INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DETECÇÃO DE ANOMALIAS EM DADOS HISTÓRICOS DE CRIPTOMOEDAS

FULGÊNCIO THIERRY GOMES DA SILVA

Cajazeiras
2025

FULGÊNCIO THIERRY GOMES DA SILVA

DETECÇÃO DE ANOMALIAS EM DADOS HISTÓRICOS DE CRIPTOMOEDAS

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto.

Cajazeiras
2025

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

S586d Silva, Fulgêncio Thierry Gomes da.

Detecção de anomalias em dados históricos de criptomoedas / Fulgêncio Thierry Gomes da Silva. – Cajazeiras, 2025.
75f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas.) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador: Prof. Me. Francisco Paulo de Freitas Neto.

1. Desenvolvimento de sistemas. 2. Mercado financeiro. 3. Criptomoedas. 4. Detecção de anomalias. I. Instituto Federal da Paraíba. II. Título.

IFPB/CZ CDU: 004.4:336.7(043.2)

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

FULGÊNCIO THIERRY GOMES DA SILVA

DETECÇÃO DE ANOMALIAS EM DADOS HISTÓRICOS DE CRIPTOMOEDAS

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto

Aprovada em: **20 de Março de 2025.**

Prof. Me. Francisco Paulo de Freitas Neto - Orientador

Prof. Dr. Fabio Gomes de Andrade - Avaliador

IFPB - Campus Cajazeiras

Prof. Me. Janderson Ferreira Dutra - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 20/03/2025 13:48:39.
- **Janderson Ferreira Dutra**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 20/03/2025 13:57:34.
- **Fabio Gomes de Andrade**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 24/03/2025 11:12:01.

Este documento foi emitido pelo SUAP em 20/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código 685310
Verificador: d3065a83e0
Código de Autenticação:

AGRADECIMENTOS

Agradeço primeiramente a Deus que me permitiu continuar até aqui.

Agradeço aos meus pais que foram as pessoas que mais me apoiaram durante toda a minha vida, não sendo diferente durante a graduação.

Agradeço a todos os membros do grupo “fã clube matteus ferraz” cada piada, cada meme e cada discussão infundada proveniente do mesmo me ajudaram um pouco a acabar com o estresse dessa caminhada.

Agradeço aos meus colegas de curso que sempre estiveram enfrentando as dificuldades junto comigo.

Agradeço ao meu orientador, Prof. Me. Francisco Paulo de Freitas Neto, pela paciência e tempo dedicados a me ajudar e por aceitar enfrentar esse trabalho junto a mim.

”Se você não gosta do seu destino, não aceite. Em vez disso, tenha a coragem de mudá-lo do jeito que você quer que seja.”

Naruto Uzumaki, Naruto

RESUMO

O mercado de criptomoedas é conhecido por sua natureza volátil e desregulada, tornando a identificação de padrões anômalos uma tarefa desafiadora, mas de extrema importância. O presente trabalho explora a detecção de anomalias em dados históricos de criptomoedas utilizando os algoritmos de Machine Learning, em específico, *Robust Covariance*, *One-Class SVM*, *Isolation Forest*, *Local Outlier Factor* e o método estatístico Z-score. Após uma revisão da fundamentação teórica sobre criptomoedas, mercado de criptomoedas, *Machine Learning* e algoritmos de detecção de anomalias, foi realizada a coleta de dados históricos de criptomoedas da plataforma *Investing*. Esses dados foram utilizados para treinar e avaliar os modelos. Após o treinamento dos modelos, foram realizadas análises de desempenho e comparação entre os algoritmos. Também foram criados recursos de visualização dos dados para auxiliar na interpretação dos resultados.

Palavras-chave: Criptomoedas. Machine Learning. Algoritmos. Detecção de Anomalias.

ABSTRACT

The cryptocurrency market is known for its volatile and unregulated nature, making the identification of anomalous patterns a challenging yet crucial task. This study explores anomaly detection in historical cryptocurrency data using Machine Learning algorithms, specifically Robust Covariance, One-Class SVM, Isolation Forest, Local Outlier Factor, and the statistical method Z-score. After a theoretical review covering cryptocurrencies, the cryptocurrency market, Machine Learning, and anomaly detection algorithms, historical cryptocurrency data was collected from the Investing platform. This data was used to train and evaluate the models. Following the training phase, performance analyses and comparisons between the algorithms were conducted. Additionally, data visualization tools were created to support the interpretation of the results.

Keywords: Cryptocurrencies. Machine Learning. Algorithms. Anomaly Detection.

LISTA DE FIGURAS

Figura 1 – Gráfico de Capitalização de Mercado de Cripto Total 11
Figura 2 – Exemplos de Anomalias em Diferentes Tipos de Dados 13
Figura 3 – *Isolation Forest* através de árvores 26
Figura 4 – Dataframe *Bitcoin* 36
Figura 5 – Cotação do *Bitcoin* ao longo dos anos a serem testados nos modelos (2019 – 2022) 37
Figura 6 – Cotação do *Ether* ao longo dos anos a serem testados nos modelos (2019 – 2022) 38
Figura 7 – Cotação do *Litecoin* ao longo dos anos a serem testados nos modelos (2019 – 2022) 39
Figura 8 – Gráfico de dispersão resultado dos métodos aplicados no Bitcoin . . 47
Figura 9 – Gráfico de dispersão resultado dos métodos aplicados no Ethereum 48
Figura 10 – Gráfico de dispersão resultado dos métodos aplicados no Litecoin . 49
Figura 11 – Gráfico com percentual de anomalias detectadas no Bitcoin 50
Figura 12 – Gráfico com percentual de anomalias detectadas no Ethereum . . . 51
Figura 13 – Gráfico com percentual de anomalias detectadas no Litecoin 53
Figura 14 – Gráfico Silhouette Score no Bitcoin 55
Figura 15 – Gráfico Silhouette Score no Ethereum 56
Figura 16 – Gráfico Silhouette Score no Litecoin 58
Figura 17 – Gráfico P-Value no Bitcoin 60
Figura 18 – Gráfico KS-Statistcs no Bitcoin 60
Figura 19 – Gráfico P-Value no Ethereum 61
Figura 20 – Gráfico KS-Statistcs no Ethereum 62
Figura 21 – Gráfico P-Value no Litecoin 63
Figura 22 – Gráfico KS-Statistcs no Litecoin 64

LISTA DE TABELAS

Tabela 1 – Percentual de Anomalias Detectadas no Bitcoin por Método 51
Tabela 2 – Percentual de Anomalias Detectadas no Ethereum por Método . . . 52
Tabela 3 – Percentual de Anomalias Detectadas no Litecoin por Método 53
Tabela 4 – Silhouette Score para Detecção de Anomalias no Bitcoin 55
Tabela 5 – Silhouette Score para Detecção de Anomalias no Ethereum 57
Tabela 6 – Silhouette Score para Detecção de Anomalias no Litecoin 58

LISTA DE ABREVIATURAS E SIGLAS

CDF	Cumulative Distribution Function
CSV	Comma-separated values
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
eps	Epsilon
ETH	Ether
FAST-MCD	FAST-mínimum covariance determinant
FN	Falsos negativos
FP	Falsos positivos
IForest	Isolation Forest
KS	Kolmogorov-Smirnov
KS-Statistic	Kolmogorov-Smirnov-Statistc
LOF	Local Outlier Factor
MinPts	Mínimo de pontos
One-Class SVM	One-Class Support Vector Machine
P2P	Peer-to-peer
SVMs	Support Vector Machines
VP	Verdadeiros Positivos

SUMÁRIO

1 INTRODUÇÃO 11

1.1 Objetivos 14

1.1.1 Objetivo Geral 14

1.1.2 Objetivos Específicos 14

1.2 Trabalhos Relacionados 15

1.3 Organização do documento 17

2 FUNDAMENTAÇÃO TEÓRICA 18

2.1 Definição e História das Criptomoedas 18

2.2 Bitcoin 19

2.3 Ethereum 20

2.4 Litecoin 20

2.5 Funcionamento das Transações de Criptomoedas 21

2.6 Machine Learning 22

2.6.1 Aprendizado Supervisionado 23

2.6.2 Aprendizado Não Supervisionado 23

2.6.3 Aprendizado Por Reforço 24

2.7 Algoritmos e Técnicas na Detecção de Anomalias 25

2.7.1 Isolation Forest 25

2.7.2 Robust Covariance 26

2.7.3 One-Class Support Vector Machine 27

2.7.4 Local Outlier Factor 28

2.7.5 Z-Score 29

2.8 Métricas Para Avaliação de Modelos não Supervisionados 29

2.8.1 Percentual de Anomalias 29

2.8.2 Silhouette Score 30

2.8.3 Teste de Kolmogorov-Smirnov 31

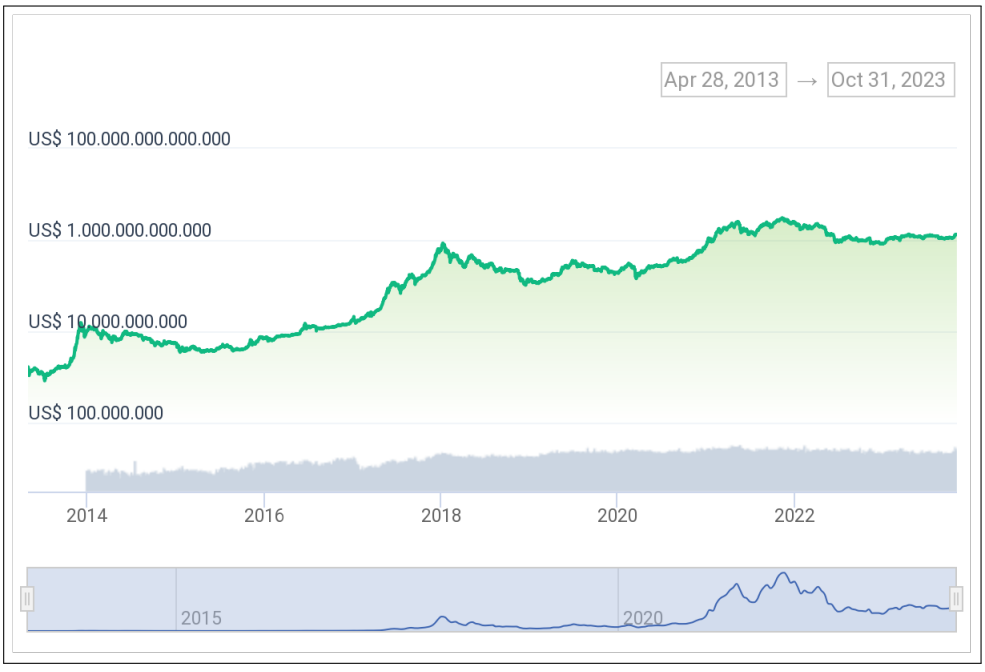
2.9 Tecnologias Utilizadas 32
2.9.1 Python 32
2.9.2 Bibliotecas 33
2.9.3 Jupyter Notebook 34
3 METODOLOGIA 35
3.1 Base de dados 35
3.2 Pré-processamento dos dados 39
3.3 Aplicação das técnicas de detecção de anomalias 42
3.4 Avaliação dos métodos 44
4 RESULTADOS 47
4.1 Anomalias encontradas 47
4.2 Metricas 49
5 CONSIDERAÇÕES FINAIS 67
REFERÊNCIAS 68

1 INTRODUÇÃO

Em 2008 o mundo conhecia a primeira das criptomoedas, o *Bitcoin* (NAKAMOTO, 2008), que foi o ponto de partida para a formação de um novo mercado de investimentos, o mercado de criptomoedas, que vem crescendo exponencialmente com o passar dos anos, atraindo uma grande variedade de investidores devido ao seu alto potencial de retorno (EDERLI et al., 2021). Contudo, o mercado de criptomoedas tem uma natureza um tanto quanto instável e desregulada, tornando-se também predisposto a anomalias, que podem ser descritas como as flutuações de preço extremas às quais o mercado é submetido.

A figura 1 mostra claramente o forte crescimento do mercado de criptomoedas ao longo de cerca de dez anos, entre 2013 e 2023. É possível perceber que a capitalização total do setor saiu da casa dos milhões e chegou a ultrapassar 3 trilhões de dólares em seu auge, por volta de 2021. Esse avanço expressivo reflete o aumento do interesse mundial por ativos digitais, impulsionado por moedas como o *Bitcoin* e o *Ethereum*, além do desenvolvimento de tecnologias como os contratos inteligentes (SZABO, 1996) e as finanças descentralizadas (ALI, 2024). Mesmo com quedas e correções no mercado, especialmente a partir de 2022, o gráfico mostra que o setor cripto vem se consolidando como uma parte importante da economia global, com sinais de crescimento contínuo no longo prazo.

Figura 1 – Gráfico de Capitalização de Mercado de Cripto Total

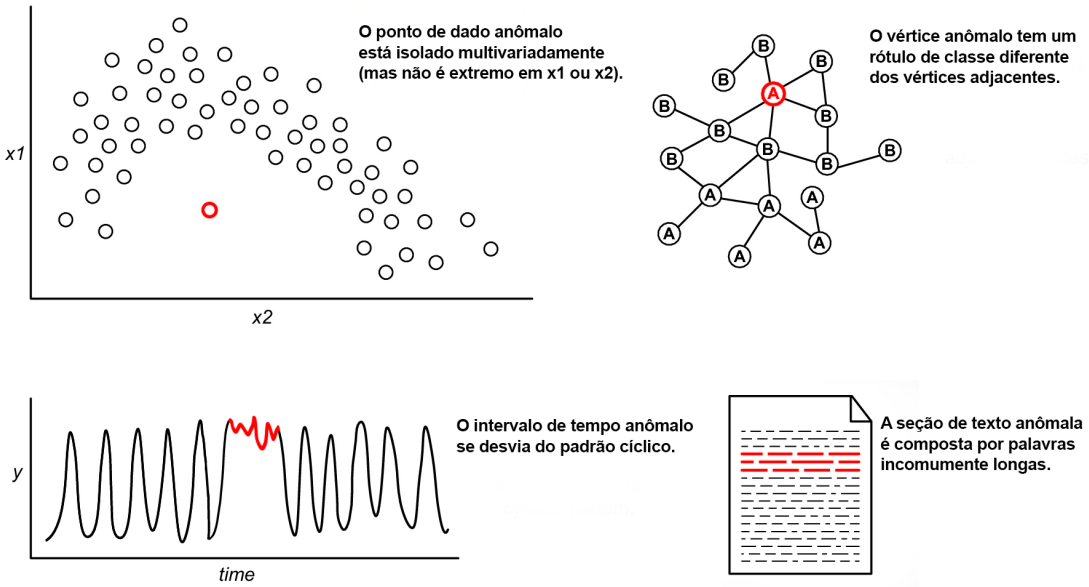


Fonte: (COINGECKO, 2023)

Outliers, ou valores atípicos, são dados que fogem do padrão esperado e se destacam por estarem muito distantes da maioria das outras observações. Eles podem representar erros, ruídos ou até situações raras que merecem atenção. Em análises de dados, principalmente em tarefas de classificação, esses pontos fora da curva podem interferir nos resultados e prejudicar o desempenho dos modelos. *Outliers* são vistos como instâncias que se comportam de maneira diferente da maioria, exigindo técnicas específicas para serem identificados corretamente (FREITAS, 2019). Neste trabalho, esses outliers são tratados como anomalias, e o foco está justamente em encontrá-los nas séries temporais de preços de criptomoedas, buscando identificar comportamentos incomuns que possam indicar mudanças importantes ou instabilidades no mercado.

A Figura 2 mostra, de forma visual e simples, como anomalias podem aparecer em diferentes tipos de dados. No primeiro exemplo, vemos um ponto que, embora não seja extremo em nenhum eixo isoladamente, está isolado do restante dos dados quando olhamos para as variáveis em conjunto. Em seguida, o gráfico de rede mostra um nó com uma classificação diferente dos seus vizinhos, indicando que algo ali foge do padrão esperado. No exemplo de série temporal, há uma clara quebra em um padrão cíclico, sugerindo que algo incomum aconteceu naquele intervalo de tempo. Por fim, no exemplo com texto, a anomalia aparece como um trecho com palavras muito maiores do que as demais. Esses exemplos ajudam a entender que anomalias são situações que não seguem o comportamento geral dos dados, e no contexto deste TCC, são fundamentais para detectar variações inesperadas no preço de criptomoedas, o que pode indicar mudanças importantes no mercado ou comportamentos atípicos de investidores.

Figura 2 – Exemplos de Anomalias em Diferentes Tipos de Dados



Fonte: (FOORTHUIS, 2021)

Anomalias são situações, eventos ou dados que fogem do padrão esperado dentro de um conjunto de informações. Elas podem representar desde simples erros ou ruídos até sinais importantes, como fraudes, falhas em sistemas ou mudanças inesperadas. Esses desvios podem aparecer de diferentes formas, dependendo do tipo de dado analisado, como casos isolados, comportamentos fora do contexto ou padrões estranhos em grupo (FOORTHUIS, 2021). O estudo de Foorthuis (2018) traz uma forma mais organizada de entender esses desvios, levando em conta o tipo e a estrutura dos dados, além da complexidade da anomalia. Compreender essas diferenças é essencial para aplicar métodos eficazes de detecção, especialmente em contextos como o deste trabalho, que busca identificar comportamentos atípicos nos preços de criptomoedas ao longo do tempo.

Tratando-se de um conjunto de dados, pode-se definir anomalias, ou valores atípicos, como os pontos de dados que se destacam do restante do conjunto. O processo de detecção dessas anomalias refere-se à identificação desses pontos não conformes (CHANDOLA et al., 2009).

A detecção de anomalias por meio de técnicas de *machine learning* emerge como uma área de estudo crucial, destacada por diversos pesquisadores (AMIRZADEH et al., 2022). Essa abordagem revela-se promissora devido à capacidade do *machine learning* em realizar previsões assertivas e aprender padrões de alta complexidade (AMARAL, 2016). Nesse contexto, a aplicação de algoritmos de detecção de anomalias

torna-se indispensável para compreender e monitorar eventos anômalos que podem causar volatilidade.

O estudo da utilização desses algoritmos em dados históricos de criptomoedas revela-se crucial para identificar padrões anômalos e, assim, compreender os fatores que influenciam a volatilidade do mercado. Conduzir uma análise aprofundada sobre como esses algoritmos operam nos dados específicos do mercado de criptomoedas é de suma importância (COINGECKO, 2023). Isso implica avaliar a eficácia de diferentes algoritmos em uma abordagem que lida com um grau elevado de instabilidade, característico das inúmeras flutuações de preço extremas no mercado de criptomoedas.

Portanto, detectar anomalias no mercado de criptomoedas aplicando métodos de *machine learning* é uma área de pesquisa que eventualmente ajudará os investidores a tomarem decisões mais assertivas que minimizem os riscos relacionados ao investimento nesse mercado, transmitindo mais segurança aos investidores.

Este Trabalho de Conclusão de Curso (TCC) investiga como o *machine learning* pode ser usado para detectar anomalias no mercado de criptomoedas. O objetivo foi aplicar e testar algoritmos especializados nessa tarefa, como o *Isolation Forest* (LIU et al., 2008), que é amplamente utilizado para identificar padrões incomuns em diversos tipos de dados. Para isso, foram analisadas as cotações de criptomoedas, como o *Bitcoin*, ao longo dos anos, buscando identificar variações atípicas e entender a eficácia desses métodos em um mercado conhecido por sua alta volatilidade.

1.1 OBJETIVOS

Seção dedicada aos objetivos abordados durante o desenvolvimento deste trabalho.

1.1.1 Objetivo Geral

Aplicar métodos como algoritmos de machine e metodos estatisticos learning para a detecção de anomalias em dados históricos de criptomoedas e avaliar a eficácia dos modelos na identificação de padrões atípicos.

1.1.2 Objetivos Específicos

O trabalho conta com os seguintes objetivos específicos.

- Identificar e analisar os padrões e fatores associados a anomalias em séries temporais de criptomoedas, considerando aspectos como volatilidade, variações

atípicas e comportamento do mercado.

- Compreender, com base na literatura, os principais métodos e técnicas de *machine learning* e estatisticos aplicados à detecção de anomalias em séries temporais, avaliando suas vantagens e limitações.
- Comparar o desempenho dos modelos testados, analisando métricas de avaliação para determinar as abordagens mais eficazes na identificação de padrões incomuns.
- Fornecer informações relevantes para aplicações práticas, auxiliando investidores e pesquisadores na detecção precoce de comportamentos anômalos no mercado de criptomoedas.

1.2 TRABALHOS RELACIONADOS

Na seção atual, abordaremos trabalhos que possuem relação com o conteúdo abordado no presente trabalho.

No trabalho de Pinto (2023), foi conduzida uma revisão de literatura sobre métodos e técnicas de detecção de anomalias em sistemas financeiros. Nesse estudo, destacou-se a importância das abordagens de detecção de anomalias para aprimorar sistemas de tomada de decisão, reduzir riscos na performance econômica e minimizar custos para os consumidores. O autor desenvolveu um *framework* de classificação por meio de códigos para sistematizar as principais técnicas e conhecimentos relacionados à detecção de anomalias em sistemas financeiros, além de identificar diversas lacunas de pesquisa. Foram destacadas três principais áreas que devem ser exploradas para o desenvolvimento da detecção de anomalias em sistemas financeiros: uma base de dados comum, testes com diferentes dimensões de dados e indicadores de efetividade dos modelos de detecção.

