

Universidade Federal de São Carlos – UFSCar
Programa de Pós-Graduação em Ciência da Computação
João Mateus Arcolini Emílio

Classificação de Retornos Utilizando Dados de Alta Frequência no Mercado de Bitcoins

São Carlos - SP

Março/2020

João Mateus Arcolini Emílio

Classificação de Retornos Utilizando Dados de Alta Frequência no Mercado de Bitcoins

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Aprendizado de Máquina.

Orientador: Prof. Dr. Ricardo A. S. Fernandes

São Carlos - SP

Março/2020



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato João Mateus Arcolini Emílio, realizada em 10/03/2020:

Prof. Dr. Ricardo Augusto Souza Fernandes
UFSCar

Prof. Dr. Diego Furtado Silva
UFSCar

Prof. Dr. Fábio Anderson Silva Borges
UESPI

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Fábio Anderson Silva Borges e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ão) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Prof. Dr. Ricardo Augusto Souza Fernandes

Agradecimentos

Primeiramente, agradeço a Deus por me guiar nas mais belas oportunidades e nunca desistir.

Agradeço à minha família e amigos por estarem comigo em todos os momentos. Por me dar forças e total apoio na busca pelos meus sonhos.

Quero agradecer meu orientador, Prof. Ricardo pela enorme paciência e oportunidade durante esses anos de pesquisa.

“Tirar conclusões precipitadas é eficaz se houver grande probabilidade de que as conclusões estejam corretas e se o custo de um ocasional erro for aceitável”

D. Kahneman

Resumo

No mercado de criptomoedas, o Bitcoin destaca-se como a mais aceita e negociada no mundo. Entretanto, devido à sua alta volatilidade, a predição de comportamentos de preços, em especial a classificação de tendência, torna-se um desafio e é de grande interesse aos investidores. Neste sentido, nos últimos anos, alguns estudos foram propostos com base em técnicas de aprendizado de máquina com a intenção de classificar sinais viáveis de negociação a partir da série de retornos do Bitcoin. Assim, o presente trabalho explora comportamentos de tendência do Bitcoin, buscando classificar seu retorno por meio de dados de alta frequência referentes às ordens do livro de ofertas em termos de dólar americano (BTCUSD) e euro (BTCEUR). A metodologia proposta busca prever retornos com precisão, criando oportunidades que deem melhor suporte às estratégias de negociação. Para tanto, a partir das séries históricas do Bitcoin, foram extraídos indicadores técnicos (comumente empregados em mercados financeiros), variáveis de mercado e a própria série de retornos para diferentes intervalos temporais. Cabe mencionar que os dados possuem uma frequência de no máximo um segundo para cada atualização de todo livro de ofertas, caracterizando assim a alta frequência. Na sequência, estes dados são submetidos às entradas de classificadores de tendência baseados em Redes Neurais Artificiais e XGBoost, os quais foram treinados e validados em três meses de negociação, de Janeiro a Março de 2019, compondo aproximadamente 5 milhões de atualizações do mercado para cada moeda base. Em uma primeira análise foram observados os comportamentos dos modelos para cada moeda base, sendo possível notar que o mercado para BTCUSD se demonstra mais eficiente em relação à BTCEUR. Consequentemente, os modelos de aprendizado de máquina obtiveram resultados mais confiáveis e estáveis ao longo do tempo para BTCUSD. Partindo desse resultado inicial, buscou-se validar os modelos preditivos para distintas configurações de treinamento e validação. Nesta segunda análise, notou-se que a classificação das tendências mostra-se mais precisa para intervalos curtos (entre 1 e 3 minutos) em ambas as moedas base, o que é justificável devido a alta volatilidade nos preços do Bitcoin em curtos intervalos de tempo. Em uma terceira análise, observou-se que à medida que a quantidade de dias no conjunto de treinamento é incrementada, em alguns casos, nota-se uma perda gradativa de desempenho na classificação (entre 1% e 2%). Ainda, em uma quarta análise, aplicou-se *one-vs.-rest* para cada comportamento de tendência, verifica-se um aumento de desempenho considerável na classificação em relação ao caso base (entre 1% e 7%).

Palavras-chaves: Bitcoin, alta frequência, aprendizado de máquina, microestrutura de mercado, negociação algorítmica.

Abstract

In the cryptocurrency market, Bitcoin stands out as the most accepted traded in the world. However, due to its high volatility, the prediction of price behaviors, in special, the trend classification, becomes a challenge and valuable to investors. In this sense, in recent years, some studies have been proposed based on machine learning techniques to classify reliable trading signals from the Bitcoin return series. Thus, the present work explores Bitcoin's trend behaviors, seeking to classify its returns using high-frequency data from limit order book in terms of US dollar (BTCUSD) and euro (BTCEUR). The proposed methodology seeks to accurately predict returns, creating opportunities that better support trading strategies. Therefore, from the Bitcoin historical series, technical indicators (commonly used in financial markets), market variables and the own series of returns for different time intervals were extracted. It is worth mentioning that the data have a frequency of at most one second for each update of the entire order book, thus characterizing the high-frequency. Subsequently, these data are submitted to the inputs of trend classifiers based on Artificial Neural Networks and XGBoost, which were trained and validated in three months of trading, from January to March 2019, composing approximately 5 million market updates for each currency base. In a first analysis, the behaviors of the models for each base currency were observed, it is possible to note that the market for BTCUSD proves to be more efficient related to BTCEUR. Consequently, the machine learning models obtained more reliable and stable results over time for BTCUSD. Based on this initial result, sought to validate the predictive models for different training and validation configurations. In this second analysis, it was noted that the trend classification is more accurate for short intervals (between 1 and 3 minutes) in both base currencies, which is justified due to the high volatility in Bitcoin prices in short intervals. In a third analysis, it was observed that as we increased the number of training days for classifiers, in some cases, was noted a gradual loss in classification (between 1% and 2%). Still, in a fourth analysis, *one-vs.-rest* was applied to each trend behavior, there is a considerable performance increase in the classification in relation to the base case (between 1% and 7%).

Key-words: Bitcoin, high-frequency, machine learning, market microstructure, algorithmic trading.

Lista de ilustrações

Figura 1 – Mercado de capitalização do Bitcoin (USD) ao longo dos anos.	23
Figura 2 – Visão geral da metodologia proposta.	29
Figura 3 – Distribuição da série de log-retorno para $k = \{1, 3, 5, 10\}$ minutos em BTCUSD.	33
Figura 4 – Distribuição da série de log-retorno para $k = \{1, 3, 5, 10\}$ minutos em BTCEUR.	33
Figura 5 – Distribuição de <i>bid ask volume imbalance</i> no livro de ofertas para os níveis $L = \{5, 10, 50, 100\}$	35
Figura 6 – Definição dos conjuntos de treinamento e validação.	39
Figura 7 – Distribuição dos resultados para a $F1$ com $k = 1$ minuto em termos de BTCUSD e BTCEUR.	46
Figura 8 – Distribuição das classes de dados com $m = 5$ dias e $k = 1$ minuto em termos de BTCUSD e BTCEUR para ambos os conjuntos de treinamento e validação, em 82 dias de negociação.	48
Figura 9 – Distribuição dos resultados para a $F1$ com $k = \{3, 5, 10\}$ minutos em termos de BTCUSD.	50
Figura 10 – Distribuição dos resultados para a $F1$ com $k = (3, 5, 10)$ minutos em termos de BTCEUR.	51
Figura 11 – Distribuição dos resultados para a $F1$ com $k = 1$ minuto e $m = \{10, 15\}$ dias em termos de BTCUSD.	53
Figura 12 – Distribuição dos resultados para a $F1$ com $k = 1$ minuto e $m = (10, 15)$ dias em termos de BTCEUR.	54

Lista de tabelas

Tabela 1 – Métricas de avaliação de desempenho.	43
Tabela 2 – Resultados para BTCUSD com $m = 5$ dias e $k = 1$ minuto. Média e desvio padrão para 82 dias de negociação.	45
Tabela 3 – Resultados para BTCEUR com $m = 5$ dias e $k = 1$ minuto. Média e desvio padrão para 83 dias de negociação.	46
Tabela 4 – Resultados para BTCUSD com $m = 5$ dias e variando $k = \{3, 5, 10\}$ minutos. Média e desvio padrão para 82 dias de negociação.	49
Tabela 5 – Resultados para BTCEUR com $m = 5$ dias e variando $k = \{3, 5, 10\}$ minutos. Média e desvio padrão para 83 dias de negociação.	49
Tabela 6 – Resultados para BTCUSD variando $m = \{10, 15\}$ dias e $k = 1$ minuto. Média e desvio padrão para 82 dias de negociação.	52
Tabela 7 – Resultados para BTCEUR variando $m = \{10, 15\}$ dias e $k = 1$ minuto. Média e desvio padrão para 83 dias de negociação.	52
Tabela 8 – Resultados – <i>One-vs.-Rest</i> – para BTCUSD com $m = 5$ dias e $k = 1$ minuto. Média e desvio padrão para 82 dias de negociação.	55
Tabela 9 – Resultados – <i>One-vs.-Rest</i> – para BTCEUR com $m = 5$ dias e $k = 1$ minuto. Média e desvio padrão para 82 dias de negociação.	55

Sumário

1	Introdução	19
1.1	Contextualização e Motivação	19
1.2	Hipótese	20
1.3	Objetivos	20
1.4	Contribuições da Dissertação	21
1.5	Organização do Trabalho	21
2	Revisão da Literatura	23
2.1	Mercado de Bitcoin	23
2.2	Negociação Algorítmica e Eficiência de Mercado	24
2.3	Previsão nos Mercados	26
3	Metodologia	29
3.1	Coleta dos Dados	29
3.2	Pré-processamento dos Dados	30
3.2.1	Períodos de Análise e Amostragem	31
3.2.2	Rotulação dos Retornos Visando a Classificação de Tendência	31
3.2.3	Variáveis de Mercado Provenientes do Livro de Ofertas a Limite	34
3.2.4	Indicadores Técnicos de Tendência	36
3.3	Definição dos Conjuntos de Treinamento e Validação	38
3.4	Algoritmos de Aprendizado de Máquina	39
3.4.1	Redes Neurais Artificiais	40
3.4.2	XGBoost – <i>Extreme Gradient Boosting</i>	41
3.5	Métricas de Avaliação de Desempenho	42
3.6	Detalhes de Implementação da Metodologia Proposta	43
4	Experimentos Computacionais	45
4.1	Experimento 1 - Caso base (E1)	45
4.2	Experimento 2 - Variação de Intervalos de Previsão (E2)	48
4.3	Experimento 3 - Variação da Quantidade de Dias de Treinamento (E3)	52
4.4	Experimento 4 - <i>One-vs.-Rest</i> (E4)	54
5	Conclusões e Trabalhos Futuros	57
	Referências	59

1 Introdução

1.1 Contextualização e Motivação

Desde a primeira vez que foi introduzido por [Nakamoto \(2008\)](#), o Bitcoin (BTC) tem recebido considerável atenção entre investidores e pesquisadores do mundo todo. Devido a sua natureza única, o Bitcoin pode ser negociado *online* e trocado pelas principais moedas do mercado (e.g. Dólar, Euro, Reais, etc.) ([DIMPFL, 2017](#)). O mercado de Bitcoin foi criado em 2009, mas o comércio ativo começou apenas em 2013, que é bastante recente em relação às outras moedas. No entanto, suas características (simplicidade, transparência, alta volatilidade e valorização nos últimos anos), tornou o Bitcoin e o mercado de criptomoedas bastante popular entre investidores e pesquisadores. A partir do Bitcoin, surgiram outras criptomoedas semelhantes, tais como Litecoin e Ethereum.

Atualmente, um dos principais pontos de interesse é se o preço do Bitcoin é previsível devido à sua alta volatilidade em curtos períodos de tempo ([SENSOY, 2018](#)), o que contradiz as hipóteses relacionadas à eficiência do mercado ([MALKIEL; FAMA, 1970](#)). No entanto, com o avanço da negociação algorítmica (processo sistemático de inserção de ordens no livro de ofertas) ([CHABOUD et al., 2014](#)) e a modernização dos mercados nos últimos anos, surgem novas hipóteses sobre como processar e tomar decisões de forma consistente, minimizando riscos nas operações relacionadas a custo e incerteza, bem como maximizando lucros nas negociações. Neste contexto, sofisticadas estratégias de negociação utilizam microestrutura de mercado para realizar análises quantitativas de dados provenientes do livro de ofertas em alta frequência ([ALDRIDGE, 2009](#)). Assim, o objetivo é estudar a formação de preços em altas frequências, ou seja, determinar as variáveis ou condições de mercado que sejam responsáveis por alterações no preço quando a eficiência do mercado é menor ([ALDRIDGE, 2013](#)). Portanto, a quantidade de informações sobre o livro de ofertas neste cenário é muito maior se comparada à tradicional baixa frequência dos dados que formam os *candles* diários.

A partir desse estudo, é possível analisar o livro de ofertas, de forma quantitativa, com o objetivo de inferir comportamentos de preços futuros. Neste sentido, algoritmos de Aprendizado de Máquina (AM), especificamente Redes Neurais Artificiais (RNAs) e modelos baseados em *ensemble*, e.g. o XGBoost ([CHEN; GUESTRIN, 2016](#)), se destacam em capturar o comportamento não linear e altamente volátil dos mercados, em contraste aos modelos estatísticos lineares tradicionais, como ARMA (*Autoregressive Moving Average*) e ARIMA (*Autoregressive Integrated Moving Average*) ([BOX et al., 2015](#)).

1.2 Hipótese

Atualmente, na literatura, existem muitos esforços para a previsão de comportamentos de preços dos mercados de criptomoedas (NAKANO; TAKAHASHI; TAKAHASHI, 2018; SHINTATE; PICHL, 2019; PENG et al., 2018). No entanto, grande parte desses trabalhos utilizam poucas informações do livro de ofertas para validar seus métodos ao longo do tempo sendo que, na maioria das vezes, tais modelagens podem não generalizar as previsões. Sendo assim, a pesquisa proposta nessa dissertação de mestrado tem a hipótese de que os dados são de suma importância para a validação do problema. Neste contexto, foram coletados dados em alta frequência do livro de ofertas do Bitcoin, dos quais foram extraídos indicadores técnicos, variáveis de mercado e a própria série de retornos para diferentes intervalos temporais. A partir dessa base de dados, foram realizados experimentos computacionais, por meio de algoritmos de AM, com a intenção de comprovar a hipótese supramencionada.

1.3 Objetivos

O objetivo deste trabalho é modelar comportamentos de preços do mercado de Bitcoin em termos de dólares Americanos (USD) e euro (EUR) utilizando dados históricos do livro de ofertas em alta frequência. Para tal, foram construídos diversos classificadores de tendência utilizando como entradas séries históricas de retornos, indicadores técnicos e variáveis de mercado. Cabe mencionar que a microestrutura de mercado oferece todo instrumental para descrever variáveis como liquidez, volume, volatilidade e *spread* em relação à formação e descoberta de preço. Ao final, os classificadores foram treinados em capturar sinais relevantes de tendência do mercado, contribuindo com a tomada de decisão em estratégias de negociação.

