &\neq \delta(t_F, p). \end{aligned} \quad (4.21)$$

Essa configuração indica que a perna anterior e a vítima realizaram operações na mesma direção, enquanto a perna posterior realizou a operação inversa no mesmo pool. Quando nenhum pool satisfaz essas condições, o candidato não é confirmado pela regra semântica.

A confirmação caracteriza uma estrutura de negociação compatível com *sandwich*, mas não exige, como condição, a demonstração de lucro líquido. A avaliação econômica depende dos ativos movimentados, das taxas de execução e de outros fluxos eventualmente produzidos, sendo tratada posteriormente no dossiê do evento.

4.5.6.2 Validação dos Candidatos de Deslocamento

Para cada candidato de deslocamento são comparadas a transação do atacante t_A e a transação da vítima t_V . Inicialmente, verifica-se se ambas apresentam o mesmo seletor de função, correspondente aos quatro primeiros bytes da calldata. Como essa igualdade não garante parâmetros equivalentes, também é calculada a similaridade entre os conteúdos das chamadas.

Após a remoção do seletor, cada chamada é dividida em palavras de 32 bytes:

$$X_A = (x_{A,1}, x_{A,2}, \dots, x_{A,m}), \quad X_V = (x_{V,1}, x_{V,2}, \dots, x_{V,n}). \quad (4.22)$$

A similaridade é definida por:

$$\text{sim}(t_A, t_V) = \frac{\sum_{j=1}^{\min(m,n)} \mathbb{I}(x_{A,j} = x_{V,j})}{\max(m, n, 1)}. \quad (4.23)$$

A comparação utiliza igualdade exata entre palavras na mesma posição, e o maior comprimento no denominador penaliza diferenças no número de parâmetros codificados.

Um evento é classificado como confirmado quando:

$$\begin{aligned} \text{selector}(t_A) &= \text{selector}(t_V), \\ \text{sim}(t_A, t_V) &\geq 0,80, \\ \text{status}(t_A) &= \text{sucesso}, \\ \text{status}(t_V) &= \text{falha}. \end{aligned} \quad (4.24)$$

Essa combinação representa uma chamada anterior fortemente semelhante à da vítima, executada com sucesso e seguida pela falha da operação concorrente. A classificação **confirmed** corresponde ao atendimento dos critérios operacionais definidos pelo procedimento. Não demonstra, isoladamente, que a execução anterior tenha causado a falha da vítima, pois não é realizada reexecução contrafactual das transações em ordens alternativas.

Quando seletor e similaridade satisfazem os critérios, a transação anterior é executada com sucesso e ambas as transações apresentam evidência de interação com um mesmo pool, mas não é observado efeito adverso suficiente sobre a vítima, o evento é classificado como **probable**. Essa categoria preserva candidatos com elevada compatibilidade semântica sem equipará-los aos casos confirmados.

Os demais candidatos são rejeitados pela regra semântica. Essa rejeição indica que as evidências recuperadas não sustentaram a hipótese estrutural segundo os critérios definidos, e não que as transações sejam necessariamente independentes sob mecanismos não contemplados pelo procedimento.

4.5.6.3 Situações produzidas pela validação

As regras anteriores produzem cinco situações possíveis, sintetizadas na Tabela 9. Eventos **confirmed** satisfazem os critérios estruturais e semânticos definidos; eventos **probable** apresentam compatibilidade relevante, mas evidência insuficiente para confirmação estrita. As demais situações registram rejeição, ausência de enriquecimento ou dependência de evidência histórica de mempool.

Para os eventos confirmados, os papéis das transações são preservados explicitamente. O valor atribuído à vítima indica apenas que ela não desempenha o papel de atacante naquele evento; não representa uma classificação geral de normalidade.

Tabela 9 – Situações da validação semântica.

Situação	Interpretação
<i>confirmed</i>	As evidências estruturais e semânticas exigidas para o padrão foram satisfeitas.
<i>probable</i>	O evento apresenta elevada compatibilidade, mas não possui evidência suficiente para confirmação estrita.
<i>rejected</i>	A correspondência estrutural não foi sustentada pelos dados enriquecidos.
<i>pending enrichment</i>	Uma ou mais transações necessárias não estavam disponíveis no conjunto enriquecido.
<i>requires mempool evidence</i>	O padrão não pode ser confirmado somente com as informações dos blocos confirmados.

Fonte: Elaborada pelo autor.

A validação semântica permanece distinta da adjudicação final. Suas regras são determinísticas e produzem evidências estruturadas, posteriormente reunidas em um dossiê para consolidação e auditoria.

4.5.7 Adjudicação e Verificação das Evidências

Os eventos classificados como confirmados ou prováveis são organizados em um dossiê auditável que reúne, por evento, as evidências estruturais, semânticas e econômicas utilizadas na decisão. São preservados os participantes, posições, resultados de execução, chamadas, pools, fluxos de tokens e justificativas produzidas pelo protocolo.

A análise dos fluxos considera a entidade econômica associada ao atacante, incluindo o remetente e, quando aplicável, o contrato executor envolvido na operação. Os fluxos observáveis de tokens e os custos de gás são utilizados como evidência complementar, sem substituir os critérios estruturais e semânticos. Assim, uma estrutura de inserção pode ser confirmada mesmo quando a rentabilidade líquida não puder ser determinada de forma completa.

A adjudicação é realizada por um protocolo determinístico sobre as evidências organizadas no dossiê. Eventos de inserção semanticamente confirmados permanecem como confirmados; a qualidade da evidência é considerada mais elevada quando há superávit observável em WETH após o custo de gás e intermediária quando a valoração econômica permanece incompleta. Eventos de deslocamento confirmados permanecem confirmados, enquanto aqueles semanticamente classificados como prováveis preservam essa decisão.

A Tabela 10 resume a política empregada.

Como controle de integridade, um procedimento independente recalcula as evidências diretamente a partir do dossiê e das transações enriquecidas, sem utilizar a decisão persistida para determinar o resultado esperado. Para inserção, são novamente verificadas

Tabela 10 – Política de adjudicação dos eventos.

Padrão	Situação semântica	Decisão	Qualidade
Inserção	Confirmado	Confirmado	Alta quando existe superávit observável de WETH após o gás; média nos demais casos.
Deslocamento	Confirmado	Confirmado	Alta.
Deslocamento	Provável	Provável	Média.
Supressão	Requer evidência de <i>mempool</i>	Não adjudicado	Não aplicável.

Fonte: Elaborada pelo autor.

a ordem das transações, a identidade das pernas externas, o sucesso das execuções, o pool comum e as direções das operações. Para deslocamento, são recalculados a ordem, o seletor, a similaridade da calldata, os resultados das execuções e, quando aplicável, a interação com o mesmo pool.

A decisão recalculada é comparada àquela registrada. Inconsistências de cobertura, persistência ou reprodução das evidências impedem a aprovação da respectiva janela para a etapa de consolidação.

Essa verificação constitui um controle da implementação e da integridade dos artefatos, mas não representa uma segunda fonte independente de verdade. Como os critérios metodológicos permanecem os mesmos, o procedimento detecta inconsistências de processamento, mas não elimina as limitações das regras utilizadas.

4.5.8 Consolidação dos Rótulos Semânticos

Após a adjudicação e a verificação das decisões, os eventos das diferentes janelas temporais são consolidados em referências semânticas utilizadas na avaliação dos modelos. Essa etapa converte decisões definidas no nível do evento em papéis associados às transações participantes.

São construídas duas referências complementares, denominadas **strict** e **sensitivity**. Elas não correspondem a modelos ou limiares distintos, mas a diferentes políticas para tratar a incerteza presente nos eventos adjudicados.

A referência **strict** inclui somente eventos de inserção e deslocamento adjudicados como confirmados. Seja e um evento e $D(e)$ sua decisão:

$$Z_e^{(\text{strict})} = \begin{cases} 1, & \text{se } D(e) = \text{confirmado e } \text{tipo}(e) \in \{\text{inserção, deslocamento}\}, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.25)$$

A referência **sensitivity** acrescenta os eventos de deslocamento adjudicados como prováveis:

$$Z_e^{(\text{sensitivity})} = \begin{cases} 1, & \text{se } D(e) \in \{\text{confirmado, provável}\} \text{ e } \text{tipo}(e) \in \{\text{inserção, deslocamento}\}, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.26)$$

A diferença entre as referências decorre, portanto, dos eventos de deslocamento prováveis. Os candidatos de supressão não recebem rótulos positivos em nenhuma das políticas, pois sua confirmação permanece dependente de evidência histórica de mempool.

Para cada evento elegível, as transações são associadas aos respectivos papéis. As pernas anterior e, no caso de inserção, posterior do atacante são tratadas como exemplos positivos. A vítima pertencente ao mesmo evento constitui um negativo contextual. Formalmente, para uma referência

$r \in \{\text{strict, sensitivity}\}$:

$$y_{t,e}^{(r)} = \begin{cases} 1, & \text{se } Z_e^{(r)} = 1 \text{ e } t \text{ possui papel de atacante,} \\ 0, & \text{se } Z_e^{(r)} = 1 \text{ e } t \text{ possui papel de vítima,} \\ \emptyset, & \text{se } e \text{ não é elegível na referência } r. \end{cases} \quad (4.27)$$

O valor zero possui significado estritamente contextual: indica que a transação desempenha o papel de vítima em um evento elegível, e não que represente uma operação normal em sentido geral. Transações sem papel adjudicado permanecem não rotuladas e não são convertidas implicitamente em exemplos negativos.

Tabela 11 – Referências semânticas consolidadas.

Elemento	Definição
Referência <i>strict</i>	Eventos de inserção ou deslocamento adjudicados como confirmados.
Referência <i>sensitivity</i>	Eventos confirmados e deslocamentos adjudicados como prováveis.
Positivo	Transação com papel de atacante anterior ou atacante posterior em evento elegível.
Negativo contextual	Transação com papel de vítima no mesmo evento elegível.
Não rotulado	Transação que não participa de um papel adjudicado na referência considerada.
Conflito	Transação associada simultaneamente aos papéis positivo e negativo; permanece sem rótulo binário consolidado.
Supressão	Excluída das referências por exigir evidência histórica de <i>mempool</i> .

Fonte: Elaborada pelo autor.

A referência **strict** prioriza os eventos sustentados pelos critérios mais restritivos, enquanto **sensitivity** permite avaliar o efeito da inclusão dos deslocamentos prováveis. A comparação entre ambas constitui, portanto, uma análise de sensibilidade à incerteza dos rótulos.

Os artefatos consolidados preservam a relação entre eventos, papéis e transações, bem como a proveniência das decisões utilizadas na consolidação. As referências são construídas independentemente dos escores e das sinalizações dos modelos e não participam do ajuste dos pesos, do pré-processamento ou do cálculo original dos limiares. Na avaliação semântica, os rótulos do recorte de validação são utilizados para selecionar a configuração de atributos e, posteriormente, o limiar entre $Q99$, $Q99,5$ e $Q99,9$. Os rótulos do teste final são utilizados exclusivamente para avaliar as escolhas realizadas na validação.

4.5.9 Protocolo de Avaliação com os Rótulos Semânticos

As referências consolidadas são utilizadas para avaliar se os modelos atribuem escores de anomalia sistematicamente diferentes aos papéis identificados nos eventos. A avaliação é realizada separadamente para Autoencoder e Isolation Forest e para as configurações de atributos definidas no protocolo experimental.

A unidade básica é a transação participante de um evento elegível. Transações com papel **attacker_front** ou **attacker_back** constituem a classe positiva, enquanto aquelas com papel **victim** constituem negativos contextuais. Transações sem papel semântico e ocorrências com conflito de papéis não são convertidas em negativos.

Para um modelo m , recorte temporal t e política de rotulação r , o conjunto avaliável é definido por:

$$D_t^{(m,r)} = \left\{ \left(y_i^{(r)}, s_i^{(m)} \right) \mid y_i^{(r)} \in \{0, 1\}, \text{conflito}_i = 0 \right\}. \quad (4.28)$$

A política **strict** utiliza somente os eventos confirmados, enquanto **sensitivity** incorpora também os deslocamentos classificados como prováveis. Os resultados são apresentados separadamente para as duas políticas, permitindo avaliar sua sensibilidade à incerteza da referência semântica.

A seleção dos modelos é realizada exclusivamente no recorte **validacao_2025_pre_ppectra**. Para cada algoritmo e política de rotulação, seleciona-se a configuração de atributos com maior PR-AUC nessa janela. A PR-AUC é priorizada devido ao desbalanceamento entre as classes e ao interesse na ordenação da classe positiva(22, 23).