Diferente do estudo de Pinto (2023), que busca melhorar a tomada de decisão em sistemas financeiros de forma geral por meio da detecção de anomalias e da redução de custos para os consumidores, este trabalho foca especificamente no mercado de criptomoedas. A proposta aqui é aplicar técnicas de detecção de anomalias em um contexto marcado por alta volatilidade, explorando os desafios e particularidades desse ambiente em comparação com outros setores do sistema financeiro.

Ozer e Sakar (2022) sugeriram um sistema de negociação fundamentado em *machine learning* para gerenciar a volatilidade e os riscos no mercado de criptomoedas. Isso foi motivado pelo crescimento acelerado do mercado de criptomoedas, que

desperta o interesse de diversos investidores. Eles destacaram os riscos ligados à volatilidade do mercado de criptomoedas, decorrentes de notícias especulativas e ações imprevistas de grandes investidores. Constataram que flutuações rápidas e intensas de preços ou padrões atípicos podem impactar negativamente a eficiência dos sinais técnicos em sistemas de negociação baseados em *machine learning*, prejudicando a generalização do modelo. Concluiu-se que a fase de detecção de anomalias amplia significativamente o retorno sobre o investimento para estratégias de negociação apoiadas em *machine learning*. Demonstrou-se que, durante períodos de alta volatilidade, o sistema de negociação se torna mais rentável em comparação com o modelo padrão e a estratégia de comprar e reter.

Tendo o trabalho de Ozer e Sakar (2022) como uma das fontes de inspiração devido ao seu sistema desenvolvido para integrar a detecção de anomalias em um sistema de negociação automatizado, tendo negociações práticas como abordagem, o atual trabalho destaca a comparação de algoritmos com o intuito de escolher o mais adequado entre diferentes algoritmos usados na detecção de anomalias em dados históricos de criptomoedas.

Em Dokuz et al. (2020) foi proposto um estudo no qual foi realizada a detecção de anomalias nos preços do *Bitcoin* com base em mudanças no preço e na variação percentual em relação ao dia anterior, utilizando um conjunto de dados abrangendo o período de 2012 a 2019. Utilizando o algoritmo DBSCAN (ESTER et al., 1996) e um método estatístico para identificar padrões anômalos nos preços do *Bitcoin*. Ambos os métodos foram eficazes na detecção de anomalias, porém o DBSCAN se demonstra superior em relação ao método estatístico, especialmente na detecção de anomalias próximas às mudanças diárias normais de preço.

Focando na detecção de anomalias no preço do Bitcoin, o trabalho de Dokuz et al. (2020) apresenta um estudo direcionado a uma única criptomoeda, no qual foi realizada uma comparação entre o algoritmo mencionado e um método estatístico baseado na suposição de que os dados seguem uma distribuição normal, para identificar qual era mais eficaz. Buscando uma maior abrangência do tema, o presente trabalho tem foco semelhante, mas com o diferencial de que a detecção de anomalias ocorre em diferentes criptomoedas, utilizando diferentes tipos de algoritmos. Tendo como um dos objetivos identificar o algoritmo que mais se destaca nesse contexto, ou seja, o que apresenta o melhor desempenho, visando encontrar aquele com maior potencial para o propósito ao qual foi submetido.

1.3 ORGANIZAÇÃO DO DOCUMENTO

A estrutura deste trabalho está dividida em mais três capítulos. O Capítulo 2 aborda a fundamentação teórica, apresentando pesquisas e conceitos científicos já estabelecidos que embasam o tema do estudo. No Capítulo 3, são detalhadas as etapas da implementação da pesquisa, explicando o processo e as metodologias utilizadas. O Capítulo 4 apresenta os resultados obtidos, analisando os dados e discutindo os achados da pesquisa. Por fim, o Capítulo 5 traz as considerações finais, destacando as principais conclusões e as expectativas em relação ao desenvolvimento do estudo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 DEFINIÇÃO E HISTÓRIA DAS CRIPTOMOEDAS

As criptomoedas, representando uma inovação no mundo financeiro, têm uma história que remonta às décadas de 80 e 90. Os primeiros conceitos e tentativas de estabelecer novas moedas virtuais emergiram no final do século passado. Essas inovações foram projetadas com a intenção de substituir o dinheiro convencional (GRIFFITH, 2014). *David Chaum*, o fundador da empresa *Digicash*, foi um pioneiro ao usar criptografia para garantir a segurança das transações, introduzindo a moeda *eCash*, que, embora não seja idêntica às criptomoedas atuais, marcou o início desse desenvolvimento (CHUEN, 2015).

Apesar de ter desfrutado de algum sucesso, a empresa tomou decisões erradas e faliu, levando à descontinuação da moeda. No entanto, o sistema mostrou potencial ao empregar criptografia e assinaturas cegas *(Blind Signatures)* para proteger a identidade dos usuários (CHUEN, 2015). Após a queda da *Digicash*, outras empresas, como o *PayPal*, exploraram ideias de dinheiro digital. No início do século XXI, as moedas digitais de ouro, garantidas por depósitos em barras de ouro, começaram a ganhar destaque (GRIFFITH, 2014).

O sistema *e-Gold* foi uma das primeiras plataformas de pagamento eletrônico relevantes e contribuiu com tecnologias usadas até hoje em sistemas de *e-commerce*. No entanto, a crise financeira global de 2008 afetou as moedas digitais e as plataformas de pagamento, levando ao encerramento do sistema *e-Gold*. Nesse momento de incerteza, o interesse em criptomoedas não diminuiu. A teoria por trás das moedas digitais ofereceu soluções para os problemas econômicos apresentados pela crise (CHUEN, 2015).

Os entusiastas acreditavam que as criptomoedas, com sua geração regulada e quantidade limitada, poderiam ser uma alternativa aos problemas da economia tradicional, onde os bancos centrais tentavam resolver questões com a geração desigual de dinheiro (CHUEN, 2015). Nesse contexto, em 2008, surgiu a pioneira das criptomoedas modernas, o famoso *Bitcoin*.

Para entender as criptomoedas, é importante destacar que elas não se limitam a ser apenas um sistema de pagamentos virtual, mas representam uma evolução além que concede aos usuários maior autonomia sobre ativos financeiros (BRETERNITZ et al., 2008). O *Bitcoin*, por exemplo, é uma forma de dinheiro que não é emitida por

nenhum governo, permitindo transações globais pela internet.

O diferencial das criptomoedas é a sua base matemática, que as torna transparentes e auditáveis, garantindo a confiabilidade e integridade das transações (NAKAMOTO, 2008). Nesse sentido, as criptomoedas representam um marco na história das finanças, oferecendo um novo paradigma de moeda e transações financeiras.

2.2 BITCOIN

O *Bitcoin*, criado em 2008 e implantado em 2009 por Satoshi Nakamoto, é a criptomoeda mais proeminente do mundo. A essência do *Bitcoin* é a criação de um sistema de pagamento eletrônico baseado em criptografia, permitindo que duas partes transacionem diretamente, eliminando a necessidade de intermediários. Isso resulta em transações computadorizadas que protegem vendedores de fraudes, e essas transações são registradas e mantidas cronologicamente em uma rede P2P (*Peer-to-peer*), proporcionando maior segurança (NAKAMOTO, 2008).

O *Bitcoin* é impulsionado por algoritmos de validação, incluindo a função *hash*, que mapeia dados variáveis em formatos fixos e únicos. Esta tecnologia de *hash* é amplamente usada e essencial para garantir a segurança e integridade das transações (PAAR; PELZL, 2009).

A moeda virtual *Bitcoin* cresceu organicamente, substituindo o sistema de escambo e evoluindo das moedas mercadorias, como gado, sal, cacau, prata e ouro, para uma moeda digital que não é emitida nem regulamentada pelo governo (CARVALHO et al., 2015). À medida que as relações comerciais se tornam cada vez mais digitais, o *Bitcoin* ganha destaque como uma forma segura e acessível de pagamento em todo o mundo, pois não é emitido por nenhum governo (RIBEIRO, 2015).

Em 2011, a revista Forbes publicou um artigo que desencadeou o crescimento acelerado das moedas digitais, em especial do *Bitcoin*, que atingiu seu pico de valorização em dezembro de 2017, chegando a cerca de USD 17,6 mil por *Bitcoin*. As preocupações com fraudes no sistema de criptografia foram substituídas por questões sobre a conversão de *Bitcoins* em moedas nacionais, como euro, dólar e iene (DINIZ, 2017).

Embora haja questionamentos legítimos sobre o potencial uso do *Bitcoin* em atividades ilegais, é importante entender que essa criptomoeda é apenas uma parte de um sistema muito maior. A tecnologia subjacente que sustenta o *Bitcoin* é o *blockchain*, uma inovação essencial que revolucionou como as transações são registradas e

validadas de maneira segura e transparente (SCHWAB; MIRANDA, 2016).

2.3 ETHEREUM

A *Ethereum*, criada em julho de 2015, é uma inovação notável na tecnologia blockchain. Ao contrário da *Bitcoin*, a *Ethereum* transcende o mero conceito de uma moeda digital e se transforma em uma plataforma de software descentralizada, como observado por Reiff (2023). A característica definidora da *Ethereum* é sua capacidade de implementar contratos inteligentes e aplicativos descentralizados que funcionam sem qualquer tempo de inatividade, fraude, controle ou interferência de terceiros.

O diferencial da *Ethereum* reside na sua própria linguagem de programação que opera em sua *blockchain*, permitindo que os desenvolvedores criem e executem aplicativos distribuídos. Todas as ações na plataforma *Ethereum* são impulsionadas por seu *token* criptográfico nativo, conhecido como *ether* (ETH). O ETH desempenha o papel de “combustível” para a execução de comandos na plataforma, sendo utilizado pelos desenvolvedores para a construção e operação de aplicativos na rede. Além disso, o ETH é negociado como uma moeda digital nas bolsas, à semelhança de outras criptomoedas (BUTERIN, 2023).

Vale ressaltar que a *Ethereum*, representa uma *blockchain* de propósito geral programável para uma ampla variedade de tarefas. Ela difere substancialmente da *blockchain* do *Bitcoin*, pois sua moeda nativa, o ETH, não é apenas uma criptomoeda, mas é usado como meio de pagamento pela execução de código na rede *Ethereum*. Isso torna a *Ethereum* uma máquina virtual descentralizada que não apenas mantém registros, mas também executa aplicativos descentralizados, incluindo Contratos Inteligentes (ANTONOPOULOS; WOOD, 2018). Essa versatilidade tecnológica permitiu o desenvolvimento de diversos tipos de aplicações descentralizadas, abrangendo desde carteiras de criptomoedas até sistemas financeiros e jogos.

2.4 LITECOIN

O *Litecoin*, criado em outubro de 2011 por Charles Lee, um ex-funcionário da *Google*, é uma criptomoeda notável que compartilha muitas semelhanças com o *Bitcoin*. No entanto, o *Litecoin* tem suas próprias características distintas que o tornam uma entidade única no mundo das criptomoedas.

O *Litecoin*, uma criptomoeda de código aberto lançada em plataformas colaborativas de desenvolvimento (BRADBURY, 2013), destaca-se por um atributo significativo: o tempo de mineração para cada bloco, estimado em apenas 2,5 minutos. Essa

característica confere ao *Litecoin* uma velocidade superior à do *Bitcoin*, agilizando o processamento das transações em sua rede.

Ao contrário do objetivo de substituir a mineração de *Bitcoin*, o *Litecoin* foi criado com a ideia de permitir a mineração conjunta de *Bitcoin* e *Litecoin*. Ambas as criptomoedas passam pelo evento de *halving* a cada quatro anos, reduzindo pela metade as recompensas de mineração. O algoritmo utilizado pelo *Litecoin* para estabelecer o processo matemático de mineração é o *Scrypt*, um algoritmo de mineração que garante a criptografia e a segurança da moeda (PERCIVAL, 2009).

2.5 FUNCIONAMENTO DAS TRANSAÇÕES DE CRIPTOMOEDAS

As transações de criptomoedas são um processo fundamental dentro desse ecossistema financeiro digital. Para entender como funcionam, é necessário observar o papel central desempenhado pelo *blockchain*, o registro público que mantém o histórico de todas as transações (BORGES, 2018).

Nesse contexto, o *blockchain* é como um grande banco de dados que registra criptograficamente todas as transações. Cada nova transação é agrupada em blocos e adicionada ao *blockchain*, criando um histórico imutável (TOMÉ, 2019). A segurança desse sistema é garantida pela descentralização e pela imutabilidade das informações. Uma vez que uma transação está no *blockchain*, ela não pode mais ser alterada (MEIRA et al., 2020).

O processo de verificação das transações não é realizado por uma entidade central, mas sim por diversos usuários da rede que utilizam poder computacional para resolver problemas criptográficos e verificar as operações. Isso elimina a necessidade de intermediários e reduz a possibilidade de fraudes (BORGES, 2018).

Uma das vantagens das transações com criptomoedas é a ausência de regulamentação central, o que pode reduzir os custos de transação e promover o comércio internacional (BÖHME et al., 2015). As transações com criptomoedas costumam ter taxas baixas, representando uma economia em relação aos cartões de crédito (MUXFELDT, 2021).

Apesar de apresentarem inovações significativas, as criptomoedas ainda enfrentam diversas desvantagens. Entre os principais pontos negativos estão a alta volatilidade dos preços, que gera insegurança para investidores, e a ausência de regulamentação clara, o que pode facilitar práticas ilícitas, como lavagem de dinheiro e evasão fiscal (MARTINS; VAL, 2016). Além disso, a complexidade tecnológica e a dependência

de conhecimento técnico dificultam o acesso para grande parte da população, limitando seu uso mais amplo (DALL'AGNOL, 2022).

2.6 MACHINE LEARNING

Machine Learning, é uma área da computação que tem ganhado destaque significativo nas últimas décadas, à medida que a quantidade de dados disponíveis continua a crescer em diversas áreas (SMOLA, 2008). Esses sistemas têm o objetivo de aprender com os dados e tomar decisões com o mínimo de intervenção humana, tornando-se uma opção altamente relevante em uma variedade de aplicações, desde pesquisa na *web* até detecção de fraudes.

A essência do *machine learning* reside na capacidade de adquirir conhecimento a partir de dados, permitindo que as máquinas identifiquem padrões e informações que seriam difíceis de discernir com métodos de análise convencionais (AMARAL, 2016). Isso se assemelha ao processo de aprendizado humano, em que a experiência da vida é o que entrega conhecimento para os humanos. No caso das máquinas, os dados são sua forma de experiência, e é por meio deles que as máquinas aprendem a resolver problemas e aprimorar seu desempenho (SACOMANO et al., 2018).

A capacidade de generalização é um aspecto fundamental do *machine learning*, à medida que os algoritmos procuram criar modelos que sejam aplicáveis a novos dados com precisão. Isso se traduz em uma capacidade de inferir lógica a partir dos dados para obter conclusões genéricas (MONARD; BARANAUSKAS, 2003).

Para que o processo de *machine learning* seja eficaz, é necessário um conjunto de treinamento robusto e uma cuidadosa seleção de variáveis e atributos nos dados. O volume de dados é essencial, uma vez que modelos baseados em *machine learning* dependem de uma ampla gama de exemplos para induzir hipóteses confiáveis (CARVALHO et al., 2011).

Além disso, o pré-processamento desempenha um papel fundamental na preparação dos dados para o *machine learning*. Isso envolve a limpeza de dados defeituosos, mal formatados e irrelevantes, garantindo que o algoritmo funcione com assertividade (CARVALHO et al., 2011).

Os algoritmos de *machine learning* não estão restritos a um único método, eles abrangem várias abordagens e técnicas para se adequarem a diferentes tipos de problemas. As principais categorias incluem aprendizado supervisionado e não supervisionado, com tarefas como classificação, regressão, agrupamento e associação

(CARVALHO et al., 2011). Essas técnicas são fundamentais para a resolução de problemas complexos em diversas áreas, como detecção de fraudes e mecanismos de recomendação.

Conforme a tecnologia avança, as indústrias também evoluem, e o *machine learning* desempenha um papel fundamental na Quarta Revolução Industrial, conhecida como Indústria 4.0 (SACOMANO et al., 2018). A integração de sistemas *cyber-físicos*, a Internet das Coisas e a análise de *Big Data* são elementos centrais desse novo paradigma industrial (PEREIRA; SIMONETTO, 2018).

2.6.1 Aprendizado Supervisionado

O Aprendizado Supervisionado é uma das abordagens mais amplamente adotadas e bem-sucedidas em *machine learning* (MÜLLER; GUIDO, 2016). Esse tipo de abordagem é caracterizado pela presença de um "supervisor" que previamente conhece as respostas desejadas do sistema para determinadas entradas (BISHOP; NASRABADI, 2006).

No Aprendizado Supervisionado, o objetivo é desenvolver um modelo capaz de mapear as variáveis de entrada para as variáveis de saída, utilizando exemplos rotulados. Essa relação entre as variáveis é representada por um modelo construído a partir dos dados disponíveis (MAIMON; ROKACH, 2005).

As variáveis em um problema de Aprendizado Supervisionado podem assumir diversas formas, incluindo variáveis contínuas, categóricas ou binárias, dependendo do contexto do problema (KOTSIANTIS et al., 2007). Para treinar um modelo desse tipo é necessário ter acesso aos dados de referência que definem o padrão desejado para as entradas.

Essa abordagem é essencial para inúmeras aplicações por permitir que as máquinas aprendam a tomar decisões com base em experiências passadas e na orientação fornecida pelo "supervisor". O processo de aprendizado supervisionado envolve a comparação da saída do algoritmo com as respostas conhecidas do supervisor, ajustando o modelo para minimizar os erros entre as previsões do algoritmo e as respostas esperadas (BISHOP; NASRABADI, 2006).

2.6.2 Aprendizado Não Supervisionado

O Aprendizado Não Supervisionado, em contraste com o aprendizado supervisionado, é uma abordagem em *machine learning* que lida com dados desprovidos de rótulos ou variáveis de resposta. Em vez de buscar modelar a relação entre variáveis

preditoras e uma variável resposta, o foco do Aprendizado Não Supervisionado é identificar associações e padrões entre as observações, visando agrupar observações semelhantes (JOHNSON et al., 2002).

Essa abordagem é altamente útil quando se lida com conjuntos de dados que não possuem valores rotulados para orientar o processo de aprendizado da máquina (GROUP, 2023). Em vez disso, a máquina processa e compara os dados por conta própria, ajustando e agrupando os dados quando apropriado.

A detecção de anomalias é outra aplicação valiosa do aprendizado não supervisionado, amplamente utilizada em diversas áreas, como na identificação de falhas em motores elétricos, detecção de fraudes bancárias e problemas médicos (SOARES, 2022). Ela se concentra em identificar eventos que não se conformam com padrões esperados em um conjunto de dados.

No entanto, vale ressaltar que o aprendizado não supervisionado é geralmente mais desafiador do que o supervisionado, por ser mais subjetivo e não contar com um mecanismo universalmente aceito para avaliação e validação dos resultados, uma vez que não há respostas verdadeiras conhecidas nos dados (JAMES et al., 2013).

2.6.3 Aprendizado Por Reforço

O Aprendizado por Reforço é uma abordagem fundamental na área de *Machine Learning* que lida com a tomada de decisões sequenciais em ambientes dinâmicos. Diferentemente do aprendizado supervisionado, onde os dados de treinamento vêm com rótulos específicos, o Aprendizado por Reforço envolve um agente que interage com um ambiente, aprendendo por tentativa e erro (SUTTON; BARTO, 2018).

Em um cenário de Aprendizado por Reforço, o agente está encarregado de selecionar ações para maximizar a recompensa acumulada. O ambiente em que ele opera pode ser desconhecido, e o agente deve explorá-lo para aprender a melhor estratégia de ação (SIGAUD; BUFFET, 2013). Em vez de determinar ações corretas ou incorretas, o agente é recompensado com base em quão próximo ele está de atingir seu objetivo (SUTTON; BARTO, 2018).

Algoritmos desse tipo, como o *Q-learning* (WATKINS, 1989), funcionam com base em uma tabela de *Q-values*, que representa a expectativa de recompensa para cada par de estado e ação. À medida que o agente interage com o ambiente, essa tabela é atualizada com base nas recompensas recebidas. Além disso, algoritmos de Aprendizado por Reforço Profundo representam as *Q-values* como redes neurais, permitindo que lidem com espaços de estados e ações de alta dimensão (MNIH et al.,

2015)

A aplicabilidade do Aprendizado por Reforço se estende a várias áreas, incluindo jogos de alto nível, como o *AlphaGo* e jogos *Atari*, onde demonstrou a capacidade de superar o desempenho humano em ambientes complexos (GRABHER, 2021). Além disso, ele é uma escolha adequada quando os sistemas de decisão são repetitivos e geram grandes volumes de dados de treinamento para otimização.

2.7 ALGORITMOS E TÉCNICAS NA DETECÇÃO DE ANOMALIAS

Essa seção é dedicada a mencionar e explicar os algoritmos e as técnicas utilizadas na produção deste trabalho.