Neste contexto, este trabalho não irá priorizar a construção de todo um sistema de negociação completo, visto que o foco aqui é analisar de forma quantitativa o livro de ofertas utilizando dados históricos e realizar previsões de tendências de preços do Bitcoin utilizando algoritmos de AM.

Como objetivos específicos deste trabalho, pode-se citar:

- Coletar dados de alta frequência de todo o livro de ofertas do Bitcoin;
- Modelar e implementar diversos indicadores relacionados a oscilações de preços e variáveis de mercado para descrever comportamentos de tendência;
- Modelar e implementar diferentes algoritmos de AM que são referências na literatura; e

- Avaliar comportamentos dos algoritmos de AM em diferentes intervalos de previsão e janelas de treinamento.

1.4 Contribuições da Dissertação

Esta dissertação de mestrado busca contribuir para o avanço do estado da arte no que diz respeito à previsão de comportamentos de preço do Bitcoin, mostrando que a mesma pode ser realizada de forma eficiente e robusta por algoritmos de AM sobre dados obtidos em alta frequência. Ademais, busca-se demonstrar que a abordagem proposta poderá ser utilizada como suporte a agentes tomadores decisões em termos de estratégias de negociação em alta frequência.

1.5 Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma:

Capítulo 2 descreve os trabalhos relacionados, onde são apresentadas as principais características do mercado de Bitcoin, como ativo e moeda. A partir disso, são apresentados aspectos fundamentais sobre a negociação algorítmica e seus benefícios. Por fim, são apresentados conceitos relacionados à previsão de comportamentos de preços dos mercados de criptomoedas, considerando o uso de algoritmos de AM.

Capítulo 3 apresenta a abordagem proposta nesta dissertação, bem como os experimentos realizados. A princípio, descreve-se a coleta dos dados e as variáveis que compõem a base de dados. Em seguida, define-se o conjunto de dados a partir de *snapshots* do livro de ofertas (BTCUSD e BTCEUR) e a construção de diversas séries de mercado. No final do capítulo, são apresentados os aspectos inerentes aos algoritmos de AM utilizados e todo o processo de execução experimental, parametrização e métricas de validação.

Capítulo 4 destina-se a apresentar os resultados experimentais obtidos por meio da abordagem proposta. Inicialmente, foi proposta uma configuração inicial para experimentação, em que todos os resultados obtidos foram analisados para ambos os mercados considerados, ou seja, BTCUSD e BTCEUR. Dessa forma, buscou-se demonstrar comportamentos dos modelos de AM em diferentes condições de mercado. Ao final do capítulo são mostrados os diversos testes executados para diferentes janelas de treinamento e intervalos de previsão.

Capítulo 5 apresenta as conclusões obtidas após a realização dos experimentos em relação às hipóteses previamente definidas. Ainda são propostos trabalhos futuros que podem ser executados pelo grupo de pesquisa a fim de suprir algumas lacunas existentes no âmbito da presente pesquisa.

2 Revisão da Literatura

2.1 Mercado de Bitcoin

Desde o início de sua comercialização, em 2010, até o final de 2017, o Bitcoin cresceu com um perfil exponencial e atingiu um pico de USD 320 bilhões em capitalização de mercado. Esse comportamento pode ser observado por meio de análise gráfica a partir da Figura 1. Sua crescente adoção como um investimento se deu pela grande inovação que a criptomoeda proporciona em relação à segurança e transparência nas negociações, ao mesmo tempo que promove garantias de privacidade entre os comerciantes, sem a necessidade de um intermediador para realizar as trocas. Esse conceito é denominado Blockchain, que consiste em uma rede distribuída de negociação sem intermediários para realizar e validar uma transação, muito menos alguém para cobrar altas taxas nas operações. As transações são verificadas pela rede, que checa a precisão da transação mais recente em relação ao registro do total de transações. A transação é subsequentemente adicionada à rede e a informação é redistribuída para outros nós (NAKAMOTO, 2008).

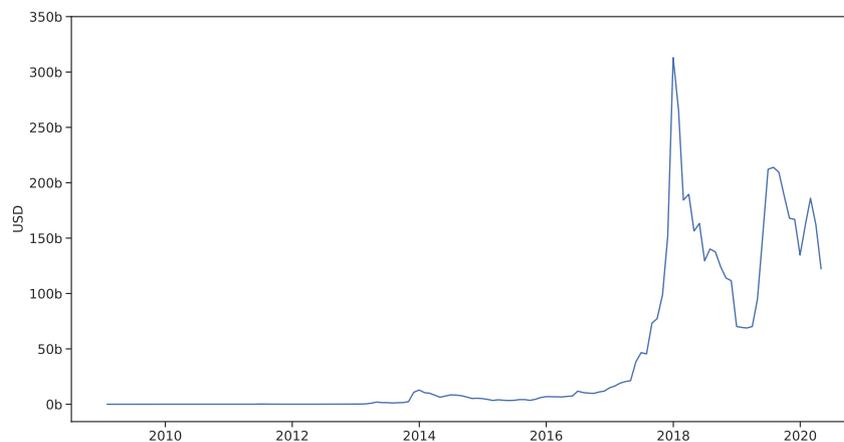


Figura 1 – Mercado de capitalização do Bitcoin (USD) ao longo dos anos.

Atualmente, pesquisas científicas questionam o uso do Bitcoin como uma moeda ou ativo (especulativo) para investimentos (BAUR; HONG; LEE, 2018; CHEAH; FRY, 2015). Neste sentido, alguns pesquisadores não consideram o Bitcoin como uma moeda, visto que ela possui alta volatilidade em comparação a outras moedas e mercados tradicionais. Em relação ao mercado de criptomoedas, há pesquisas que definem o mercado de Bitcoin como ineficiente, porém esse perfil poderá mudar ao longo dos anos (DIMPFL, 2017), visto que

essa característica está diretamente relacionada às diferentes modalidades de negociação do ativo (SENSOY, 2018).

A negociação no mercado de Bitcoins é semelhante àquelas dos mercados financeiros tradicionais. Em particular, as plataformas de negociação não atuam como formadores de mercado, porém os *traders* lançam ordens a mercado ou limitadas, que compõem o livro de ofertas. As ordens a mercado são imediatamente executadas em relação às ofertas remanescentes, enquanto as ordens a limite entram no livro de ofertas a menos que possam ser executadas imediatamente. Entretanto, não há leilões ou interrupções de volatilidade, pois o mercado opera durante 24 horas e 7 dias por semana. O tamanho mínimo do *tick*¹ é de 0,01 unidades da moeda base (DIMPFL, 2017).

2.2 Negociação Algorítmica e Eficiência de Mercado

O crescimento de investimentos em negociação de ações pelos mercados financeiros nos últimos anos, transformou-se em uma busca contínua por melhores ferramentas de monitoramento e gerenciamento de ordens nos livros de ofertas. Com a evolução dos computadores, a negociação algorítmica tornou-se um importante ferramental para os *traders*, trazendo maior consistência, velocidade e precisão nas operações, principalmente no auxílio da previsão de comportamentos dos mercados, visando maximizar lucros e minimizar riscos. A partir da negociação algorítmica, surge um amplo interesse em entender os potenciais impactos de determinadas variáveis na dinâmica de mercado e sua relação com a formação de preço, volatilidade e custos, ou seja, a eficiência de mercado.

Segundo Aldridge (2009), esta prática trouxe grandes benefícios, em especial, maior eficiência na formação de preço e estabilidade dos mercados. Por definição, a negociação algorítmica consiste em um processo sistemático de inserção de ordens no livro de ofertas, que são pré-determinadas por outra parte do processo de negociação ou por um gerente de portfólio (CHABOUD et al., 2014).

Portanto, nos mercados financeiros atuais, algoritmos fazem HFT (*High Frequency Trading*) com o principal objetivo de obter pequenos lucros em cada negociação, buscando realizar milhares de negociações por dia, com precisão de milissegundos (ALDRIDGE, 2009). Nesse formato de operação, o volume negociado na compra e venda de ativos em um único dia é maior quando comparado ao volume médio negociado por *traders* comuns (HAGSTRÖMER; NORDÉN, 2013), ou seja, aqueles que fazem operações diárias ou intra-diárias. Dessa forma, estratégias desta modalidade tendem a ter um controle maior sobre o risco nas negociações. Isso acontece porque o valor em carteira negociado é menor em cada operação, criando uma dependência menor sobre as oscilações de preço dos ativos

¹ Nos mercados financeiros, o tamanho do *tick* (ou *tick size*) é o menor incremento com o qual o preço dos ativos negociados podem se mover.

(SILVA et al., 2014). Dentre os benefícios das estratégias de HFT nos mercados financeiros, pode-se destacar: (i) aumento da eficiência do mercado; (ii) maior liquidez; (iii) inovação tecnológica; e (iv) estabilidade dos mercados (ALDRIDGE, 2009).

Nesse contexto, estratégias de HFT necessitam analisar o livro de ofertas da forma mais granular possível e encontrar oportunidades de negociações favoráveis. Em Aldridge (2009), discute-se que essas oportunidades são em grande parte uma função dos dados que as identificam. Nesse sentido, quanto maior a frequência dos dados, mais oportunidades podem aparecer. De acordo com Dacorogna et al. (2001), os dados de alta frequência abriram um novo campo de exploração e trouxeram alguns comportamentos que não podiam ser observados em frequências mais baixas.

Formalmente, dados de alta frequência são coleções de *ticks* sequenciais de eventos mais recentes que ocorrem no mercado (livro de ofertas), ou seja, última cotação, negociação, informação de volume, liquidez, *spread*, entre outras variáveis de mercado (MADHAVAN, 2000). Conforme mostrado por Aldridge (2009), dados de *tick* seguem usualmente as seguintes variáveis:

- Instante de tempo (ou *timestamp*) do último evento ocorrido no mercado;
- Código identificador do ativo;
- Indicadores de variáveis de mercado que tal ativo carrega:
 - Preço de compra (*bid*);
 - Preço de venda (*ask*);
 - Volume disponível para compra;
 - Volume disponível para venda;
 - Preço da última ordem negociada (*last trade price*); e
 - Tamanho da última ordem negociada (*last trade size*).

Todas essas informações compõem o livro de ofertas. Dessa forma, se torna essencial a utilização de dados de *tick* para o estudo de comportamentos dos mercados, em especial, para o estudo do processo da formação de preço de um ativo. Nos mercados financeiros, em geral, várias hipóteses são levantadas sobre a eficiência do mercado (MALKIEL; FAMA, 1970), que consiste em, dada qualquer interferência interna ou externa no mercado, os preços são instantaneamente ajustados. No entanto, em altas frequências, o processo de formação do preço é dificilmente instantâneo, pois o mercado tenta encontrar um valor ótimo que corresponde a essa mudança. Tal conceito é denominado microestrutura de mercado (ALDRIDGE, 2009). Especificamente, o objetivo é entender quais

fatores latentes estão relacionados com as definições de preços ao longo do tempo, incluindo variáveis estáticas (e.g. taxas) e dinâmicas (e.g. volume, liquidez, volatilidade e *spread*) (MADHAVAN, 2000).

Diversos mercados de criptomoedas oferecem uma plataforma de negociação algorítmica para seus *traders*, o que torna essencial analisar a eficiência de mercado dessas criptomoedas em um nível intra-diário. Nesse contexto, existem trabalhos na literatura que comprovam a ineficiência de mercado do Bitcoin e identificam a formação de preço em altas frequências (SENSOY, 2018; CIAIAN; RAJCANIOVA; KANCS, 2016).

No trabalho de Sensoy (2018), o objetivo foi analisar a formação de preço em altas frequências em termos de BTCUSD e BTCEUR. Foram construídas séries de log-retorno com frequências de 15, 20, 30, 40 e 45 minutos para testes comparativos sobre a eficiência relativa do mercado. Para as análises, foram utilizadas janelas de tempo deslizantes semanais não sobrepostas, em um total de 270 semanas (Janeiro de 2013 a Março de 2018). Ainda empregaram a entropia de permutação, proposta em Bandt e Pompe (2002), para testes de dependência dos log-retornos para medir a eficiência do Bitcoin. O autor notou que, em termos de BTCUSD, a formação de preço é suavemente mais eficiente do que BTCEUR. Além disso, em altas frequências, a eficiência dos preços é menor.

Em Ciaian, Rajcaniova e Kancs (2016), o objetivo foi estudar a formação de preço relacionada a fatores econômicos (e.g. oferta, demanda e especulação de preço) em curto e médio prazos. Os autores observaram que tais fatores, principalmente a oferta e demanda, são predominantes na formação do preço.

2.3 Previsão nos Mercados

A previsão de movimentações de preço nos mercados financeiros, tanto para ativos como para moedas ou criptomoedas é uma tarefa de grande valor econômico. Entretanto, tal tarefa mostra-se complexa de se realizar com precisão e de forma consistente para garantir previsões que consigam generalizar seus comportamentos, o que seria inconsistente de acordo com as hipóteses sobre a eficiência de mercado (MALKIEL; FAMA, 1970). Técnicas de previsão de séries temporais, análise técnica e fundamentalista, são aplicadas corriqueiramente na literatura para modelar tais comportamentos de preços e movimentações no mercado financeiro (BOX et al., 2015; ZUO; KITA, 2012; ALDRIDGE, 2009). No entanto, estas técnicas podem, na maioria dos casos, não generalizar de forma consistente previsões do mercado (SILVA et al., 2014).

Técnicas de aprendizado de máquina têm se mostrado como ferramentas de predição capazes de generalizar e realizar previsões de forma consistente, além de apresentarem desempenho maior se comparadas aos modelos estatísticos (como por exemplo ARMA ou ARIMA). Dessa forma, as tomadas de decisões de envio de ordens de com-

pra e venda podem ser otimizadas, objetivando lucro nas negociações. A utilização de aprendizado de máquina para previsão de preços nos mercados tradicionais é comumente encontrada na literatura, pois tais técnicas atingem ótimos resultados (SILVA et al., 2014; OLIVEIRA; NOBRE; ZARATE, 2013; MARTINEZ et al., 2009; LABOISSIERE; FERNANDES; LAGE, 2015). A previsão de preços do Bitcoin pode ser considerada análoga a outras tarefas de previsão de séries temporais financeiras. Entretanto, ainda existem poucos trabalhos relevantes sobre previsão de comportamentos de preços relacionados a criptomoedas (NAKANO; TAKAHASHI; TAKAHASHI, 2018; SHINTATE; PICHL, 2019; SHAH; ZHANG, 2014; PENG et al., 2018).