Após a seleção da configuração, são comparadas as três políticas de limiar previamente definidas, $Q99$, $Q99,5$ e $Q99,9$. Entre elas, seleciona-se aquela com maior F_1 na mesma janela de validação. Portanto, os rótulos semânticos não são utilizados para

estimar um novo valor de corte contínuo; eles apenas determinam, na validação, a escolha entre políticas de limiar previamente estabelecidas.

A configuração e o limiar selecionados são então aplicados sem alterações ao recorte `teste_final_2025`, que permanece fora de todas as decisões de seleção.

A avaliação dos escores contínuos utiliza ROC-AUC e PR-AUC, sendo esta última operacionalizada pela *Average Precision* (AP). Para as decisões binárias produzidas pelo limiar selecionado, são calculadas *precision*, *recall*, especificidade, valor preditivo negativo, F_1 , acurácia balanceada, coeficiente de correlação de Matthews e taxa de falsos positivos, além da matriz de confusão e da proporção de observações sinalizadas.

Os intervalos de confiança de 95% de ROC-AUC e PR-AUC são estimados por *bootstrap* estratificado por classe, com reamostragem de transações com reposição, preservando separadamente as quantidades de atacantes e vítimas contextuais. São utilizadas 500 replicações, e os limites são obtidos pelos percentis 2,5% e 97,5% das distribuições empíricas resultantes(32). A reamostragem não é agrupada por evento ou bloco, de modo que esses intervalos não incorporam explicitamente a dependência entre transações pertencentes ao mesmo contexto.

Além das métricas globais, é avaliada a concentração dos atacantes nas frações superiores do ranqueamento. Para o top 1%, 5% e 10% dos maiores escores são calculadas *precision*, *recall* e *lift*. Nesse contexto, o *lift* compara a proporção de atacantes na região selecionada com sua prevalência no conjunto semântico avaliável. Valores superiores a 1 indicam enriquecimento da classe positiva no topo do ranqueamento.

Como atacantes e vítimas pertencem aos mesmos eventos, também é realizada uma avaliação pareada. Para um evento e , uma transação atacante a e sua vítima v , define-se a margem:

$$\Delta_{e,a,v}^{(m)} = s_{e,a}^{(m)} - s_{e,v}^{(m)}. \quad (4.29)$$

Uma margem positiva indica que o modelo atribuiu maior escore ao atacante do que à vítima correspondente. A taxa de superioridade pareada é definida por:

$$S_{\text{par}}^{(m)} = \frac{\sum_{(e,a,v)} \mathbb{I}[\Delta_{e,a,v}^{(m)} > 0]}{N_{\text{pares}}}. \quad (4.30)$$

Também é calculada a superioridade no nível do evento, considerada satisfeita somente quando todas as transações atacantes associadas ao evento apresentam escore superior ao da vítima correspondente:

$$S_{\text{evento}}^{(m)} = \frac{\sum_e \mathbb{I}\left[\min_{a \in A_e} (s_{e,a}^{(m)} - s_{e,v}^{(m)}) > 0\right]}{N_{\text{eventos}}}. \quad (4.31)$$

Além dessas taxas, são registradas a média e a mediana das margens entre os escores dos atacantes e das vítimas.

As métricas são calculadas para o conjunto agregado e, de forma descritiva, separadamente para inserção e deslocamento. A seleção de configurações e limiares é realizada somente no conjunto agregado, evitando que subpopulações reduzidas determinem as decisões do protocolo.

A interpretação dessas métricas permanece condicionada à população de eventos recuperada e validada pelo procedimento descrito, conforme discutido na subseção seguinte.

4.5.10 Limitações da Validação Semântica

A validação semântica acrescenta uma camada de interpretação aos escores produzidos pelo Autoencoder e pelo Isolation Forest, mas não constitui uma referência completa sobre a ocorrência de *front-running* na Ethereum. Seus resultados estão condicionados à população recuperada pelas regras estruturais, à natureza dos rótulos construídos e às evidências disponíveis para a adjudicação.

A primeira limitação decorre da geração e da validação dos candidatos. Seja U o universo de transações amostradas, $C \subset U$ o subconjunto recuperado pelas regras estruturais e $E \subseteq C$ o conjunto com papéis semânticos elegíveis e sem conflitos na política considerada. As métricas caracterizam a diferenciação entre atacantes e vítimas contextuais em E , e não o desempenho sobre todo o universo U . Um evento real que não satisfaça os critérios estruturais ou não reúna evidências suficientes para validação não participa dessa avaliação, independentemente do escore atribuído pelos modelos. Consequentemente, o *recall* calculado se refere aos atacantes rotulados em E , não à capacidade de recuperar todos os ataques existentes na amostra. Essas métricas também não permitem estimar a prevalência global de *front-running* na rede.

A segunda limitação refere-se à composição das classes. As vítimas são utilizadas como negativos contextuais por desempenharem, dentro dos eventos elegíveis, papel distinto daquele atribuído aos atacantes. Isso não significa que sejam transações normais em sentido estatístico, benignas em sentido geral ou livres de outros comportamentos anômalos. As métricas medem, portanto, a diferenciação entre papéis dentro da população avaliada, e não uma classificação geral entre transações maliciosas e benignas.

A terceira limitação está associada às evidências utilizadas na validação. Calldata, recibos e logs ampliam a interpretação dos candidatos, mas não reconstroem necessariamente toda a execução ou a intenção econômica dos agentes. Contratos intermediários, proxies, `delegatecall`, chamadas internas, agregadores e interfaces indisponíveis podem limitar a interpretação semântica. Além disso, compatibilidade estrutural e semântica não constitui, isoladamente, prova da intenção econômica que originou uma sequência transacional.

Os rótulos também dependem das regras determinísticas empregadas na adjudicação. Essa estratégia favorece a reprodutibilidade, mas não equivale a um padrão-ouro produzido por múltiplos especialistas independentes. As referências **strict** e **sensitivity** tornam parte dessa incerteza explícita: a primeira restringe a avaliação aos eventos confirmados, enquanto a segunda incorpora deslocamentos prováveis e, conseqüentemente, amplia a cobertura sob critérios menos restritivos.

Deve-se considerar ainda a dependência entre as observações. Atacantes e vítimas pertencentes ao mesmo evento não são estatisticamente independentes, e uma vítima pode ser comparada a mais de uma transação atacante. A avaliação pareada explora diretamente essa estrutura, mas os intervalos de confiança obtidos por *bootstrap* estratificado no nível das transações podem não representar integralmente a dependência intrínseca aos eventos. Por essa razão, esses intervalos são interpretados como medidas descritivas de incerteza.

Por fim, os rótulos semânticos são utilizados somente após o treinamento dos modelos e a produção dos escores. Autoencoder e Isolation Forest permanecem modelos não supervisionados de detecção de anomalias. Entretanto, a escolha da configuração e do limiar na avaliação semântica utiliza os rótulos da validação, constituindo uma seleção orientada por essa referência, sem reajuste dos modelos. O teste final permanece separado dessas decisões. A avaliação investiga, portanto, se escores produzidos sem rótulos no treinamento permitem priorizar transações posteriormente associadas aos papéis de atacante e vítima.

Essas restrições delimitam a interpretação dos resultados: a avaliação permite examinar a utilidade dos escores para priorização dentro da população de candidatos estruturais recuperados, mas não permite concluir que toda anomalia corresponda a um ataque, que todos os ataques existentes tenham sido recuperados ou que as métricas representem a prevalência global de *front-running* na Ethereum. Limitações de validade externa relacionadas à amostragem e à evolução temporal da rede são retomadas no Capítulo 6.

4.6 Considerações Finais

Este capítulo apresentou o protocolo utilizado para avaliar e interpretar os escores de anomalia produzidos pelo Autoencoder e pelo Isolation Forest. A avaliação foi organizada em camadas complementares, preservando a distinção entre anomalia estatística e evidências externas ou semânticas associadas às transações.

Inicialmente, os modelos foram examinados a partir das distribuições dos escores, das taxas de sinalização e de sua estabilidade temporal, seguidas pela análise de concordância entre os métodos. O Label Cloud introduziu uma referência externa independente do treinamento, permitindo investigar a associação entre os escores e endereços historicamente relacionados a atividades de MEV. Em uma etapa mais específica, regras estruturais de

inserção, deslocamento e supressão foram aplicadas independentemente das sinalizações dos modelos, e os candidatos compatíveis foram submetidos a enriquecimento e validação semântica.

A construção das referências **strict** e **sensitivity** permitiu avaliar se os escores não supervisionados priorizam transações associadas aos papéis de atacante em relação às vítimas dos mesmos eventos, sem transformar os modelos em classificadores supervisionados de *front-running*. O capítulo seguinte apresenta os resultados obtidos por essas diferentes perspectivas de avaliação, comparando os algoritmos, as configurações de atributos, o comportamento temporal das sinalizações e sua relação com as evidências externas e semânticas.

5 RESULTADOS E DISCUSSÃO

Este capítulo apresenta e discute os resultados obtidos com o Autoencoder e o Isolation Forest nas configurações de atributos e janelas temporais definidas no protocolo experimental. A análise abrange o comportamento dos escores e das taxas de sinalização, a comparação entre os modelos e sua relação com duas fontes externas de avaliação: os rótulos históricos do Label Cloud e os rótulos semânticos derivados da validação dos candidatos a *front-running*. As decisões de seleção de configurações e limiares são realizadas exclusivamente na validação, preservando o teste final para aplicação das escolhas previamente estabelecidas.

5.1 Visão Geral dos Resultados Experimentais

O experimento utiliza aproximadamente 5% dos blocos, com todas as transações dos blocos selecionados, totalizando 47.545.805 transações em 261.206 blocos. As sete janelas são não sobrepostas e não apresentam duplicação de identificadores. A Tabela 12 resume sua composição.

Tabela 12 – Transações por janela temporal do experimento.

Janela	Transações	Papel experimental
estresse_pre_dencun_2024	4.153.783	Estresse histórico
transicao_dencun_2024	1.232.305	Transição de protocolo
treino_2024_pos_dencun	16.080.647	Treinamento
validacao_2025_pre_pectra	7.248.601	Validação
transicao_pectra_2025	2.108.033	Transição de protocolo
teste_final_2025	14.127.430	Teste final
estresse_fusaka_2025	2.595.006	Estresse futuro
Total	47.545.805	—

Fonte: Elaborada pelo autor.

As cinco configurações de atributos foram avaliadas pelos dois algoritmos. Seus escores brutos não são diretamente comparáveis, pois possuem escalas distintas. A análise conjunta considera posições relativas, decisões por limiares e sobreposição dos conjuntos sinalizados. O treinamento utiliza exclusivamente a janela pós-Dencun de 2024. A validação pré-Pectra calibra os limiares e seleciona configurações e cortes na avaliação semântica, mantendo o teste final separado dessas escolhas.

Os resultados abrangem convergência, estabilidade temporal, concordância, associação com o Label Cloud e ordenação dos papéis semânticos. As janelas de estresse e transição complementam a avaliação temporal, sem serem utilizadas para redefinir as escolhas aplicadas ao teste.

5.2 Resultados do Autoencoder

O Autoencoder foi treinado separadamente nas cinco configurações, utilizando exclusivamente atributos *on-chain*. A análise distingue a convergência do ajuste, a resposta temporal dos escores e o efeito da ablação das taxas monetárias absolutas.

5.2.1 Convergência do Treinamento

Das 16.080.647 transações de treinamento, 14.472.583 foram utilizadas no ajuste dos pesos e 1.608.064 na validação interna determinística. As cinco configurações completaram 200 épocas. A Tabela 13 informa a dimensionalidade, a época da menor perda observada e seu valor.

Tabela 13 – Resumo do treinamento do Autoencoder.

Configuração	Atributos	Época da menor perda	Perda de validação
A completa	13	199	0,02341
B completa	21	200	0,12106
B-ST	18	195	0,06995
C completa	23	195	0,12255
C-ST	19	199	0,08414

Fonte: Elaborada pelo autor.

A Figura 8 apresenta as curvas de treinamento e validação interna.