2.7.1 Isolation Forest

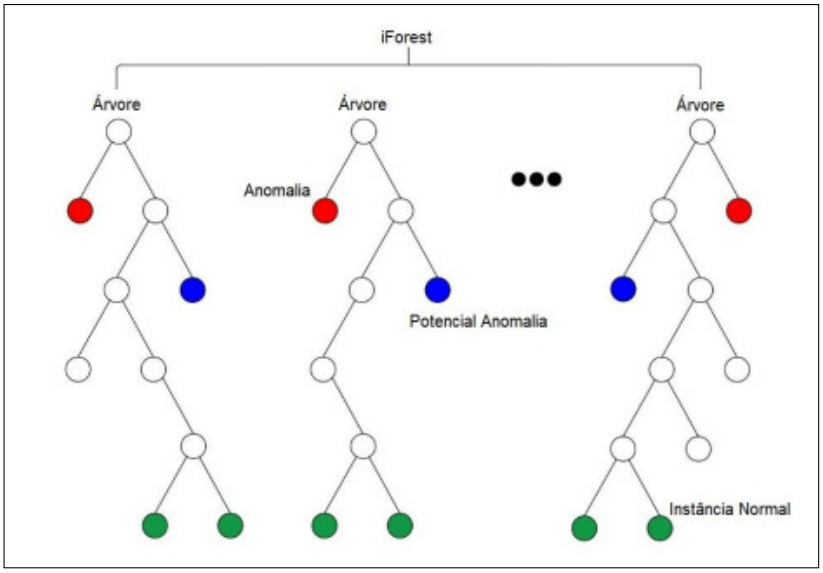
O *Isolation Forest* (*IForest*) é um algoritmo de aprendizado não supervisionado amplamente utilizado na detecção de anomalias, conhecido por seu desempenho eficiente (LIU et al., 2008).

Esse algoritmo se baseia na construção de árvores de decisão e na premissa de que as anomalias são exceções raras e, portanto, estão isoladas das observações normais. Para criar uma árvore no *IForest*, o processo ocorre da seguinte forma: primeiro, são escolhidas aleatoriamente duas características do modelo. Em seguida, um único ponto de dados é selecionado aleatoriamente, e os dados são divididos por um valor aleatório que varia entre os limites máximos e mínimos de uma das características escolhidas (LIU et al., 2008).

As árvores de isolamento são construídas com base na posição do ponto selecionado em relação ao valor de divisão. Quando um ponto de dados se encontra abaixo ou à esquerda da divisão (ou precisamente na linha de divisão), a árvore de decisão se ramifica para a esquerda. Por outro lado, quando o ponto está acima ou à direita da divisão, a árvore de decisão se expande no sentido direito. O número de divisões realizadas pelo algoritmo em uma instância específica é denominado o “comprimento do caminho”.

Conforme observado, é típico que anomalias apresentem um comprimento do caminho mais curto em comparação com as observações normais (LIU et al., 2008). A figura 3 demonstra o funcionamento do *IForest* por meio de um grafo.

Figura 3 – *Isolation Forest* através de árvores



Fonte: (CAVALCANTE, 2022)

A detecção de anomalias com o *IForest* é um processo de duas etapas. Primeiro, no estágio de treinamento, são construídas árvores de isolamento usando subamostras do conjunto de treinamento. Em seguida, no estágio de teste, as instâncias de teste são passadas através das árvores de isolamento para obter uma pontuação de anomalia para cada instância. O resultado é que o *IForest* isola as anomalias em vez de criar padrões entre elas. Caminhos mais curtos indicam anomalias, enquanto caminhos mais longos são característicos das amostras normais (LIMA, 2022).

2.7.2 Robust Covariance

O *Robust Covariance* sendo o *Elliptic Envelope* permite identificar os parâmetros principais de uma grande distribuição geral de dados. Ele baseia-se no princípio de que os dados seguem uma distribuição normal ou Gaussiana multivariada subjacente (ZIEGEL, 2003). De fato, através da aplicação deste método, traça-se uma elipse que englobe a maioria dos dados, com exceção dos candidatos a anomalias; de modo que esses se tornam anômalos. O tamanho e a forma desse círculo são estimados pelo método *FAST-mínimum covariance determinant* (ESTIMATOR, 1999).

O método *FAST-MCD* funciona selecionando subconjuntos de amostras não sobrepostos e calculando a média e a matriz de covariância de cada conjunto em todas as dimensões das variáveis. Em seguida, esses subconjuntos são ordenados conforme a distância de Mahalanobis (MCLACHLAN, 1999), que mede o quão distante um ponto está da média de uma distribuição, em termos do número de desvios padrão.

A equação 1 (MAHALANOBIS, 1936) define matematicamente a distância de Mahalanobis, tendo x como ponto de dado, μ a média da distribuição e C^{-1} é a matriz inversa de covariância.

$$\text{Mahalanobis} = d_M = \sqrt{(x-\mu)^T C^{-1} (x-\mu)} \tag{1}$$

O método *FAST-MCD* repete o processo de selecionar subconjuntos, recalcular médias, covariâncias e distâncias até que a matriz de covariância estabilize. Por fim, o subconjunto cuja matriz de covariância tenha o menor determinante define a elipse final. Essa elipse abrange a maioria dos dados originais, com uma fração menor excluída como *outliers* (AMIR; PRASETYO, 2020).

Os pontos que caem dentro da elipse são rotulados como *"inliers"*, valores normais, enquanto aqueles fora dela são classificados como *"outliers"* ou anomalias. Esses pontos fora do limite podem ser descartados ou tratados separadamente (AMIR; PRASETYO, 2020).

2.7.3 One-Class Support Vector Machine

O algoritmo *One-Class Support Vector Machine* (SCHÖLKOPF et al., 2001), ou *One-Class SVM*, é uma técnica de *machine learning* comumente usada em detecção de anomalias ou classificação quando há apenas uma classe de dados disponível, ou quando há dados altamente desbalanceados. É uma adaptação do SVM binário (RÄTSCH et al., 2000) que lida com problemas onde apenas uma classe de dados é conhecida.

O principal objetivo do *One-Class SVM* é encontrar uma região compacta no espaço de características que envolva a maioria dos pontos de dados da classe conhecida. separando-os da origem ou de um ponto de referência no espaço de características. Essa região é definida por um hiperplano que maximiza a margem entre os dados e a origem (LIU et al., 2024). O *One-Class SVM* se torna útil em cenários onde os dados são desbalanceados, como em detecção de fraudes, diagnóstico médico ou detecção de falhas.

Os dados são mapeados para um espaço de características de alta dimensão usando uma função de kernel, geralmente o kernel Gaussiano (LIU et al., 2011). Esse mapeamento permite que os dados sejam linearmente separáveis no espaço de características.

O *One-Class SVM* tenta encontrar um hiperplano no espaço de características que maximize a margem entre os dados e a origem.

A equação 2 (LIU et al., 2024) define o hiperplano, onde w é o vetor de pesos, $\Phi(x)$ é a função de mapeamento para o espaço de características e ρ é o termo de bias.

$$\text{Hiperplano} = \langle w, \Phi(x) \rangle = \rho \tag{2}$$

O *One-Class SVM* usa a função de perda hinge (SCHÖLKOPF et al., 2001) para penalizar os pontos que estão do lado errado do hiperplano. A função de perda hinge é convexa, facilitando a otimização, mas a torna sensível a *outliers* e ruídos, por penalizar fortemente os pontos mal classificados (LIU et al., 2024).

2.7.4 Local Outlier Factor

O *Local Outlier Factor* (LOF) é um algoritmo de detecção de anomalias que considera a densidade local dos dados para identificar *outliers*. Diferente de métodos tradicionais, como o *Z-score*, que analisam anomalias com base em estatísticas globais, o LOF detecta outliers em relação à densidade da região onde um ponto está inserido. Isso é especialmente útil para conjuntos de dados com clusters de densidades variadas, onde um ponto pode parecer normal em um contexto global, mas ser uma anomalia em seu ambiente imediato (BREUNIG et al., 2000).

Para calcular o LOF, o algoritmo considera alguns conceitos-chave. O primeiro é a k-distância (BREUNIG et al., 2000), que define a distância entre um ponto e seu k-ésimo vizinho mais próximo, delimitando a região ao redor do ponto. Depois, vem a distância de alcance (BREUNIG et al., 2000), que ajusta essa métrica para evitar variações bruscas na vizinhança, garantindo uma análise mais estável. A partir daí, o algoritmo calcula a densidade local de alcance (BREUNIG et al., 2000), que mede o quão densa é a vizinhança de um ponto, considerando as distâncias médias até seus k-vizinhos.

Por fim, o LOF é obtido ao comparar a densidade local de um ponto com a de seus vizinhos. Se o valor de LOF estiver próximo de 1, significa que a densidade do ponto é similar à dos vizinhos, indicando um comportamento normal. Já valores maiores que 1 indicam que o ponto é significativamente menos denso que seus vizinhos, sugerindo que ele seja uma anomalia. Esse método torna o LOF uma abordagem poderosa para detectar outliers em cenários onde a densidade dos dados não é

uniforme (BREUNIG et al., 2000).

2.7.5 Z-Score

O *Z-Score* é uma medida estatística que indica o quão distante um ponto de dados está da média, em termos de desvios padrão. Ele é muito utilizado para detectar valores anômalos, especialmente quando os dados seguem uma distribuição aproximadamente normal (Estatística Fácil, 2024).

A equação 3 (TRIOLA, 2018) representa a fórmula utilizada para calcular o *Z-score*, onde X é o valor que analisaremos, μ representa a média do conjunto de dados e, finalmente, σ é o desvio padrão.

$$\text{Z-Score} = \frac{X - \mu}{\sigma} \tag{3}$$

o *Z-score* ajuda a identificar anomalias ao comparar a posição de um ponto em relação à distribuição dos dados. Se o valor absoluto do *Z-score* $(|Z|)$ for maior que um certo limite, ele pode ser considerado atípico. Normalmente, se $|Z| > 3$, significa que o dado é extremamente raro: menos de 0,3$\%$ dos casos em uma distribuição normal. Já se $|Z| > 2$, o ponto pode ser potencialmente incomum, abrangendo cerca de 5$\%$ dos dados. Essa abordagem simples e eficaz faz do *Z-Score* uma ferramenta amplamente utilizada para detectar padrões fora do esperado.

2.8 MÉTRICAS PARA AVALIAÇÃO DE MODELOS NÃO SUPERVISIONADOS

2.8.1 Percentual de Anomalias

O Percentual de Anomalias é uma métrica que quantifica a proporção de dados ou eventos classificados como anômalos em relação ao total de dados analisados. No contexto de sistemas de detecção de anomalias, essa métrica pode ser usada para (MATOS et al., 2009):

- Avaliar a sensibilidade do sistema, quantas anomalias foram detectadas.
- Verificar a taxa de falsos positivos, quantos dados normais foram erroneamente classificados como anômalos.
- Comparar a eficácia de diferentes algoritmos ou métodos de detecção.

Em Matos et al. (2009), é ressaltado que o percentual de anomalias pode variar conforme o tipo de dado analisado. Em sistemas de segurança, por exemplo,

as anomalias são raras e representam menos de 0,01% dos casos, enquanto no monitoramento industrial, onde falhas e variações são mais comuns, esse percentual pode chegar a 5%.

2.8.2 Silhouette Score

O *Silhouette Score* é uma métrica de avaliação de desempenho usada para avaliar a qualidade de agrupamentos em algoritmos de *clustering*, como o *K-means* (GAMA, 2022). Essa métrica fornece uma medida de quão bem um objeto foi classificado em seu próprio *cluster* em comparação com outros *clusters*.

O Silhouette Score é calculado para cada um dos pontos de dados e tem uma variação entre -1 e 1 (JANUZAJ et al., 2023):

- **Valor próximo de 1:** O ponto está bem posicionado dentro de seu cluster, distante dos outros clusters.
- **Valor próximo de 0:** O ponto está na fronteira entre dois clusters, ou seja, pode pertencer a ambos.
- **Valor negativo -1:** O ponto pode ter sido mal classificado em seu cluster, por estar mais próximo de um cluster diferente.

A equação 4 (ROUSSEEUW, 1987) representa o *Silhouette Score*, i é um ponto de dado, $a(i)$ calcula a distância média entre o ponto i e todos os outros pontos do mesmo cluster, $b(i)$ determina a menor distância média entre o ponto i e todos os pontos de qualquer outro cluster ao qual i não pertence.

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{4}$$

Geralmente, calculamos o *Silhouette Score* médio avaliando cada ponto de dados individualmente e tirando uma média dos resultados obtidos para todos os pontos analisados no conjunto de dados em questão.

Um resultado elevado sugere que os grupos estão claramente definidos e separados entre si, uma indicação positiva da eficácia do algoritmo de agrupamento utilizado no processo de análise dos dados. Por outro lado, e, em contrapartida, ao resultado anteriormente mencionado, um valor baixo pode sugerir que os grupos possam estar se sobrepondo ou que o modo como eles foram configurados não é o

mais ideal para representar adequadamente as relações existentes entre os dados analisados. (JANUZAJ et al., 2023)

2.8.3 Teste de Kolmogorov-Smirnov

O teste de *Kolmogorov-Smirnov* é um teste estatístico não paramétrico utilizado para comparar distribuições de probabilidade. Ele pode ser aplicado de duas formas (OHUNAKIN et al., 2024):

- KS unidirecional *one-sample*: Compara uma amostra de dados com uma distribuição teórica de referência, como a normal ou exponencial.
- KS bidirecional *two-sample*: Compara duas amostras diferentes para verificar se elas seguem a mesma distribuição.

O teste KS calcula uma estatística chamada D, que representa a maior diferença entre as funções de distribuição acumulada das amostras comparadas. Quanto maior esse valor, maior a discrepância entre as distribuições (OHUNAKIN et al., 2024).

Esse teste é útil para avaliar se os escores de anomalia atribuídos por um algoritmo seguem distribuições distintas entre dados normais e anômalos. A ideia é que, se o modelo for eficaz, os escores dos dados normais devem estar concentrados em valores baixos próximos de 0, enquanto os dados anômalos devem ter escores mais altos próximos de 1. O teste KS mede a diferença entre essas duas distribuições, e um valor de D próximo de 1 indica que o algoritmo separa bem os dados normais das anomalias, enquanto um D próximo de 0 sugere que o modelo não consegue diferenciá-los corretamente (OHUNAKIN et al., 2024).

De acordo com Ohunakin et al. (2024):

- O algoritmo atribui um score a cada ponto de dado, indicando a probabilidade de ele ser uma anomalia.
- Os dados são divididos em dois grupos: um contendo os pontos considerados normais e outro com aqueles suspeitos de serem anômalos.
- As CDFs dos dois grupos são comparadas:
  - O teste KS calcula a maior diferença entre essas distribuições.
  - Se a diferença for pequena, $D \approx 0$, significa que ambos os grupos têm distribuições semelhantes, indicando que o modelo não separa bem as anomalias.

– Se a diferença for grande, $D \approx 0$, significa que os grupos seguem distribuições bem distintas, indicando um bom desempenho na detecção de anomalias.

2.9 TECNOLOGIAS UTILIZADAS

Para desenvolver este trabalho, o *Python* foi escolhido como a linguagem de programação para a análise de dados. O processo foi aprimorado com o uso da biblioteca Pandas, que facilitou a organização e manipulação dos dados, e pela *NumPy*, que auxiliou nas operações numéricas. Para métodos matemáticos avançados e análise estatística, utilizou-se a *SciPy*. Já a aplicação dos algoritmos de aprendizado de máquina ficou a cargo da *Scikit-learn*, e o *Matplotlib* foi essencial para criar gráficos que tornaram os resultados mais claros e fáceis de interpretar. Juntas, essas ferramentas tornaram todo o processo mais eficiente, desde a implementação dos algoritmos até a visualização final dos resultados.

2.9.1 Python

O *Python* é uma linguagem de programação muito utilizada na análise de dados, graças à sua simplicidade, versatilidade e grande variedade de bibliotecas. A linguagem tem um papel fundamental no processamento de grandes volumes de dados, sendo uma das principais escolhas em projetos de big data e ciência de dados. Além disso, sua capacidade de trabalhar com diferentes tipos de dados e se integrar facilmente com outras tecnologias a torna ainda mais eficiente e prática (SIEGEL, 2018).

O *python* possui diversas ferramentas voltadas para a análise de dados, dentre elas estão algumas principais como as bibliotecas, Pandas, usado para organizar e manipular tabelas de dados, *NumPy* essencial para operações numéricas, *Matplotlib* e *Seaborn* que facilitam a criação de gráficos e visualizações. O estudo de Kabir e Ahmed (2024) demonstra como essas ferramentas tornam o processo de explorar, transformar e visualizar os dados mais rápido e eficiente, contribuindo para análises mais ágeis e precisas.

Por ter uma sintaxe simples e intuitiva, o *python* permite que até iniciantes realizem análises complexas. Guimaraes (2024) ressalta a importância de bibliotecas como o *Scikit-Learn*, muito usada em projetos de aprendizado de máquina, e o *Statsmodels*, voltado para análises estatísticas, enfatizando o papel do *Jupyter Notebooks* como uma plataforma interativa que facilita o desenvolvimento de projetos de análise de dados, permitindo combinar código, gráficos e relatórios em um só lugar.

2.9.2 Bibliotecas

A biblioteca Pandas é uma das ferramentas mais populares para quem trabalha com dados em *python*. Ele oferece estruturas como *DataFrames* e Series, que tornam mais simples organizar e analisar informações em formato de tabela (REZEK, 2024). Com ela, fica simples ler, limpar, transformar e agrupar dados de maneira eficiente. Quando o volume de dados é muito grande, usar técnicas como dividir o processamento em partes menores ou aplicar operações em lote ajuda a manter o desempenho rápido e fluido (YON; MARJORAM, 2019).

NumPy é uma biblioteca fundamental para quem trabalha com ciência de dados e computação científica em *python*. Ela facilita operações com *arrays* multidimensionais e cálculos matemáticos de forma rápida e eficiente (VIRTANEN et al., 2020). O projeto é mantido por uma comunidade ativa e colaborativa, garantindo sua evolução constante e seu uso em áreas como inteligência artificial, simulações e análise de dados (MOSS, 2019).

O Scikit-learn é uma das bibliotecas mais usadas para aprendizado de máquina em Python, conhecida por sua interface simples e fácil de usar. Com ela, é possível implementar algoritmos clássicos como SVMs, Random Forests e Regressão Linear de forma prática e eficiente (PEDREGOSA et al., 2011). Seu design modular permite que usuários iniciantes e avançados desenvolvam modelos de maneira eficiente. Na obra de Géron (2022), a biblioteca é abordada tanto em conceitos básicos, como classificação e regressão, até técnicas mais avançadas, como *ensembles* e *Gradient Boosting*.

Plotly é uma biblioteca do *python* muito usada para criar gráficos interativos e visualizações de dados de alta qualidade. Ela funciona com base em *JavaScript*, o que permite gerar gráficos dinâmicos que podem ser exibidos diretamente no navegador e integrados facilmente ao *Jupyter Notebook*. Comparado a outras ferramentas, como *Bokeh* e *Altair*, o *plotly* se destaca por sua facilidade de uso e pela simplicidade em criar gráficos interativos com poucas linhas de código (XIAO et al., 2023).

Focado em visualização, o *Matplotlib* é uma biblioteca indispensável para visualização de dados em *python*, muito usada pela sua flexibilidade e capacidade de personalização. Inspirada no *MATLAB*, ela permite criar desde gráficos simples até visualizações interativas e publicações de alta qualidade (HUNTER, 2007). Comparado a outras ferramentas, como *Seaborn* e *Plotly*, o *Matplotlib* se destaca pelo controle detalhado dos elementos visuais, sendo a escolha favorita em ambientes acadêmicos e científicos (ZHAO et al., 2023). Além de gráficos básicos, como linhas e barras, a biblioteca também suporta animações e pode ser integrada a interfaces gráficas, como

o *PyQt*, tornando as aplicações mais dinâmicas e interativas (MCGREGGOR, 2015).

O *SciPy* é uma biblioteca essencial para quem trabalha com computação científica em *python*, especialmente em análises estatísticas. Seu módulo *stats* oferece diversas ferramentas, como distribuições estatísticas, testes de hipóteses e cálculos de medidas descritivas (HOFMANN et al., 2001). Comparado a pacotes do *R*, como stats e lme4, o SciPy se destaca pela integração com outras bibliotecas do ecossistema Python e pelo bom desempenho em grandes volumes de dados (EROSHEVA et al., 2022). Com o tempo, a biblioteca evoluiu e passou a incluir funções mais avançadas, como testes não paramétricos e cálculos de correlação, tornando-se uma ferramenta indispensável tanto para pesquisadores quanto para profissionais de ciência de dados (VIRTANEN et al., 2020).

2.9.3 Jupyter Notebook

O *Jupyter Notebook* é uma ferramenta interativa muito usada para análise de dados, permitindo combinar código, explicações escritas e gráficos em um único lugar. Ele surgiu a partir do projeto *IPython*, que trouxe a ideia de um ambiente onde o código pode ser testado e visualizado de forma dinâmica (PÉREZ; GRANGER, 2007). Com o tempo, o Jupyter evoluiu para suportar várias linguagens de programação, sendo o Python a mais popular. A estrutura de blocos facilita a execução e modificação de pequenos trechos de código, tornando o processo mais ágil e organizado.