Um dos primeiros trabalhos significativos que aplicam aprendizado de máquina para a previsão de comportamentos de preços do Bitcoin foi proposto por Nakano, Takahashi e Takahashi (2018). O objetivo foi explorar a previsão de movimentações de preços (alta e baixa) para o próximo período de 15 minutos em uma estratégia de negociação que utiliza configurações viáveis de custos de execução (taxas e *spread*). Para isso, os autores construíram um sistema de negociação utilizando Redes Neurais Artificiais para classificação de sinais de negociação (positivos ou negativos) a partir da série de retornos e indicadores técnicos de tendência. Em suas análises, não foram consideradas hipóteses relacionadas à falta de liquidez no mercado, visto que tal variável é indispensável para simular o comportamento real das negociações a partir de dados históricos e validar efetivamente uma estratégia de negociação. Ainda, os resultados demonstraram que a abordagem utilizada melhora significativamente o desempenho de uma estratégia *Buy-and-Hold* (B&H).

Outro trabalho relevante sobre a classificação de tendências no mercado de Bitcoin foi publicado por Shintate e Pichl (2019). Neste trabalho, os autores também utilizaram RNAs com o objetivo de classificar tendências futuras (em alta, baixa e estática) da série de log-retorno. Para tanto, empregaram um conjunto de dados (entre 13 de Junho de 2013 e 18 de Março de 2017) no formato OHLC (*Open, High, Low e Close*) com precisão de 1 minuto do mercado de Bitcoin da China (Okcoin) em termos de BTCUSD e BTCCNY. A metodologia proposta obteve a precisão de 0.5717 e sensibilidade (ou *recall*) de 0.5538 em BTCUSD. Em termos de BTCCNY, a metodologia apresentou precisão de 0.5581 e sensibilidade de 0.5182.

Ainda sobre a utilização de aprendizado de máquina em criptomoedas, no trabalho de Peng et al. (2018) é proposta uma abordagem denominada *Support Vector Regression GARCH* (ou SVR-GARCH) para realizar previsões de volatilidade em criptomoedas, na qual se demonstra superior ao tradicional método GARCH (BAUWENS; LAURENT; ROMBOUS, 2006).

Todos os trabalhos citados motivam a investigação da previsão de comportamentos de preços no mercado de Bitcoins em alta frequência proposta nessa dissertação de mes-

trado. Entretanto, o foco será explorar aspectos relacionados à microestrutura de mercado com o objetivo de capturar tendências futuras no processo de formação de preços.

3 Metodologia

Este capítulo é destinado a detalhar a metodologia utilizada neste trabalho, a qual pode ser visualizada pela Figura 2, que a resume em 7 etapas: (1) coleta dos dados de mercado; (2) rotulação de retornos para generalizar comportamentos de tendência; (3) definição de variáveis de mercado; (4) construção de indicadores técnicos de tendência; (5) definição dos conjuntos de treinamento e validação utilizando janelas deslizantes; (6) aplicação dos modelos de AM, especificamente RNA e XGBoost; e (7) avaliação de desempenho dos modelos de AM ao longo do tempo.

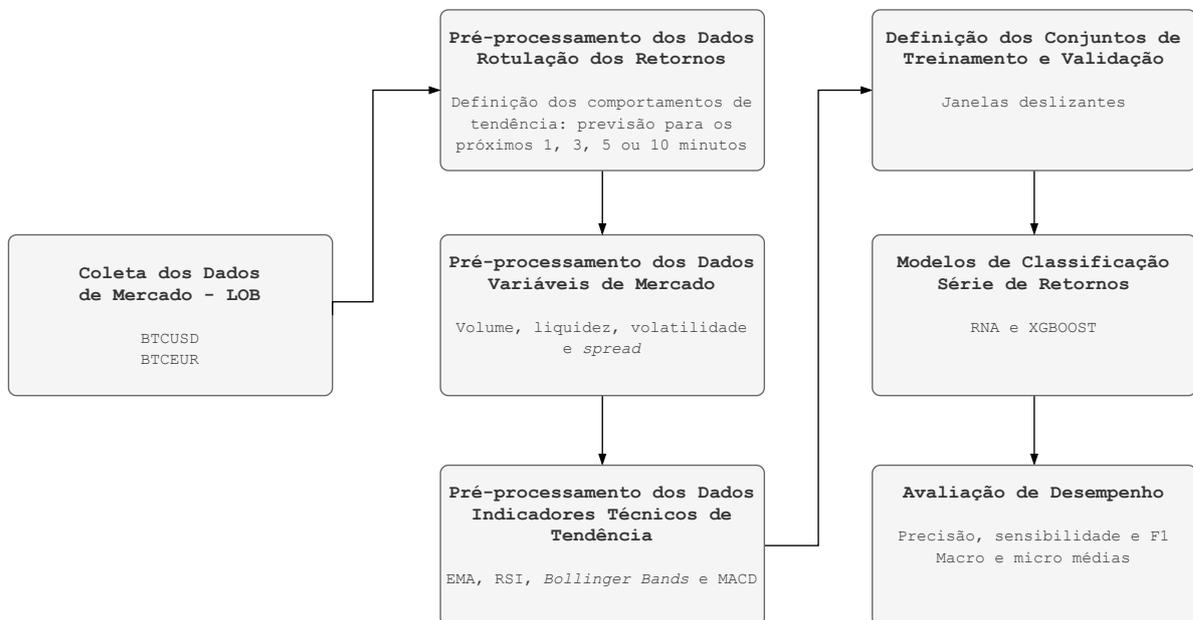


Figura 2 – Visão geral da metodologia proposta.

3.1 Coleta dos Dados

Os dados consistem de *snapshots* coletados do livro de ofertas do mercado Bitstamp por meio de uma conexão realizada via *WebSocket*² (WS). Cabe comentar que o Bitstamp foi escolhido pois, conforme mostrado por (DIMPFL, 2017), ele é um dos mercados mais líquidos em termos de BTCUSD e BTCEUR.

O uso de WS permite que o próprio servidor da aplicação, neste caso o Bitstamp, envie todo evento ocorrido no livro de ofertas aos seus clientes. Esta arquitetura permite a coleta de dados do mercado em tempo real. Atualmente, o Bitstamp disponibiliza uma precisão de um segundo para *snapshots* do livro de ofertas e microssegundos para inserção

de ordens. Sua documentação está disponível em [Bitstamp.net/websocket/v2](https://bitstamp.net/websocket/v2).

Tratando-se de uma grande quantidade de dados, é necessário que seja implementada uma arquitetura capaz de armazenar estes dados de alta frequência e que a mesma seja consistente, dado a dependência temporal entre cada evento ocorrido no mercado. Além disso, a perda de dados desencadeada não deve ocorrer. Tais restrições são essenciais para a execução dos experimentos. Neste contexto, foi implementada uma estrutura do tipo fila (FIFO, do inglês *First In, First Out*), em que dois agentes, um produtor e outro consumidor, são responsáveis pela persistência dos dados. Ambos os agentes funcionam em processos distintos, sendo a fila com acessos compartilhados entre eles. A estrutura de fila implementada suporta toda a semântica de bloqueio necessária. O produtor (Bitstamp) envia os dados de mercado para esta fila a todo segundo, enquanto o consumidor retira e persiste dados da fila a cada uma hora de coleta.

Dessa forma, toda a coleta de dados do mercado acontece de forma consistente. Todo o processamento e persistência ocorrem em servidores da *Amazon Web Services*³, garantindo que a coleta ocorra 24 horas por dia e em 7 dias na semana e que os dados sejam persistidos em segurança.

Os dados foram coletados entre Janeiro e Março de 2019, considerando *snapshots* do livro de ofertas com precisão de um segundo. Cada *snapshot* contém um conjunto de ordens a limite (LOB, do inglês *Limit Order Book*) com as duas filas de prioridade, compra (*bid*) e venda (*ask*), de tamanho 100, onde cada posição $i = 1, \dots, 100$, ou prioridade no livro, contém uma ordem com seu preço de compra $P_{bid}^{(i)}$ ou venda $P_{ask}^{(i)}$ em uma moeda base (USD e EUR) e a quantidade em Bitcoin (BTC). Cada *snapshot* pode ser identificado por um *timestamp* t em que foi atualizado o livro de ofertas. Assim, considerando os três meses de LOB coletados, BTCUSD e BTCEUR totalizaram 4979091 e 5439370 registros, respectivamente. Dessa forma, pode-se definir como $P_{bid,t}^{(1)}$ o melhor preço de compra e $P_{ask,t}^{(1)}$ como o melhor preço de venda em um instante de tempo t . Ainda, é possível definir $\mathbb{T}_{usd} = \{t; t = 1, \dots, 4979091\}$ e $\mathbb{T}_{eur} = \{t; t = 1, \dots, 5439370\}$ como sendo os conjuntos de dados utilizados nos experimentos computacionais. Entretanto, será utilizado \mathbb{T} para denotar ambos os conjuntos.

3.2 Pré-processamento dos Dados

Após a coleta dos dados descrita na Seção 3.1, os dados são inseridos em coleções no MongoDB⁴ e indexados pelo *timestamp* em que cada atualização do livro de ofertas ocorreu. Dessa forma, foi possível analisar o mercado de Bitcoins temporalmente, mantendo a consistência de um grande volume de dados. A partir disso, foram construídos conjuntos

² Tecnologia que permite a comunicação bidirecional por canais *full-duplex* sobre um único *socket* de comunicação (FETTE; MELNIKOV, 2011).

³ Plataforma de serviços de computação em nuvem oferecida pela [Amazon.com](https://amazon.com)

de dados para diferentes períodos de negociação, levando em consideração variáveis de mercado e indicadores técnicos de tendência.

3.2.1 Períodos de Análise e Amostragem

Visando efetuar as previsões de tendência para um próximo período de negociação, foi necessário definir períodos de análises e realizar amostragens para a criação das séries de mercado. Na literatura, é comum encontrar trabalhos que utilizam uma amostragem de dados sumarizados do mercado, no formato OHLC (*candlestick*) (NAKANO; TAKAHASHI; TAKAHASHI, 2018; PENG et al., 2018; SHINTATE; PICHL, 2019). No entanto, cada período em um *candlestick* é único, ou seja, cada período analisado irá sintetizar um intervalo de tempo de eventos mutuamente exclusivos do mercado.

Nesse contexto, utiliza-se uma abordagem baseada em janela deslizante (*sliding window*) que considera um conjunto de dados como sendo os pares $\mathbf{d} = (\mathbf{i}_t, \mathbf{o}_t)$, em que \mathbf{i}_t e \mathbf{o}_t denotam respectivamente vetores do espaço de entrada e saída no instante de tempo t para $t \in \mathbb{T}$. Assim, dada uma sequência de dados em \mathbf{d} , uma amostragem por janela deslizante pode ser descrita como sendo o intervalo $[t - k, t] \forall t \in \mathbb{T}$, onde k denota o tamanho do período de negociação em minutos que será analisado para um *timestamp* t . A Equação 3.1 define uma sequência para o processo de amostragem utilizado nas análises, em que $n = |\mathbb{T}|$:

$$S = ((t - k, t), (t - k + 1, t + 1), (t - k + 2, t + 2), \dots, (t - k + n, t + n)). \quad (3.1)$$

Dessa forma, dado um período que é representado por um *candlestick*, pode-se obter as mesmas variáveis de OHLC dos preços, porém os intervalos não irão sintetizar eventos exclusivos. Portanto, pode-se dizer que para cada *tick* é criado um novo período a ser analisado. Além disso, é possível analisar variáveis como volume e volatilidade dos preços dentro do período em questão.

Analogamente, pode-se definir um intervalo de previsão para o próximo período como sendo $[t, t + k] \forall t \in \mathbb{T}$. Define-se $S_{t,k}$ como sendo um período no instante de tempo t de tamanho k minutos. Neste sentido, são construídas todas as séries relacionadas às variáveis de mercado e indicadores visando a previsão/classificação de tendência.

3.2.2 Rotulação dos Retornos Visando a Classificação de Tendência

Conforme previamente mencionado, o objetivo desse trabalho é classificar sinais relevantes de tendência para o próximo período de negociação utilizando dados de alta

⁴ Banco de dados *NoSQL* orientado a documentos de código aberto, multiplataforma e escrito na linguagem C++. Documentação disponível em [MongoDB.com](https://www.mongodb.com).

frequência. Em geral, dados de mercados financeiros são analisados utilizando retornos. Um retorno é a diferença entre duas cotações de preços subsequentes normalizadas em relação ao preço anterior. Assim, um retorno simples $R_t^{(1)}$ no tempo t é definido por:

$$R_t^{(1)} = \frac{p_t^{(1)} - p_{t-1}^{(1)}}{p_{t-1}^{(1)}} = \frac{p_t^{(1)}}{p_{t-1}^{(1)}} - 1. \quad (3.2)$$

sendo que $p_t^{(1)}$ denota o preço médio da moeda base (USD ou EUR) e é definido como:

$$p_t^{(1)} = \frac{P_{bid,t}^{(1)} + P_{ask,t}^{(1)}}{2}, \quad (3.3)$$

onde $P_{bid}^{(1)}$ é o melhor preço da compra e $P_{ask}^{(1)}$ é o melhor preço de venda. No entanto, grande parte da literatura se baseia no log dos retornos, conforme mostrado em [Aldridge \(2009\)](#). Dessa forma, os retornos simples subjacentes e os preços utilizados para calcular os retornos simples seguem uma distribuição log-normal. As distribuições log-normais refletem as distribuições reais dos preços dos ativos, visto que os preços são geralmente positivos. Isso não ocorreria em uma distribuição normal, pois a mesma permitiria a ocorrência de valores negativos. Assim, o log-retorno $r_t^{(1)}$ no tempo t pode ser definido como:

$$r_t^{(1)} = \ln R_t^{(1)} = \ln p_t^{(1)} - \ln p_{t-1}^{(1)}. \quad (3.4)$$

A partir do processo de amostragem definido pela Equação 3.1, pode-se construir séries de log-retornos para qualquer intervalo definido por k da seguinte forma:

$$r_{t,k}^{(1)} = \ln p_t^{(1)} - \ln p_{t-k}^{(1)}. \quad (3.5)$$

Seguindo essa premissa, foram construídas quatro séries de log-retornos considerando $k = \{1, 3, 5, 10\}$ para cada moeda base (BTCUSD e BTCEUR). Portanto, tem-se oito séries de log-retornos e, conseqüentemente, tem-se oito distribuições log-normais que podem ser visualizadas por meio das Figuras 3 e 4.