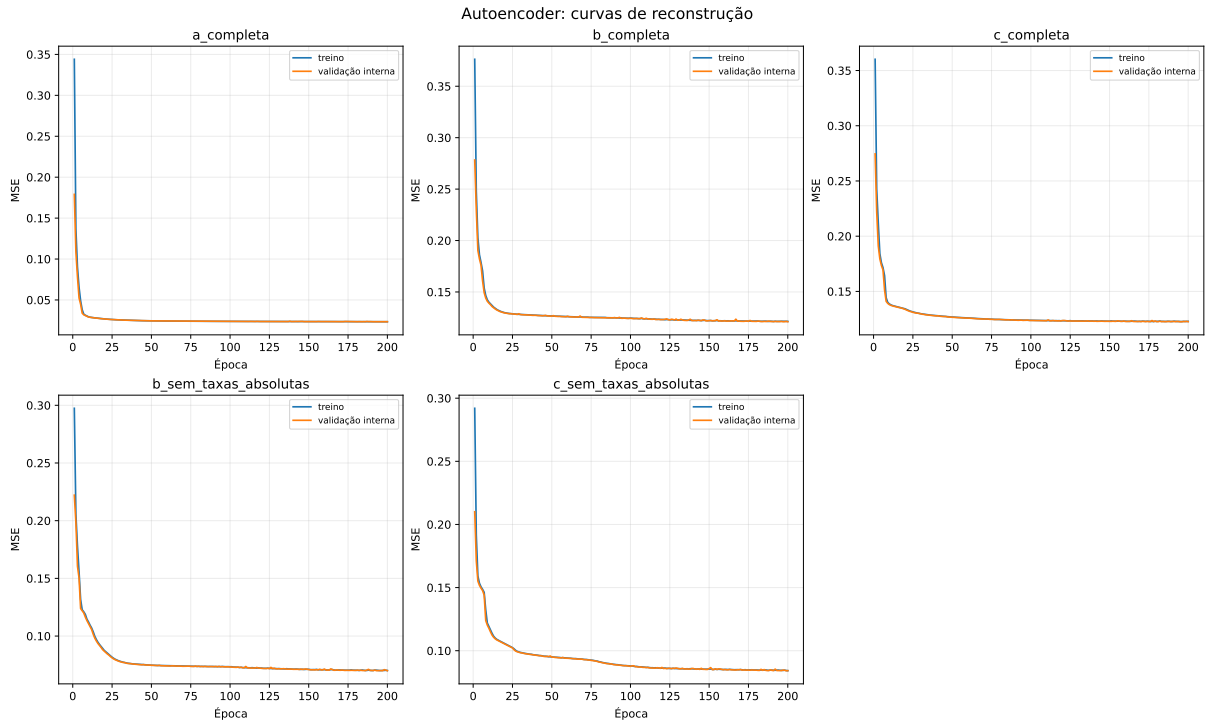


Figura 8 – Perdas de treinamento e validação do Autoencoder.

Fonte: Elaboração própria a partir dos resultados experimentais.

As perdas diminuem rapidamente nas primeiras épocas e os ganhos posteriores se tornam menores. Os mínimos de validação ocorreram entre as épocas 195 e 200. Isso caracteriza a convergência observada, sem demonstrar, isoladamente, ausência de sobreajuste. A época do mínimo registrado não deve ser confundida com o último *checkpoint* aceito pelo critério de melhoria mínima. Os valores de MSE também não ordenam diretamente as configurações, pois dependem da composição e da dimensionalidade das matrizes. O tempo de ajuste em GPU variou de 135,46 a 144,45 segundos, excluindo preparação dos dados e produção dos escores.

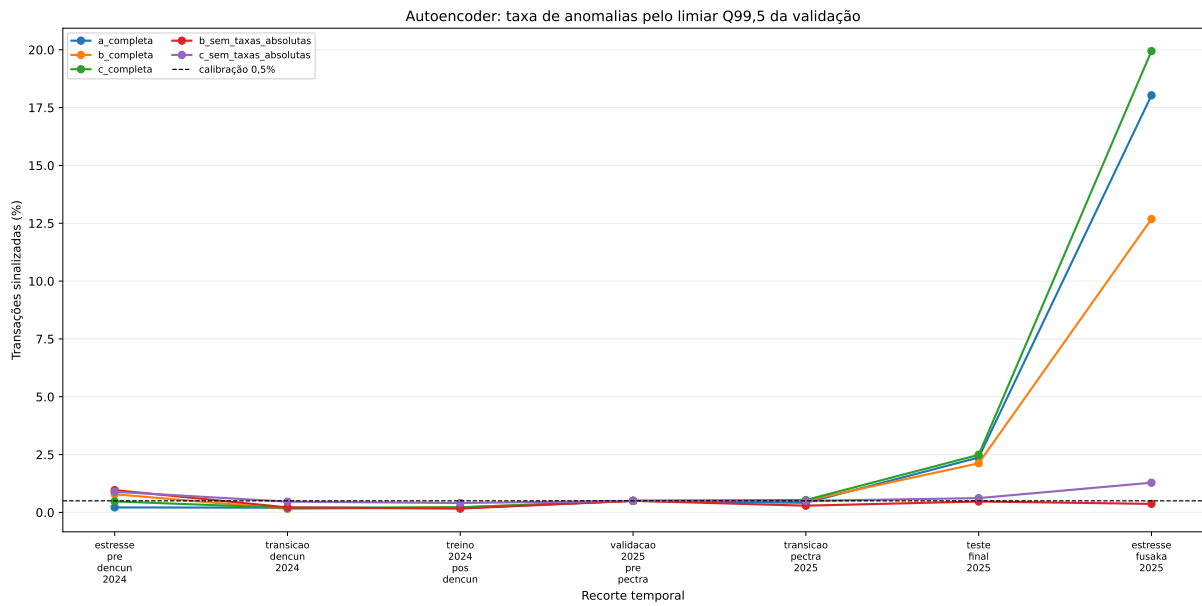
5.2.2 Taxas de Sinalização ao Longo do Tempo

Os limiares são calculados separadamente por configuração na validação pré-Pectra. No corte $Q_{99,5}$, aproximadamente 0,5% das transações dessa janela são sinalizadas por construção. A Tabela 14 apresenta as taxas nas sete janelas. B-ST e C-ST designam as versões sem taxas absolutas. Os valores de corte variaram de 0,28062 em A completa a 0,65753 em B completa, sem constituir medidas diretamente comparáveis de desempenho.

Tabela 14 – Percentual de transações sinalizadas pelo Autoencoder no limiar $Q99,5$.

Config.	Pré-Dencun	Dencun	Treino	Validação	Pectra	Teste	Fusaka
A completa	0,21%	0,20%	0,18%	0,50%	0,42%	2,37%	18,03%
B completa	0,79%	0,15%	0,22%	0,50%	0,51%	2,12%	12,68%
B-ST	0,97%	0,22%	0,16%	0,50%	0,29%	0,47%	0,37%
C completa	0,48%	0,19%	0,22%	0,50%	0,54%	2,49%	19,94%
C-ST	0,89%	0,46%	0,41%	0,50%	0,50%	0,62%	1,28%

Fonte: Elaborada pelo autor.

Figura 9 – Sinalização do Autoencoder por configuração e janela no corte $Q99,5$.

Fonte: Elaboração própria a partir dos resultados experimentais.

Nas configurações completas A, B e C, as taxas de Fusaka atingiram, respectivamente, 18,03%, 12,68% e 19,94%. C completa apresentou a maior expansão. Esses aumentos revelam mudança na resposta a distribuições posteriores, não crescimento equivalente de ataques. O erro de reconstrução pode aumentar em decorrência de transformações gerais do ambiente transacional.

5.2.3 Efeito da Remoção das Taxas Absolutas

A ablação reduziu a expansão prospectiva das sinalizações. No teste final e em Fusaka, B completa sinalizou 2,12% e 12,68%, contra 0,47% e 0,37% de B-ST. C completa alcançou 2,49% e 19,94%, enquanto C-ST registrou 0,62% e 1,28%. A presença das taxas absolutas esteve, portanto, associada a maior sensibilidade às distribuições futuras.

Essa associação é compatível com a variação de grandezas monetárias diante da demanda e das condições da rede, mas não identifica sua causa. A redução também não é uniforme em todos os recortes. B-ST foi mais conservadora nas janelas futuras, enquanto C-

ST apresentou maior expansão em Fusaka. Maior estabilidade não implica necessariamente maior capacidade de priorização, que é examinada pelas avaliações externas e semânticas.

5.3 Resultados do Isolation Forest

O Isolation Forest foi ajustado separadamente para as cinco configurações de atributos. Diferentemente do Autoencoder, seu treinamento não envolve épocas sucessivas ou uma função de reconstrução. O método constrói árvores aleatórias que isolam progressivamente as observações, atribuindo maiores escores às transações separadas por caminhos mais curtos.

5.3.1 Ajuste dos Modelos

Das 16.080.647 transações de treinamento, uma seleção determinística por dispersão de identificadores reteve 501.045 registros, aproximadamente 3,12% dessa janela. O tamanho resulta de uma regra de seleção com alvo aproximado de 500 mil observações, e não de um corte exato. Cada floresta contém 200 árvores, com até 8.192 observações por árvore.

O ajuste variou de 2,97 a 3,40 segundos entre as configurações. Esses tempos abrangem apenas a construção das florestas, não a preparação dos dados nem a produção dos escores para as 47.545.805 transações. Utilizou-se o negativo de `score_samples`, de modo que escores maiores indiquem maior anomalia. A classificação final depende dos quantis calculados na validação, não de `contamination`.

5.3.2 Distribuição Temporal dos Escores

A Figura 10 mostra a evolução dos quantis. Em A completa, a mediana na validação, no teste e em Fusaka foi, respectivamente, 0,4547, 0,4870, 0,5239. Os valores de $Q_{99,5}$ foram 0,6265, 0,6460, 0,6598. Como o corte permanece fixado na validação, o deslocamento da cauda altera a proporção de transações sinalizadas.

As curvas de sobrevivência da Figura 11 mostram a proporção de transações acima de cada nível de escore. Curvas mais afastadas indicam mudanças de distribuição. Nas versões sem taxas absolutas, a maior proximidade entre os períodos é compatível com menor variação prospectiva, sem garantir invariância temporal.

5.3.3 Taxas de Sinalização

Os limiares são calculados separadamente por configuração na validação pré-Pectra. No corte $Q_{99,5}$, aproximadamente 0,5% das transações dessa janela são sinalizadas por construção. A Tabela 15 apresenta as taxas nas sete janelas. B-ST e C-ST designam as versões sem taxas absolutas.

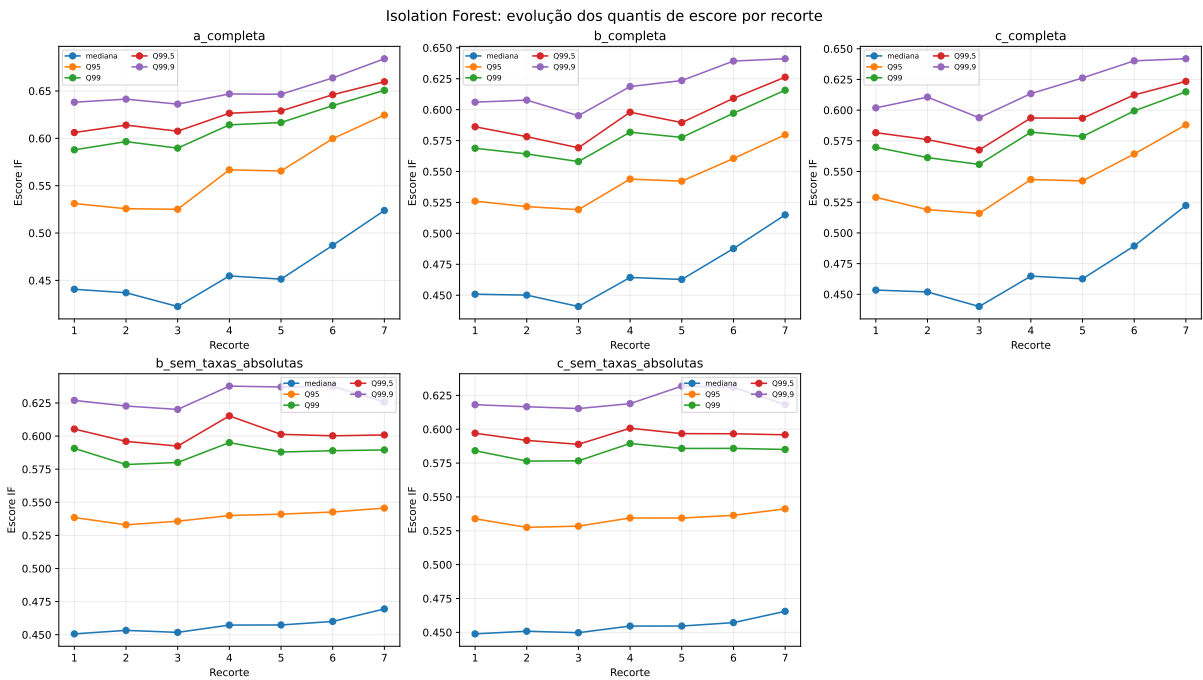


Figura 10 – Quantis dos escores do Isolation Forest. Recortes: 1 – Pré-Dencun; 2 – Transição Dencun; 3 – Treinamento Pós-Dencun; 4 – Validação Pré-Pectra; 5 – Transição Pectra; 6 – Teste final; 7 – Fusaka.