No contexto da análise de dados, o *Jupyter Notebook* brilha ao integrar bibliotecas poderosas como *Pandas*, *NumPy* e *Matplotlib*, permitindo importar, limpar, transformar e visualizar informações em tempo real (DOMBROWSKI et al., 2023). Além disso, a opção de incluir explicações em formato Markdown torna o documento mais didático, facilitando a comunicação dos resultados. Seja para projetos acadêmicos ou para o mercado, o Jupyter se destaca por sua interface simples e o formato compartilhável, que torna o trabalho colaborativo e acessível a todos.

3 METODOLOGIA

A metodologia deste trabalho foi organizada em etapas que ajudaram a guiar todo o processo de desenvolvimento do projeto. Tudo começou com a definição da ideia, onde foi escolhido o tema e definido o que seria abordado no estudo. Em seguida, partimos para a construção do referencial teórico, reunindo e estudando materiais que ajudaram a entender melhor o que já foi pesquisado sobre detecção de anomalias e como isso se aplica ao mercado de criptomoedas.

Com a base teórica pronta, foi realizada a coleta dos dados, buscando informações históricas sobre o preço de criptomoedas como Bitcoin, Ethereum e Litecoin, que seriam usadas ao longo do trabalho. Depois disso, veio a seleção dos algoritmos, onde foram escolhidos modelos de aprendizado de máquina conhecidos por sua eficiência na detecção de anomalias, como o Isolation Forest e o DBSCAN.

Na etapa seguinte, foi feito o treinamento dos modelos, usando os dados coletados para ensinar os algoritmos a identificar padrões incomuns. Com os resultados obtidos, foram criadas visualizações para facilitar a análise, usando gráficos e representações visuais que tornaram as informações mais claras e acessíveis. Por fim, todas essas etapas foram registradas na construção do documento, reunindo os detalhes do processo, os resultados alcançados e as conclusões do estudo.

3.1 BASE DE DADOS

A plataforma *Investing* (INVESTING, 2023) foi a base de dados escolhida para coletar os dados a serem utilizados durante o processo de produção deste trabalho. A investing.com, em si, trata-se de uma plataforma de mercado financeiro que está disponível globalmente, dispondo de inúmeros serviços em 44 idiomas diferentes. Com aproximadamente 21 milhões de usuários mensais, de acordo com Similarweb (2023) o site é destaque, ocupando a 4ª posição no ranking de sites de finanças mais visitados. Fundada em 2007, a investing.com evoluiu ao ponto de se tornar uma ferramenta confiável para *traders* e investidores, onde há a possibilidade de obter dados em tempo real, cotações, gráficos e notícias de última hora. Abrangendo bolsa de valores, *commodities*, criptomoedas, entre outros (INVESTING, 2023).

A escolha da *investing* como base de dados se deu por motivos de confiabilidade e gratuidade, por se tratar de uma plataforma que dá acesso gratuito e ilimitado a ferramentas avançadas do mercado financeiro, incluindo uma base de dados sólida sobre as criptomoedas abordadas neste trabalho, já a questão da confiabilidade se

dá devido ao enorme número de usuários espalhados pelo mundo que utilizam a plataforma. E, por fim, a facilidade com a qual os dados são retirados da plataforma, sendo somente necessário para isso selecionar uma data de início e de término da série temporal a ser analisada e fazer o download dos dados em formato CSV.

Os dados retirados e utilizados da plataforma foram dados históricos diários das criptomoedas utilizadas para desenvolver o trabalho que contém as seguintes informações agrupadas em colunas (SENNA; SOUZA, 2023):

- Preço de Abertura (**Abertura**): Valor da criptomoeda no início do dia.
- Preço de Fechamento (**Último**): Valor registrado no final do dia.
- Preço Máximo (**Máxima**): Maior valor atingido ao longo do dia.
- Preço Mínimo (**Mínima**): Menor valor registrado durante o dia.
- Volume de Negociação (**Vol.**): Quantidade total negociada no período.
- Variação Percentual Diária (**Var%**): Percentual de alteração no preço em relação ao dia anterior.

A figura 4 mostra uma parte do conjunto de dados do *Bitcoin* após as transformações, com informações históricas sobre preços e volumes de negociação após o pré-processamento. O *dataset* inclui dados como data, preço de fechamento (último), preço de abertura, máximos e mínimos diários, volume negociado e a variação percentual (Var%). Esse exemplo ilustra como os dados estão organizados e destaca a volatilidade do *Bitcoin*, com variações percentuais expressivas em curtos períodos. Além do *Bitcoin*, o trabalho também usa dois outros conjuntos de dados, um para o *Ethereum* e outro para o *Litecoin*, que têm os mesmos parâmetros. Isso permite uma análise comparativa entre essas criptomoedas, oferecendo uma visão mais completa do comportamento do mercado, levando em conta as características de cada ativo o dataset está organizado em ordem decrescente de acordo com a coluna da data.

Figura 4 – Dataframe *Bitcoin*

	Data	Último	Abertura	Máxima	Mínima	Vol.	Var%
0	2022-01-01	47738.0	46217.5	47917.6	46217.5	31240.0	3.29
1	2021-12-31	46219.5	47123.3	48553.9	45693.6	58180.0	-1.92
2	2021-12-30	47123.3	46470.7	47901.4	46003.0	60960.0	1.42
3	2021-12-29	46461.7	47548.4	48121.7	46127.8	63920.0	-2.28
4	2021-12-28	47545.2	50703.4	50703.8	47345.7	74390.0	-6.18

Fonte: Elaborado pelo autor

Com os dados que foram utilizados retirados da plataforma *Investing* os mesmos foram submetidos às respectivas transformações de dados para se adequar ao contexto do trabalho. Onde foram criados gráficos para facilitar a compreensão do histórico de dados. Os notebooks utilizados para as transformações estão disponíveis no Github [1]

A figura 5 mostra um gráfico de *candlestick* que acompanha a cotação do *Bitcoin* em dólares entre janeiro de 2019 e janeiro de 2022. As velas vermelhas indicam dias em que o preço de fechamento foi menor do que o de abertura, sinalizando queda, enquanto as velas verdes representam dias de valorização. É possível notar um crescimento expressivo no preço do Bitcoin a partir do final de 2020, atingindo seu maior valor histórico em 2021, seguido por um período de forte volatilidade, com oscilações constantes. Essa visualização ajuda a entender as variações no mercado de criptomoedas e destaca o comportamento imprevisível do Bitcoin ao longo do tempo.

Figura 5 – Cotação do *Bitcoin* ao longo dos anos a serem testados nos modelos (2019 – 2022)

Fonte: Elaborado pelo autor

A figura 6 mostra um gráfico de *candlestick* que acompanha a cotação do *Ethereum* em dólares entre janeiro de 2019 e janeiro de 2022. Assim como o *Bitcoin* 5, o gráfico exibe períodos de valorização e queda, representados por velas verdes e

[1] https://github.com/FThierryDev/TCC.git

vermelhas, respectivamente. Até meados de 2020, o preço do *Ethereum* se manteve relativamente estável, com variações moderadas. No entanto, a partir do final de 2020, a criptomoeda passou por um crescimento significativo, seguindo uma tendência parecida com a do *Bitcoin* e atingindo seu pico histórico em 2021, quando superou os 4.000 dólares. Apesar dessa semelhança, o *Ethereum* apresentou oscilações ainda mais intensas em certos momentos, possivelmente influenciadas por fatores próprios do seu ecossistema, como atualizações na rede e o aumento do uso de contratos inteligentes, que impactaram diretamente sua volatilidade no mercado.

Figura 6 – Cotação do *Ether* ao longo dos anos a serem testados nos modelos (2019 – 2022)

Fonte: Elaborado pelo autor

A figura 7 mostra um gráfico de candlestick que representa a cotação do *Litecoin* em dólares entre janeiro de 2019 e janeiro de 2022. Assim como nos gráficos anteriores 5 6, as velas verdes indicam momentos de valorização, enquanto as vermelhas representam períodos de queda. Até 2020, o *Litecoin* teve um crescimento moderado em comparação com as outras criptomoedas, com variações menos intensas. No entanto, a partir do final de 2020, ele seguiu a mesma tendência de forte valorização observada no *Bitcoin* e no *Ethereum*, atingindo seu pico em 2021, quando ultrapassou 400 dólares. Apesar desse movimento semelhante, o *Litecoin* se mostrou ainda mais volátil, registrando picos de alta e quedas bruscas em curtos períodos. Essa instabilidade pode estar relacionada à sua menor adoção institucional e à maior influência do mercado especulativo sobre seu preço.

Figura 7 – Cotação do *Litecoin* ao longo dos anos a serem testados nos modelos (2019 – 2022)

Fonte: Elaborado pelo autor

Ao observar os dados de 2020 e 2021, fica claro que houve grandes oscilações nas criptomoedas analisadas, com momentos de forte valorização seguidos por quedas bruscas. No entanto, cada uma delas se comportou de maneira diferente: o *Bitcoin* teve variações mais controladas, enquanto o *Ethereum* e o *Litecoin* passaram por períodos de volatilidade ainda mais intensos. Apesar dessas diferenças, foi apenas com a aplicação dos modelos que conseguimos identificar anomalias específicas em cada uma das séries históricas, revelando padrões atípicos que não eram tão evidentes à primeira vista nos gráficos.

3.2 PRÉ-PROCESSAMENTO DOS DADOS

Antes da aplicação das técnicas de detecção de anomalias propostas, foi necessária a realização de uma preparação com os dados utilizados, visando a garantia da interpretação correta dos dados por parte dos métodos de detecção utilizados para a realização do experimento. O processo se deu por etapas, sendo elas 3, conversão de tipos, normalização e a divisão dos dados.

Durante a etapa de conversão de tipos, as datas foram convertidas para o formato *datetime*, e os valores numéricos passaram por ajustes, como a remoção de caracteres indesejados, como pontos e vírgulas. Além disso, valores no formato "K"(milhares) e "M"(milhões) na coluna de volume foram convertidos para seus equivalentes numéricos (AMARAL et al., 2017).

O Algoritmo 1 apresenta o tratamento realizado nos dados utilizados para o desenvolvimento desse trabalho. Utilizando o bitcoin como exemplo, foi realizado um processo de pré-processamento detalhado.

- Ajustando o formato das datas para que fossem reconhecidas corretamente.
- Padronizando os valores de preço como último, abertura, máxima e mínima, transformando números com vírgulas e pontos em valores decimais simples.
- Para o volume de negociação, que aparecia com abreviações como *K* para mil e *M* para milhão, foi desenvolvido um método para converter os símbolos em números inteiros multiplicando por *K* para 1.000 ou *M* para 1.000.000.
- Ajuste da coluna de variação percentual, removendo o símbolo *%* e substituindo vírgulas por pontos, além de preencher eventuais dados faltantes com zero.

Algoritmo 1 – Conversão de Tipos

```
# Convertendo para os tipos de dados corretos
dados_bitcoin['Data'] = pd.to_datetime(dados_bitcoin['Data'], format="%d.%m.%Y")

# Conversao de colunas numericas (substituindo separadores de milhares e decimais)
dados_bitcoin['Último'] = pd.to_numeric(dados_bitcoin['Último'].str.replace('.', '').str.replace(',', '.'))
dados_bitcoin['Abertura'] = pd.to_numeric(dados_bitcoin['Abertura'].str.replace('.', '').str.replace(',', '.'))
dados_bitcoin['Máxima'] = pd.to_numeric(dados_bitcoin['Máxima'].str.replace('.', '').str.replace(',', '.'))
dados_bitcoin['Mínima'] = pd.to_numeric(dados_bitcoin['Mínima'].str.replace('.', '').str.replace(',', '.'))

# Função para converter valores com 'K' (milhares) e 'M' (milhões)
def converter_k_e_m_para_numero(value):
    value = value.replace(',', '.')  # Substitui vírgula por ponto decimal
    if 'K' in value:
        return float(value.replace('K', '')) * 1000
    elif 'M' in value:
        return float(value.replace('M', '')) * 1000000
    else:
        return float(value)

# Aplicando a conversão na coluna de volume
dados_bitcoin['Vol.'] = dados_bitcoin['Vol.'].apply(converter_k_e_m_para_numero)

# Conversão da variação percentual, removendo '%' e tratando valores nulos
dados_bitcoin['Var%'] = dados_bitcoin['Var%'].str.replace(',', '.').str.rstrip('%').astype('float')
dados_bitcoin['Var%'].fillna(0, inplace=True)
```

Fonte: Elaborado pelo autor

A etapa da normalização garante que todas as variáveis sejam comparáveis, os dados foram escalonados utilizando o *MinMaxScaler*, que transforma os valores para um intervalo comum, entre 0 e 1 (JAISWAL, 2024).

Na divisão dos dados, 80% foram usados para treinamento e 20% para teste, mantendo a ordem temporal original, sem embaralhar os eventos, para preservar a sequência natural no tempo (RUPPERT, 2004). A escolha dos hiperparâmetros e a forma como os dados foram separados seguiram as diretrizes do estudo Gonçalves

(2023), com algumas pequenas adaptações para se ajustarem melhor ao conjunto de dados utilizado.

3.3 APLICAÇÃO DAS TÉCNICAS DE DETECÇÃO DE ANOMALIAS

Para treinamento dos modelos, excluindo o *Z-Score* por se tratar de um método estatístico, foi feita a divisão dos dados coletados a partir da base de dados em dois tipos, dados de treinamento (80%) e dados de teste (20%) (RUPPERT, 2004), onde os dados de treinamento são aqueles que foram utilizados para o modelo poder aprender o que é “normal” nessa abordagem de anomalias em séries históricas de criptomoedas e os de teste foram os dados utilizados para conferir a eficácia dos modelos.

O método *Robust Covariance* (EllipticEnvelope) assume que os dados seguem uma distribuição normal Gaussiana e utiliza uma técnica chamada covariância robusta para identificar anomalias. Essa abordagem cria uma elipse ao redor dos dados considerados normais, considerando a dispersão deles. Se um ponto estiver fora desse limite, ele é classificado como uma anomalia.

Já o *One-Class SVM* define um limite ao redor dos dados considerados "normais". Ele utiliza uma técnica chamada *kernel* para transformar os dados em um espaço de alta dimensão, onde a maioria dos pontos normais fica separada da origem. Dessa forma, qualquer ponto que fuja desse padrão pode ser identificado como uma anomalia.

O *Isolation Forest* usa árvores de decisão para separar os pontos do conjunto de dados de forma aleatória. Como as anomalias costumam ser diferentes do restante dos dados, elas são isoladas com menos divisões, ou seja, em menos passos na árvore. O algoritmo mede a quantidade média de divisões necessárias e, com base nisso, identifica os pontos que provavelmente são *outliers*.

O algoritmo *Local Outlier Factor* (LOF) analisa o quão denso é um determinado ponto em relação aos seus vizinhos. Se um ponto estiver em uma região menos densa do que os demais ao seu redor, ele é classificado como uma possível anomalia.

Por fim, o *Z-score* se tratando do único método estatístico utilizado, mede o quão distante um valor está da média de um conjunto de dados, em termos de desvios padrão. Quanto maior o *Z-score*, mais o ponto se diferencia do padrão geral, podendo ser considerado uma anomalia.

O Algoritmo 2 demonstra o passo a passo para a aplicação dos métodos para a análise, foram escolhidas seis variáveis financeiras essenciais, Último, Abertura, Máxima, Mínima, Vol e Var%. Essas variáveis ajudam a entender a dinâmica do

mercado, incluindo volatilidade, liquidez e tendências, sendo fundamentais para detectar comportamentos fora do comum.

Algoritmo 2 – Aplicação dos metodos

```
    # Selecionar as colunas de interesse para a detecção de anomalias
cols_in = ['Último', 'Abertura', 'Máxima', 'Mínima', 'Vol.', 'Var%']

    # Definição dos métodos de detecção de anomalias
anomaly_algorithms = {
    "Robust covariance": EllipticEnvelope(),
    "One-Class SVM": svm.OneClassSVM(nu=0.15, kernel="rbf", gamma
       =0.1),
    "Isolation Forest": IsolationForest(random_state=42),
    "Local Outlier Factor": LocalOutlierFactor(n_neighbors=35,
       novelty=True),
    "Z-score": stats.zscore
}

    # Normalizando os dados
scaler = MinMaxScaler()
train_scaled = scaler.fit_transform(train_df[cols_in])
test_scaled = scaler.transform(test_df[cols_in])

    # Treinamento e aplicação dos algoritmos
results = {}
for name, algorithm in anomaly_algorithms.items():
    if name == "Local Outlier Factor":
        algorithm.fit(train_scaled)
        y_pred = algorithm.predict(test_scaled)
        y_pred = y_pred == -1
    elif name == "Z-score":
        y_pred = [np.abs(stats.zscore(test_scaled[:, i])) > 1.5 for i
            in range(test_scaled.shape[1])]
        df_zscore = pd.DataFrame(y_pred).T
        y_pred = df_zscore.any(axis=1).values
    else:
        algorithm.fit(train_scaled)
        y_pred = algorithm.predict(test_scaled)
        y_pred = y_pred == -1

    results[name] = y_pred
```

Fonte: Elaborado pelo autor

Para as variáveis poderem ser comparadas justamente, foi utilizada a técnica de normalização com *MinMaxScaler*, do *scikit-learn*. Esse método ajusta os valores para ficarem no intervalo de 0 a 1. A normalização foi feita separadamente para os

conjuntos de treino e teste, garantindo que os dados do teste não fossem influenciados por informações do treino, evitando assim qualquer vazamento de dados.

Para cada método, excluindo o *Z-Score*, que por se tratar de um método estatístico passa diretamente pela fase de testes, dispensando o treinamento, o processo segue duas etapas:

1. **Treinamento**: O método aprende o padrão dos dados normais a partir do conjunto de treino normalizado, excluindo o Z-Score que não precisa da fase de treinamento.
2. **Predição**: O método analisa os dados do conjunto de teste e identifica possíveis anomalias.

3.4 AVALIAÇÃO DOS MÉTODOS

Para validação, os resultados obtidos pelos métodos foi feito o uso de métricas de avaliação, sendo elas, Percentual de Anomalias, *Silhouette Score* e *Teste de Kolmogorov-Smirnov (KS)*, sendo estas métricas amplamente utilizadas na avaliação de modelos de *machine learning* e, para facilitar a visualização dos resultados, foram utilizados gráficos.

O Percentual de Anomalias mostra a quantidade de dados considerados fora do padrão em relação ao total analisado. Esse número pode variar dependendo do algoritmo usado e das características dos dados. No mercado financeiro, um percentual baixo geralmente indica eventos raros e importantes, enquanto um percentual mais alto pode refletir uma maior volatilidade (CUNHA, 2024).

O *Silhouette Score* é uma métrica usada para avaliar a qualidade dos grupos formados em algoritmos de aprendizado não supervisionado. Ele verifica o quão bem cada ponto se encaixa em seu próprio grupo em comparação com outros grupos. O valor varia de -1 a 1: quanto mais próximo de 1, mais bem definidos e separados estão os grupos, valores próximos de 0 indicam que os grupos estão sobrepostos, e valores negativos sugerem que alguns pontos foram classificados no grupo errado (JANUZAJ et al., 2023).

O Teste de *Kolmogorov-Smirnov* é uma técnica estatística usada para comparar se uma amostra segue uma distribuição específica ou se duas amostras têm distribuições semelhantes. Ele mede a maior diferença entre as curvas de distribuição acumulada dos dados analisados.

O resultado do teste inclui o p-value (MIOLA; MIOT, 2021), que indica a probabilidade de observar essa diferença caso as distribuições sejam iguais. Se o p-value for menor que um limite pré-definido, significa que há uma diferença significativa entre as distribuições.

O Algoritmo 3 demonstra a aplicação das métricas de avaliação nos métodos utilizados no trabalho para a detecção de anomalias, onde foi utilizado um laço de repetição para realizar um loop pelas métricas em cada um dos algoritmos por meio de um dicionário com a chave sendo o nome do algoritmo e um *array* com as anomalias detectadas pelo mesmo para aplicar as métricas nos dados recebidos.

Algoritmo 3 – avaliação dos metodos

```
    # Avaliação dos algoritmos
p_values = {}
p_stats_ = {}
percent = {}
silhouette_scores = {}

for name, y_pred in results.items():
    # Percentual de anomalias
    percent[name] = np.mean(y_pred)

    # Silhouette score para amostras não-anômalas
    labels = [1 if not i else -1 for i in y_pred]
    if len(np.unique(labels)) > 1:
        silhouette_scores[name] = silhouette_samples(test_scaled,
           labels).mean()
    else:
        silhouette_scores[name] = 0

    # Kolmogorov-Smirnov test
    normal = test_df[~y_pred]
    anomaly = test_df[y_pred]
    for col in cols_in:
        if not normal.empty and not anomaly.empty:
            stats_, pvalue = stats.ks_2samp(normal[col], anomaly[col
               ])
            p_values[f"{col}_{name}"] = pvalue
            p_stats_[f"{col}_{name}"] = stats_
```

Fonte: Elaborado pelo autor

Para tornar as informações do projeto mais fáceis de compreender, foram utilizados três tipos de gráficos: *candlestick*, dispersão e barras. O gráfico de *candlestick* mostra como os preços variam ao longo do tempo, destacando valores como abertura,

fechamento, máxima e mínima. O gráfico de dispersão auxilia na identificação de padrões e relações entre variáveis, enquanto o gráfico de barras permite comparar quantidades de forma simples e clara. Juntos, esses gráficos tornam a análise dos dados mais visual e intuitiva.