Ao analisar as distribuições log-normais de cada uma das séries, nota-se que grande parte dos log-retornos para todos os k estão muito próximos de zero, caracterizando condições de estabilidade no mercado entre períodos de negociação. Dessa forma, pode-se classificar os log-retornos em percentis q_β para diferentes valores de β e definir diferentes condições de mercado, criando dados que representem a tendência (rotulada como alta,

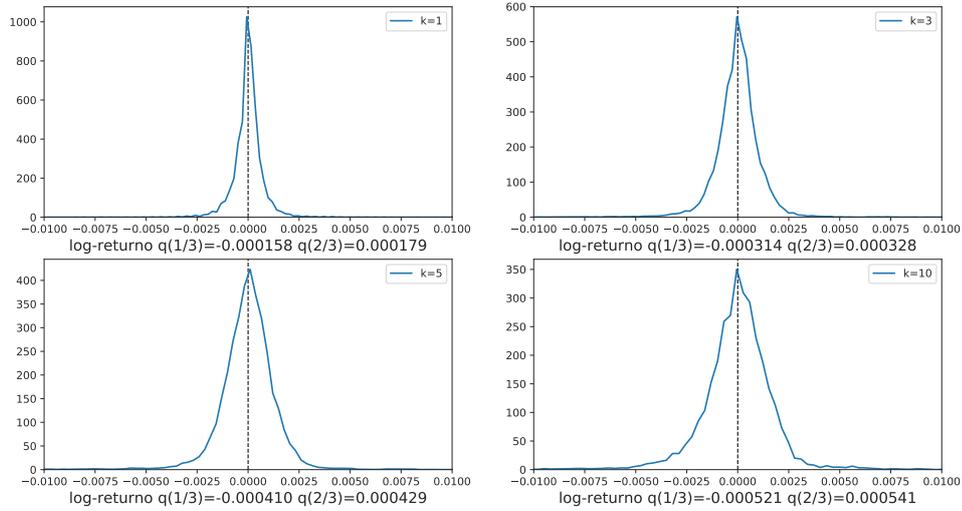


Figura 3 – Distribuição da série de log-retorno para $k = \{1, 3, 5, 10\}$ minutos em BTCUSD.

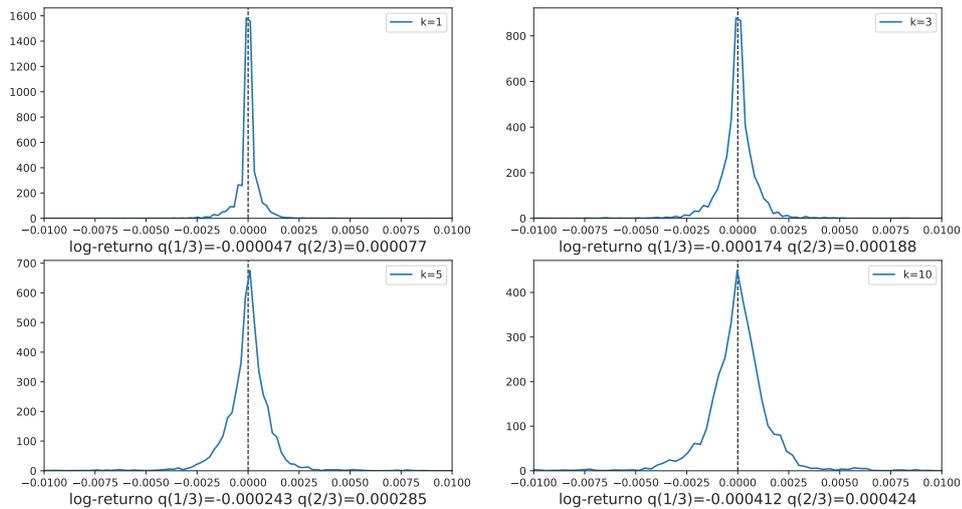


Figura 4 – Distribuição da série de log-retorno para $k = \{1, 3, 5, 10\}$ minutos em BTCEUR.

estável ou baixa). Assim, tais rótulos irão compor o espaço de saída $\mathbf{o}_{t,k}$ e podem ser descritos como:

$$\mathbf{o}_{t,k} = \begin{cases} (1, 0, 0) & , q_{2/3} \leq r_{t,k}^{(1)}, \\ (0, 1, 0) & , q_{1/3} \leq r_{t,k}^{(1)} < q_{2/3}, \\ (0, 0, 1) & , q_{1/3} < r_{t,k}^{(1)}. \end{cases} \quad (3.6)$$

Entretanto, é válido mencionar que o objetivo dessa dissertação é classificar a

série de log-retorno para o próximo período de negociação no intervalo $[t, t + k] \forall t \in \mathbb{T}$. Dessa forma, deve-se entender que $\mathbf{o}_{t,k}$ se refere ao próximo período de negociação. Além disso, $(\mathbf{o}_{i,t,k})_{i,\dots,C}, (C = 3) \in (0, 1)$ pode ser interpretado como sendo a probabilidade que o próximo log-retorno terá de pertencer a cada classe, onde:

$$\mathbf{o}_{t,k} = \begin{cases} -1 & , \text{ se } \max \{(\mathbf{o}_{i,t,k})_{i,\dots,C}\} = \mathbf{o}_{1,t,k}, \\ 1 & , \text{ se } \max \{(\mathbf{o}_{i,t,k})_{i,\dots,C}\} = \mathbf{o}_{3,t,k}, \\ 0 & , \text{ caso contrário.} \end{cases} \quad (3.7)$$

As Figuras 3 e 4 ainda mostram que para BTCEUR o mercado sofreu pouca variação nas frequências mais altas em relação a BTCUSD. De fato, entre um período e outro, não houve volatilidade suficiente para aumento nos retornos. No entanto, encontrar bons valores de retornos para otimizar e dar suporte a uma estratégia de negociação é essencial. Os log-retornos que estão entre $q_{1/3}$ e $q_{2/3}$ são úteis para remover ruídos inerentes às condições de alta ou baixa das séries de retornos, possibilitando uma melhor classificação da tendência. Vale lembrar que β é apenas um parâmetro e pode ser ajustado de acordo com a necessidade de uma estratégia de negociação.

3.2.3 Variáveis de Mercado Provenientes do Livro de Ofertas a Limite

A partir dos dados de LOB coletados, foi possível extrair as séries que definem as variáveis de mercado (entradas). Portanto, define-se um vetor \mathbf{i}_t como sendo:

$$\mathbf{i}_t = (r_t^{(1)}, sp_t^{(1)}, vi_t^{(L)}) \in \mathbb{R}^6, \quad (3.8)$$

onde $r_t^{(1)}$ é o log-retorno (definido pela Equação 3.4), $sp_t^{(1)}$ é o *bid-ask spread* que pode ser representado por:

$$sp_t^{(1)} = \frac{|P_{bid,t}^{(1)} - P_{ask,t}^{(1)}|}{p_t^{(1)}}, \quad (3.9)$$

e $vi_t^{(L)}$ é o *bid-ask volume imbalance* que pode ser obtido por meio de (NÃO EXISTE NA LITERATURA, ARGUMENTAR!!!):

$$vi_t^{(L)} = \frac{V_{bid,t}^{(L)} - V_{ask,t}^{(L)}}{V_{bid,t}^{(L)} + V_{ask,t}^{(L)}} \in [-1, 1], \quad (3.10)$$

sendo que o parâmetro L representa o nível do livro de ofertas a ser considerado para o cálculo do volume ($vi_t^{(L)}$). Assim, pode-se definir que $V_{bid,t}^{(L)} = \sum_{i=1}^L BTC_{bid}^{(i)}$ é o volume em BTC para a compra e $V_{ask,t}^{(L)} = \sum_{i=1}^L BTC_{ask}^{(i)}$ é o volume em BTC para a venda, ambos obtidos até o nível $L = \{5, 10, 50, 100\}$.

Na literatura, geralmente o *spread* é definido como sendo apenas a distância entre o melhor preço da compra ($P_{bid}^{(1)}$) e o melhor preço de venda ($P_{ask}^{(1)}$). No entanto, na Equação 3.9, o *spread* é definido como um percentual calculado em relação ao preço médio, pois no cálculo do custo real relacionado ao *spread* não necessariamente devem ser levados em consideração valores reais da compra e venda. A porcentagem do *spread* de compra e venda fornece um bom indicador de liquidez no mercado, visto que quanto menor esse valor, maior será a presença de liquidez.

Além de quantificar a liquidez, tem-se o *bid-ask volume imbalance* que é um indicador de desequilíbrio de volume, responsável por relacionar ordens a limite na compra e venda, conforme a Equação 3.10. Para o cálculo do volume foram consideradas as quantidades de BTC para cada nível do livro de ofertas. Dessa forma, considera-se que se o volume em BTC é baixo em um determinado nível do livro de ofertas, tanto na compra quanto na venda, a liquidez nesse mesmo nível seguirá o mesmo comportamento. Quanto mais próximo o $vi^{(L)}$ estiver de 1, maior será a quantidade de BTC que está sendo negociada na compra. Assim, $vi^{(L)}$ próximo de -1 representa negociações de venda e próximo de 0 demonstra equilíbrio nas negociações.

Dadas as características supramencionadas do *bid ask volume imbalance*, buscou-se analisar a distribuição de seus dados em relação aos L níveis do livro de ofertas, conforme mostrado na Figura 5.

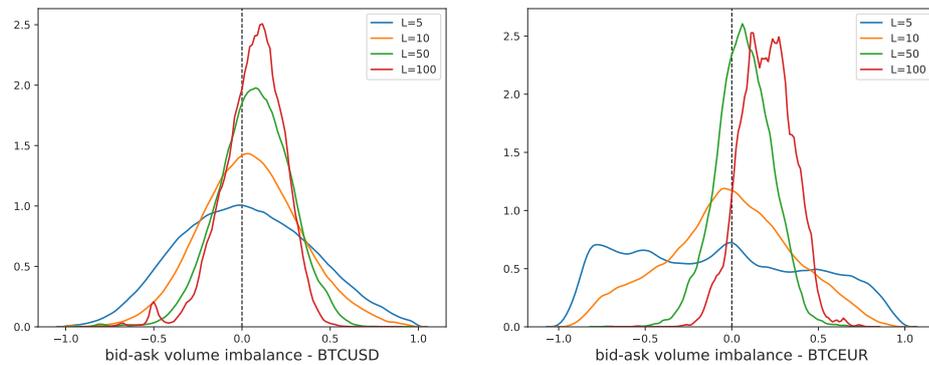


Figura 5 – Distribuição de *bid ask volume imbalance* no livro de ofertas para os níveis $L = \{5, 10, 50, 100\}$.

Nota-se que, à medida que L aumenta, menor é a variância, ou seja, mais conservadora será a métrica. Outra observação importante, é o fato de que as ordens que se encontram no topo do livro de ofertas, geralmente são aquelas que movimentam o mercado. No entanto, em termos de BTCEUR, o mercado se mostrou um tanto irregular em comparação aos mesmos níveis em termos de BTCUSD, o que pode caracterizar instabilidade e falta de liquidez no mercado. Ainda, pode-se afirmar que em altas frequências, o mercado para BTCEUR é menos eficiente em relação a BTCUSD, visto que a eficiência de

mercado está atrelada ao comportamento do mercado frente a um fator de desequilíbrio no livro de ofertas. Após a análise dos gráficos, considerou-se apenas o *bid-ask volume imbalance* para o nível $L = 10$ do livro de ofertas, visto que os demais níveis demonstram distribuições irregulares e, portanto, não apresentam a mesma relevância em termos de análise de tendência.

Após essas análises, utilizando o processo de amostragem definido pela Equação 3.1, além da série de retornos, criou-se diversas outras séries relacionadas às variáveis de mercado. Dessa forma, o vetor de dados de entrada $\mathbf{i}_{t,k}$ passa a ser definido como:

$$\mathbf{i}_{t,k} = \left(r_{t,k}^{(1)}, v_{t,k}^{(1)}, vi_{t,k}^{(10)}, sp_{t,k}^{(1)} \right) \in \mathbb{R}^4, \quad (3.11)$$

onde $r_{t,k}^{(1)}$ é o log-retorno (definido pela Equação 3.5), $v_{t,k}^{(1)}$ expressa a volatilidade que é definida como:

$$v_{t,k}^{(1)} = \frac{1}{T-1} \sum_{j=t-k}^T (r_j^{(1)} - \mu)^2, \quad (3.12)$$

$$\mu = \frac{1}{T} \sum_{j=t-k}^T r_j^{(1)}, \quad (3.13)$$

$$r_j^{(1)} = \ln p_j^{(1)} - \ln p_{j-1}^{(1)}, \quad (3.14)$$

considerando que $j > 1$ e $T = |S_{t,k}|$. Ainda, $vi_{t,k}^{(10)}$ e $sp_{t,k}^{(1)}$ são respectivamente calculados por:

$$vi_{t,k}^{(10)} = \frac{1}{T} \sum_{j=t-k}^T vi_j^{(10)}, \quad (3.15)$$

$$sp_{t,k}^{(1)} = \frac{1}{T} \sum_{j=t-k}^T sp_j^{(1)}. \quad (3.16)$$

Cabe comentar que, por meio da Equação 3.12, a volatilidade pode ser analisada a partir dos log-retornos no período $S_{t,k}$. Neste sentido, quanto maior a volatilidade, maior será a variação nos preços e instabilidade no mercado.

3.2.4 Indicadores Técnicos de Tendência

Na Seção 3.2.3 foram definidas as variáveis e séries de mercado obtidas do LOB, com o objetivo de classificar futuros retornos e obter a tendência do mercado. Entretanto,

para compor o conjunto de dados também foram considerados os indicadores técnicos e estatísticos de tendência. A partir disso, o vetor de entradas $\mathbf{i}_{t,k}$ passa a ser definido como:

$$\mathbf{i}_{t,k} = \left(r_{t,k}^{(1)}, (EMA_{M,t,k})_M, (RSI_{M,t,k})_M, (MACD_{M_1,M_2,t,k})_{M_1,M_2}, (BB_{M,\alpha,t,k})_{M,\alpha} \right) \in \mathbb{R}^{25} \quad (3.17)$$

onde $M = \{12, 24, 48\}$, representando o M -ésimo período de uma média móvel. Por meio dessa consideração, calcula-se a Média Móvel Exponencial (EMA, do inglês *Exponential Moving Average*), conforme segue:

$$EMA_{M,t,k} = (EMAr_{M,t,k}, EMAv_{M,t,k}, EMAvi_{M,t,k}, EMAsp_{M,t,k}) \in \mathbb{R}^4, \quad (3.18)$$

$$EMAr_{M,t,k} = \alpha_M r_{t-1,k}^{(1)} + (1 - \alpha_M) EMAr_{M,t-1,k}, \quad (3.19)$$

$$EMAv_{M,t,k} = \alpha_M v_{t-1,k}^{(1)} + (1 - \alpha_M) EMAv_{M,t-1,k}, \quad (3.20)$$

$$EMAvi_{M,t,k} = \alpha_M vi_{t-1,k}^{(10)} + (1 - \alpha_M) EMAvi_{M,t-1,k}, \quad (3.21)$$

$$EMAsp_{M,t,k} = \alpha_M v_{t-1,k}^{(1)} + (1 - \alpha_M) EMAsp_{M,t-1,k}, \quad (3.22)$$

onde α_M é especificado como $\alpha_M = 2/(1 + M)$. Médias móveis são utilizadas como indicadores estatísticos de tendência de séries temporais (HYNDMAN; ATHANASOPOULOS, 2018). Especificamente, médias móveis exponenciais têm como objetivo dar maior importância para eventos mais recentes no cálculo das médias.