Fonte: Elaboração própria a partir dos resultados experimentais.

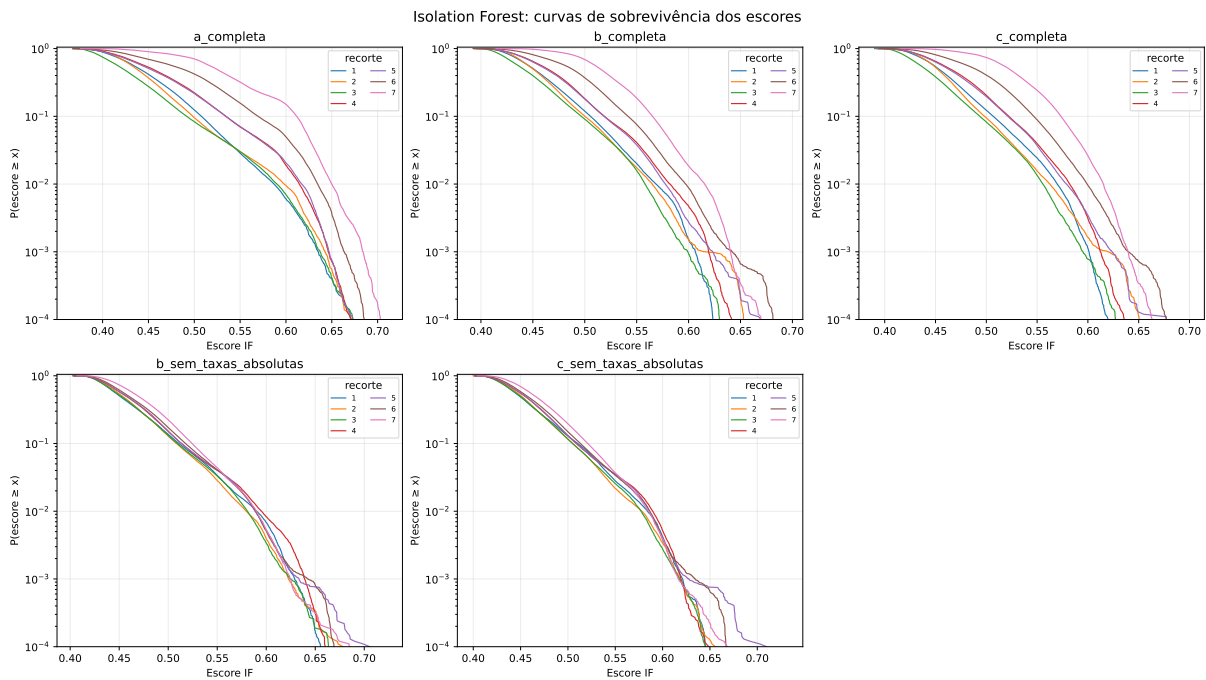


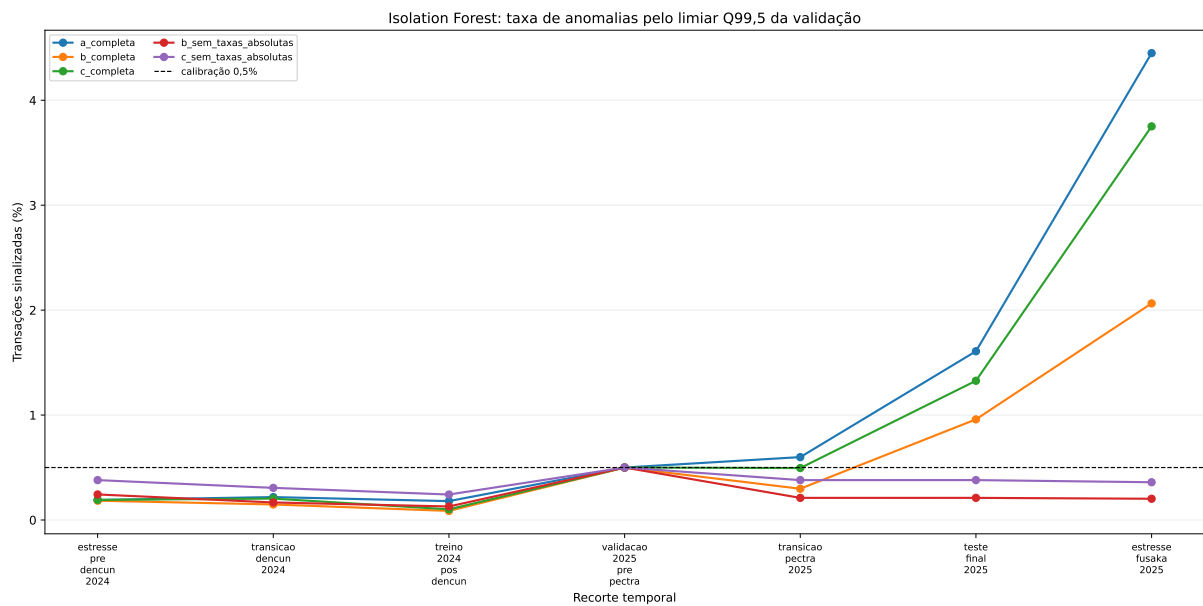
Figura 11 – Curvas de sobrevivência dos escores do Isolation Forest. A numeração dos recortes corresponde à apresentada na Figura 10.

Fonte: Elaboração própria a partir dos resultados experimentais.

Tabela 15 – Percentual de transações sinalizadas pelo Isolation Forest no limiar $Q99,5$.

Config.	Pré-Dencun	Dencun	Treino	Validação	Pectra	Teste	Fusaka
A completa	0,19%	0,22%	0,18%	0,50%	0,60%	1,61%	4,45%
B completa	0,18%	0,15%	0,09%	0,50%	0,30%	0,96%	2,06%
B-ST	0,24%	0,17%	0,13%	0,50%	0,21%	0,21%	0,20%
C completa	0,19%	0,20%	0,10%	0,50%	0,49%	1,33%	3,75%
C-ST	0,38%	0,31%	0,24%	0,50%	0,38%	0,38%	0,36%

Fonte: Elaborada pelo autor.

Figura 12 – Sinalização do Isolation Forest por configuração e janela no corte $Q99,5$.

Fonte: Elaboração própria a partir dos resultados experimentais.

A completa passou de aproximadamente 0,5% na validação para 1,61% no teste e 4,45% em Fusaka. B e C completas apresentaram a mesma tendência geral, com magnitudes diferentes. A resposta temporal também no Isolation Forest indica que a expansão das sinalizações não se restringe ao mecanismo de reconstrução do Autoencoder.

5.3.4 Efeito da Remoção das Taxas Absolutas

No teste e em Fusaka, B-ST sinalizou 0,21% e 0,20%, enquanto C-ST registrou 0,38% e 0,36%. As respectivas versões completas apresentaram taxas maiores, mostrando menor expansão prospectiva após a ablação. Isso não significa redução universal das sinalizações em todos os períodos nem superioridade discriminativa. A estabilidade agregada e a concentração de evidências relevantes são critérios distintos.

5.4 Comparação e Concordância entre Autoencoder e Isolation Forest

A comparação considera os conjuntos sinalizados por $Q99,5$, calibrado separadamente por método e configuração. A união e a interseção são medidas descritivas de concordância, não requisitos para gerar os candidatos estruturais. Estes são produzidos a partir das relações entre transações dos blocos selecionados, independentemente de ultrapassarem os limiares dos modelos.

5.4.1 Comparação das Taxas de Sinalização

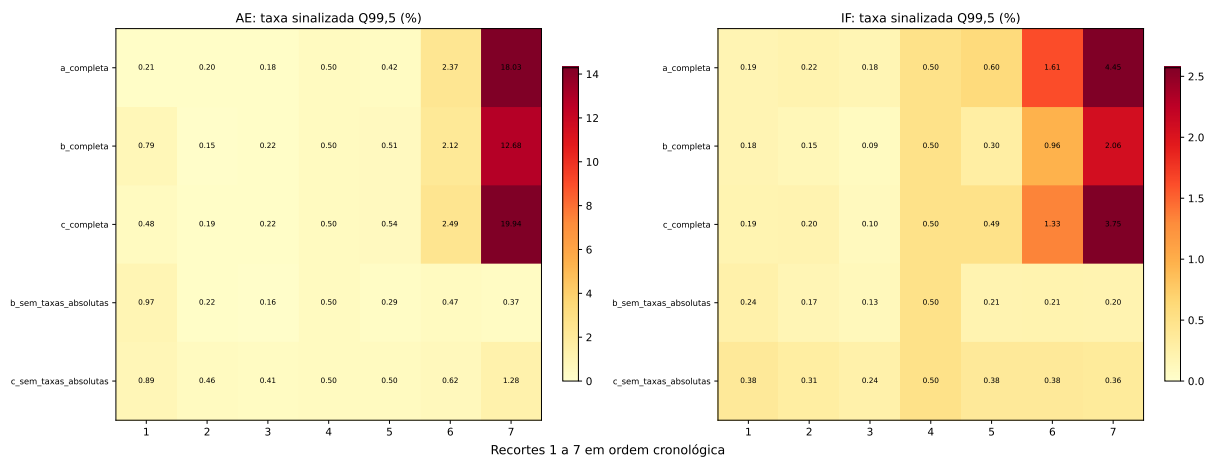


Figura 13 – Sinalização do Autoencoder e Isolation Forest no limiar $Q99,5$.

Fonte: Elaboração própria a partir dos resultados experimentais.

Os dois métodos apresentaram maior expansão prospectiva nas configurações completas, mas com magnitudes diferentes. Em B completa, o Autoencoder sinalizou 12,68% de Fusaka, contra 2,06% do Isolation Forest. As escalas de cores dos dois painéis são distintas, devendo-se comparar os valores numéricos das células. Tendências agregadas semelhantes não implicam identificação das mesmas transações, como mostra a análise de Jaccard.

5.4.2 Concordância entre os Modelos

Tabela 16 – Jaccard entre Autoencoder e Isolation Forest no limiar $Q99,5$.

Config.	Pré-Dencun	Dencun	Treino	Validação	Pectra	Teste	Fusaka
A completa	0,026	0,027	0,043	0,110	0,147	0,113	0,086
B completa	0,162	0,060	0,103	0,183	0,214	0,070	0,039
B-ST	0,187	0,148	0,130	0,437	0,340	0,173	0,207
C completa	0,149	0,091	0,071	0,079	0,132	0,161	0,120
C-ST	0,040	0,027	0,036	0,026	0,039	0,042	0,048

Fonte: Elaborada pelo autor.

Na validação, Jaccard variou de 0,026 em C-ST a 0,437 em B-ST. No teste, variou de 0,042 a 0,173, com máximo em B-ST. A maior parte das transações sinalizadas não foi compartilhada entre os algoritmos.

Em B completa na janela Fusaka, o Autoencoder sinalizou 329.109 transações e o Isolation Forest, 53.568. A interseção reuniu 14.209, resultando em Jaccard de 0,039. A expansão das sinalizações de um método não corresponde necessariamente à expansão do conjunto identificado pelo outro.

5.4.3 Concordância entre Configurações de Atributos

No Autoencoder, a concordância entre configurações na validação variou de 0,014 a 0,312. O mínimo ocorreu entre A completa e B-ST, com 973 transações compartilhadas. O máximo ocorreu entre B completa e B-ST. Essas diferenças mostram que a representação modifica o conjunto priorizado mesmo com a mesma taxa nominal de sinalização.

No Isolation Forest, a concordância entre configurações na validação variou de 0,038 a 0,553. O mínimo ocorreu entre A completa e B-ST, com 2.657 transações compartilhadas. O máximo ocorreu entre B completa e B-ST. Essas diferenças mostram que a representação modifica o conjunto priorizado mesmo com a mesma taxa nominal de sinalização.

5.4.4 Diferenças entre os Mecanismos de Detecção

O Autoencoder prioriza dificuldade de reconstrução e o Isolation Forest, facilidade de isolamento. A concordância reduzida indica que os métodos não são intercambiáveis, mas não comprova falha ou complementaridade benéfica por si só. Da mesma forma, uma interseção elevada não confirma ataques. A distinção entre sinalizações compartilhadas e exclusivas permite descrever suas respostas sem condicionar a geração estrutural de candidatos à união dos modelos.