O gráfico de *candlestick*, ou gráfico de velas, é uma ferramenta visual utilizada na análise técnica de mercados financeiros para representar variações de preços de ativos em um determinado período. Sua popularidade deve-se à capacidade de sintetizar informações complexas visualmente, auxiliando *traders* na interpretação do sentimento do mercado (FERNANDES et al., 2015).

O gráfico de dispersão, ou diagrama de dispersão, é uma forma visual de mostrar a relação entre duas variáveis numéricas. Cada ponto no gráfico representa um par de valores, o que permite observar como uma variável varia em relação à outra. Esse tipo de gráfico é muito útil para identificar padrões, tendências e possíveis conexões entre os dados analisados (MARTINS, 2014).

O gráfico de barras é uma forma visual simples e eficaz de comparar diferentes categorias ou grupos de dados. Ele usa barras retangulares, cujo tamanho varia conforme o valor que representam, facilitando a análise e a comparação das informações. Esse tipo de gráfico é muito utilizado para mostrar dados qualitativos ou quantitativos discretos, sendo indispensável em áreas como educação, estatística e pesquisa científica (MARTINS, 2018).

Todos os gráficos criados serão apresentados no Capítulo 4 que aborda os resultados deste trabalho.

4 RESULTADOS

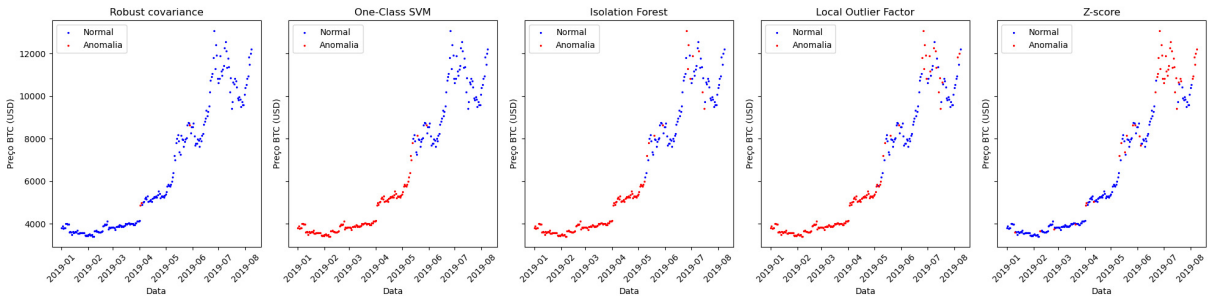
Neste capítulo são apresentados os resultados obtidos a partir das métricas de avaliação de desempenho obtidas pelos experimentos feitos com métodos de detecção de anomalias usando os dados históricos das criptomoedas, Bitcoin, Ethereum e Litecoin.

4.1 ANOMALIAS ENCONTRADAS

Cada método se comportou de formas diferentes, considerando a criptomoeda em que era aplicado, tendo em vista que a volatilidade da criptomoeda não tem um parâmetro pré-definido, ou seja, o *bitcoin* poderia estar perdendo valor enquanto o *ethereum* ganhava valor. Outro ponto a ser considerado é a diferença nos valores das criptomoedas, o *bitcoin* é a moeda com maior valor utilizado, fazendo com que o comportamento que os métodos apresentaram para ele não seja o mesmo que apresentaram para as outras criptomoedas.

A figura 8 demonstra os resultados das aplicações das técnicas nos dados do Bitcoin por meio de um gráfico de dispersão onde as amostras consideradas normais são azuis e as consideradas anômalas são destacadas em vermelho. Visivelmente no gráfico, o *Robust Covariance* tem um número de anomalias destacadas extremamente baixo em comparação com os outros métodos. Isso ocorre devido à natureza das criptomoedas de não seguir um padrão fixo e previsível em conjunto com a configuração dos hiperparâmetros, fazendo com que o *robust couvariance* acabe considerando grande parte dessas variações como normais, deixando de marcar muitos pontos como anômalos porque o algoritmo assume que os dados seguem uma distribuição gaussiana (AMIR; PRASETYO, 2020).

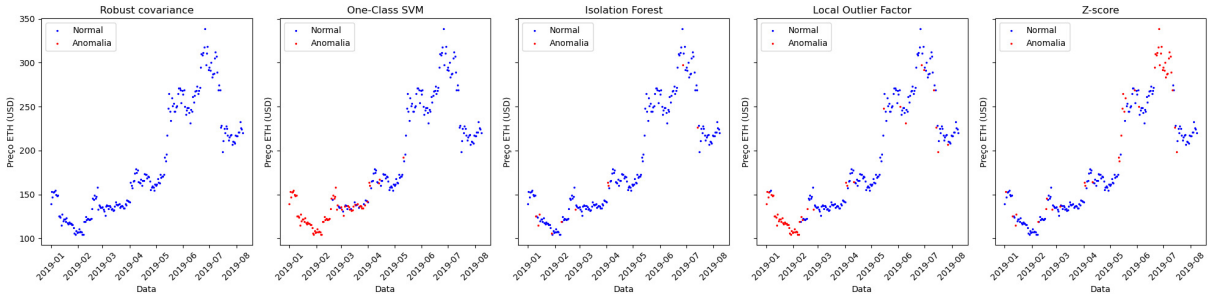
Figura 8 – Gráfico de dispersão resultado dos métodos aplicados no Bitcoin



Fonte: Elaborado pelo autor

A figura 9 representa o gráfico de dispersão construído com a aplicação das técnicas, assim como no *bitcoin*. As amostras em azul são amostras normais e as vermelhas, anomalias. Novamente, o comportamento do *robust couvariance* se demonstra ineficiente para detecção nos experimentos testados nesse trabalho em padrões que demonstram grande volatilidade como o mercado de criptomoedas, só que, diferentemente dos resultados com o *bitcoin* no *ethereum* o algoritmo não conseguiu identificar uma anomalia sequer.

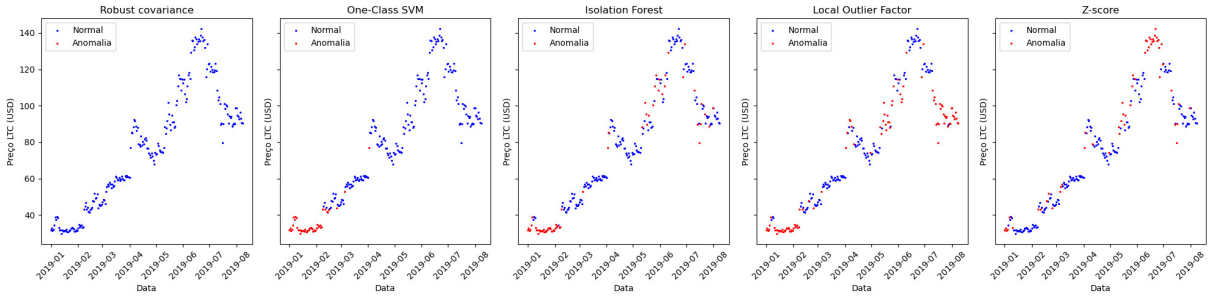
Figura 9 – Gráfico de dispersão resultado dos métodos aplicados no Ethereum



Fonte: Elaborado pelo autor

A figura 10, por fim, apresenta o mesmo gráfico utilizado nas outras duas criptomoedas, onde amostras em azul são normais e em vermelho anomalias. por fim o último também demonstra que o *robust couvariance* pode não ser o algoritmo mais indicado para detectar anomalias no mercado de criptomoedas, já que ele não conseguiu identificar nenhuma anomalia nos testes com o Litecoin.

Figura 10 – Gráfico de dispersão resultado dos métodos aplicados no Litecoin



Fonte: Elaborado pelo autor

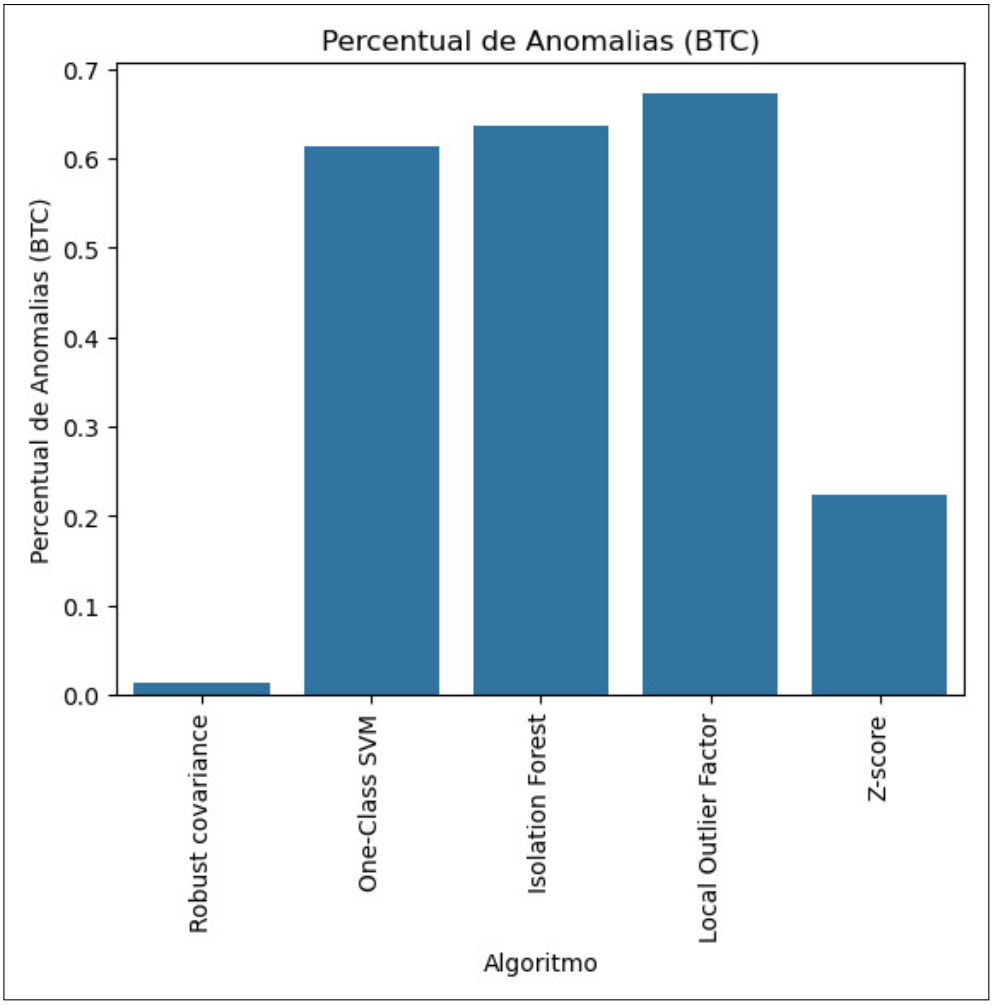
Os três ativos analisados demonstram uma tendência de crescimento ao longo do período, mas com oscilações típicas da volatilidade do mercado de criptomoedas. A quantidade de anomalias detectadas variou conforme o algoritmo utilizado.O *Robust Covariance (Elliptic Envelope)* foi o mais conservador, identificando poucas anomalias no caso do bitcoin e nenhuma anomalia nos ativos Ethereum e Litecoin, enquanto o *One-Class SVM* marcou muitas no início das séries, sendo mais sensível a pequenas variações. O *Isolation Forest* e o *Local Outlier Factor* conseguiram um equilíbrio melhor, identificando principalmente picos e quedas mais significativas. Já o *Z-Score* detectou inúmeras anomalias, especialmente nos valores mais extremos. Os momentos de alta volatilidade e mudanças bruscas nos preços foram os que mais resultaram na marcação de anomalias, principalmente pelos métodos *Isolation Forest*, *Local Outlier Factor* e *Z-Score*.

4.2 METRICAS

As métricas utilizadas para avaliação dos algoritmos em cada cenário foram abordadas por meio de gráficos e tabelas para facilitar o entendimento dos resultados, sendo elas: Percentual de Anomalias, *Silhouette Score* e *Kolmogorov-Smirnov test*

A figura 11 representa um gráfico de barras que compara o percentual de anomalias detectadas pelos métodos de detecção de anomalias utilizados para os experimentos com o Bitcoin. O eixo *X* representa os algoritmos utilizados, enquanto o eixo *Y* mostra a porcentagem de pontos identificados como anômalos.

Figura 11 – Gráfico com percentual de anomalias detectadas no Bitcoin



Fonte: Elaborado pelo autor

A tabela 1 é resultado das informações presentes no gráfico da figura 11, ela apresenta a quantidade de pontos marcados como anômalos por cada método de detecção utilizado. Os valores estão indicados com $\pm$, indicando que se tratam de valores aproximados e podem variar um pouco dependendo da execução do método e da distribuição dos dados.

Como já era esperado, análise dos métodos de detecção de anomalias mostrou que o *Robust Couvariance* foi o mais conservador, identificando somente $\pm$2% dos pontos como anômalos. Isso significa que ele considera a maioria das variações como normais, podendo não ser ideal para séries temporais voláteis como as de criptomoedas. Já os métodos *One-Class SVM*, *Isolation Forest* e *Local Outlier Factor* foram muito mais sensíveis, detectando $\pm$61%, $\pm$63% e $\pm$68% de anomalias, respectivamente. Esses algoritmos são bons para identificar oscilações no mercado, mas podem acabar classificando muitas variações naturais como anomalias, aumentando o risco de falsos

Tabela 1 – Percentual de Anomalias Detectadas no Bitcoin por Método

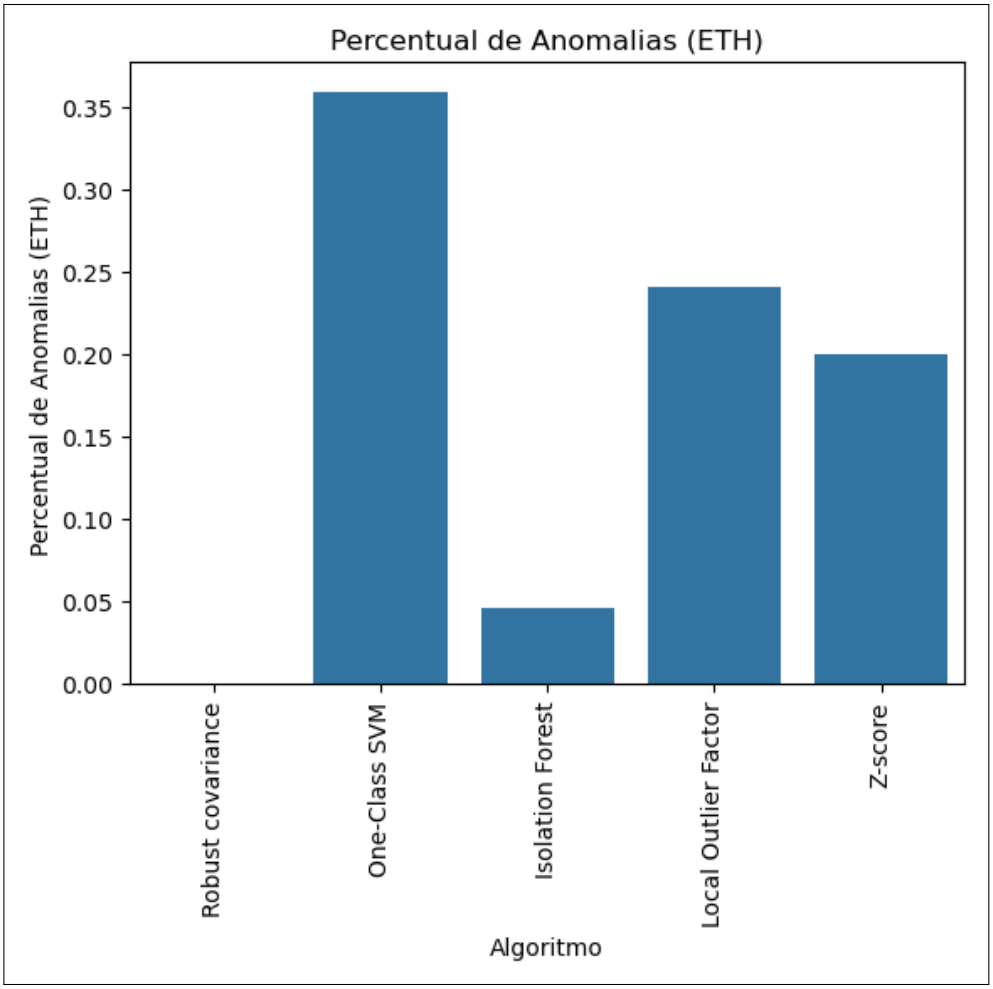
Método	Percentual de Anomalias (%)
Robust Covariance	±2%
One-Class SVM	±61%
Isolation Forest	±63%
Local Outlier Factor	±68%
Z-Score	±25%

Fonte: Elaborado pelo autor

positivos. O Z-Score, por sua vez, teve um desempenho intermediário, marcando ±25% dos pontos como anômalos, sendo mais equilibrado e focando em valores extremos sem exagerar na detecção.

A figura 12 apresenta o gráfico de barras que demonstra o percentual de anomalias detectadas pelos algoritmos aplicados aos dados do Ethereum. O eixo *X* representa os algoritmos utilizados, enquanto o eixo *Y* mostra a porcentagem de pontos identificados como anômalos.

Figura 12 – Gráfico com percentual de anomalias detectadas no Ethereum



Fonte: Elaborado pelo autor

A tabela 2 são os resultados que compõem o gráfico da figura 12, a mesma representa as anomalias identificadas nos dados do *ethereum* pelos métodos em valores numéricos aproximados.

Tabela 2 – Percentual de Anomalias Detectadas no Ethereum por Método

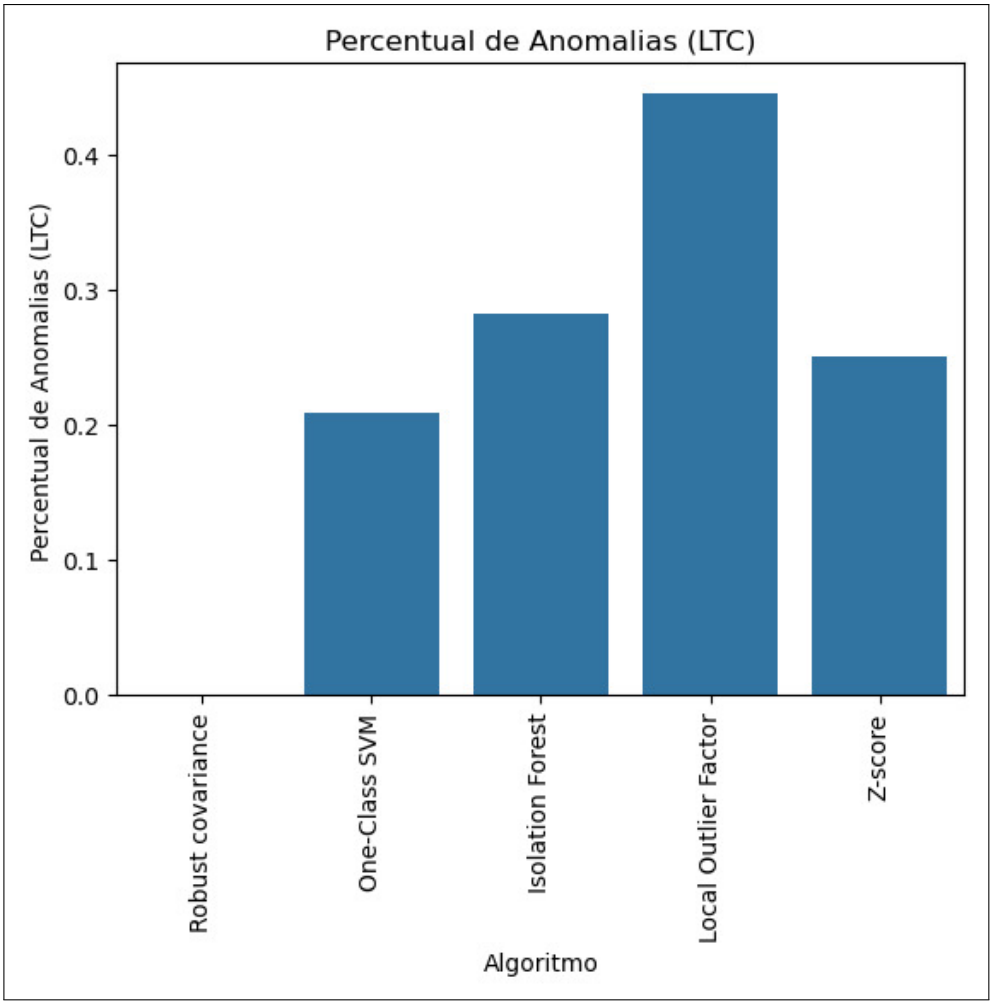
Método	**Percentual de Anomalias (%)**
Robust Covariance	0%
One-Class SVM	$\pm$35%
Isolation Forest	$\pm$5%
Local Outlier Factor	$\pm$25%
Z-Score	$\pm$20%

Fonte: Elaborado pelo autor

O *One-Class SVM* foi o método que mais marcou anomalias, identificando $\pm$35% dos pontos como atípicos, o que pode indicar um alto número de falsos positivos. Já o *Local Outlier Factor* e o *Z-Score* tiveram um comportamento mais equilibrado, detectando entre $\pm$20% e $\pm$25% das anomalias, sendo úteis para capturar mudanças no mercado sem exagerar na marcação. O *Isolation Forest*, por sua vez, foi o mais conservador, identificando menos de $\pm$10% das observações como anômalas, o que sugere que ele focou somente em eventos realmente fora do padrão. Por fim, o *Robust Couvariance* não encontrou anomalias, ao assumir que os dados seguem um comportamento estável, podendo não ser ideal dada a natureza volátil dos dados.