Outro indicador utilizado foi o RSI (*Relative Strength Index*), introduzido por Wilder (1978). Tal indicador mede a magnitude das alterações recentes dos preços para avaliar as condições de sobre-compra (*overbought*) ou sobre-venda (*oversold*), conforme a Equação 3.23:

$$\begin{aligned} RSI_{M,t,k} &= PR_{M,t,k} / (PR_{M,t,k} + NR_{M,t,k}), \\ PR_{M,t,k} &= \sum_{j=1}^M \max \{ r_{t-j,k}^{(1)}, 0 \}, \\ NR_{M,t,k} &= \sum_{j=1}^M \max \{ -r_{t-j,k}^{(1)}, 0 \}. \end{aligned} \quad (3.23)$$

em que $PR_{M,t,k}$ denota o total de períodos de alta (retornos positivos) e $NR_{M,t,k}$ o total de períodos de baixa (retornos negativos). Especificamente, se $RSI > 0.7$, o ativo é interpretado como *overbought*. Se $RSI < 0.3$, então o ativo é considerado *oversold*.

Na sequência, foi calculado o indicador MACD (*Moving Average Convergence Divergence*) que é definido como a diferença entre uma EMA de curto e outra de longo prazo, conforme:

$$MACD_{M_1, M_2, t, k} = EMAr_{M_1, t, k} - EMAr_{M_2, t, k}, \quad (3.24)$$

sendo que $M_1 < M_2$. Vale comentar que a EMA foi calculada em conformidade com a Equação 3.19, utilizando $(M_1, M_2) = \{(2, 12), (2, 24), (4, 24)\}$.

O último indicador calculado é denominado por BB (*Bollinger Bands*), introduzido por Bollinger (1992). Tal indicador é corriqueiramente utilizado em análises técnicas de ativos, sendo definido por duas bandas de volatilidade, uma superior $UB_{M, \alpha, t, k}$ e outra inferior $LB_{M, \alpha, t, k}$, obtidas a partir de uma EMA que é definida pela Equação 3.20, onde α é a distância entre as bandas. O cálculo do BB é dado por:

$$\begin{aligned} BB_{M, \alpha, t, k} &= (UB_{M, \alpha, t, k}, LB_{M, \alpha, t, k}) \in \mathbb{R}^2, \\ UB_{M, \alpha, t, k} &= EMAr_{M, t, k} + \alpha EMAv_{M, t, k}, \\ LB_{M, \alpha, t, k} &= EMAr_{M, t, k} - \alpha EMAv_{M, t, k}. \end{aligned} \quad (3.25)$$

Nesta dissertação, empregou-se um $\alpha = 2$. Destaca-se que, em termos de comportamento de mercado, quando há maior volatilidade, as bandas aumentam. Por outro lado, as bandas se contraem em períodos menos voláteis.

3.3 Definição dos Conjuntos de Treinamento e Validação

Com base nos espaços de entrada, $\mathbf{i}_{t,k}$, e de saída, $\mathbf{o}_{t,k}$, definidos na Seção 3.2 para o conjunto de dados $\mathbf{d} = (\mathbf{i}_{t,k}, \mathbf{o}_{t,k})$, torna-se possível definir os conjuntos de treinamento e validação que serão empregados pelos algoritmos de AM.

As separações dos conjuntos de treinamento de validação foram feitas de forma semelhante ao processo de amostragem definido na Seção 3.2.1, ou seja, baseado em janela deslizante. Neste caso, os intervalos são definidos em m dias de treinamento e n dias de validação. Inicialmente, definiu-se $m = 5$ e $n = 1$, conforme exemplificado na Figura 6.

Neste exemplo, considera-se D_1, D_2, D_3, D_4 e D_5 como conjuntos de treinamento dos modelos, enquanto D_6 é o conjunto utilizado para validação na primeira iteração. A janela se move para a direita em um dia por vez, enquanto que o dia do lado esquerdo é descartado. Para cada deslocamento, foram calculadas:

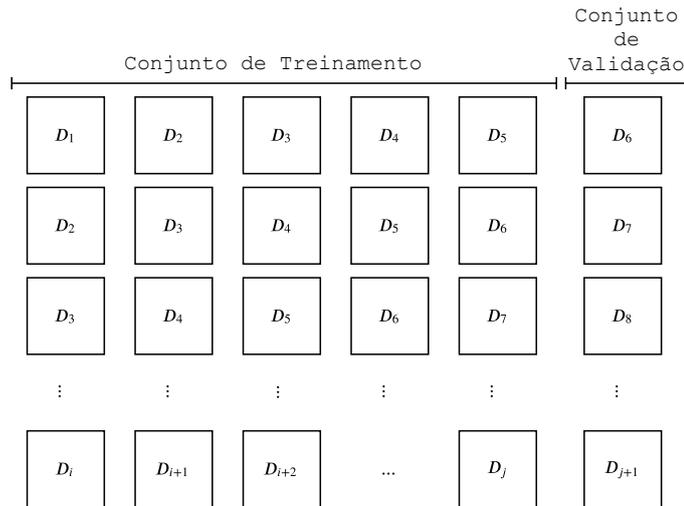


Figura 6 – Definição dos conjuntos de treinamento e validação.

- Estatísticas dos percentis referentes às tendências definidas pela Equação 3.6 utilizando dados de treinamento;
- Estatísticas para preenchimento de posições nulas nos conjuntos (ou *imputation*) a partir dos dados de treinamento. Foi utilizada a mediana para cada uma das colunas e os dados ausentes foram substituídos;
- Estatísticas para normalização dos dados (*zscore*) definida por $z = \frac{x-\mu}{\sigma}$, em que x é o valor de uma amostra a ser normalizada, μ é a média da amostra de treinamento, e σ o desvio padrão.

A partir dessas estatísticas, foram aplicadas transformações tanto nos dados de treinamento quanto nos de validação na ordem especificada. Para os experimentos foram aplicados diferentes valores para m , especificamente $m = \{5, 10, 15\}$ dias. Assim, foram gerados diferentes modelos com diversas configurações de treinamento, em que pôde ser avaliada a efetividade de cada um com diversas métricas de análise de desempenho, definidas na Seção 3.5.

3.4 Algoritmos de Aprendizado de Máquina

Todos os modelos de AM ajustados têm como objetivo generalizar comportamentos de preços e efetuar previsões de tendência para o próximo período de negociação $\mathbf{o}_{t,k}$ desejado, a partir de dados de entrada $\mathbf{i}_{t,k}$. Foram utilizadas duas abordagens de AM diferentes, visando analisá-las de forma comparativa. Assim, empregou-se uma RNA com arquitetura de múltiplas camadas e outra abordagem baseada em árvores, o XGBoost

(CHEN; GUESTRIN, 2016). De forma geral, para a metodologia proposta, os modelos de AM têm como objetivo descrever o seguinte mapeamento não linear:

$$f : \mathbf{i}_{t,k} \in \mathbb{R}^{25} \mapsto \mathbf{o}_{t,k} \in (0, 1)^3. \quad (3.26)$$

3.4.1 Redes Neurais Artificiais

Como mencionado no Capítulo 2.3, RNAs são amplamente utilizadas da literatura na previsão de séries de preços de ativos em mercados financeiros. Especificamente, o Perceptron de Múltiplas Camadas (MLP, do inglês *Multilayer Perceptron*) tem sido o algoritmo mais utilizado, devido à sua capacidade de generalização e precisão nas previsões de séries temporais (PAL; MITRA, 1992; NAKANO; TAKAHASHI; TAKAHASHI, 2018). O modelo de MLP proposto nessa dissertação consiste em 10 camadas ($L = 10$), i.e., entrada, ocultas e de saída, onde o número de nós (ou unidades) para a ℓ -ésima camada ($\ell = 1, \dots, 10$), denotado por $N^{[\ell]}$, é especificado da seguinte forma:

$$(N^{[\ell]})_{\ell=1,\dots,10} = (25, 40, 35, 30, 25, 20, 15, 10, 5, 3). \quad (3.27)$$

A partir da Equação 3.26, a MLP é descrita como:

$$\begin{aligned} \mathbf{x}^{[1]} &= \mathbf{i}_{t,k}, \\ \mathbf{x}^{[\ell]} &= \mathbf{h}^{[\ell]}(\mathbf{y}^{[\ell]}), \mathbf{y}^{[\ell]} = \mathbf{W}^{[\ell]} \mathbf{x}^{[\ell-1]} + \mathbf{b}^{[\ell]}, \ell = 2, \dots, L, \\ \mathbf{o}_{t,k} &= \mathbf{x}^{[L]} \end{aligned} \quad (3.28)$$

onde:

- $\mathbf{y}^{[\ell]} \in \mathbb{R}^{N^{[\ell]}}$, $\mathbf{W}^{[\ell]} \in \mathbb{R}^{N^{[\ell]} \times N^{[\ell-1]}}$, $\mathbf{b}^{[\ell]} \in \mathbb{R}^{N^{[\ell]}}$, $\mathbf{x}^{[\ell-1]} \in \mathbb{R}^{N^{[\ell-1]}}$, $\ell = 2, \dots, L$;
- $\mathbf{h}^{[\ell]} : \mathbb{R}^{N^{[\ell]}} \rightarrow \mathbb{R}^{N^{[\ell]}}$, $\ell = 2, \dots, L - 1$;
- $\mathbf{h}^{[L]} : \mathbb{R}^{N^{[L]}} \rightarrow (0, 1)^{N^{[L]}}$.

Além disso, as funções de ativação das camadas ocultas ($\ell = 2, \dots, L - 1$ camadas) e a camada de saída (a L -ésima camada), i.e., $\mathbf{h}^{[\ell]}$ ($\ell = 2, \dots, L - 1$) e $\mathbf{h}^{[L]}$, são as seguintes: linear retificado (*rectified linear*, ReLU), introduzidas por Glorot, Bordes e Bengio (2011), e *softmax* (GOODFELLOW; BENGIO; COURVILLE, 2016), respectivamente. A principal razão para utilizar ReLU é que a mesma permite contornar do problema do gradiente de fuga (ou *gradient vanish problem*), como é demonstrado em Glorot, Bordes e Bengio (2011). Como função de perda, empregou-se a entropia cruzada (ou *cross entropy*). A partir disso, pode-se definir que a MLP é uma função f não linear com os parâmetros $\theta = (\mathbf{W}^{[\ell]}, \mathbf{b}^{[\ell]})_{\ell=2,\dots,L}$, sendo que tais parâmetros são otimizados utilizando

gradiente estocástico (ou *stochastic gradient descent*, SGD) (KINGMA; BA, 2014). Ainda, foram utilizados *minibatches* com tamanho de 450 e uma quantidade de 500 épocas de treinamento para todas as configurações de execução da metodologia.

3.4.2 XGBoost – *Extreme Gradient Boosting*

O XGBoost é um dos mais recentes algoritmos de AM criados que se destacou devido à sua simplicidade de implementação, ajuste e poder preditivo para diferentes problemas. XGBoost significa *Extreme Gradient Boosting* onde o termo *Gradient Boosting* se origina de Friedman (2001). Basicamente, algoritmos de impulso (ou *boosting*) são métodos de *ensemble* que combinam um conjunto de modelos fracos para formar um único modelo forte (FREUND; SCHAPIRE et al., 1996) da seguinte forma:

$$\hat{\mathbf{y}}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^K f_k(\mathbf{x}_i), \quad (3.29)$$

$$\mathbf{x}_i = \mathbf{i}_{t,k}, \mathbf{o}_{t,k} = \mathbf{y}_i, \hat{\mathbf{o}}_{t,k} = \hat{\mathbf{y}}_i.$$

onde f_k é um modelo fraco e K é o número de modelos que compõem o *ensemble*. Em se tratando de um impulso por árvores (ou *tree boosting*), os modelos base são compostos por árvores de decisão que podem ser utilizadas tanto para regressão quanto para classificação. Em um certo nível, o XGBoost pode ser considerado como um *ensemble* de árvores de decisão. Dito isso, pode-se definir uma função objetivo e encontrar parâmetros ϕ que minimizam a seguinte função de perda:

$$L(\phi) = \sum_{i=1}^n l(\hat{\mathbf{y}}_i, \mathbf{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (3.30)$$

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2$$

onde $\Omega(f_k)$ é a complexidade do k -ésimo modelo de árvore, n é o tamanho da amostra, T é a quantidade de nós folhas das sub-árvores, ω é o peso de cada nó folha, γ controla a extensão da penalidade de complexidade em T , e λ controla o grau da regularização de f_k . No entanto, minimizar tal função de perda, dado pela Equação 3.30, com métodos tradicionais é um tanto mais complexo do que simplesmente utilizar o gradiente. Dessa forma, segundo Chen e Guestrin (2016), o XGBoost adiciona f_t que otimiza a função de perda original da seguinte forma:

$$L^t = \sum_{i=1}^n l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad (3.31)$$

de forma que $\hat{y}_i^{(t-1)}$ é a pontuação para a i -ésima amostra de treinamento na t -ésima iteração e f_t é um modelo base para o mesmo t . Fazendo isso, fixa-se o ajuste e adiciona-

se uma nova árvore por iteração. Dessa forma, é possível utilizar o gradiente de forma simplificada e objetiva com foco nos erros mais recentes em relação às iterações anteriores. O processo de corte e ramificação de cada nó folha é feito por maximização do ganho de informação (CHEN; GUESTRIN, 2016).

O XGBoost também pode sub-amostrar entre colunas, o que reduz a relevância de cada classificador base do *ensemble*, sendo outro parâmetro de ajuste para generalização dos modelos. Nesse contexto, a presente metodologia fixou as sub-amostras tanto para as colunas quanto para as instâncias (amostras) de treinamento em 80%, termo de regularização λ igual a 5 e um total de $K = 500$ classificadores base. Além disso, a altura máxima das sub-árvores foi fixada em 3.

3.5 Métricas de Avaliação de Desempenho

Em problemas de classificação binária, geralmente, a correção de uma classificação pode ser avaliada calculando-se o número de exemplos classificados corretamente (verdadeiros positivos, vp), o número de exemplos classificados corretamente que não pertencem à classe (verdadeiros negativos, vn), e exemplos que ou foram atribuídos incorretamente à classe positiva (falsos positivos, fp) ou que não foram classificados como exemplos da classe de dados em questão (falsos negativos, fn). A partir disso, a literatura (SOKOLOVA; LAPALME, 2009) define três principais métricas de desempenho que podem ser sumarizadas da seguinte forma:

- Precisão – o número de exemplos positivos classificados corretamente dividido pelo número de exemplos rotulados pelo modelo como positivos;
- Sensibilidade – o número de exemplos positivos classificados corretamente dividido pelo número de exemplos positivos nos dados;
- F1 – uma combinação dos anteriores.