5.5 Resultados da Avaliação Externa com o Label Cloud

A avaliação externa verifica se transações originadas por endereços historicamente associados a `mev_bot` recebem escores superiores às demais. Esses rótulos não participaram do treinamento ou da calibração dos limiares. No teste final, 250.616 das 14.127.430 transações tiveram correspondência, com prevalência operacional de 1,7740%. Na validação, foram 63.315 correspondências, aproximadamente 0,8735%.

A associação é estabelecida no nível do endereço. Sua presença não comprova MEV ou *front-running* em cada transação, nem sua ausência demonstra normalidade. Os negativos constituem uma hipótese operacional desta avaliação auxiliar.

5.5.1 Capacidade de Ordenação dos Escores

A Tabela 17 distingue métricas contínuas, ROC-AUC e PR-AUC, das métricas calculadas no corte $Q99,5$. A PR-AUC é operacionalizada como *average precision*.

Tabela 17 – Avaliação com o Label Cloud no teste final.

Método	Configuração	ROC-AUC	PR-AUC	<i>Precision</i>	<i>Recall</i>	F_1	MCC
AE	A completa	0,6796	0,0247	0,0000	0,0000	0,0000	-0,0209
IF	A completa	0,9054	0,1237	0,1039	0,0942	0,0988	0,0835
AE	B completa	0,7264	0,0385	0,0100	0,0120	0,0109	-0,0086
IF	B completa	0,9538	0,2856	0,4456	0,2409	0,3127	0,3190
AE	B-ST	0,7960	0,1817	0,3752	0,0998	0,1577	0,1865
IF	B-ST	0,8308	0,0587	0,0000	0,0000	0,0000	-0,0062
AE	C completa	0,7095	0,0493	0,0951	0,1335	0,1111	0,0937
IF	C completa	0,9449	0,2902	0,4047	0,3028	0,3464	0,3400
AE	C-ST	0,8526	0,0775	0,0541	0,0188	0,0279	0,0217
IF	C-ST	0,8175	0,0536	0,0002	0,0000	0,0001	-0,0082

Fonte: Elaborada pelo autor.

IF B completa apresentou a maior ROC-AUC, 0,9538. IF C completa obteve a maior PR-AUC, 0,2902, seguido de B completa, 0,2856. No Autoencoder, C-ST alcançou a maior ROC-AUC, 0,8526, mas B-ST apresentou maior PR-AUC, 0,1817. Assim, a configuração de melhor ordenação depende da métrica considerada.

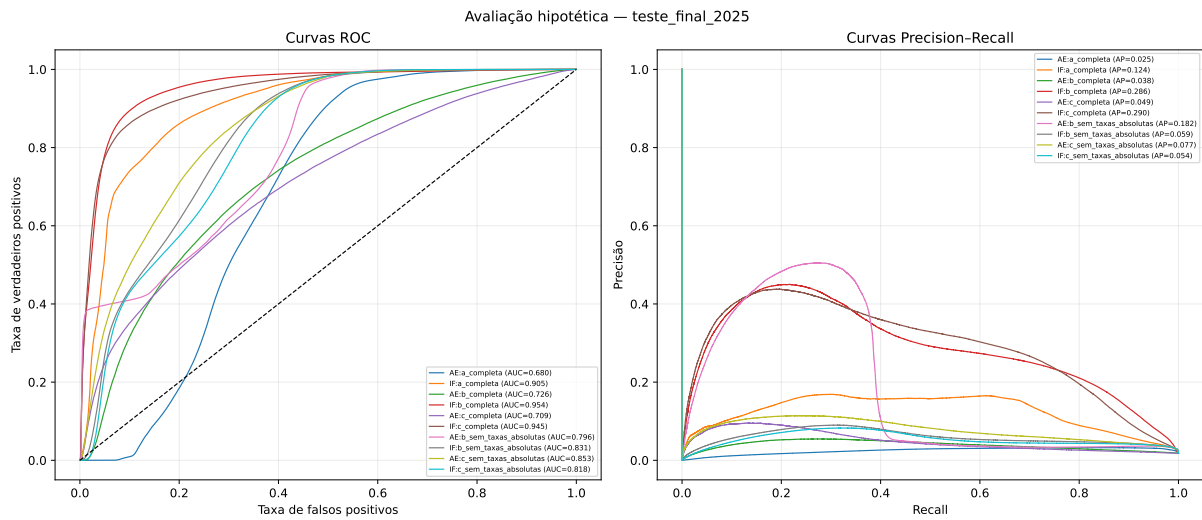


Figura 14 – Curvas ROC e *Precision-Recall* da avaliação com o Label Cloud no teste final.

Fonte: Elaboração própria a partir dos resultados experimentais.

A remoção das taxas absolutas favoreceu essas métricas do Autoencoder nas famílias B e C, enquanto no Isolation Forest suas versões completas superaram as respectivas ablações. A configuração A do Autoencoder apresentou ROC-AUC de 0,6796, mas PR-AUC de apenas 0,0247, evidenciando a importância de considerar a baixa prevalência e a concentração efetiva de positivos.

5.5.2 Resultados nos Limiares Predefinidos

No corte $Q99,5$, IF C completa apresentou o maior F_1 , recuperando 75.874 das 250.616 correspondências, com 111.588 falsos positivos operacionais. A *precision* foi 0,4047, a *recall*, 0,3028, a especificidade, 0,9920, e o MCC, 0,3400.

Nessa configuração, $Q99$ produziu *precision* de 0,3454, *recall* de 0,4416 e F_1 de 0,3876. Em $Q99,9$, a *precision* foi 0,4033 e a *recall* caiu para 0,1073. A restrição do corte reduziu a cobertura sem assegurar aumento monotônico da *precision*. Esses cortes foram definidos sem otimização pelos rótulos do Label Cloud.

Os resultados das ablações do Isolation Forest também mostram que uma ordenação global informativa pode coexistir com baixa recuperação no corte extremo. Por isso, as métricas contínuas não substituem a avaliação do ponto de operação.

5.5.3 Enriquecimento no Topo dos Escores

A Figura 15 apresenta o *lift*, razão entre a proporção de positivos na faixa inspecionada e a prevalência no conjunto completo. Valores superiores a 1 indicam concentração acima da linha de base, não confirmação de ataques.

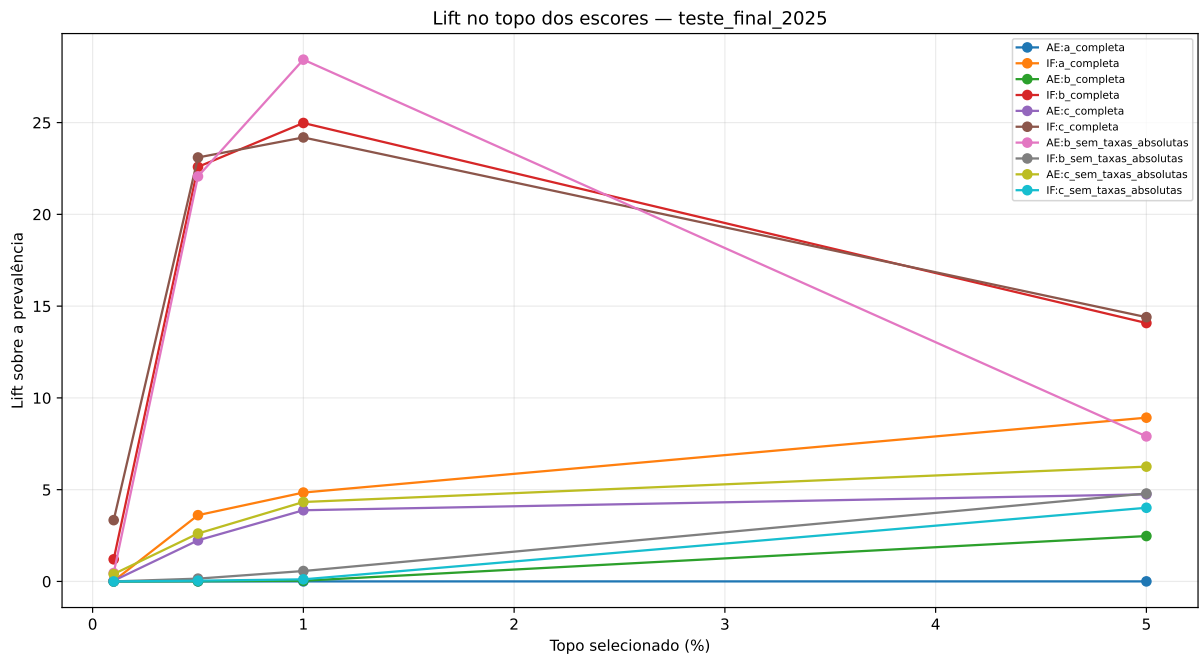


Figura 15 – *Lift* nas frações superiores dos escores no teste final.

Fonte: Elaboração própria a partir dos resultados experimentais.

No 1% superior, AE B-ST apresentou o maior *lift*, 28,43. Entre 141.275 transações, 71.248 tinham correspondência com o Label Cloud, com *precision* de 0,5043 e *recall* de 0,2843. IF B e C completas alcançaram *lift* de 24,98 e 24,19 nessa faixa.

Ao ampliar a inspeção para 5%, IF C completa recuperou 71,99% das correspondências, com *lift* de 14,40. A maior cobertura veio acompanhada de menor concentração. Os resultados não são monotônicos em todas as faixas: positivos associados à referência externa não ocupam necessariamente a cauda mais extrema dos escores.

5.5.4 Interpretação da Evidência Externa

IF C completa destacou-se na PR-AUC, no F_1 de $Q99,5$ e na cobertura do topo de 5%. AE B-ST concentrou mais correspondências no topo de 1%, enquanto C-ST apresentou maior ROC-AUC entre os Autoencoders. Essas diferenças reforçam que não há uma configuração universalmente superior.

O Label Cloud fornece evidência auxiliar de associação com agentes catalogados. Como sua cobertura é parcial e a rotulação não descreve o papel de cada transação, essas métricas não estimam desempenho definitivo de detecção de *front-running*. A avaliação seguinte aproxima a análise das relações entre atacantes e vítimas nos eventos adjudicados.

5.6 Resultados da Validação Estrutural e Semântica de Front-Running

A avaliação verifica se os escores priorizam atacantes em relação às vítimas contextuais dos eventos adjudicados. As vítimas não representam a população geral de transações normais. A geração estrutural não depende das sinalizações dos modelos, mantendo esta referência independente do treinamento e dos cortes de anomalia.

Nas sete janelas, foram adjudicados 20.611 eventos, dos quais 9.091 confirmados e incluídos na política estrita. A consolidação reuniu 47.628 transações semânticas únicas por janela e hash, sendo 26.826 avaliáveis na política estrita e 45.390 na de sensibilidade. As contagens de transações diferem das de eventos e pares, pois uma mesma transação pode participar de múltiplas relações. Eventos prováveis são incluídos apenas na sensibilidade, e conflitos de papel são excluídos da população avaliável.

5.6.1 Cobertura dos Rótulos Semânticos

Tabela 18 – Cobertura dos rótulos semânticos na validação e no teste final.

Janela	Política	Transações semânticas	Avaliáveis	Atacantes	Vítimas	Cobertura
Validação	Estrita	6.578	5.170	3.425	1.745	78,60%
Validação	Sensibilidade	6.578	6.424	4.052	2.372	97,66%
Teste final	Estrita	8.838	7.118	4.709	2.409	80,54%
Teste final	Sensibilidade	8.838	8.700	5.500	3.200	98,44%

Fonte: Elaborada pelo autor.

No teste final, a prevalência de atacantes foi de 66,16% na política estrita e 63,22% na sensibilidade. A elevada proporção decorre da composição relacional, com eventos

contendo mais de um atacante por vítima. A cobertura da tabela refere-se às transações semânticas da própria janela, não à totalidade da rede.

As PR-AUC devem ser interpretadas em relação a essas linhas de base, não diretamente comparadas com o Label Cloud, cuja prevalência operacional no teste foi de aproximadamente 1,77%. As duas avaliações possuem populações e unidades conceituais de rotulação distintas.

5.6.2 Seleção das Configurações na Validação

A maior PR-AUC na validação selecionou AE C-ST e IF B completa em ambas as políticas. Na estrita, as PR-AUC de validação foram 0,7524 e 0,8737. Na sensibilidade, foram 0,7099 e 0,8315. O teste final recebeu essas configurações sem nova seleção.