A figura 13 apresenta o gráfico com o percentual de anomalias detectadas no *litecoin* com os métodos utilizados para a detecção.

Figura 13 – Gráfico com percentual de anomalias detectadas no Litecoin



Fonte: Elaborado pelo autor

A tabela 3 são os resultados que compõem o gráfico da figura 13, a mesma representa as anomalias identificadas nos dados do *litecoin* pelos métodos em valores numéricos aproximados.

Tabela 3 – Percentual de Anomalias Detectadas no Litecoin por Método

Método	**Percentual de Anomalias (%)**
Robust Covariance	0%
One-Class SVM	±22%
Isolation Forest	±28%
Local Outlier Factor	±42%
Z-Score	±24%

Fonte: Elaborado pelo autor

O *Robust Covariance* não detectou anomalias, considerando que os dados

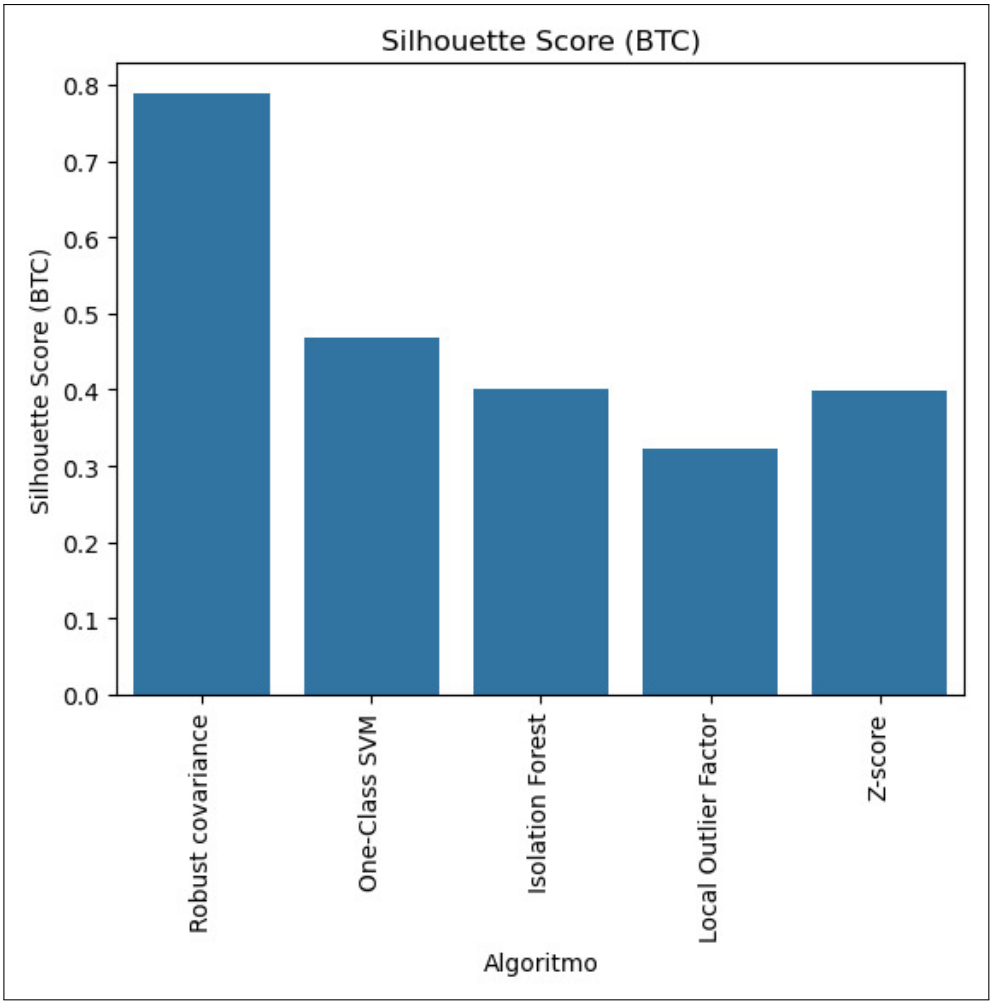
seguem um padrão normal. O *One-Class SVM* marcou cerca de $\pm$22% das observações como anômalas, mas pode estar capturando variações naturais como se fossem eventos incomuns, gerando possíveis falsos positivos. Já o *Isolation Forest* teve um desempenho mais equilibrado, detectando $\pm$28% das anomalias e conseguindo focar melhor em eventos realmente atípicos. O *Local Outlier Factor* foi o mais sensível, classificando $\pm$42% dos pontos como anômalos, o que pode significar que ele está incluindo oscilações normais do mercado. Por fim, o *Z-Score* identificou $\pm$24% das anomalias, concentrando-se nos valores mais extremos, mas sem exagerar na marcação.

A análise dos algoritmos de detecção de anomalias revelou diferenças significativas no comportamento das criptomoedas *Bitcoin*, *Ethereum* e *Litecoin*. O *Robust Couvariance* foi o método mais conservador, quase não identificando anomalias, especialmente no *Ethereum* e no *Litecoin*, onde o método não conseguiu identificar uma única anomalia. O *One-Class SVM* mostrou-se bastante sensível, detectando inúmeras anomalias, o que pode indicar uma alta taxa de falsos positivos. Já o *Isolation Forest* teve um desempenho mais equilibrado, conseguindo diferenciar melhor os eventos realmente atípicos das variações normais do mercado.

O *Local Outlier Factor* foi o método mais agressivo, identificando muitas anomalias, o que pode incluir até mesmo oscilações naturais como eventos incomuns, tornando-o um dos mais sensíveis, mas também mais propenso a falsos positivos. O *Z-Score*, por outro lado, apresentou um comportamento intermediário, sendo útil para capturar valores extremos sem exagerar na marcação de anomalias. No geral, algoritmos mais sensíveis, como *Local Outlier Factor* e *One-Class SVM*, detectam mais anomalias, enquanto métodos mais seletivos, como *Isolation Forest* e *Z-Score*, oferecem um equilíbrio melhor, reduzindo a possibilidade de falsos positivos e tornando-se opções mais adequadas para análises mais precisas.

A figura 14 um gráfico de barras que mostra os resultados do *Silhouette Score* para os métodos de detecção de anomalias aplicados nos dados do *Bitcoin*.

Figura 14 – Gráfico Silhouette Score no Bitcoin



Fonte: Elaborado pelo autor

A tabela 4 apresenta em forma de valores os resultados da tabela 14 do *Silhouette Score* para cada método de detecção de anomalias aplicado ao *Bitcoin*.

Tabela 4 – Silhouette Score para Detecção de Anomalias no Bitcoin

Método	**Silhouette Score**
Robust Covariance	±0.80
One-Class SVM	±0.45
Isolation Forest	±0.40
Local Outlier Factor	±0.30
Z-Score	±0.40

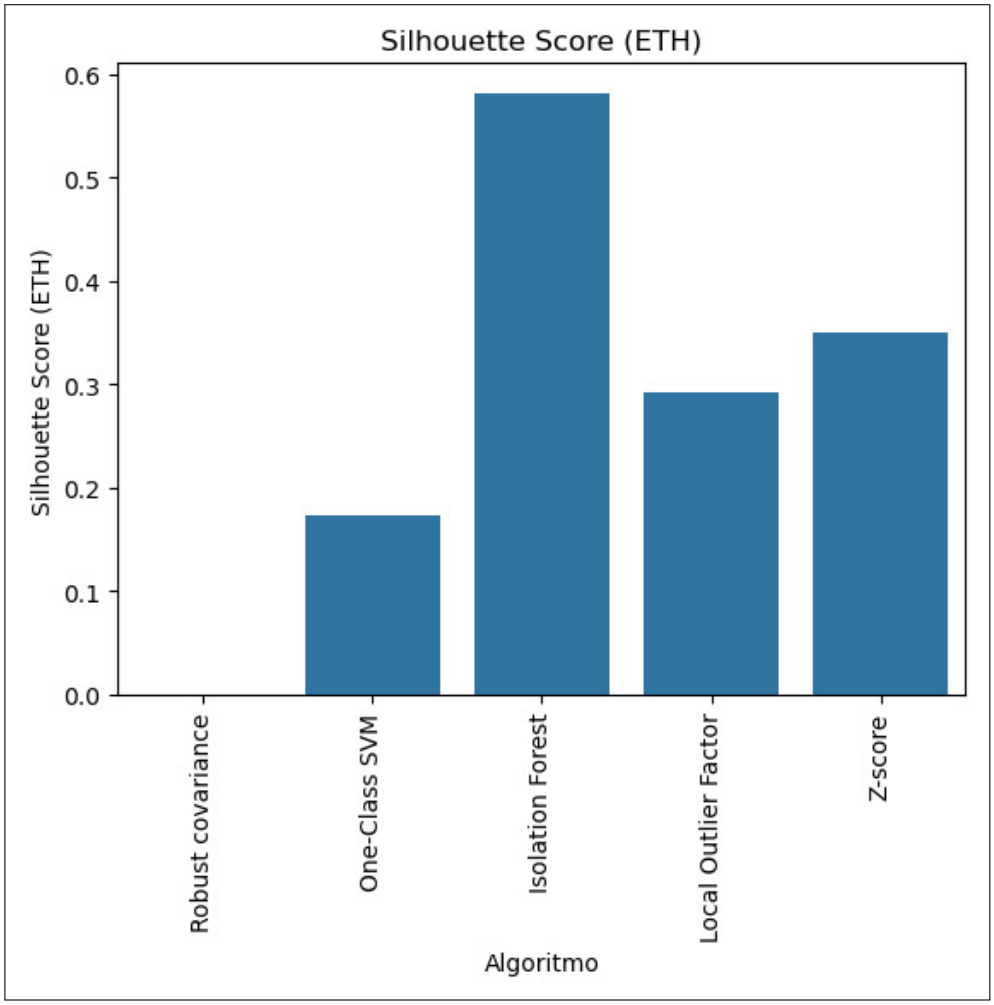
Fonte: Elaborado pelo autor

Nos resultados com os dados do *bitcoin* o *Robust Covariance* obteve o maior *Silhouette Score* ±0.80, indicando que conseguiu a melhor separação entre anomalias

e dados normais, mas isso provavelmente se deve ao seu critério mais restritivo, que identifica um número muito menor de anomalias. O *One-Class SVM*, com um *Silhouette Score* de ± 0.45, e o *Isolation Forest*, com ± 0.40, apresentaram uma separação moderada, sugerindo que seus critérios de detecção são menos rígidos e permitem capturar mais variações no mercado. O *Local Outlier Factor* teve o pior desempenho, com um *Silhouette Score* de ± 0.30, o que indica que algumas das anomalias detectadas podem estar misturadas com os dados normais, aumentando a chance de falsos positivos. O *Z-Score*, com um *Silhouette Score* de ± 0.40, apresentou um desempenho intermediário, sugerindo que sua abordagem estatística pode não ser a mais eficiente para identificar padrões mais complexos no comportamento do Bitcoin.

A figura 15 representa um gráfico que demonstra os resultados do *Silhouette Score* para os métodos de detecção de anomalias aplicados nos dados do *Ethereum*.

Figura 15 – Gráfico Silhouette Score no Ethereum



Fonte: Elaborado pelo autor

A tabela 5 apresenta em forma de valores os resultados da tabela 15 do *Silhouette Score* para cada método de detecção de anomalias aplicado ao *Bitcoin*.

Tabela 5 – Silhouette Score para Detecção de Anomalias no Ethereum

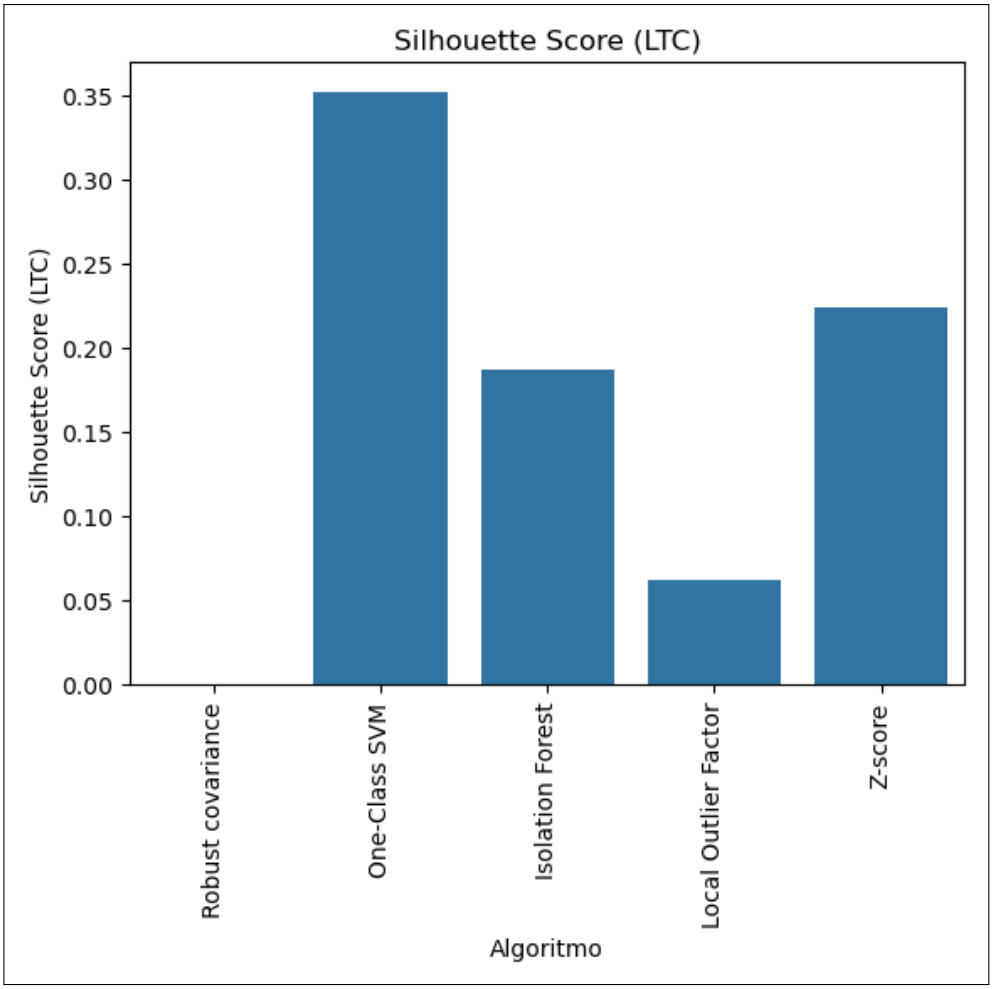
Método	**Silhouette Score**
Robust Covariance	-
One-Class SVM	$\pm$0.18
Isolation Forest	$\pm$0.59
Local Outlier Factor	$\pm$0.30
Z-Score	$\pm$0.38

Fonte: Elaborado pelo autor

O *Isolation Forest* foi o método que conseguiu a melhor separação entre anomalias e dados normais $\pm$0.60, indicando que foi mais eficiente em distinguir padrões incomuns no *Ethereum*. O *Z-Score* $\pm$0.38 e o *Local Outlier Factor* $\pm$0.30 tiveram um desempenho intermediário, sugerindo que conseguiram identificar algumas anomalias, mas sem uma separação tão clara entre os grupos. Já o *One-Class SVM* $\pm$0.18 teve a pior separação, indicando que muitas das anomalias detectadas podem estar misturadas com os dados normais, aumentando o risco de falsos positivos. O *Robust Covariance* não teve impacto na análise, por não identificar anomalias, não gerando o agrupamento necessário para calcular o *Silhouette Score*.

A figura 16 representa um gráfico que demonstra os resultados do *Silhouette Score* para os métodos de detecção de anomalias aplicados nos dados do *Litecoin*.

Figura 16 – Gráfico Silhouette Score no Litecoin



Fonte: Elaborado pelo autor

A tabela 6 apresenta em forma de valores os resultados do gráfico 16 do *Silhouette Score* para cada método de detecção de anomalias aplicado ao *Litecoin*.

Tabela 6 – Silhouette Score para Detecção de Anomalias no Litecoin

Método	**Silhouette Score**
Robust Covariance	-
One-Class SVM	± 0.35
Isolation Forest	± 0.18
Local Outlier Factor	± 0.06
Z-Score	± 0.22

Fonte: Elaborado pelo autor

Dados os resultados apresentados no *Litecoin* o *One-Class SVM* foi o método que teve a melhor separação entre anomalias e dados normais ± 0.35, mostrando que

conseguiu identificar padrões incomuns de forma mais clara. O *Z-Score* $\pm$0.22 e o *Isolation Forest* $\pm$0.18 tiveram um desempenho intermediário, conseguindo distinguir as anomalias, mas sem uma separação tão precisa. Já o *Local Outlier Factor* $\pm$0.06 apresentou o pior resultado, indicando que as anomalias identificadas podem estar misturadas com os dados normais, aumentando a chance de falsos positivos. Por último, o *Robust Covariance* não teve impacto na análise, por não detectar nenhuma anomalia.

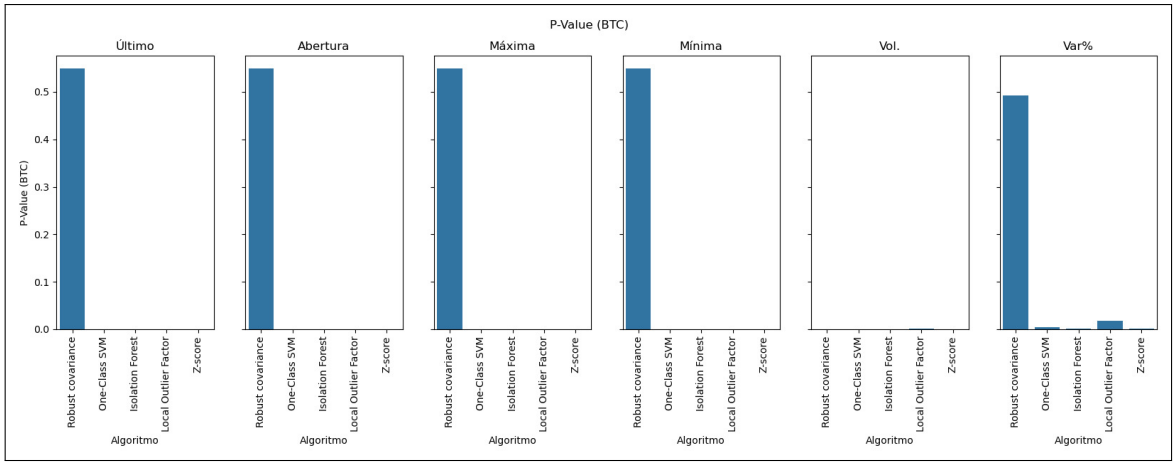
A análise do Silhouette Score para cada criptomoeda demonstrou que a separação entre dados normais e anômalos varia bastante entre as criptomoedas. No *Bitcoin*, o *Robust Covariance* teve o melhor desempenho $\pm$0.80, mas não foi relevante para o *Ethereum* e o *Litecoin*, pois não detectou anomalias nos dados. No *Ethereum*, o *Isolation Forest* se destacou $\pm$0.59, conseguindo diferenciar bem os padrões normais dos eventos atípicos. Já no *Litecoin*, o *One-Class SVM* foi o método mais eficiente $\pm$0.35, indicando que conseguiu capturar melhor as anomalias nessa criptomoeda.

Métodos como *Z-Score* e *Local Outlier Factor* tiveram um desempenho intermediário, com valores entre $\pm$0.30 e $\pm$0.40, mostrando que eles conseguiram separar as anomalias dos dados normais de forma razoável, mas sem tanta precisão. O *Local Outlier Factor* teve o pior desempenho no Litecoin *Local Outlier Factor*0.05), sugerindo que teve dificuldades para diferenciar anomalias reais de variações naturais do mercado. No geral, cada criptomoeda respondeu melhor a um método específico: *Isolation Forest* foi mais eficiente para o *Ethereum*, *Robust Covariance* para o Bitcoin no contexto da métrica do *silhouette score* e *One-Class SVM* para o Litecoin, reforçando a importância de escolher o algoritmo mais adequado para cada tipo de dado e comportamento do mercado.

O resultado do *Kolmogorov-Smirnov test* fica dividido em duas métricas, o *p-value* e a *KS-Statistics*. O *p-value* indica a probabilidade de que as amostras analisadas sigam a distribuição esperada; um valor baixo sugere que as distribuições são significativamente diferentes. A *KS-Statistics* mede a maior diferença entre as funções de distribuição acumulada das amostras; quanto maior esse valor, maior a divergência entre as distribuições comparadas (COMMUNITY, 2025).

A figura 17 representa os resultados do *p-value* para cada atributo da amostra de dados do *Bitcoin* para todos os métodos utilizados para detectar anomalias.