Como demonstrado no Capítulo 3, a metodologia proposta consiste em, basicamente, classificar tendências em três classes de dados distintas, sendo um problema de aprendizado supervisionado multiclasse definido por \mathbf{d} . Nesse contexto, a tabela 1 apresenta todas as medidas utilizadas para avaliar o desempenho dos modelos. Para cada classe individual de dados C_i , sendo $i = 1, 2, 3$, a avaliação é definida por vp_i , fn_i , vn_i e fp_i que são calculados a partir das contagens para C_i . Em Sokolova e Lapalme (2009), os autores descrevem que o desempenho da classificação para problemas multiclasse é usualmente avaliado em duas formas: (i) utilizando-se a média das mesmas medidas calculadas para C_1, \dots, C_l (macro-média, indexado por M); ou (ii) realiza-se a soma cumulativa das

quantidades de vp_i , fn_i , vn_i , fp_i e então calcula-se tal medida de desempenho (micro-média, indexado por μ). A macro-média trata todas as classes igualmente, enquanto que a micro-média favorece as classes com maiores quantidades.

Nessa dissertação serão consideradas as métricas de avaliação de desempenho descritas de forma sumarizada na Tabela 1. Os resultados dos experimentos computacionais serão apresentados no Capítulo 4.

Tabela 1 – Métricas de avaliação de desempenho.

Métrica	Equação	Foco da avaliação
<i>Acurácia</i>	$\frac{\sum_{i=1}^l \frac{vp_i + vn_i}{vp_i + fn_i + fp_i + vn_i}}{l}$	Eficácia média por classe de um classificador
<i>Precisão$_{\mu}$</i>	$\frac{\sum_{i=1}^l vp_i}{\sum_{i=1}^l (vp_i + fp_i)}$	Assertividade dos rótulos da classe de dados com aqueles definidos pelos classificadores, se calculado a partir de somas das decisões para cada comportamento de tendência
<i>Sensibilidade$_{\mu}$</i>	$\frac{\sum_{i=1}^l vp_i}{\sum_{i=1}^l (vp_i + fn_i)}$	Eficácia de um classificador para identificar rótulos de uma classe de dados se calculado a partir das somas de decisões para cada comportamento de tendência
<i>F1$_{\mu}$</i>	$2 \frac{Precisão_{\mu} Sensibilidade_{\mu}}{Precisão_{\mu} + Sensibilidade_{\mu}}$	Relações entre os rótulos positivos dos dados e aqueles definidos por um classificador com base nas somas das decisões por comportamento de tendência
<i>Precisão$_M$</i>	$\frac{\sum_{i=1}^l \frac{vp_i}{vp_i + fp_i}}{l}$	Assertividade média por classe dos rótulos da classe de dados com aqueles definidos pelos classificadores
<i>Sensibilidade$_M$</i>	$\frac{\sum_{i=1}^l \frac{vp_i}{vp_i + fn_i}}{l}$	Eficácia média por classe de um classificador para identificar rótulos de classe
<i>F1$_M$</i>	$2 \frac{Precisão_M Sensibilidade_M}{Precisão_M + Sensibilidade_M}$	Relações entre os rótulos positivos dos dados e aqueles definidos por um classificador com base na média por classe

3.6 Detalhes de Implementação da Metodologia Proposta

Durante todo o processo de implementação, buscou-se construir estruturas que sejam facilmente replicáveis para todas as etapas da metodologia. Dessa forma, optou-se por utilizar alguns conceitos de programação funcional para criar conjuntos de funções e tipos para os algoritmos de AM e para as transformações nos conjuntos de dados, incluindo algumas estatísticas. Em contraste com algumas linguagens imperativas, as

computações são descritas em um formato de sequência de tarefas e depois executadas, onde podem mudar de estado (LIPOVACA, 2011). Na programação puramente funcional, como *Haskell*, não se descreve o que fazer com tais computações, mas uma definição do que elas são, como por exemplo um conjunto de funções e transformações em um conjunto de dados. Ao final, torna-se possível replicar as mesmas transformações nos dados, treinamentos e validações dos modelos para todos os dias de negociação da base de dados de forma independente.

Todas as partes do projeto, desde a captura dos dados, até os algoritmos de AM foram implementadas em *Python*. Para as estruturas funcionais, utilizou-se uma biblioteca denominada *Toolz*. Para as RNAs empregou-se especificamente o *Keras.io*, que é amplamente difundido na literatura e pode ser integrada ao *TensorFlow.org*. Para o *XGBoost*, considerou-se sua implementação oficial disponível em *XGBoost.ai*.

⁵ Disponível em toolz.readthedocs.io.

4 Experimentos Computacionais

Este capítulo sumariza todos os experimentos computacionais realizados mediante a metodologia proposta para a classificação de retornos em todos os dias de negociação definidos em \mathbb{T} . A princípio, são apresentados os resultados para o caso base (E1), comparando RNAs e XGBOOST para ambas as moedas (USD e EUR). Na sequência, são comparados os desempenhos dos modelos de AM para diferentes intervalos de previsão de tendência (E2). Após isso, foi variada a quantidade de dias que compõem o conjunto de treinamento (E3) e, desta forma, realiza-se uma nova análise comparativa em torno dos modelos de AM. Por último, foi ajustado um classificador por comportamento de tendência, abordagem definida como *one-vs.-rest* (E4), ou seja, para cada classificador, a classe em questão é ajustada contra todas as outras classes. Para todos os experimentos, considerou-se $n = 1$, ou seja, a previsão ocorre sempre para o próximo dia de negociação. É importante notar que, para cada m e k , tem-se uma nova configuração de execução dos experimentos em diferentes conjuntos definidos em \mathbf{d} .

4.1 Experimento 1 - Caso base (E1)

Para o caso base da metodologia proposta, fixou-se $m = 5$ dias de treinamento para a previsão do próximo dia de negociação com intervalo de $k = 1$ minuto, ou seja, a classificação de tendência ocorre de minuto em minuto a partir de um instante de tempo t . Nas Tabelas 2 e 3 são apresentados os resultados dos modelos de AM (RNA e XGBOOST) para ambas as moedas, USD e EUR respectivamente, considerando ainda a média e o desvio padrão para os três meses de dados.

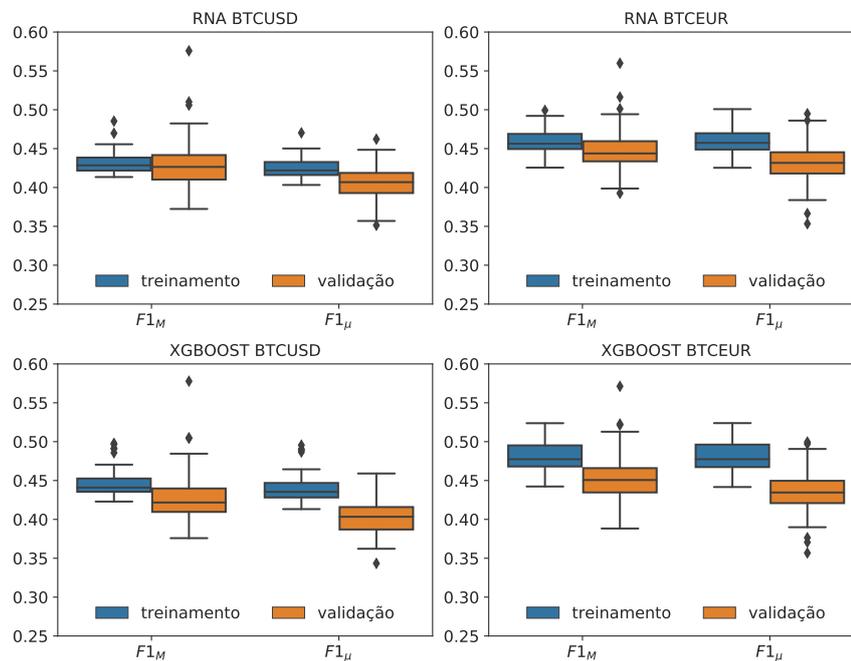
Tabela 2 – Resultados para BTCUSD com $m = 5$ dias e $k = 1$ minuto. Média e desvio padrão para 82 dias de negociação.

Métrica	RNA		XGBOOST	
	Treinamento	Validação	Treinamento	Validação
$Precisão_M$	0,4286±0,0128	0,4144±0,0205	0,4455±0,0158	0,4112±0,0204
$Precisão_\mu$	0,4312±0,0132	0,4299±0,0315	0,4460±0,0162	0,4270±0,0310
$Sensibilidade_M$	0,4315±0,0132	0,4119±0,0200	0,4462±0,0162	0,4088±0,0204
$Sensibilidade_\mu$	0,4312±0,0132	0,4299±0,0315	0,4460±0,0162	0,4270±0,0310
$F1_M$	0,4215±0,0126	0,4060±0,0211	0,4397±0,0164	0,4024±0,0219
$F1_\mu$	0,4312±0,0132	0,4299±0,0315	0,4462±0,0162	0,4270±0,0310

Na Figura 7 podem ser visualizados os resultados do experimento E1, de forma a melhor compreender a distribuição de F1.

Tabela 3 – Resultados para BTCEUR com $m = 5$ dias e $k = 1$ minuto. Média e desvio padrão para 83 dias de negociação.

Métrica	RNA		XGBOOST	
	Treinamento	Validação	Treinamento	Validação
$Precisão_M$	0,4610±0,0173	0,4344±0,0237	0,4817±0,0188	0,4398±0,0251
$Precisão_\mu$	0,4593±0,0162	0,4479±0,0274	0,4793±0,0173	0,4526±0,0303
$Sensibilidade_M$	0,4596±0,0163	0,4317±0,0245	0,4793±0,0173	0,4356±0,0259
$Sensibilidade_\mu$	0,4593±0,0161	0,4478±0,0274	0,4793±0,0173	0,4526±0,0303
$F1_M$	0,4593±0,0166	0,4308±0,0248	0,4791±0,0177	0,4350±0,0265
$F1_\mu$	0,4593±0,0162	0,4478±0,0274	0,4793±0,0173	0,4526±0,0302

Figura 7 – Distribuição dos resultados para a $F1$ com $k = 1$ minuto em termos de BTCUSD e BTCEUR.

Ao analisar estes primeiros resultados, é visível que para BTCUSD os resultados se mostram mais consistentes, com um desvio padrão menor se comparado a BTCEUR. Ainda, em termos de BTCUSD, obteve-se uma máxima de 0,4299 para a $F1_\mu$ no conjunto de validação. No entanto, para o BTCEUR, o desempenho nos conjuntos de validação tendem a ser melhores, com uma máxima de 0,4526 para a $F1_\mu$. A nível dos algoritmos de AM, a RNA tende a ser mais estável e sofre menos com sobreajustes (*overfitting*) nos conjunto de treinamento do que o XGBOOST. De forma geral, em termos de BTCEUR, os modelos sofreram mais com sobreajustes do que para BTCUSD.

Verificando a Figura 7, ao avaliar a média de cada métrica para os conjuntos de validação, é notável que os melhores resultados são obtidos para a micro-média, ou

seja, os modelos tendem a classificar melhor comportamentos de tendência ao considerar que todas as classes possuem as mesmas distribuições e balanceamento. No entanto, as diferenças para a macro-média é consideravelmente pequena, o que pode indicar uma positiva capacidade de generalização dos modelos de AM.

Outro ponto importante, também notável na Figura 7, é que apesar de os resultados para BTCUSD serem melhores do que para BTCEUR, este comportamento pode estar diretamente relacionado com as condições de mercado em ambas as moedas-base. Conforme mencionado na Seção 3.2, a partir do volume negociado em diferentes níveis do livro de ofertas e *spread*, o mercado para euro se mostra menos líquido e com uma eficiência de mercado menor quando comparado ao mercado para dólar, principalmente nas frequências maiores. Nesse contexto, os cortes utilizados para as classificações de tendência a partir dos percentis $q_{1/3}$ e $q_{2/3}$ são muito próximos de zero (ver Figuras 3 e 4), aproximando o problema em simplesmente classificar retornos positivos ou negativos. Consequentemente, os resultados para BTCEUR são melhores e ao mesmo tempo instáveis, com um sobreajuste maior nos conjuntos de treinamento dos modelos.

As Tabelas 2 e 3, mostram o desempenho dos classificadores levando-se em consideração a macro e micro-média para todas as métricas de avaliação (verificar a Tabela 1). A partir disso, observou-se que na maioria dos casos, ocorreram divergências em ambas as abordagens, macro e micro, principalmente nos conjuntos de validação. Isso ocorreu pelo desbalanceamento existente entre as classes de dados em cada um dos conjuntos. Na Figura 8, é possível verificar tal desbalanceamento em ambas as moedas base (dólar e euro). Em geral, nos conjuntos de treinamento, não houve desbalanceamento considerável, enquanto que nos conjuntos de validação, os desvios foram maiores, o que justifica essa diferença entre a macro e micro. Vale lembrar que a macro trata todas as classes igualmente, e a micro favorece as classes com maiores quantidades.

Em comparação com trabalhos citados na Seção 2.3, avaliou-se a acurácia definida pela Tabela 1 para a RNA em termos de BTCUSD com uma média de 0,4299 no conjunto de validação. Em Nakano, Takahashi e Takahashi (2018), a maior acurácia foi de 0,4400 com 52 mil *candles*, e em Shintate e Pichl (2019) com 0,5559 em aproximadamente 120 mil *candles*. A partir disso, é visto que o atual experimento possui um desempenho próximo e inferior ao dos trabalhos citados. No entanto, tal validação ocorreu em aproximadamente 5 milhões de pontos, o que gera oportunidades e comportamentos nunca vistos em comparação com os outros trabalhos. Vale ainda destacar que a metodologia do presente trabalho se mantém estável e com baixo sobreajuste nos conjuntos de treinamento nos 3 meses de negociação.

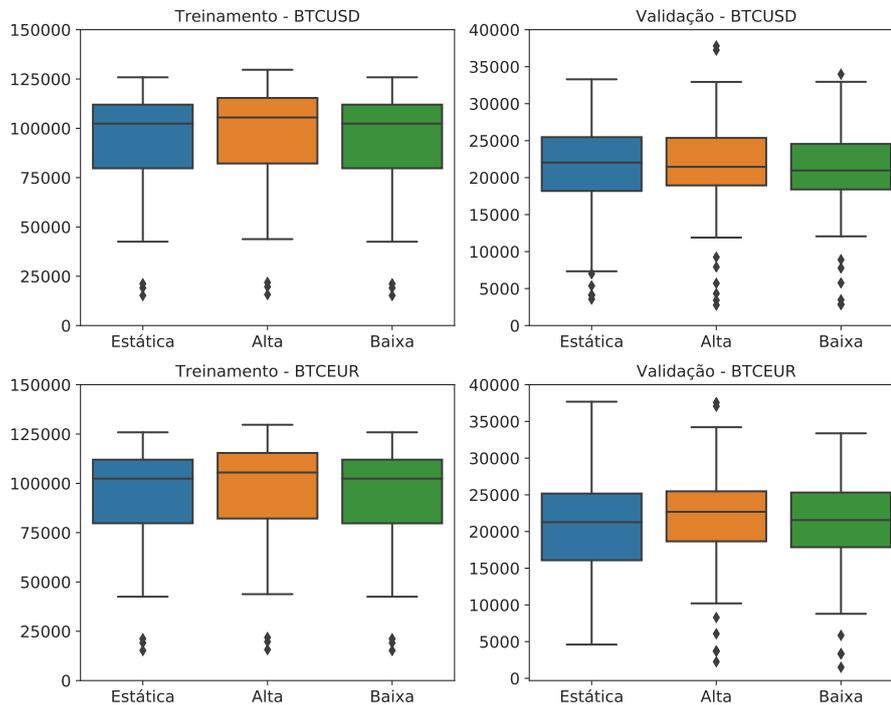


Figura 8 – Distribuição das classes de dados com $m = 5$ dias e $k = 1$ minuto em termos de BTCUSD e BTCEUR para ambos os conjuntos de treinamento e validação, em 82 dias de negociação.