Tabela 19 – Configurações selecionadas e resultados no teste final.

Política	Método	Configuração	PR-AUC val.	ROC-AUC teste	PR-AUC teste	Sup. pareada
Estrita	AE	C-ST	0,7524	0,6779	0,7542	0,7126
Estrita	IF	B completa	0,8737	0,7503	0,8453	0,7850
Sensibilidade	AE	C-ST	0,7099	0,6487	0,7178	0,6700
Sensibilidade	IF	B completa	0,8315	0,7182	0,8147	0,7261

Fonte: Elaborada pelo autor.

Os cortes foram escolhidos separadamente, pelo maior F_1 na população semântica da validação entre os três quantis previamente calibrados. Essa seleção não utiliza o teste e não transforma o ajuste dos modelos em treinamento supervisionado.

5.6.3 Discriminação no Teste Final

Na política estrita, AE C-ST alcançou ROC-AUC de 0,6779, com intervalo de confiança de 95% entre 0,6625 e 0,6913, e PR-AUC de 0,7542, com intervalo entre 0,7421 e 0,7657. IF B completa apresentou ROC-AUC de 0,7503, com intervalo entre 0,7367 e 0,7609, e PR-AUC de 0,8453, com intervalo entre 0,8338 e 0,8557.

Entre os modelos selecionados na validação, IF B completa apresentou valores superiores aos de AE C-ST nessas métricas. A figura 16 apresenta todas as configurações para comparação descritiva, mas eventuais valores superiores de outras configurações no teste não redefinem a seleção. Os intervalos foram obtidos por reamostragem estratificada de transações, não por um teste pareado da diferença entre algoritmos. Sua separação não substitui uma análise que considere a dependência entre transações dos mesmos eventos.

Na sensibilidade, AE C-ST apresentou ROC-AUC de 0,6487 e PR-AUC de 0,7178. IF B completa alcançou 0,7182 e 0,8147. A inclusão de prováveis altera a população e reduz a discriminação observada, sem demonstrar que todos esses eventos sejam incorretos.

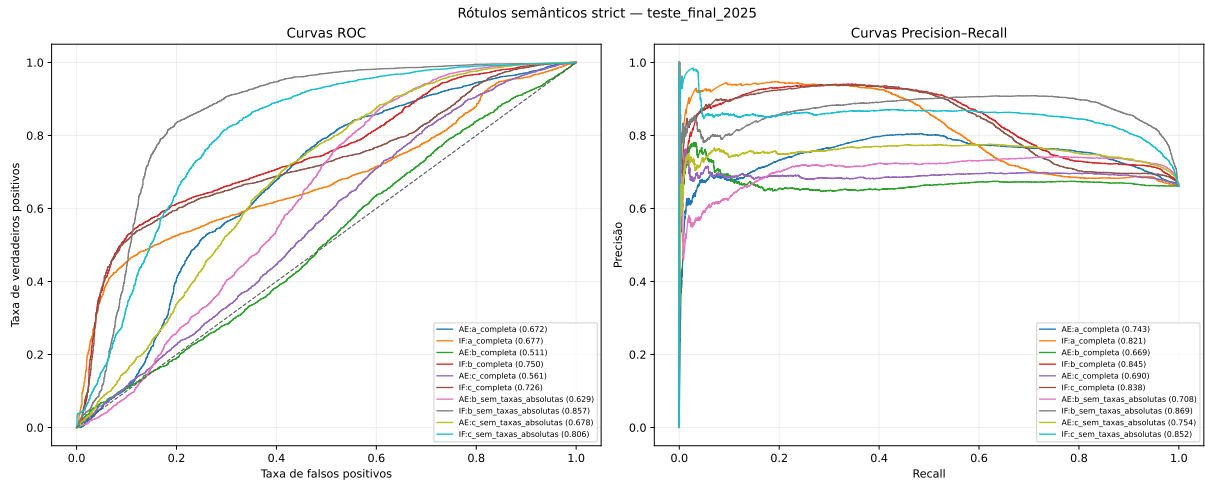


Figura 16 – Curvas ROC e *Precision-Recall* da política estrita no teste final.

Fonte: Elaboração própria a partir dos resultados experimentais.

Na curva ROC, maior aproximação do canto superior esquerdo indica melhor separação. Na curva *Precision-Recall*, a referência é a prevalência estrita de 0,6616. As PR-AUC de 0,7542 e 0,8453 mostram ganho sobre essa referência dentro da população adjudicada, não desempenho sobre toda a Ethereum.

5.6.4 Desempenho dos Limiares de Anomalia

$Q99,5$ permanece o corte de referência das análises de sinalização e concordância. Para a decisão semântica final, contudo, $Q99$ apresentou o maior F_1 de validação em ambos os modelos e políticas. Portanto, os resultados selecionados não devem ser apresentados como se utilizassem obrigatoriamente $Q99,5$.

No teste estrito, AE C-ST sinalizou 128 atacantes e 48 vítimas com $Q99$, apresentando *precision* de 0,7273, *recall* de 0,0272 e F_1 de 0,0524. IF B completa sinalizou 444 atacantes e 52 vítimas, com *precision* de 0,8952, *recall* de 0,0943 e F_1 de 0,1706. No corte de referência $Q99,5$, as *recall* foram menores, 0,0104 e 0,0157, respectivamente.

Na sensibilidade, $Q99$ resultou em *recall* de 0,0251 no AE e 0,0807 no IF. A baixa recuperação mostra que ordenar atacantes acima de vítimas não implica colocá-los entre os extremos da população global. Os cortes favorecem uma inspeção restrita, mas não recuperam a maior parte dos atacantes adjudicados.

5.6.5 Concentração de Atacantes no Topo

No 1% superior da população estrita, composto por 72 transações, AE C-ST recuperou 52 atacantes, com *precision* de 0,7222 e *lift* de 1,0917. IF B completa recuperou 54, com *precision* de 0,7500 e *lift* de 1,1337.

Nos 10% superiores, entre 712 transações, o AE recuperou 530 atacantes, com *precision* de 0,7444 e *recall* de 0,1126. O IF recuperou 652, com *precision* de 0,9157 e *recall* de 0,1385. Nessa população, o IF concentrou mais atacantes nas duas faixas, embora ampliar o topo não assegure redução monotônica da *precision*.

O *lift* é mais modesto que na avaliação com o Label Cloud devido à prevalência estrita de 0,6616. A interpretação deve considerar a linha de base e a quantidade de transações inspecionadas, sem confundir topo percentual com limiar global congelado.

5.6.6 Comparação Pareada entre Atacantes e Vítimas

A avaliação pareada compara cada atacante com a vítima do mesmo evento. No teste estrito, AE C-ST atribuiu escore maior ao atacante em 3.357 dos 4.711 pares, ou 71,26%. IF B completa atingiu 3.698 pares, ou 78,50%. A contagem de pares supera os 4.709 atacantes únicos porque alguns participam de mais de um evento.

Em 2.409 eventos, todos os atacantes superaram a vítima em 65,46% dos casos no AE e 59,65% no IF. As medianas das diferenças de escore foram positivas, 0,0346 e 0,0457, mas suas magnitudes não são comparáveis entre algoritmos com escalas distintas.



Figura 17 – Superioridade dos escores dos atacantes sobre as vítimas na política estrita.

Fonte: Elaboração própria a partir dos resultados experimentais.

Na Figura 17, cada célula informa a fração de pares em que o atacante recebeu escore maior. Valores superiores a 0,5 favorecem essa ordenação, sem constituir probabilidade de ataque. A sensibilidade reduziu a superioridade pareada para 0,6700 no AE e 0,7261 no IF.

O IF apresentou maior superioridade por par, mas o AE maior superioridade completa por evento. Não há contradição: um evento com múltiplos atacantes só satisfaz a segunda condição quando todos superam a vítima. A agregação por par e a condição conjunta por evento respondem a perguntas diferentes.

5.6.7 Resultados por Tipo de Ataque

Inserção dominou o teste estrito: 6.904 das 7.118 transações avaliáveis pertenciam a essa categoria, com 4.602 atacantes e 2.302 vítimas em 2.302 eventos. Os pares de inserção totalizaram 4.604, devido à reutilização de atacantes em alguns eventos.

Nesse subconjunto, AE C-ST alcançou ROC-AUC de 0,6981, PR-AUC de 0,7731 e superioridade pareada de 72,76%. IF B completa apresentou 0,7835, 0,8951 e 80,30%, respectivamente. Todos os atacantes superaram a vítima em 68,20% dos eventos no AE e 62,38% no IF.

Deslocamento reuniu 107 eventos, com um atacante e uma vítima por evento. AE e IF apresentaram ROC-AUC de 0,1915 e 0,0220. O atacante superou a vítima em apenas 7 pares no AE e 1 no IF, correspondendo a 6,54% e 0,93%. As medianas das diferenças foram negativas, -0,0559 e -0,0882. A ordenação foi, portanto, predominantemente inversa à esperada, não uniformemente nula.

Não houve eventos de supressão confirmados com papéis suficientes para esta avaliação. A evidência positiva agregada sustenta principalmente a priorização nos eventos de inserção, sem generalização uniforme para os demais padrões.

5.6.8 Síntese da Avaliação Semântica

AE C-ST e IF B completa, selecionados na validação, mantiveram discriminação no teste. O IF apresentou maior ROC-AUC, PR-AUC e superioridade por par. O AE apresentou maior proporção de eventos em que todos os atacantes superaram a vítima. Os quantis globais recuperaram poucos atacantes, mesmo quando a ordenação relativa foi informativa.

A evidência positiva concentrou-se em inserção. Deslocamento apresentou ordenação predominantemente inversa e supressão não pôde ser avaliada. Essas conclusões são condicionadas às regras, à adjudicação automatizada e à população de atacantes e vítimas contextuais, não constituindo estimativas de desempenho sobre a rede completa.

5.7 Discussão Integrada dos Resultados

A interpretação distingue anomalia estatística, associação com conhecimento externo e compatibilidade com padrões transacionais. Um escore elevado não comprova atividade maliciosa. A Tabela 20 resume as evidências e seus limites.

Tabela 20 – Síntese das evidências do protocolo experimental.

Dimensão	Resultado principal	Interpretação
Estabilidade temporal	Maior sinalização futura nas configurações completas.	Cortes fixos respondem a mudanças de distribuição, sem medir aumento de ataques.
Ablação	B-ST e C-ST reduziram a expansão prospectiva.	Estabilidade não equivale a discriminação, e a remoção pode eliminar informação útil.
Concordância AE–IF	Sobreposição reduzida em várias janelas.	Os modelos priorizam conjuntos distintos, sem comprovar correção por concordância.
Label Cloud	IF C completa liderou PR-AUC, e AE B-ST o <i>lift</i> no topo de 1%.	Associação com agentes catalogados, não confirmação transacional de ataques.
Rótulos semânticos	Entre os modelos selecionados, IF B completa apresentou maior discriminação e superioridade por par, enquanto AE C-ST apresentou maior superioridade completa por evento.	Evidência condicionada aos papéis adjudicados, principalmente de inserção.
Limiares globais	<i>Q99</i> foi selecionado na validação, mas apresentou baixa <i>recall</i> .	Ordenação relativa e raridade individual não são equivalentes.
Tipos de ataque	Inserção apresentou evidência favorável. Deslocamento, ordenação inversa. Supressão não foi avaliável.	As métricas agregadas não representam desempenho uniforme entre padrões.

Fonte: Elaborada pelo autor.

5.7.1 Resposta à Questão de Pesquisa

Os resultados sustentam que atributos exclusivamente *on-chain* permitem produzir sinais de desvio e priorizar investigações, dentro do escopo experimental. Nenhum rótulo externo ou semântico foi incorporado ao treinamento. A resposta não equivale à identificação automática e definitiva de ataques: o significado de uma anomalia depende da representação e do contexto (14).

As associações externas e a ordenação dos papéis semânticos evidenciam utilidade dos escores. Entretanto, nem toda observação extrema corresponde a um ataque e muitos atacantes adjudicados não ultrapassam os cortes globais. A interpretação requer relações entre transações e evidências de execução.

5.7.2 Estabilidade Temporal e Escolha dos Atributos

A maior sinalização prospectiva nas configurações completas foi observada nos dois métodos, especialmente em Fusaka. A análise diagnóstica dos 23 atributos, tomando o treinamento como referência, identificou mudança elevada em três atributos na validação, cinco no teste e sete em Fusaka, conforme os critérios descritivos de PSI e KS aproximado. Entre os atributos com maiores mudanças destacaram-se `effective_gas_price_log`, `max_fee_per_gas_log` e `transaction_fee_paid_log`. Esses resultados contextualizam a expansão das sinalizações, sem atribuir causalidade às atualizações do protocolo, pois uso da rede, aplicações e condições econômicas evoluem simultaneamente.