Figura 17 – Gráfico P-Value no Bitcoin



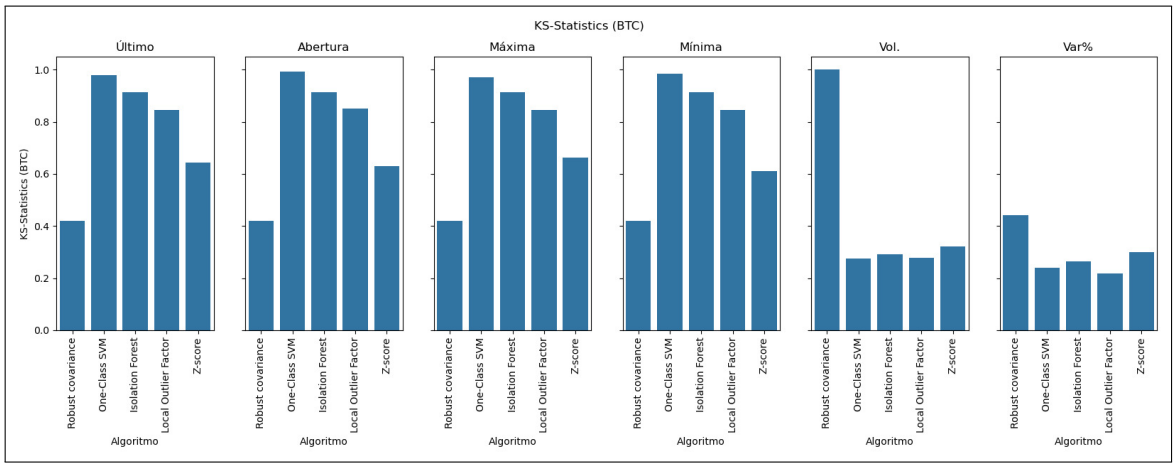
Fonte: Elaborado pelo autor

O *Robust Covariance* apresentou altos valores de *P-Value* para todos os atributos, indicando que os dados analisados não diferem significativamente de uma distribuição normal. Demonstrando por que esse método quase não detectou anomalias, já que por característica própria ele considera a maioria dos dados como pertencentes ao comportamento esperado.

Em contrapartida, os demais métodos tiveram *P-Values* muito baixos, sugerindo que identificaram mudanças significativas nos dados. Levando esses algoritmos a marcar um número maior de anomalias ao reconhecerem padrões fora do esperado com mais facilidade.

A figura 18 representa os resultados do *KS-Statistcs* para cada atributo da amostra de dados do *Bitcoin* para todos os métodos utilizados para detectar anomalias.

Figura 18 – Gráfico KS-Statistcs no Bitcoin



Fonte: Elaborado pelo autor

O *One-Class SVM* e o *Isolation Forest* tiveram os maiores valores de *KS-Statistic*, indicando que conseguiram identificar padrões bem diferentes entre os dados normais e as anomalias. Isso sugere que esses métodos foram mais sensíveis na separação dos pontos fora do padrão.

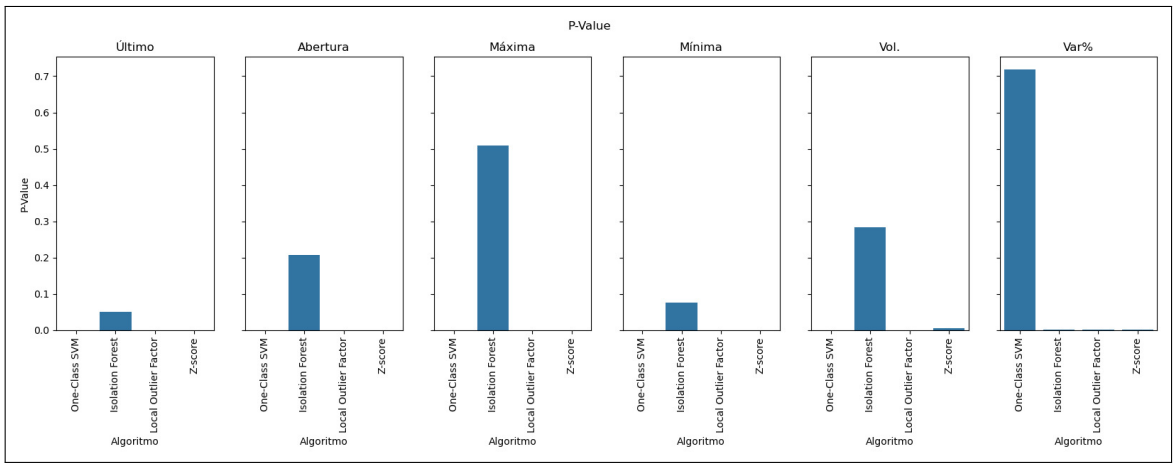
Com valores intermediários de *KS-Statistic* o *Local Outlier Factor* e o *Z-Score*, o que significa que suas detecções de anomalias ainda se desviam da distribuição original dos dados, mas em menor grau do que *One-Class SVM* e *Isolation Forest*.

o *Robust Covariance* teve os menores valores de *KS-Statistic*, mostrando que seus resultados estão mais alinhados com a distribuição original dos dados. Isso faz sentido, já que ele apresentou altos valores de P-Value, indicando que quase não detectou anomalias e considerou a maioria dos dados dentro do comportamento esperado.

O *Robust Covariance* apresentou *P-Values* altos e valores baixos de *KS-Statistic*, o que indica que ele não alterou significativamente a distribuição original dos dados e, por isso, detectou poucas anomalias. Já o *One-Class SVM* e o *Isolation Forest* tiveram *P-Values* baixos e valores altos de *KS-Statistic*, mostrando que esses métodos identificaram padrões bem diferentes dos dados normais. Isso sugere que foram mais rigorosos na detecção de anomalias, marcando mais pontos como fora do padrão. Por outro lado, o *Local Outlier Factor* e o *Z-Score* ficaram em um nível intermediário, o que indica que detectaram anomalias sem alterar tanto a distribuição estatística dos dados, equilibrando sensibilidade e precisão.

A figura 19 representa os resultados do *p-value* para cada atributo da amostra de dados do *Ethereum* para todos os métodos utilizados para detectar anomalias.

Figura 19 – Gráfico P-Value no Ethereum



Fonte: Elaborado pelo autor

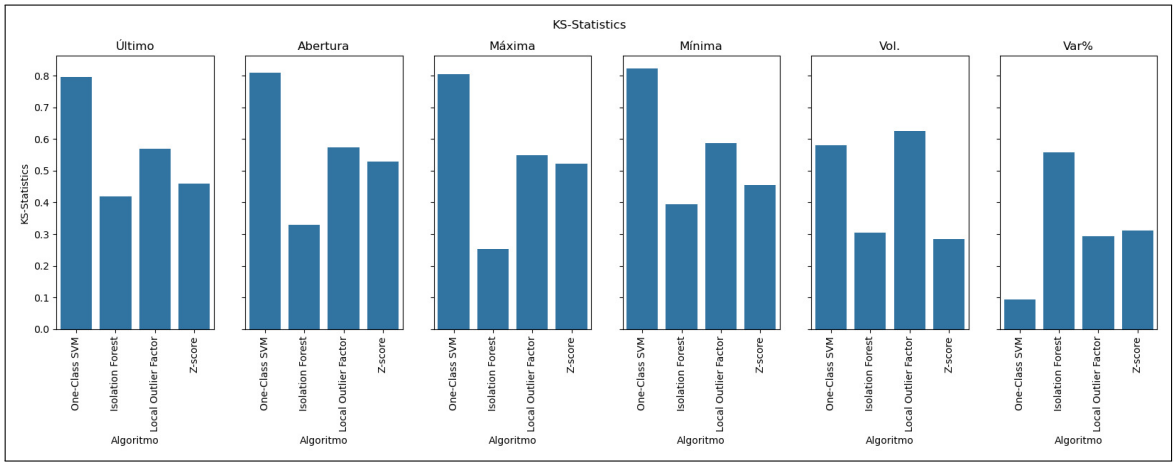
Os métodos *One-Class SVM*, *Local Outlier Factor* e *Z-Score* tiveram *P-Values* muito baixos na maioria dos atributos do *Ethereum*, o que significa que eles consideraram os dados bastante diferentes de uma distribuição normal. Isso indica que esses algoritmos identificaram muitas anomalias, sendo mais rigorosos na detecção.

O *Isolation Forest*, por outro lado, apresentou *P-Values* um pouco mais altos em algumas variáveis, como Abertura e Volume, sugerindo que ele foi um pouco mais seletivo e não marcou tantas anomalias quanto os outros métodos mais agressivos.

Já o *Robust Covariance* não aparece nos gráficos porque não detectou anomalias relevantes no *Ethereum* o que faz com que o código não consiga aplicar as métricas a ele por não ter valores anômalos detectados. Indicando que, segundo esse método, os dados não se desviaram significativamente da distribuição original, mantendo sua estrutura estatística quase inalterada.

A figura 20 representa os resultados do *KS-Statistcs* para cada atributo da amostra de dados do *Ethereum* para todos os métodos utilizados para detectar anomalias.

Figura 20 – Gráfico KS-Statistcs no Ethereum



Fonte: Elaborado pelo autor

O *One-Class SVM* apresentou os maiores valores de *KS-Statistcs* em quase todos os atributos analisados, o que mostra que ele identificou padrões bem diferentes dos dados normais e foi bastante agressivo na marcação de anomalias.

O *Isolation Forest* também teve valores altos de *KS-Statistcs*, especialmente nas métricas Máxima e Abertura, indicando que detectou um número significativo de anomalias, mas ainda assim de forma um pouco mais equilibrada em comparação ao One-Class SVM.

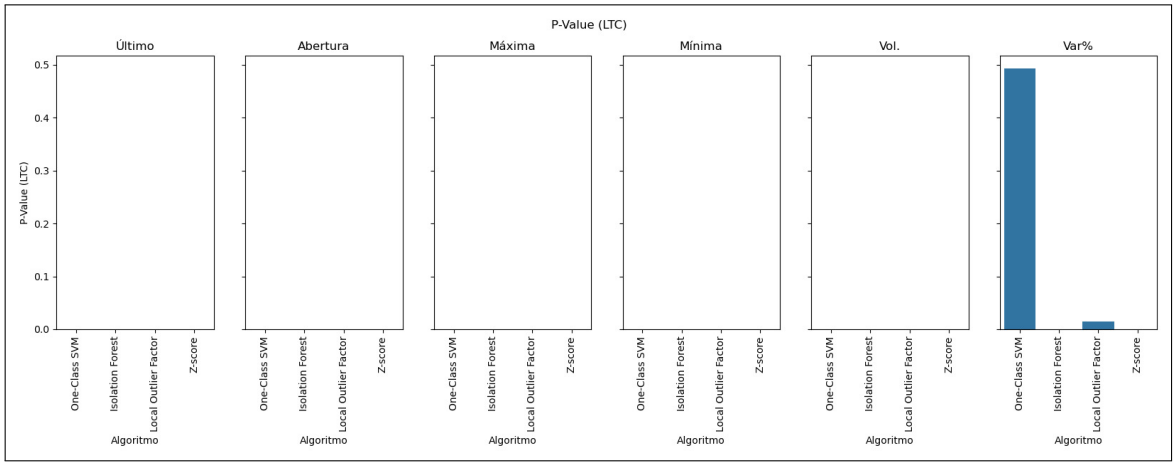
Já o *Local Outlier Factor* e o *Z-Score* tiveram valores de *KS-Statistcs* intermediários, sugerindo que suas detecções alteraram a distribuição dos dados, mas sem exageros. Isso significa que eles marcaram anomalias de forma mais moderada, sem modificar tanto a estrutura dos dados.

Como demonstrado no gráfico, o *Robust Couvariance* não tem resultados para o Ethereum por não detectar anomalias na amostra de dados da criptomoeda.

Os algoritmos *One-Class SVM*, *Local Outlier Factor* e *Z-Score* tiveram P-Values muito baixos, o que indica que marcaram muitas anomalias e alteraram significativamente a distribuição dos dados. Isso se reflete nos altos valores de *KS-Statistcs*, mostrando que esses métodos detectaram padrões bem diferentes dos dados normais. O *Isolation Forest* apresentou um comportamento mais equilibrado, com *P-Values* um pouco mais altos e valores de *KS-Statistcs* ainda elevados, sugerindo que ele conseguiu identificar anomalias de forma significativa, mas sem ser tão agressivo quanto os outros métodos. O *Robust Couvariance* não teve medidas para o *Ethereum*.

A figura 21 representa os resultados do *p-value* para cada atributo da amostra de dados do *Litecoin* para todos os métodos utilizados para detectar anomalias.

Figura 21 – Gráfico P-Value no Litecoin



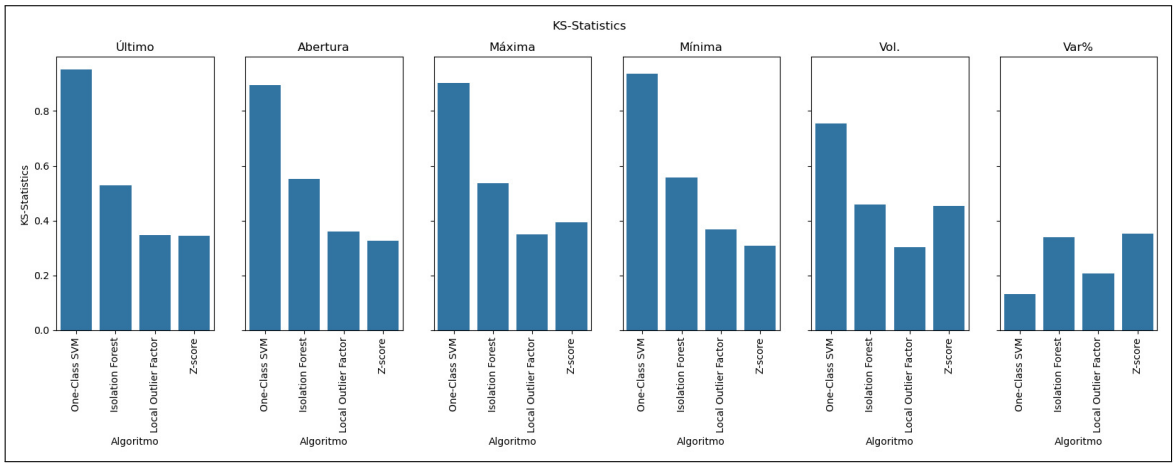
Fonte: Elaborado pelo autor

Os métodos *One-Class SVM*, *Isolation Forest*, *Local Outlier Factor* e *Z-Score* apresentaram *P-Values* muito baixos para quase todos os atributos analisados. Isso indica que esses algoritmos rejeitam fortemente a ideia de que os dados seguem a distribuição original, sugerindo que encontraram uma grande quantidade de anomalias.

Entre eles, o *One-Class SVM* se destacou na variável Variação Percentual

Var%, apresentando um *P-Value* um pouco mais alto. Isso pode indicar que, nessa métrica específica, ele foi mais seletivo na marcação de anomalias, ao contrário do que ocorreu nas demais variáveis. Nenhum dos métodos analisados teve *P-Values* altos o suficiente para sugerir que os dados mantiveram sua distribuição original, o que reforça que a detecção de anomalias foi significativa na maioria dos casos.

Figura 22 – Gráfico KS-Statistcs no Litecoin



Fonte: Elaborado pelo autor

O *One-Class SVM* apresentou os maiores valores de *KS-Statistcs* em praticamente todos os atributos. Isso indica que ele identificou padrões muito diferentes dos dados normais, sendo o método mais agressivo na detecção de anomalias.

O *Isolation Forest* também mostrou valores elevados de *KS-Statistcs*, especialmente para os atributos de Preço Máximo e Preço de Abertura. Isso sugere que ele identificou anomalias significativamente, mas com um critério mais equilibrado em comparação ao *One-Class SVM*.

Já o *Local Outlier Factor* e o *Z-Score* tiveram valores intermediários de *KS-Statistcs*, o que indica que alteraram a distribuição dos dados, mas de forma mais controlada. Isso significa que marcaram anomalias, porém sem modificar os padrões estatísticos dos dados de maneira tão intensa quanto os métodos mais agressivos.

Como já era esperado, o *Robust Couvariance* não tem resultados por não detectar anomalias na abordagem do *Litecoin*.

A comparação entre os gráficos de *P-Value* e *KS-Statistics* mostra uma relação clara: os métodos que tiveram *P-Values* baixos como *One-Class SVM*, *Isolation Forest*, *Local Outlier Factor* e *Z-Score* também apresentaram altos valores de *KS-Statistics*.

Isso significa que eles identificaram muitas anomalias e alteraram significativamente a distribuição dos dados. O *Isolation Forest* teve um comportamento um pouco mais equilibrado, com *P-Values* um pouco mais altos e valores de *KS-Statistics* ainda elevados. Isso sugere que ele conseguiu detectar anomalias de forma eficiente, mas sem ser tão agressivo quanto os outros métodos. *Robust Couvariance* não representa relevância no *KS-Statistics* nos dados do *Litecoin*

A análise dos gráficos de *P-Value* e *KS-Statistics* para *Bitcoin*, *Ethereum* e *Litecoin* revelou um padrão consistente entre os diferentes algoritmos de detecção de anomalias. Métodos como *One-Class SVM*, *Isolation Forest*, *Local Outlier Factor* e *Z-Score* apresentaram *P-Values* muito baixos na maioria das variáveis analisadas, indicando que rejeitam fortemente a hipótese de que os dados seguem a distribuição original, sugerindo uma alta detecção de anomalias. Isso se reflete nos altos valores de *KS-Statistics* observados nesses mesmos métodos, mostrando que as anomalias detectadas causaram um desvio significativo na distribuição dos dados. O *Isolation Forest* se destacou por apresentar um equilíbrio maior, detectando anomalias significativamente, mas sem alterar drasticamente a estrutura dos dados. Já o *Robust Covariance* manteve *P-Values* elevados e *KS-Statistics* baixos para o *Bitcoin*, confirmando que quase não identificou anomalias e preservou a distribuição original dos dados, já nas outras duas amostras *Ethereum* e *Litecoin* não chegou a detectar nenhuma anomalia.

Os resultados das métricas de avaliação, como o percentual de anomalias detectadas e o *Silhouette Score*, deixam claro que não existe um único método perfeito para detectar anomalias. Cada algoritmo encontra um equilíbrio diferente entre a sensibilidade e a precisão ao identificar padrões incomuns, o que impacta diretamente os resultados da análise.

Entre os métodos analisados, o *One-Class SVM* e o *Local Outlier Factor* se destacaram como os mais agressivos, identificando muitas anomalias. No entanto, essa abordagem mais intensa pode gerar uma quantidade considerável de falsos positivos, o que pode prejudicar a utilidade da detecção em alguns casos. O *Isolation Forest*, por sua vez, se mostrou mais equilibrado, especialmente no caso do *Ethereum*, sendo capaz de identificar anomalias de forma eficaz sem alterar muito a distribuição dos dados. Ele se destacou por detectar padrões incomuns sem exagerar na quantidade de anomalias, tornando-se uma escolha mais equilibrada. O *Z-Score* teve um desempenho intermediário, identificando algumas anomalias, mas com menos eficácia em comparação ao *Isolation Forest*. Já o *Robust Covariance* foi o método menos eficiente, detectando poucas anomalias e mantendo os dados muito próximos à distribuição original, o que limita sua aplicabilidade.

Ao avaliar a qualidade dos agrupamentos usando o *Silhouette Score*, o *Isolation Forest* se destacou novamente no *Ethereum*, sendo o método mais eficiente para esse ativo. Por outro lado, o *One-Class SVM* foi o que obteve melhor desempenho para o *Litecoin*, indicando que ele se adapta melhor às características dessa criptomoeda. Isso destaca a importância de escolher o método de detecção de anomalias levando em conta as características dos dados e o tipo de ativo, já que a eficácia de cada algoritmo pode variar dependendo da volatilidade e dos padrões do mercado.

Esses resultados mostram que não existe um único modelo que funcione perfeitamente em todas as situações. A escolha do melhor algoritmo depende de uma avaliação cuidadosa dos *trade-offs* entre precisão, seletividade e impacto na estrutura dos dados, além das características específicas de cada criptomoeda. Entre os métodos testados, o *Isolation Forest* foi o mais equilibrado, pois conseguiu detectar anomalias sem distorcer a distribuição dos dados. Por outro lado, o *Robust Covariance* se mostrou menos eficaz, não conseguindo identificar anomalias relevantes.

5 CONSIDERAÇÕES FINAIS

Este estudo explorou como diferentes algoritmos de aprendizado de máquina e métodos estatísticos podem ser utilizados para identificar anomalias em dados históricos de criptomoedas, ajudando a detectar padrões incomuns em um mercado altamente volátil. Foram testados os algoritmos *Isolation Forest*, *One-Class SVM*, *Local Outlier Factor*, *Robust Covariance* e *Z-Score* como método estatístico para avaliar sua capacidade de identificar desvios nos dados. Os resultados mostraram que não existe um único método ideal para todas as situações, já que cada algoritmo tem um equilíbrio diferente entre sensibilidade e precisão na detecção de anomalias. Esse equilíbrio depende do que se deseja alcançar com o algoritmo, o que significa que é importante escolher o método de acordo com as necessidades específicas de cada caso.

Nenhum dos algoritmos analisados conseguiu, sozinho, abranger com eficiência todas as criptomoedas estudadas, e muito menos conseguiria cobrir todo o mercado de criptomoedas. No entanto, os resultados indicam que a escolha do algoritmo deve considerar a volatilidade e as características específicas de cada ativo. O *Isolation Forest* se destacou como a abordagem mais equilibrada, enquanto o *One-Class SVM* e o *Local Outlier Factor* foram os mais agressivos na detecção de anomalias. Para obter melhores resultados na prática, a combinação de diferentes algoritmos pode ser uma estratégia mais eficiente, permitindo que investidores e analistas tomem decisões mais seguras e reduzam os riscos no mercado de criptomoedas.