4.2 Experimento 2 - Variação de Intervalos de Previsão (E2)

A partir do experimento E1, onde $m = 5$ dias para treinamento, varia-se os intervalos de previsão para ambos os algoritmos de AM e moedas-base, sendo $k = \{3, 5, 10\}$. As Tabelas 4 e 5 resumam os resultados (F1) para cada intervalo de previsão em termos de BTCUSD e BTCEUR, respectivamente. Além disso, nas Figuras 9 e 10, são plotadas as distribuições dos F1 para os modelos ao longo dos dias de negociação.

A partir destes resultados, nota-se que para maiores intervalos de previsão, os modelos obtiveram uma perda de desempenho considerável nos conjuntos de validação, o que condiz com a realidade do Bitcoin dado suas características de mercado, e.g. falta de liquidez e alta volatilidade nos preços em curtos períodos de tempo. A nível das moedas-base, especificamente para euro com um intervalo $k = 10$ minutos, a perda de desempenho foi a maior ($F1_{\mu} = 0,3803$ para a RNA e $F1_{\mu} = 0,3787$ para o XGBOOST) se comparado com a mesma moeda-base no intervalo de $k = 1$ minuto ($F1_{\mu} = 0,4478$ para a RNA e $F1_{\mu} = 0,4526$ para o XGBOOST).

Tabela 4 – Resultados para BTCUSD com $m = 5$ dias e variando $k = \{3, 5, 10\}$ minutos. Média e desvio padrão para 82 dias de negociação.

Métrica	RNA		XGBOOST	
	Treinamento	Validação	Treinamento	Validação
$k = 3$				
$F1_M$	0,4144±0,0139	0,3856±0,0222	0,4399±0,0241	0,3825±0,0254
$F1_\mu$	0,4191±0,0153	0,4100±0,0328	0,4454±0,0235	0,4081±0,0338
$k = 5$				
$F1_M$	0,4141±0,0186	0,3778±0,0254	0,4474±0,0308	0,3777±0,0253
$F1_\mu$	0,4198±0,0179	0,4038±0,0362	0,4525±0,0301	0,4062±0,0377
$k = 10$				
$F1_M$	0,4403±0,0178	0,3568±0,0279	0,4659±0,0347	0,3535±0,0338
$F1_\mu$	0,4432±0,0181	0,3761±0,0366	0,4699±0,0344	0,3802±0,0390

Tabela 5 – Resultados para BTCEUR com $m = 5$ dias e variando $k = \{3, 5, 10\}$ minutos. Média e desvio padrão para 83 dias de negociação.

Métrica	RNA		XGBOOST	
	Treinamento	Validação	Treinamento	Validação
$k = 3$				
$F1_M$	0,4104±0,0177	0,3824±0,0269	0,4598±0,0233	0,3886±0,0281
$F1_\mu$	0,4198±0,0154	0,4109±0,0347	0,4638±0,0236	0,4136±0,0313
$k = 5$				
$F1_M$	0,4224±0,0191	0,3731±0,0297	0,4633±0,0303	0,3770±0,0314
$F1_\mu$	0,4294±0,0183	0,3988±0,0330	0,4689±0,0293	0,4045±0,0322
$k = 10$				
$F1_M$	0,4401±0,0343	0,3518±0,0324	0,4774±0,0366	0,3514±0,0350
$F1_\mu$	0,4455±0,0337	0,3803±0,0379	0,4815±0,0362	0,3787±0,0404

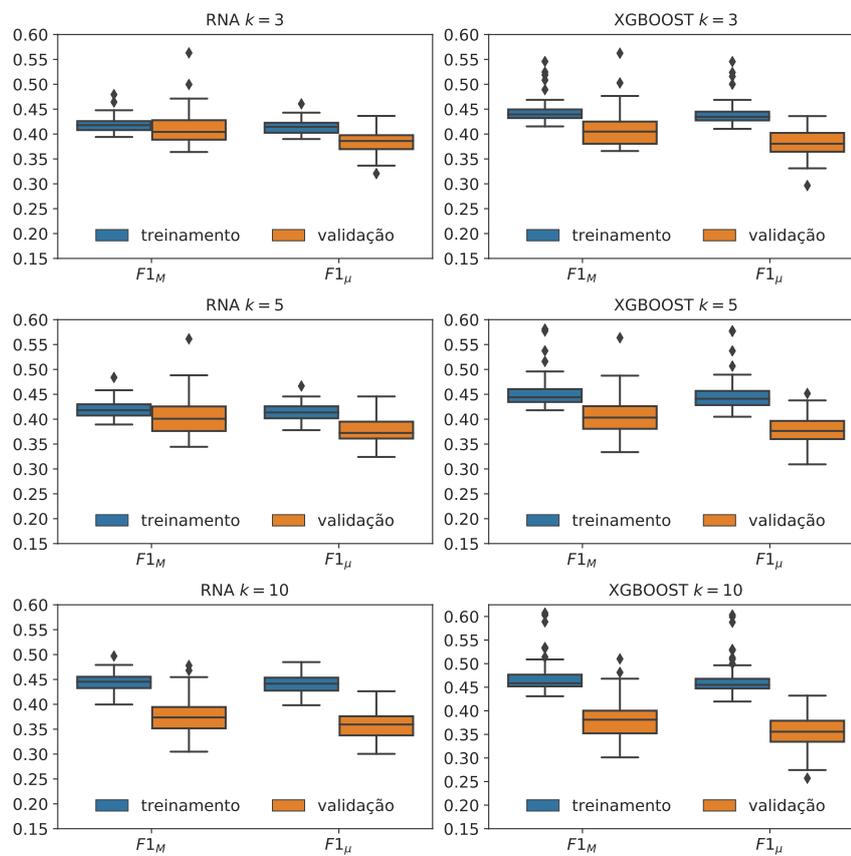


Figura 9 – Distribuição dos resultados para a $F1$ com $k = \{3, 5, 10\}$ minutos em termos de BTCUSD.

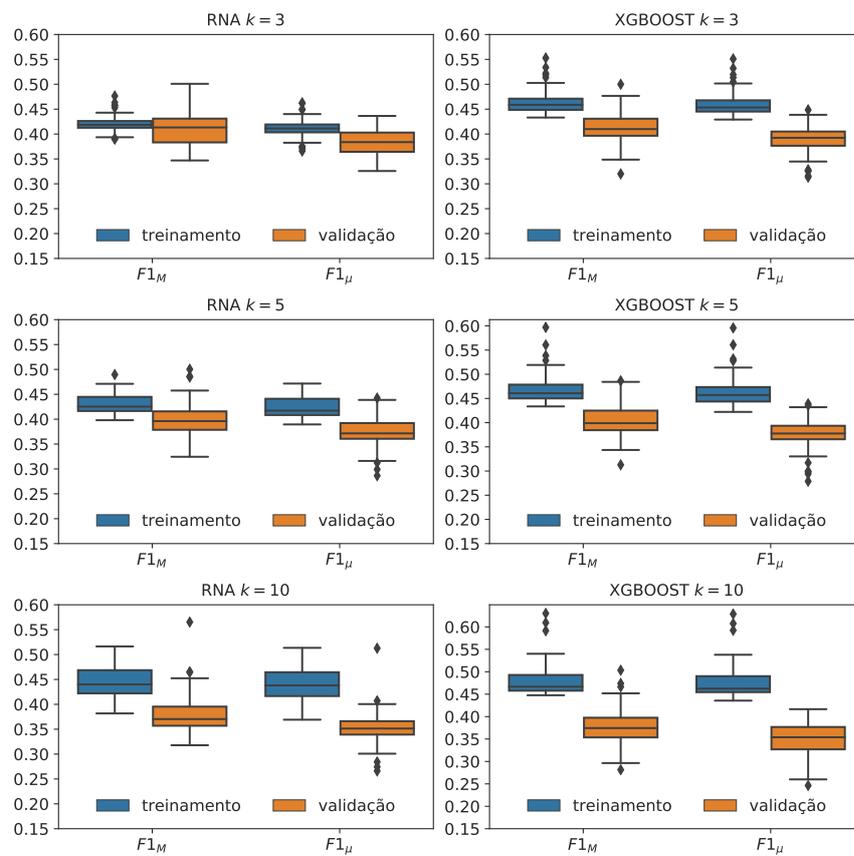


Figura 10 – Distribuição dos resultados para a $F1$ com $k = (3, 5, 10)$ minutos em termos de BTCEUR.

4.3 Experimento 3 - Variação da Quantidade de Dias de Treinamento (E3)

Aqui, os algoritmos de AM foram treinados com diferentes quantidades de dias de treinamento, especificamente $m = (10, 15)$ e $k = 1$. A hipótese para este experimento é que à medida que se aumenta a quantidade de dias do conjunto de treinamento, maior será o espaço de funções que os modelos irão mapear (LUXBURG; SCHÖLKOPF, 2011), evidenciando um poder de generalização maior. As Tabelas 6 e 7 mostram os resultados obtidos para ambas as moedas-base, BTCUSD e BTCEUR, respectivamente. Além disso, novamente são plotadas as distribuições para cada caso nas Figuras 11 e 12.

Tabela 6 – Resultados para BTCUSD variando $m = \{10, 15\}$ dias e $k = 1$ minuto. Média e desvio padrão para 82 dias de negociação.

Métrica	RNA		XGBOOST	
	Treinamento	Validação	Treinamento	Validação
$m = 10$				
$F1_M$	0,4244±0,0111	0,4030±0,0237	0,4324±0,0116	0,4019±0,0236
$F1_\mu$	0,4329±0,0120	0,4309±0,0304	0,4400±0,0127	0,4287±0,0296
$m = 15$				
$F1_M$	0,4251±0,0103	0,4019±0,0214	0,4292±0,0109	0,3990±0,0238
$F1_\mu$	0,4342±0,0118	0,4317±0,0291	0,4380±0,0130	0,4294±0,0284

Tabela 7 – Resultados para BTCEUR variando $m = \{10, 15\}$ dias e $k = 1$ minuto. Média e desvio padrão para 83 dias de negociação.

Métrica	RNA		XGBOOST	
	Treinamento	Validação	Treinamento	Validação
$m = 10$				
$F1_M$	0,4586±0,0095	0,4302±0,0241	0,4719±0,0096	0,4364±0,0246
$F1_\mu$	0,4578±0,0095	0,4465±0,0095	0,4722±0,0093	0,4540±0,0302
$m = 15$				
$F1_M$	0,4256±0,0161	0,4025±0,0336	0,4684±0,0065	0,4349±0,0251
$F1_\mu$	0,4360±0,0063	0,4291±0,0325	0,4687±0,0064	0,4533±0,0064

De forma geral, para BTCUSD, não houve alterações significativas tanto nos conjuntos de treinamento quanto nos de validação, o que demonstra uma certa consistência no mercado para o período analisado. No entanto, para BTCEUR, foram notadas algumas alterações consideráveis. Especificamente, a medida que se aumenta m (considerando os experimentos em E1 para BTCEUR), os modelos parecem sofrer menos com sobreajustes. Em contrapartida, ocorreram perdas consideráveis de desempenho nos conjuntos de

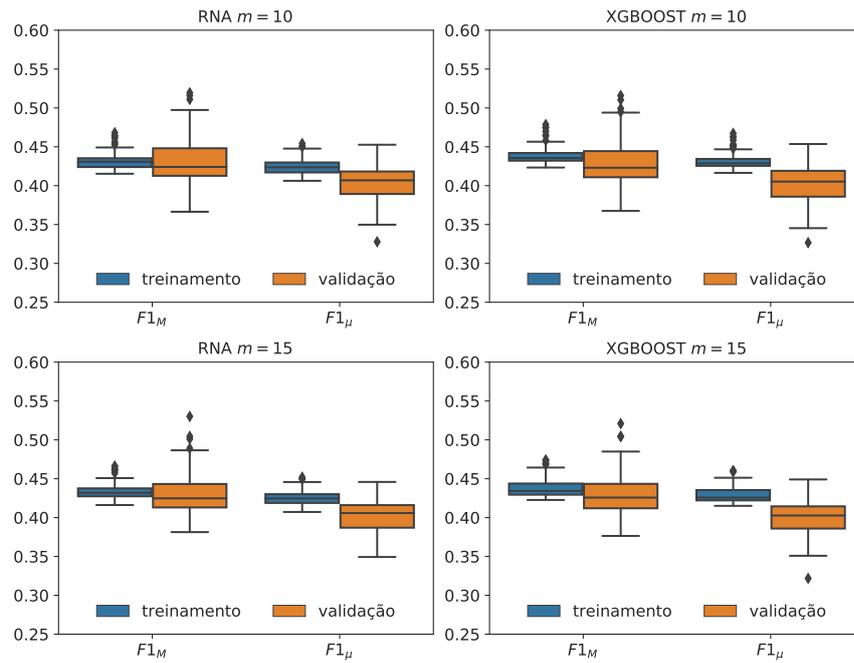


Figura 11 – Distribuição dos resultados para a $F1$ com $k = 1$ minuto e $m = \{10, 15\}$ dias em termos de BTCUSD.

validação. Em especial, para $m = 15$ com menor $F1_M = 0,4025$ e $F1_\mu = 0,4291$ para a RNA, se comparado a $m = 5$ com $F1_M = 0,4308$ e $F1_\mu = 0,4478$.