O monitoramento separado das transações do tipo 4 registrou participações de aproximadamente 0,17% na transição Pectra, 0,48% no teste e 0,61% em Fusaka. Essa categoria, ausente no treinamento e sem indicador específico nas matrizes, caracteriza uma mudança na composição transacional. Sua presença, entretanto, não demonstra que ela seja responsável pelas alterações dos escores.

A ablação reduziu a expansão futura, mas teve efeitos distintos na discriminação: favoreceu as métricas do AE nas famílias B e C do Label Cloud, enquanto reduziu as do IF. Na referência semântica, foram selecionados AE C-ST e IF B completa. A escolha dos atributos deve, portanto, considerar conjuntamente estabilidade, algoritmo e finalidade, sem pressupor uma representação universalmente superior.

5.7.3 Diferenças entre Autoencoder e Isolation Forest

Reconstrução e isolamento produziram conjuntos parcialmente distintos de transações. No Label Cloud, IF C completa liderou a PR-AUC e AE B-ST o *lift* no topo de 1%. Na avaliação semântica selecionada, IF B completa apresentou maior discriminação e superioridade por par, mas AE C-ST maior superioridade completa por evento.

Essas diferenças justificam preservar as evidências separadamente, não afirmar que um método substitua integralmente o outro. União e interseção descrevem concordância, mas não comprovam correção e não condicionam a geração dos candidatos estruturais.

5.7.4 Convergência entre as Avaliações Externas

As duas referências favoreceram configurações sem taxas no AE e completas no IF, mas não selecionaram exatamente a mesma matriz em todos os critérios. AE B-ST destacou-se no topo do Label Cloud, enquanto C-ST foi selecionada semanticamente. IF C completa liderou a PR-AUC do Label Cloud no teste, mas B completa foi escolhida pela validação semântica. A convergência é parcial, não identidade entre escolhas.

As fontes também não são intercambiáveis. O Label Cloud associa categorias históricas a endereços. A adjudicação atribui papéis transacionais segundo evidências

estruturais e semânticas. Suas prevalências de teste, aproximadamente 1,77% e 66,16% na política estrita, impedem comparar diretamente PR-AUC ou *lift*. Ambas têm cobertura limitada e não constituem um padrão-ouro independente.

5.7.5 Ordenação em Comparação com Classificação por Limiar

No teste estrito, AE C-ST apresentou ROC-AUC de 0,6779 e superioridade pareada de 71,26%, mas recuperou apenas 2,72% dos atacantes com $Q99$. IF B completa apresentou ROC-AUC de 0,7503 e superioridade de 78,50%, com *recall* de 9,43% e *precision* de 89,52% no mesmo corte selecionado.

Raridade individual e papel relacional não são equivalentes. *Front-running* depende de relações e ordenação entre operações (5, 10). Um atacante pode superar a vítima em escore sem ocupar a cauda extrema global. Os modelos mostraram-se úteis para ranquear candidatos, mas insuficientes como classificadores binários autônomos de todos os ataques.

5.7.6 Heterogeneidade entre os Padrões de Front-Running

Inserção concentrou a evidência favorável, com superioridade pareada de 72,76% no AE e 80,30% no IF. Nos 107 pares de deslocamento, as vítimas predominaram com escores maiores. Supressão não contou com eventos confirmados avaliáveis.

Essa heterogeneidade delimita as conclusões: o resultado agregado não demonstra detecção uniforme de *front-running*. Os padrões possuem estruturas diferentes (5, 10) e podem exigir atributos relacionais e critérios específicos. Ampliar a amostra não elimina, por si só, a inadequação da representação individual para certos papéis.

5.7.7 Implicações para o Uso do Pipeline

O pipeline integra fontes de evidência, não um classificador monolítico. Os modelos produzem escores, as regras geram candidatos independentemente das sinalizações, e o enriquecimento e a adjudicação acrescentam contexto. Essa separação evita confundir raridade com ataque ou restringir a investigação a endereços conhecidos.

Em novas execuções, é necessário monitorar distribuições e taxas de sinalização. Recalibrações e novas seleções devem ser documentadas em janelas anteriores ao teste. O cache compartilhado e o paralelismo RPC reduzem trabalho repetido e tempo de coleta, mas não removem cotas externas nem asseguram resultados analíticos idênticos após retreinamento.

5.8 Considerações Finais

Na execução de 5%, os dois algoritmos produziram sinais exclusivamente *on-chain*, com sensibilidade temporal e concordância dependentes dos atributos. A ablação reduziu

a expansão prospectiva sem assegurar melhor desempenho em todos os critérios.

O Label Cloud revelou associação com agentes catalogados, com ROC-AUC de até 0,9538 no IF e 0,8526 no AE. Na referência semântica estrita, AE C-ST e IF B completa alcançaram ROC-AUC de 0,6779 e 0,7503, e superioridade pareada de 71,26% e 78,50%. A maior parte dos atacantes não ultrapassou os cortes globais, reforçando o uso dos escores para priorização.

Os resultados favoráveis concentraram-se em inserção, sem extensão uniforme a deslocamento ou supressão. Anomalia estatística, associação externa e evidência semântica permanecem níveis distintos. O capítulo seguinte consolida as contribuições e as limitações desse alcance.

6 CONCLUSÕES

Esta dissertação investigou a detecção de comportamentos transacionais anômalos na Ethereum com atributos exclusivamente *on-chain*. O pipeline não supervisionado foi aplicado a 47.545.805 transações de uma amostra determinística de aproximadamente 5% dos blocos de 2024 e 2025, organizada em sete janelas cronológicas. Autoencoder e Isolation Forest foram avaliados em cinco configurações de atributos.

O treinamento foi separado das referências externas. O Label Cloud forneceu associações históricas no nível do endereço, enquanto regras estruturais, enriquecimento RPC e adjudicação automatizada produziram papéis transacionais contextualizados. Essa separação permitiu investigar a informação contida nos escores sem assumir que anomalia equivale a ataque.

6.1 Síntese da Pesquisa e Resposta à Questão Central

A questão central foi:

É possível detectar comportamentos transacionais anômalos utilizando exclusivamente informações observáveis na blockchain Ethereum?

Os resultados sustentam uma resposta afirmativa no sentido de produzir sinais estatísticos e priorizar observações para investigação, não de confirmar automaticamente ataques. A associação externa alcançou ROC-AUC de até 0,9538 no IF e 0,8526 no AE. Esses valores expressam correspondência com endereços catalogados, não a natureza de cada operação.

Na avaliação semântica estrita, AE C-ST e IF B completa foram selecionados exclusivamente na validação. No teste final, alcançaram ROC-AUC de 0,6779 e 0,7503 e PR-AUC de 0,7542 e 0,8453. O atacante recebeu escore superior ao da vítima em 71,26% e 78,50% dos pares, respectivamente. Embora o IF tenha apresentado maior superioridade por par, o AE apresentou maior proporção de eventos em que todos os atacantes superaram a vítima.

Os cortes globais recuperaram poucos atacantes, mesmo quando a ordenação relativa foi informativa. Q_{99} , escolhido pelo maior F_1 na validação, apresentou *recall* de apenas 2,72% no AE e 9,43% no IF no teste estrito. A identificação de papéis relacionais não se reduz à raridade individual de uma transação.

As configurações completas apresentaram maior expansão temporal das sinalizações, enquanto a ablação das taxas absolutas reduziu essa variação em determinadas configurações. Estabilidade e discriminação não foram equivalentes. A concordância reduzida mostrou ainda que reconstrução e isolamento priorizam conjuntos parcialmente distintos.

A evidência favorável concentrou-se em inserção. Deslocamento apresentou ordenação predominantemente inversa e supressão não contou com eventos confirmados avaliáveis. As métricas descrevem a população adjudicada de atacantes e vítimas contextuais, sem estimar desempenho sobre toda a rede. Assim, atributos *on-chain* são úteis para triagem, mas sua interpretação exige evidências relacionais e semânticas, consideração temporal e explicitação da cobertura dos rótulos.

6.2 Contribuições da Pesquisa

As contribuições são metodológicas, experimentais e técnicas. Em conjunto, estabelecem um processo rastreável que mantém as referências de avaliação externas ao treinamento não supervisionado.

6.2.1 Contribuições Metodológicas

A arquitetura distingue anomalia estatística, associação externa e evidência de padrões transacionais. O desenho cronológico separa treinamento, validação e teste e acrescenta recortes de transição e estresse, permitindo analisar mudanças de distribuição sem incorporar informações futuras ao ajuste.

A avaliação combina taxas de sinalização, concordância, métricas contínuas, cortes predefinidos, enriquecimento no topo e comparações pareadas. A seleção semântica ocorre na validação e utiliza vítimas dos próprios eventos como negativos contextuais, sem assumir normalidade de todas as transações não rotuladas. Isso delimita as métricas à população condicionada pela adjudicação.

6.2.2 Contribuições Experimentais

Foi processada uma base de 47.545.805 transações de 261.206 blocos, preservando todas as operações dos blocos selecionados. Sobre essa base, a comparação entre reconstrução e isolamento evidenciou respostas parcialmente distintas e dependentes dos atributos.

A ablação mostrou que remover taxas monetárias absolutas pode reduzir a expansão prospectiva das sinalizações, sem assegurar maior discriminação. A confrontação com duas referências externas de natureza distinta ampliou a interpretação dos escores. A decomposição por padrão de *front-running* delimitou a evidência favorável principalmente à inserção, evitando generalizar métricas agregadas a categorias com resultados inversos ou cobertura insuficiente.

6.2.3 Contribuições Técnicas

Foi implementado um pipeline reproduzível que integra preparação dos dados, matrizes experimentais, treinamento, escores, candidatos estruturais, enriquecimento, dossiês e avaliação. As configurações centralizadas e os resultados isolados por execução permitem diferentes percentuais determinísticos sem substituir artefatos anteriores.

O enriquecimento RPC é retomável, reutiliza cache compartilhado e distribui janelas entre múltiplos provedores. As evidências recuperadas permanecem vinculadas às transações e às decisões da adjudicação automatizada. Essa rastreabilidade permite auditar o percurso entre candidato, evidência e rótulo, sem transformar o escore em prova automática de ataque.

6.3 Limitações da Pesquisa

As limitações envolvem cobertura amostral, representação individual, mudanças temporais, referências externas e disponibilidade das evidências. Elas delimitam o alcance das conclusões e as condições necessárias à reprodução ou extensão do experimento.

6.3.1 Dados, Amostragem e Representação

A amostra determinística de aproximadamente 5% não representa toda a rede e pode excluir blocos contendo eventos raros. Entretanto, preserva todas as transações de cada bloco selecionado, não fragmentando deliberadamente os padrões intrabloco. A regra por módulo também não equivale a uma amostra aleatória com garantia formal de representatividade de todos os comportamentos.

Os atributos descrevem predominantemente características individuais e contexto imediato. Não representam integralmente relações entre operações, fluxos econômicos, chamadas internas e alterações de estado. Essa restrição pode ser relevante para a ordenação inversa observada em deslocamento e para a cobertura insuficiente de supressão.

As janelas de Dencun, Pectra e Fusaka são marcos analíticos, não intervenções controladas. Mudanças de uso da rede e condições econômicas impedem atribuir causalidade a uma atualização isolada. Aplicações a outros períodos ou percentuais exigem nova avaliação das distribuições, dos modelos e da calibração.

6.3.2 Modelos e Referências Externas

AE e IF não abrangem todas as formas de modelar anomalias. Os escores dependem da representação, do treinamento e da configuração, e quantis congelados não garantem estabilidade futura. A ablação evidenciou que estabilidade e discriminação podem variar em direções diferentes.

O Label Cloud classifica endereços, não cada transação, e possui cobertura parcial. Sua presença não confirma uma operação específica de MEV, nem sua ausência demonstra normalidade. A referência é um conjunto histórico cuja associação não estabelece quando o endereço passou a exercer determinada atividade. As métricas são auxiliares e dependem dessa hipótese operacional de rotulação.

6.3.3 Identificação e Validação de Front-Running

As regras estruturais e semânticas são aproximações operacionais da literatura. Compatibilidade com um padrão não comprova intenção, causalidade econômica ou observação do mempool. Contratos proxy, chamadas internas, interfaces não verificadas e lógica personalizada podem dificultar a interpretação de calldata, recibos e logs. Essas evidências não substituem integralmente rastreamentos e reconstrução de alterações de estado.