Para avaliar o desempenho dos métodos, foram utilizadas métricas como percentual de anomalias detectadas, *Silhouette Score*, teste de *Kolmogorov-Smirnov* e *P-Value*. Essas métricas ajudaram a entender o impacto de cada algoritmo na estrutura dos dados e o quão bem separadas estavam as anomalias detectadas. O *Isolation Forest* apresentou o melhor equilíbrio entre precisão e preservação da estrutura dos dados, enquanto o *Robust Covariance* se mostrou o menos eficaz, praticamente não identificando anomalias.

Este estudo abre portas para pesquisas futuras que explorem a combinação de diferentes métodos, além do uso de redes neurais e técnicas mais avançadas de aprendizado de máquina para aprimorar ainda mais a detecção de anomalias em criptomoedas. Isso pode contribuir para um entendimento mais aprofundado do comportamento do mercado e fornecer ferramentas mais eficientes para análise e tomada de decisões no setor financeiro.

REFERÊNCIAS

ALI, A. Decentralized finance (defi) and its impact on traditional banking systems: Opportunities, challenges, and future directions. **Challenges, and Future Directions (August 01, 2024)**, 2024.

AMARAL, F. **Aprenda mineração de dados: teoria e prática**. [S.l.]: Alta Books Editora, 2016. v. 1.

AMARAL, R. M. d.; QUONIAM, L.; FARIA, L. I. L. d.; LEIVA, D. R.; MILANEZ, D. H.; FIORONI, J. Ultrapassando as barreiras de conversão e tratamento de dados: indicadores de produção científica dos programas de pós-graduação em engenharia de materiais e metalúrgica. **Em Questão**, v. 23, n. 1, p. 228–253, jan. 2017. Disponível em: <https://seer.ufrgs.br/index.php/EmQuestao/article/view/66693>.

AMIR, S. B. H.; PRASETYO, B. Comparison of elliptic envelope method and isolation forest method on imbalance dataset. **Jurnal Matematika, Statistika Dan Komputasi**, v. 17, n. 1, p. 42–49, 2020.

AMIRZADEH, R.; NAZARI, A.; THIRUVADY, D. Applying artificial intelligence in cryptocurrency markets: A survey. **Algorithms**, MDPI, v. 15, n. 11, p. 428, 2022.

ANTONOPOULOS, A. M.; WOOD, G. **Mastering ethereum: building smart contracts and dapps**. [S.l.]: O'reilly Media, 2018.

BISHOP, C. M.; NASRABADI, N. M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006. v. 4.

BÖHME, R.; CHRISTIN, N.; EDELMAN, B.; MOORE, T. Bitcoin: Economics, technology, and governance. **Journal of economic Perspectives**, American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203-2418, v. 29, n. 2, p. 213–238, 2015.

BORGES, A. B. d. S. Bitcoin: internet do dinheiro e o direito. **Revista de Direito Bancário e do Mercado de Capitais**, v. 81, p. 119–139, 2018.

BRADBURY, D. **Litecoin founder Charles Lee on the origins and potential of the world's second largest cryptocurrency**. 2013. Disponível em: <https://www.coindesk.com/markets/2013/07/23/litecoin-founder-charles-lee-on-the-origins-and-potential-of-the-worlds-second-largest-cryptocurrency/>. Acesso em: 16 de outubro de 2023.

BRETERNITZ, V. J.; ALMEIDA, M. I. R. de; GALHARDI, A. C.; MACCARI, E. A. Dinheiro digital-uma implementação de micropagamentos. **Revista Ibero Americana de Estratégia**, Universidade Nove de Julho, v. 7, n. 2, p. 139–146, 2008.

BREUNIG, M. M.; KRIEGEL, H.-P.; NG, R. T.; SANDER, J. Lof: Identifying density-based local outliers. In: **Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data**. ACM, 2000. p. 93–104. Disponível em: <https://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf>.

BUTERIN, V. **Ethereum Whitepaper**. 2023. Disponível em: <https://ethereum.org/en/whitepaper/>. Acesso em: 21 de outubro de 2023.

CARVALHO, A.; FACELI, K.; LORENA, A.; GAMA, J. Inteligência artificial–uma abordagem de aprendizado de máquina. **Rio de Janeiro: LTC**, v. 2, p. 45, 2011.

CARVALHO, T. P. d. et al. Aspectos inovativos do bitcoin, microestrutura de mercado e volatilidade de preços. Universidade Federal da Paraíba, 2015.

CAVALCANTE, A. R. **Detecção de anomalias em dados meteorológicos do sertão de Pernambuco utilizando Isolation Forest e DBSCAN**. Dissertação (B.S. thesis) — Brasil, 2022.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 41, n. 3, p. 1–58, 2009.

CHUEN, D. Handbook of digital currency: Bitcoin. **Innovation, Financial**, 2015.

COINGECKO. **Gráficos de Capitalização de Mercado de Criptomoedas Globais**. 2023. Disponível em: <https://www.coingecko.com/pt/global-charts>. Acesso em: 01 de novembro de 2023.

COMMUNITY, S. **scipy.stats.kstest — Kolmogorov-Smirnov test for goodness of fit**. 2025. Disponível em: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html>. Acesso em: 08 de fevereiro de 2025. Disponível em: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html>.

CUNHA, L. L. **Anomaly detection in cryptocurrency transactions with active learning**. Tese (Doutorado), 2024.

DALL'AGNOL, N. d. A. Bitcoin: estudo sobre conhecimento e percepção da moeda virtual dos alunos do campus de vacaria-camva da universidade de caxias do sul. **Repositório Institucional da UCS**, 2022.

DINIZ, E. H. Emerge uma nova tecnologia disruptiva. **GV-executivo**, v. 16, n. 2, p. 46–50, 2017.

DOKUZ, A. Ş.; ÇELIK, M.; ECEMIŞ, A. Anomaly detection in bitcoin prices using dbscan algorithm. **European Journal of Science and Technology**, v. 2020, p. 436–443, 2020.

DOMBROWSKI, Q.; GNIADY, T.; KLOSTER, D. Introdução ao jupyter notebook. **The Programming Historian em Português**, ProgHist Ltd, 2023.

EDERLI, D. L.; PALMA, D. H. do P.; BERTONCELLO, A. G. O impacto das criptomoedas na economia. **Revista Alomorfia**, v. 5, n. 3, p. 426–437, 2021.

EROSHEVA, E. A.; MINHAS, S.; XU, G.; XU, R. Editorial: Data science meets social sciences. **Journal of Data Science**, School of Statistics, Renmin University of China, v. 20, n. 3, p. 277–278, 2022. ISSN 1680-743X.

Estatística Fácil. **O que é Z-Score? Outlier Detection (Detecção de Outliers com Z-Score)**. 2024. Acessado em: 05 fev. 2025. Disponível em: <https://estatisticafacil.org/glossario/o-que-e-z-score-outlier-detection-deteccao-de-outliers-com-z-score/>.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **kdd**. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231.

ESTIMATOR, D. A fast algorithm for the minimum covariance. **Technometrics**, v. 41, n. 3, p. 212, 1999.

FERNANDES, M. da S.; HAMBERGER, P. A. do V.; VALLE, A. C. M. do. Análise técnica e eficiência dos mercados financeiros: Uma avaliação do po-der de previsão dos padrões de candlestick. **Revista Evidenciação Contábil & Finanças**, Universidade Federal da Paraíba, v. 3, n. 3, p. 35–54, 2015.

FOORTHUIS, R. A typology of data anomalies. In: SPRINGER. **International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems**. [S.l.], 2018. p. 26–38.

____. On the nature and types of anomalies: a review of deviations in data. **International journal of data science and analytics**, Springer, v. 12, n. 4, p. 297–331, 2021.

FREITAS, I. W. S. d. Um estudo comparativo de técnicas de detecção de outliers no contexto de classificação de dados. **Universidade Federal Rural do Semi-Árido**, 2019.

GAMA, M. de P. **Bases da Análise de Grupamento:(Cluster Analysis)**. [S.l.]: Editora Dialética, 2022.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems**. [S.l.]: "O'Reilly Media, Inc.", 2022.

GONÇALVES, L. **Como avaliar modelos de detecção de anomalias não supervisionados**. 2023. Medium. Acesso em: 23 de março de 2025. Disponível em: <https://medium.com/@luanaebio/como-avaliar-modelos-de-detec%C3%A7%C3%A3o-de-anomalias-n%C3%A3o-supervisionados-ed39cb096e1b>.

GRABHER, G. J. A. Um estudo de técnicas de aprendizado por reforço para gerenciamento consciente de memória em ambientes de processamento de streams. 2021.

GRIFFITH, K. **A quick history of cryptocurrencies BBTC — before Bitcoin**. 2014. Disponível em: <https://bitcoinmagazine.com/business/quick-history-cryptocurrencies-bbtc-bitcoin-1397682630>. Acesso em: 16 de outubro de 2023.

GROUP, C. S. **What is supervised learning?** 2023. Disponível em: <https://www.spotfire.com/glossary/what-is-supervised-learning>. Acesso em: 23 de outubro de 2023.

GUIMARAES, N. Revisão sistemática de ferramentas python para análise de dados: Uma abordagem quantitativa. **Journal of Data Science and Python Applications**, v. 12, n. 3, p. 45–67, 2024. Acesso em: 15 de fevereiro de 2025. Disponível em: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5051680>.

HOFMANN, M.; GERHARDT, N.; WAGNER, A.; STOLZ, W.; KOCH, S.; RUHLE, W.; HADER, J.; MOLONEY, J.; SCHNEIDER, H.; CHOW, W. Physics of 1.3 /spl mu/m (gain)(nas)/gaas semiconductor lasers. In: **LEOS 2001. 14th Annual Meeting of the IEEE Lasers and Electro-Optics Society (Cat. No.01CH37242)**. [S.l.: s.n.], 2001. v. 1, p. 326–327 vol.1.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, v. 9, n. 3, p. 90–95, 2007.

INVESTING. **Sobre nós, investing**. 2023. Disponível em: <https://br.investing.com/about-us/>. Acesso em: 15 de novembro de 2023.

JAISWAL, S. **O que é normalização em aprendizado de máquina? Um guia abrangente para redimensionamento de dados**. 2024. Disponível em: <https://www.datacamp.com/pt/tutorial/normalization-in-machine-learning>. Acesso em: 08 de fevereiro de 2025.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. et al. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112.

JANUZAJ, Y.; BEQIRI, E.; LUMA, A. Determining the optimal number of clusters using silhouette score as a data mining technique. **International Journal of Online & Biomedical Engineering**, v. 19, n. 4, 2023.

JOHNSON, R. A.; WICHERN, D. W. et al. Applied multivariate statistical analysis. Prentice hall Upper Saddle River, NJ, 2002.

KABIR, M. A.; AHMED, M. Python for data analytics: A systematic literature review of tools, techniques, and applications. **ACADEMIC JOURNAL ON SCIENCE, TECHNOLOGY, ENGINEERING & MATHEMATICS EDUCATION**, v. 4, n. 04, p. 10–69593, 2024.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. et al. Supervised machine learning: A review of classification techniques. **Emerging artificial intelligence applications in computer engineering**, Amsterdam, v. 160, n. 1, p. 3–24, 2007.

LIMA, J. Q. d. Detecção de fraudes em cartões de crédito utilizando técnicas de aprendizado de máquina. Serra, 2022.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: IEEE. **2008 eighth ieee international conference on data mining**. [S.l.], 2008. p. 413–422.

LIU, J.; YUAN, J.; LI, C.-N. One-class svm with 0-1 loss. **Available at SSRN 5050012**, 2024.

LIU, W.; PRINCIPE, J. C.; HAYKIN, S. **Kernel adaptive filtering: a comprehensive introduction**. [S.l.]: John Wiley & Sons, 2011.

MAHALANOBIS, P. C. On the generalised distance in statistics. **Proceedings of the National Institute of Sciences of India**, v. 2, p. 49–55, 1936.

MAIMON, O.; ROKACH, L. **Data Mining and Knowledge Discovery Handbook**. [S.l.]: Springer US, Boston, MA, 2005.

MARTINS, A. N. d. G. L.; VAL, E. M. Criptomoedas: Notas sobre seu funcionamento e perpectivas institucionais no brasil e mercosul. **Revista de Direito Internacional Econômico e Tributário**, v. 11, n. 1 Jan/Jun, p. 227–252, 2016.

MARTINS, M. E. G. Diagrama ou gráfico de dispersão. **Revista de Ciência Elementar**, Casa das Ciências, v. 2, n. 3, p. 1–2, 2014.

____. Diagrama ou gráfico de barras. **Revista de Ciência Elementar**, Casa das Ciências, v. 6, n. 1, 2018.

MATOS, R.; PARDO, T. A. S.; INFANTE, R. **Avaliação de Sistemas de Detecção de Anomalias: Uma Abordagem Baseada em Métricas de Desempenho**. [S.l.], 2009. Disponível em: <https://sites.icmc.usp.br/taspardo/TechReportUFSCar2009a-MatosEtAl.pdf>.

MCGREGGOR, D. M. **Mastering matplotlib**. [S.l.]: Packt Publishing Ltd, 2015.

MCLACHLAN, G. J. Mahalanobis distance. **Resonance**, v. 4, n. 6, p. 20–26, 1999.

MEIRA, L. A.; DALL'ORA, F. S.; SANTANA, H. L. S. **Tributação de novas tecnologias e o caso das criptomoedas**. [S.l.]: Almedina São Paulo, 2020.

MIOLA, A. C.; MIOT, H. A. **P-valor e dimensão do efeito em estudos clínicos e experimentais**. [S.l.]: SciELO Brasil, 2021. e20210038 p.

MNIH, V.; KAVUKCUOGLU, K.; SILVER, D.; RUSU, A. A.; VENESS, J.; BELLEMARE, M. G.; GRAVES, A.; RIEDMILLER, M.; FIDJELAND, A. K.; OSTROVSKI, G. et al. Human-level control through deep reinforcement learning. **nature**, Nature Publishing Group, v. 518, n. 7540, p. 529–533, 2015.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003.

MOSS, J. univariateml: An r package for maximum likelihood estimation of univariate densities. **Journal of Open Source Software**, The Open Journal, v. 4, n. 44, p. 1863, 2019. Disponível em: <https://doi.org/10.21105/joss.01863>.

MÜLLER, A. C.; GUIDO, S. **Introduction to machine learning with Python: a guide for data scientists**. [S.l.]: "O'Reilly Media, Inc.", 2016.

MUXFELDT, P. **Vantagens e desvantagens das criptomoedas, como o bitcoin**. 2021. Disponível em: <https://br.ccm.net/faq/57726-vantagens-e-desvantagens-das-criptomoedas>. Acesso em: 9 de outubro de 2023.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. **Decentralized business review**, 2008.

OHUNAKIN, O. S.; HENRY, E. U.; MATTHEW, O. J.; EZEKIEL, V. U.; ADELEKAN, D. S.; OYENIRAN, A. T. Conditional monitoring and fault detection of wind turbines based on kolmogorov–smirnov non-parametric test. **Energy Reports**, v. 11, p. 2577–2591, 2024. ISSN 2352-4847. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352484724000817>.

OZER, F.; SAKAR, C. O. An automated cryptocurrency trading system based on the detection of unusual price movements with a time-series clustering-based approach. **Expert Systems with Applications**, Elsevier, v. 200, p. 117017, 2022.

PAAR, C.; PELZL, J. **Understanding cryptography: a textbook for students and practitioners**. [S.l.]: Springer Science & Business Media, 2009.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. et al. Scikit-learn: Machine learning in python. **the Journal of machine Learning research**, JMLR. org, v. 12, p. 2825–2830, 2011.

PERCIVAL, C. **Stronger key derivation via sequential memory-hard functions**. 2009. Disponível em: <http://www.tarsnap.com/scrypt/scrypt.pdf>. Acesso em: 16 de outubro de 2023.

PEREIRA, A.; SIMONETTO, E. de O. Indústria 4.0: conceitos e perspectivas para o brasil. **Revista da Universidade Vale do Rio Verde**, v. 16, n. 1, 2018.

PÉREZ, F.; GRANGER, B. E. Ipython: a system for interactive scientific computing. **Computing in science & engineering**, IEEE, v. 9, n. 3, p. 21–29, 2007.

PINTO, S. O. Revisão de literatura: abordagem de detecção de anomalias em sistemas financeiros. 2023.

RÄTSCH, G.; SCHÖLKOPF, B.; MIKA, S.; MÜLLER, K.-R. **SVM and boosting: One class**. [S.l.]: GMD-Forschungszentrum Informationstechnik, 2000.

REIFF, N. **Bitcoin vs. Ethereum: What's the Difference?** 2023. Disponível em: <https://www.investopedia.com/articles/investing/031416/bitcoin-vs-ethereum-driven-different-purposes.asp>. Acesso em: 17 de outubro de 2023.

REZEK, H. **A Guide to Data Manipulation with Python's Pandas and NumPy**. 2024. Disponível em: <https://medium.com/munchy-bytes/a-guide-to-data-manipulation-with-pythons-pandas-and-numpy-607cfc62fba7>. Acesso em: 08 de fevereiro de 2025.

RIBEIRO, D. A (r) evolução das obrigações empresariais: do escambo ao bitcoin e o anseio por uma regulamentação brasileira. **Revista da AMDE**, v. 13, p. 173–189, 2015.

ROUSSEEUW, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal of computational and applied mathematics**, Elsevier, v. 20, p. 53–65, 1987.

RUPPERT, D. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: Taylor & Francis, 2004.

SACOMANO, J. B.; GONÇALVES, R. F.; BONILLA, S. H.; SILVA, M. T. da; SÁTYRO, W. C. **Indústria 4.0**. [S.l.]: Editora Blucher, 2018.

SCHÖLKOPF, B.; PLATT, J. C.; SHAWE-TAYLOR, J.; SMOLA, A. J.; WILLIAMSON, R. C. Estimating the support of a high-dimensional distribution. **Neural computation**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 13, n. 7, p. 1443–1471, 2001.

SCHWAB, K.; MIRANDA, D. A quarta revolução industrial (edipro). **São Paulo**, 2016.

SENNA, V. d.; SOUZA, A. M. Criptomoedas e sistema financeiro: Revisão sistemática de literatura. **Revista de Administração de Empresas**, SciELO Brasil, v. 63, p. e2022–0019, 2023.

SIEGEL, I. F. **Análise de Dados e Aplicações em Python: Uma Abordagem Prática**. 2018. Repositório Institucional da Universidade Federal Fluminense (RIUFF).

SIGAUD, O.; BUFFET, O. **Markov decision processes in artificial intelligence**. [S.l.]: John Wiley & Sons, 2013.

SIMILARWEB. **Ranking dos sites de finanças mais visitados**. 2023. Disponível em: <https://www.similarweb.com/pt/top-websites/finance/>. Acesso em: 15 de novembro de 2023.

SMOLA, A. **Introduction to machine learning**. 2008.

SOARES, H. L. Detecção de anomalias no sistema aps de caminhões de carga utilizando algoritmos do tipo one-class. Serra, 2022.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: An introduction**. [S.l.]: MIT press, 2018.

SZABO, N. Smart contracts: building blocks for digital markets. **EXTROPY: The Journal of Transhumanist Thought,(16)**, v. 18, n. 2, p. 28, 1996.

TOMÉ, M. P. D. **A natureza jurídica do bitcoin**. [S.l.]: Porto Alegre: Elegantia Juris, 2019.

TRIOLA, M. F. **Estatística**. [S.l.]: Pearson, 2018.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J. et al. Scipy 1.0: fundamental algorithms for scientific computing in python. **Nature methods**, Nature Publishing Group US New York, v. 17, n. 3, p. 261–272, 2020.

WATKINS, C. J. C. H. Learning from delayed rewards. King's College, Cambridge United Kingdom, 1989.

XIAO, P.; XIAO, M.; CAI, N.; QIU, B.; ZHOU, S.; WANG, H. Adaptive hybrid framework for multiscale void inspection of chip resistor solder joints. **IEEE Transactions on Instrumentation and Measurement**, v. 72, p. 1–12, 2023.

YON, G. G. V.; MARJORAM, P. fmcmc: A friendly mcmc framework. **Journal of Open Source Software**, The Open Journal, v. 4, n. 39, p. 1427, 2019. Disponível em: <https://doi.org/10.21105/joss.01427>.

ZHAO, G.; BAN, Y.; ZHANG, Z.; SHI, Y.; LIU, H. Phase demodulation strategy based on kalman filter for sinusoidal encoders. **IEEE Sensors Journal**, v. 23, n. 10, p. 10625–10632, 2023.

ZIEGEL, E. R. **The elements of statistical learning**. [S.l.]: Taylor & Francis, 2003.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Cajazeiras - Código INEP: 25008978
Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

Entrega do trabalho de conclusão de curso

Assunto:	Entrega do trabalho de conclusão de curso
Assinado por:	Thierry Silva
Tipo do Documento:	Tese
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Fulgêncio Thierry Gomes da Silva**, **DISCENTE (202122010027) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 27/03/2025 10:20:14.

Este documento foi armazenado no SUAP em 27/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1436730
Código de Autenticação: 9d156d8415