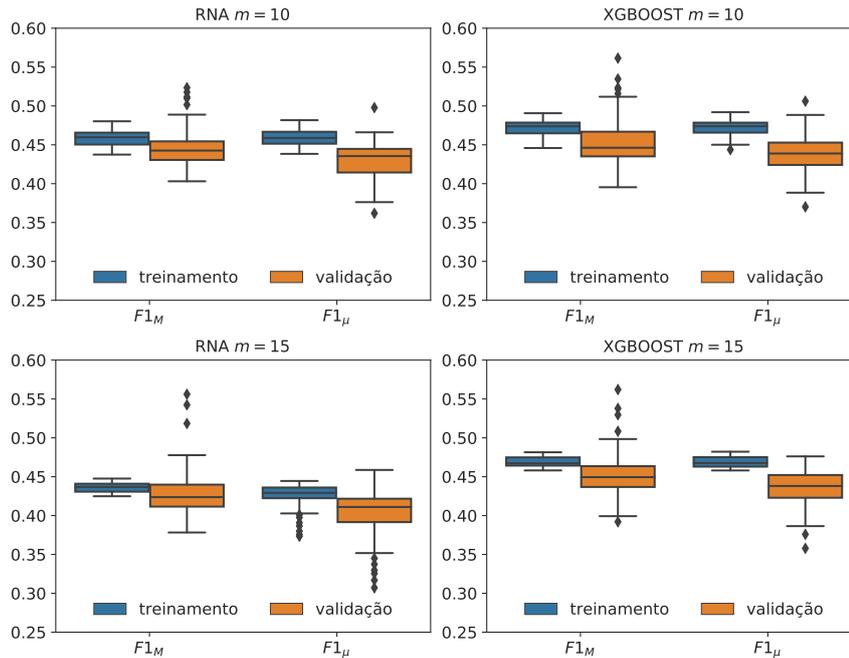


Figura 12 – Distribuição dos resultados para a $F1$ com $k = 1$ minuto e $m = (10, 15)$ dias em termos de BTCEUR.

4.4 Experimento 4 - *One-vs.-Rest* (E4)

Por último, ajustou-se um único modelo por comportamento de tendência (estática, baixa e alta) com objetivos de reduzir ambiguidades nos classificadores multiclasse (RIFKIN; KLAUTAU, 2004) e melhorar o desempenho da classificação de forma geral, abordagem denominada *one-vs.-rest*. As Tabelas 8 e 9 mostram os resultados obtidos em termos de BTCUSD e BTCEUR para $m = 5$ dias de treinamento e $k = 1$ minuto para a previsão do próximo período. Aqui, diferente dos experimentos anteriores, para validar o problema, utilizou-se como medida de desempenho a acurácia para verificar a classificação de forma geral e a $F1_M$ como uma métrica que é sensível ao desbalanceamento entre as classes de dados. Vale lembrar que, reduzir um problema de classificação multiclasse para outro de classificação binária, neste caso aplicando-se *one-vs.-rest*, irá gerar desbalanceamento nos dados.

Ao analisar estes resultados, em comparação com o experimento E1, verificou-se ganhos consideráveis em ambos os algoritmos de AM e moedas-base com uma acurácia máxima de 0,6775 (XGBoost) e $F1_M$ de 0,5050 (RNA) para a classe estática em termos de BTCUSD. Em termos de BTCEUR, a acurácia máxima foi de 0,7013 (XGBoost) e $F1_M$ de 0,5721 (XGBoost) para a classe estática. Porém, nota-se que, de forma geral, a RNA apresentou os melhores resultados. Outro ponto importante é a notável diferença entre a acurácia e a $F1_M$. Conforme previamente comentado, a $F1_M$ atende com mesma prioridade cada uma das classes, em contraste com a acurácia que representa uma média

Tabela 8 – Resultados – *One-vs.-Rest* – para BTCUSD com $m = 5$ dias e $k = 1$ minuto. Média e desvio padrão para 82 dias de negociação.

Tendência	RNA		XGBOOST	
	Treinamento	Validação	Treinamento	Validação
Estática				
<i>Acurácia</i>	0,6810±0,0082	0,6770±0,0757	0,6852±0,0093	0,6775±0,0766
<i>F1_M</i>	0,5246±0,0369	0,5050±0,0313	0,5186±0,0417	0,4908±0,0345
Alta				
<i>Acurácia</i>	0,6653±0,0028	0,6632±0,0465	0,6693±0,0051	0,6626±0,0466
<i>F1_M</i>	0,4577±0,0217	0,4520±0,0239	0,4586±0,0255	0,4460±0,0239
Baixa				
<i>Acurácia</i>	0,6733±0,0026	0,6718±0,0455	0,6761±0,0027	0,6714±0,0451
<i>F1_M</i>	0,4420±0,0212	0,4381±0,0212	0,4390±0,0178	0,4304±0,0210

Tabela 9 – Resultados – *One-vs.-Rest* – para BTCEUR com $m = 5$ dias e $k = 1$ minuto. Média e desvio padrão para 82 dias de negociação.

Tendência	RNA		XGBOOST	
	Treinamento	Validação	Treinamento	Validação
Estática				
<i>Acurácia</i>	0,7092±0,0143	0,6988±0,0777	0,7120±0,0133	0,7013±0,0790
<i>F1_M</i>	0,6014±0,0367	0,5699±0,0419	0,6038±0,0332	0,5721±0,0412
Alta				
<i>Acurácia</i>	0,6656±0,0027	0,6634±0,0542	0,6699±0,0048	0,6642±0,0542
<i>F1_M</i>	0,4682±0,0218	0,4571±0,0205	0,4506±0,0270	0,4339±0,0204
Baixa				
<i>Acurácia</i>	0,6747±0,0035	0,6732±0,0519	0,6780±0,0043	0,6733±0,0533
<i>F1_M</i>	0,4619±0,0298	0,4543±0,0261	0,4467±0,0265	0,4344±0,0266

geral.

5 Conclusões e Trabalhos Futuros

Este trabalho investigou a eficiência da classificação de tendências futuras em dados de alta frequência no mercado de Bitcoin em termos de dólar e euro utilizando técnicas de AM e microestrutura de mercado. Especificamente, foram empregados dois algoritmos, RNA e XGBOOST. A partir disso, os modelos de classificação foram construídos para a previsão de sinais relevantes de negociação, onde se espera retornos maiores entre um período e outro. Sendo tal modelagem um suporte para agentes tomadores de decisões em estratégias de negociação. Para tanto, foram consideradas variáveis de mercado (e.g. volume, liquidez e volatilidade), indicadores técnicos de tendência e a série de retornos com intervalos de 1, 3, 5 e 10 minutos.

Como resultado, foi possível validar os modelos de forma consistente ao longo do tempo em diferentes condições de mercado. Observou-se que, em vários contextos, o mercado para BTCEUR é menos eficiente do que para BTCUSD no período de dados analisado. Consequentemente, os resultados para BTCEUR se mostraram instáveis ao longo do tempo, enquanto que para BTCUSD, a metodologia se mostrou mais confiável e consistente. Além disso, a medida que se aumentou os intervalos de previsão, os modelos de AM tenderam a sofrer mais com sobreajustes nos conjuntos de treinamento. Em alguns casos, quando se aumentou a quantidade de dias de treinamento para a previsão do próximo dia de negociação, os modelos obtiveram perdas de desempenho (em termos de F1) se comparado ao caso base (experimento E1). Ainda, aplicando-se *one-vs.-rest* para cada comportamento de tendência, os resultados se mostraram consideravelmente melhores em relação ao caso base.

Como trabalhos futuros, pode-se utilizar os dados com o intuito de aprofundar a compreensão sobre o processo de formação de preços e, assim, buscar generalizar tais comportamentos em novos modelos de AM. Ademais, outra vertente a ser estudada seria baseada na criação de simulações do mercado com valores reais de liquidez e custos. Partindo-se dessas simulações, seria possível integrar tal modelagem a uma estratégia de negociação e realizar testes de rentabilidade ao longo do tempo.

Referências

- ALDRIDGE, I. *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*. Wiley, 2009. (Wiley Trading). ISBN 9780470579770. Disponível em: <<https://books.google.com.br/books?id=fEXKZAAVu4oC>>. Citado 5 vezes nas páginas 19, 24, 25, 26 e 32.
- ALDRIDGE, I. *High-frequency trading: a practical guide to algorithmic strategies and trading systems*. [S.l.]: John Wiley & Sons, 2013. Citado na página 19.
- BANDT, C.; POMPE, B. Permutation entropy: a natural complexity measure for time series. *Physical review letters*, APS, v. 88, n. 17, p. 174102, 2002. Citado na página 26.
- BAUR, D. G.; HONG, K.; LEE, A. D. Bitcoin: Medium of exchange or speculative assets? *Journal of International Financial Markets, Institutions and Money*, Elsevier, v. 54, p. 177–189, 2018. Citado na página 23.
- BAUWENS, L.; LAURENT, S.; ROMBOUTS, J. V. Multivariate garch models: a survey. *Journal of applied econometrics*, Wiley Online Library, v. 21, n. 1, p. 79–109, 2006. Citado na página 27.
- BOLLINGER, J. Using bollinger bands. *Stocks & Commodities*, v. 10, n. 2, p. 47–51, 1992. Citado na página 38.
- BOX, G. E.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015. Citado 2 vezes nas páginas 19 e 26.
- CHABOUD, A. P.; CHIQUOINE, B.; HJALMARSSON, E.; VEGA, C. Rise of the machines: Algorithmic trading in the foreign exchange market. *The Journal of Finance*, Wiley Online Library, v. 69, n. 5, p. 2045–2084, 2014. Citado 2 vezes nas páginas 19 e 24.
- CHEAH, E.-T.; FRY, J. Speculative bubbles in bitcoin markets? an empirical investigation into the fundamental value of bitcoin. *Economics Letters*, Elsevier, v. 130, p. 32–36, 2015. Citado na página 23.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: ACM. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.], 2016. p. 785–794. Citado 4 vezes nas páginas 19, 40, 41 e 42.
- CIAIAN, P.; RAJCANIOVA, M.; KANCS, d. The economics of bitcoin price formation. *Applied Economics*, Taylor & Francis, v. 48, n. 19, p. 1799–1815, 2016. Citado na página 26.
- DACOROGNA, M.; GENÇAY, R.; MULLER, U. A.; OLSEN, R.; PICTET, O. *An introduction to high-frequency finance*. [S.l.]: Elsevier, 2001. Citado na página 25.
- DIMPFL, T. Bitcoin market microstructure. 2017. Citado 4 vezes nas páginas 19, 23, 24 e 29.

- FETTE, I.; MELNIKOV, A. *The websocket protocol*. [S.l.], 2011. Citado na página 30.
- FREUND, Y.; SCHAPIRE, R. E. et al. Experiments with a new boosting algorithm. In: CITESEER. *icml*. [S.l.], 1996. v. 96, p. 148–156. Citado na página 41.
- FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, JSTOR, p. 1189–1232, 2001. Citado na página 41.
- GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. [S.l.: s.n.], 2011. p. 315–323. Citado na página 40.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016. Citado na página 40.
- HAGSTRÖMER, B.; NORDÉN, L. The diversity of high-frequency traders. *Journal of Financial Markets*, Elsevier, v. 16, n. 4, p. 741–770, 2013. Citado na página 24.
- HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. [S.l.]: OTexts, 2018. Citado na página 37.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 41.
- LABOISSIERE, L. A.; FERNANDES, R. A.; LAGE, G. G. Maximum and minimum stock price forecasting of brazilian power distribution companies based on artificial neural networks. *Applied Soft Computing*, Elsevier, v. 35, p. 66–74, 2015. Citado na página 27.
- LIPOVACA, M. *Learn you a haskell for great good!: a beginner's guide*. [S.l.]: no starch press, 2011. Citado na página 44.
- LUXBURG, U. V.; SCHÖLKOPF, B. Statistical learning theory: Models, concepts, and results. In: *Handbook of the History of Logic*. [S.l.]: Elsevier, 2011. v. 10, p. 651–706. Citado na página 52.
- MADHAVAN, A. Market microstructure: A survey. *Journal of financial markets*, Elsevier, v. 3, n. 3, p. 205–258, 2000. Citado 2 vezes nas páginas 25 e 26.
- MALKIEL, B. G.; FAMA, E. F. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, Wiley Online Library, v. 25, n. 2, p. 383–417, 1970. Citado 3 vezes nas páginas 19, 25 e 26.
- MARTINEZ, L. C.; HORA, D. N. da; PALOTTI, J. R. d. M.; MEIRA, W.; PAPPA, G. L. From an artificial neural network to a stock market day-trading system: A case study on the bm&f bovespa. In: IEEE. *2009 International Joint Conference on Neural Networks*. [S.l.], 2009. p. 2006–2013. Citado na página 27.
- NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. Working Paper, 2008. Citado 2 vezes nas páginas 19 e 23.
- NAKANO, M.; TAKAHASHI, A.; TAKAHASHI, S. Bitcoin technical trading with artificial neural network. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 510, p. 587–609, 2018. Citado 5 vezes nas páginas 20, 27, 31, 40 e 47.

- OLIVEIRA, F. A. de; NOBRE, C. N.; ZARATE, L. E. Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index—case study of petr4, petrobras, brazil. *Expert Systems with Applications*, Elsevier, v. 40, n. 18, p. 7596–7606, 2013. Citado na página 27.
- PAL, S. K.; MITRA, S. Multilayer perceptron, fuzzy sets, classification. 1992. Citado na página 40.
- PENG, Y.; ALBUQUERQUE, P. H. M.; SÁ, J. M. C. de; PADULA, A. J. A.; MONTENEGRO, M. R. The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression. *Expert Systems with Applications*, Elsevier, v. 97, p. 177–192, 2018. Citado 3 vezes nas páginas 20, 27 e 31.
- RIFKIN, R.; KLAUTAU, A. In defense of one-vs-all classification. *Journal of machine learning research*, v. 5, n. Jan, p. 101–141, 2004. Citado na página 54.
- SENSOY, A. The inefficiency of bitcoin revisited: A high-frequency analysis with alternative currencies. *Finance Research Letters*, Elsevier, 2018. Citado 3 vezes nas páginas 19, 24 e 26.
- SHAH, D.; ZHANG, K. Bayesian regression and bitcoin. In: IEEE. *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*. [S.l.], 2014. p. 409–414. Citado na página 27.
- SHINTATE, T.; PICHL, L. Trend prediction classification for high frequency bitcoin time series with deep learning. *Journal of Risk and Financial Management*, Multidisciplinary Digital Publishing Institute, v. 12, n. 1, p. 17, 2019. Citado 4 vezes nas páginas 20, 27, 31 e 47.
- SILVA, E.; CASTILHO, D.; PEREIRA, A.; BRANDAO, H. A neural network based approach to support the market making strategies in high-frequency trading. In: IEEE. *Neural Networks (IJCNN), 2014 International Joint Conference on*. [S.l.], 2014. p. 845–852. Citado 3 vezes nas páginas 25, 26 e 27.
- SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information processing & management*, Elsevier, v. 45, n. 4, p. 427–437, 2009. Citado na página 42.
- WILDER, J. W. *New concepts in technical trading systems*. [S.l.]: Trend Research, 1978. Citado na página 37.
- ZUO, Y.; KITA, E. Stock price forecast using bayesian network. *Expert Systems with Applications*, Elsevier, v. 39, n. 8, p. 6729–6737, 2012. Citado na página 26.