A adjudicação automatizada não foi validada por revisão independente de todos os eventos por múltiplos especialistas. Não se estimou concordância entre avaliadores nem taxa de erro do próprio rotulador. A distinção entre eventos confirmados e prováveis explicita parte da incerteza das evidências, mas não elimina possíveis erros das regras de validação e adjudicação. A evidência favorável concentra-se em inserção, enquanto deslocamento apresenta ordenação predominantemente inversa e supressão não dispõe de eventos confirmados avaliáveis.

Transações dos mesmos eventos não são independentes, e alguns atacantes aparecem em mais de um par. Os intervalos por reamostragem de transações não modelam integralmente essa dependência. As vítimas são negativos contextuais, não exemplos comprovadamente normais sobre toda a rede.

Consequentemente, as métricas representam separação dentro da população semanticamente adjudicada. Não estimam prevalência de ataques, *recall* sobre todos os ataques da Ethereum ou desempenho de classificação na rede completa.

6.3.4 Reprodutibilidade e Validade Externa

Código, configurações, manifestos e artefatos preservam rastreabilidade, mas a reprodução integral depende dos dados e das fontes externas. Respostas RPC, interfaces e classificações podem mudar. O cache compartilhado reduz consultas repetidas, sem eliminar a necessidade de obter evidências inéditas ou assegurar resultados idênticos após retreinamento.

A ampliação da amostra aumenta sobretudo o custo de enriquecimento RPC. O paralelismo entre janelas reduz o tempo de coleta, mas permanece condicionado às cotas e velocidades dos provedores. Serviços pagos podem ampliar a capacidade, com custo

adicional que limita sua adoção neste contexto. Não foi realizado um estudo econômico que permita quantificar essa alternativa.

Uma infraestrutura própria com Geth e cliente de consenso permitiria expor RPC local, exigindo armazenamento adequado, memória, conectividade, sincronização e manutenção (37, 38). A retenção de blocos e recibos históricos deve atender ao período investigado. Consultas ao estado de contratos em blocos antigos podem exigir retenção de estado histórico ou modo de arquivo, cujo custo varia com a configuração (39, 40). Portanto, um nó convencional não garante, por si só, cobertura de todos os métodos e períodos necessários.

Como continuidade, são pertinentes a validação por especialistas, atributos relacionais e reamostragem por eventos, além de infraestrutura RPC com cobertura histórica verificada. A validade externa permanece limitada ao período, à rede, aos atributos e às regras estudados. Os resultados demonstram utilidade para priorização, não cobertura exaustiva das estratégias de MEV.

REFERÊNCIAS

- 1 JUNG, Eunjin et al. **Data mining-based Ethereum fraud detection**. In: IEEE INTERNATIONAL CONFERENCE ON BLOCKCHAIN, 2., 2019, Atlanta. *Proceedings of the 2019 IEEE International Conference on Blockchain*. Piscataway: IEEE, 2019. p. 266–273. DOI: <<https://doi.org/10.1109/Blockchain.2019.00042>>.
- 2 OSTAPOWICZ, Michał; ŻBIKOWSKI, Kamil. **Detecting fraudulent accounts on blockchain: a supervised approach**. In: CHENG, Reynold et al. (ed.). *Web Information Systems Engineering – WISE 2019*. Cham: Springer, 2019. p. 18–31. (Lecture Notes in Computer Science, v. 11881). DOI: <https://doi.org/10.1007/978-3-030-34223-4_2>.
- 3 SAYADI, Sirine; BEN REJEB, Sonia; CHOUKAIR, Zied. **Anomaly detection model over blockchain electronic transactions**. In: INTERNATIONAL WIRELESS COMMUNICATIONS AND MOBILE COMPUTING CONFERENCE, 15., 2019, Tangier. *Proceedings of the 2019 15th International Wireless Communications and Mobile Computing Conference*. Piscataway: IEEE, 2019. p. 895–900. DOI: <<https://doi.org/10.1109/IWCMC.2019.8766765>>.
- 4 DAIAN, Philip et al. **Flash Boys 2.0: frontrunning in decentralized exchanges, miner extractable value, and consensus instability**. In: IEEE SYMPOSIUM ON SECURITY AND PRIVACY, 2020, San Francisco. *Proceedings of the 2020 IEEE Symposium on Security and Privacy*. Piscataway: IEEE, 2020. p. 910–927. DOI: <<https://doi.org/10.1109/SP40000.2020.00040>>.
- 5 ESKANDARI, Shayan; MOOSAVI, Seyedehmahsa; CLARK, Jeremy. **SoK: transparent dishonesty: front-running attacks on blockchain**. In: BRACCIALI, Andrea et al. (ed.). *Financial Cryptography and Data Security*. Cham: Springer, 2020. p. 170–189. (Lecture Notes in Computer Science, v. 11599). DOI: <https://doi.org/10.1007/978-3-030-43725-1_13>.
- 6 FARRUGIA, Steven; ELLUL, Joshua; AZZOPARDI, George. **Detection of illicit accounts over the Ethereum blockchain**. *Expert Systems with Applications*, v. 150, art. 113318, 2020. DOI: <<https://doi.org/10.1016/j.eswa.2020.113318>>.
- 7 LI, Ji et al. **A survey on blockchain anomaly detection using data mining techniques**. In: ZHENG, Zibin et al. (ed.). *Blockchain and Trustworthy Systems*. Singapore: Springer, 2020. p. 491–504. (Communications in Computer and Information Science, v. 1156). DOI: <https://doi.org/10.1007/978-981-15-2777-7_40>.
- 8 PODGORELEC, Blaž; TURKANOVIĆ, Muhamed; KARAKATIČ, Sašo. **A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection**. *Sensors*, v. 20, n. 1, art. 147, 2020. DOI: <<https://doi.org/10.3390/s20010147>>.
- 9 AL-E'MARI, Salam et al. **A labeled transactions-based dataset on the Ethereum network**. In: ANBAR, Mohammed et al. (ed.). *Advances in Cyber Security*. Singapore: Springer, 2021. p. 61–79. (Communications in Computer and Information Science, v. 1347). DOI: <https://doi.org/10.1007/978-981-33-6835-4_5>.

- 10 TORRES, Christof Ferreira; CAMINO, Ramiro; STATE, Radu. **Frontrunner Jones and the Raiders of the Dark Forest: an empirical study of frontrunning on the Ethereum blockchain.** In: USENIX SECURITY SYMPOSIUM, 30., 2021. *Proceedings of the 30th USENIX Security Symposium*. USENIX Association, 2021. p. 1343–1359. Disponível em: <https://www.usenix.org/conference/usenixsecurity21/presentation/torres>.
- 11 HEIMBACH, Lioba et al. **Ethereum’s proposer-builder separation: promises and realities.** In: ACM INTERNET MEASUREMENT CONFERENCE, 2023, Montreal. *Proceedings of the 2023 ACM Internet Measurement Conference*. New York: ACM, 2023. p. 406–420. DOI: <https://doi.org/10.1145/3618257.3624824>.
- 12 ZHANG, Yuheng et al. **FRAD: front-running attacks detection on Ethereum using ternary classification model.** In: WANG, Guojun et al. (ed.). *Ubiquitous Security*. Singapore: Springer, 2024. p. 63–75. (Communications in Computer and Information Science, v. 2034). DOI: https://doi.org/10.1007/978-981-97-1274-8_5.
- 13 LI, Dongze et al. **A Geth-based real-time detection system for sandwich attacks in Ethereum.** *Discover Computing*, v. 27, n. 1, art. 11, 2024. DOI: <https://doi.org/10.1007/s10791-024-09445-6>.
- 14 CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: a survey. *ACM Computing Surveys*, v. 41, n. 3, art. 15, 2009. DOI: 10.1145/1541880.1541882.
- 15 HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. *Science*, v. 313, n. 5786, p. 504–507, 2006. DOI: 10.1126/science.1127647.
- 16 LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation Forest. In: *2008 Eighth IEEE International Conference on Data Mining*. Pisa, Italy: IEEE, 2008. p. 413–422.
- 17 SAKURADA, M.; YAIRI, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: *Proceedings of the 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA ’14)*. New York: ACM, 2014. DOI: 10.1145/2689746.2689747.
- 18 BUTERIN, V. A next-generation smart contract and decentralized application platform. *Ethereum White Paper*, 2014.
- 19 WOOD, G. Ethereum: a secure decentralised generalised transaction ledger. *Ethereum Yellow Paper*, Shanghai version, 4 Feb. 2025.
- 20 ETHEREUM FOUNDATION. *Proof-of-stake (PoS)*. Ethereum.org, 2026. Disponível em: <https://ethereum.org/developers/docs/consensus-mechanisms/pos/>. Acesso em: 18 set. 2026.
- 21 FAWCETT, T. An introduction to ROC analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861–874, 2006. DOI: 10.1016/j.patrec.2005.10.010.
- 22 DAVIS, J.; GOADRICH, M. The relationship between Precision-Recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York: ACM, 2006. p. 233–240. DOI: 10.1145/1143844.1143874.

- 23 SAITO, T.; REHMSMEIER, M. The Precision-Recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, v. 10, n. 3, e0118432, 2015. DOI: 10.1371/journal.pone.0118432.
- 24 KAR, P.; NARASIMHAN, H.; JAIN, P. Surrogate functions for maximizing Precision at the Top. In: *Proceedings of the 32nd International Conference on Machine Learning*. Proceedings of Machine Learning Research, v. 37, p. 189–198, 2015.
- 25 SPEARMAN, C. The proof and measurement of association between two things. *The American Journal of Psychology*, v. 15, n. 1, p. 72–101, 1904. DOI: 10.2307/1412159.
- 26 THOMAS, L. C. *Consumer Credit Models: Pricing, Profit and Portfolios*. Oxford: Oxford University Press, 2009.
- 27 BERGER, V. W.; ZHOU, Y. Kolmogorov–Smirnov Tests. In: *Encyclopedia of Biostatistics*. 2. ed. Chichester: John Wiley & Sons, 2005. DOI: 10.1002/0470013192.bsa332.
- 28 LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. In: *International Conference on Learning Representations*. 2019.
- 29 JACCARD, P. Étude comparative de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, v. 37, p. 547–579, 1901. DOI: 10.5169/seals-266450.
- 30 ETHERSCAN. *Public Name Tags, Labels & Public Notes*. Etherscan Information Center, 2021. Disponível em: <<https://info.etherscan.com/public-name-tags-labels/>>. Acesso em: 19 set. 2026.
- 31 ETHEREUM FOUNDATION. *JSON-RPC API*. Ethereum.org, 2026. Disponível em: <<https://ethereum.org/developers/docs/apis/json-rpc/>>. Acesso em: 21 set. 2026.
- 32 EFRON, Bradley. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, v. 7, n. 1, p. 1–26, 1979. DOI: 10.1214/aos/1176344552.
- 33 AMAZON WEB SERVICES. **AWS Public Blockchain Data**. Registry of Open Data on AWS. Disponível em: <<https://registry.opendata.aws/aws-public-blockchain/>>. Acesso em: 21 set. 2026.
- 34 ETHEREUM FOUNDATION. **Cancun-Deneb (Dencun)**. Ethereum.org, 2024. Disponível em: <<https://ethereum.org/roadmap/dencun/>>. Acesso em: 24 set. 2026.
- 35 ETHEREUM FOUNDATION. **Pectra Mainnet Announcement**. Ethereum Foundation Blog, 23 abr. 2025. Disponível em: <<https://blog.ethereum.org/2025/04/23/pectra-mainnet>>. Acesso em: 24 set. 2026.
- 36 ETHEREUM FOUNDATION. **Fusaka Mainnet Announcement**. Ethereum Foundation Blog, 6 nov. 2025. Disponível em: <<https://blog.ethereum.org/2025/11/06/fusaka-mainnet-announcement>>. Acesso em: 24 set. 2026.
- 37 GO ETHEREUM. **Hardware requirements**. Disponível em: <<https://geth.ethereum.org/docs/getting-started/hardware-requirements>>. Acesso em: 1 out. 2026.

- 38 GO ETHEREUM. **Sync modes**. Disponível em:
<<https://geth.ethereum.org/docs/fundamentals/sync-modes>>. Acesso em: 1 out. 2026.
- 39 GO ETHEREUM. **Archive mode**. Disponível em:
<<https://geth.ethereum.org/docs/fundamentals/archive>>. Acesso em: 1 out. 2026.
- 40 GO ETHEREUM. **History Pruning**. Disponível em:
<<https://geth.ethereum.org/docs/fundamentals/historypruning>>. Acesso em: 1 out. 2026.