

Universidade Federal Fluminense

THIAGO BRUM LEITE

Modelagem Computacional para Previsão do
Preço do Bitcoin via Machine Learning e
Séries Temporais

VOLTA REDONDA

2024

THIAGO BRUM LEITE

Modelagem Computacional para Previsão do Preço do Bitcoin via Machine Learning e Séries Temporais

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Métodos Computacionais Aplicados à Matemática Estocástica/Estatística

Orientador:

Eliane da Silva Christo

Coorientador:

Cecilia Toledo Hernandez

UNIVERSIDADE FEDERAL FLUMINENSE

VOLTA REDONDA

2024

Ficha catalográfica automática - SDC/BEM
Gerada com informações fornecidas pelo autor

L533m Leite, Thiago Brum
Modelagem Computacional para Previsão do Preço do Bitcoin
via Machine Learning e Séries Temporais / Thiago Brum Leite. -
2024.
103 f.

Orientador: Eliane da Silva Christo.
Coorientador: Cecilia Toledo Hernandez.
Dissertação (mestrado)-Universidade Federal Fluminense,
Escola de Engenharia Industrial e Metalúrgica de Volta
Redonda, Volta Redonda, 2024.

1. Métodos Computacionais. 2. Matemática Estocástica. 3.
Probabilidade e Estatística. 4. Séries Temporais. 5.
Produção intelectual. I. Christo, Eliane da Silva,
orientadora. II. Hernandez, Cecilia Toledo, coorientadora.
III. Universidade Federal Fluminense. Escola de Engenharia
Industrial e Metalúrgica de Volta Redonda. IV. Título.


CDD - XXX

Modelagem Computacional para Previsão do Preço do Bitcoin via Machine Learning e Séries Temporais

Thiago Brum Leite


Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Métodos Computacionais Aplicados à Matemática Estocástica/Estatística.

Aprovada por:

Documento assinado digitalmente
 **ELIANE DA SILVA CHRISTO**
Data: 10/05/2024 10:17:33-0300
Verifique em <https://validar.iti.gov.br>


Prof. Eliane da Silva Christo, D.Sc. / MCCT-UFF

(Orientador)


Documento assinado digitalmente
 **CECILIA TOLEDO HERNANDEZ**
Data: 14/05/2024 15:16:03-0300
Verifique em <https://validar.iti.gov.br>

Prof. Cecilia Toledo Hernandez D.Sc. / MCCT-UFF

(Coorientador)

Documento assinado digitalmente
 **FLAVIA VITAL JANUZZI**
Data: 13/05/2024 08:08:11-0300
Verifique em <https://validar.iti.gov.br>

Prof. Flavia Vital Januzzi D.Sc. / FACC-UFJF

Documento assinado digitalmente
 **WESLEY LUIZ DA SILVA ASSIS**
Data: 11/05/2024 18:36:20-0300
Verifique em <https://validar.iti.gov.br>

Prof. Wesley Luiz da Silva Assis D.Sc. / MCCT-UFF

Volta Redonda, 06 de Maio de 2024.

"A matemática é o alfabeto com o qual Deus escreveu o universo."

Galileu Galilei.

Agradecimentos

Gostaria de expressar minha sincera gratidão a todas as pessoas que contribuíram de maneira significativa para a realização deste trabalho de dissertação. Esta jornada acadêmica foi marcada por inúmeros desafios, aprendizados e, acima de tudo, pela colaboração e apoio de pessoas notáveis.

Primeiramente, agradeço a Deus por me conceder o fôlego da vida e pela preciosa oportunidade de crescer e me desenvolver como ser humano e pesquisador ao longo da trajetória do mestrado.

Agradeço à minha orientadora, Eliane da Silva Christo, pela orientação dedicada, sabedoria e paciência ao longo deste processo. Suas valiosas sugestões e insights foram fundamentais para o desenvolvimento deste estudo.

Minha gratidão se estende aos colegas de curso, cuja troca de conhecimento e experiências tornou esta jornada mais enriquecedora e motivadora.

Agradeço também à minha família, especialmente à minha esposa e filhos, que foram verdadeiros alicerces e meus maiores incentivadores para concluir este curso.

À Universidade Federal Fluminense, aos professores do MCCT, que muito me ensinaram e agregaram para este projeto. Ao coordenador, Prof. Panters Rodrigues Bermudez, e à dedicada secretária Camila de Souza e Silva, expresso meu reconhecimento pela condução, organização, diligência, celeridade e excelência em todas as demandas administrativas do curso.

Lembro com apreço e saudade do Prof. Diomar Cesar Lobão (em memória), cujo incentivo e apoio foram cruciais para superar desafios iniciais do curso.

Por fim, dedico este trabalho aos professores, pesquisadores e instituições que, por meio de seus estudos e contribuições científicas, pavimentaram o caminho para que esta pesquisa pudesse ser realizada.

A todos, o meu mais profundo agradecimento. Este trabalho não seria possível sem a colaboração e apoio de cada um de vocês.

Resumo

O dinheiro, uma qualidade intrínseca a diferentes bens, variando em graus conforme tempo e local, representa um conceito fundamental na economia moderna. Ao longo da história, diversos ativos desempenharam o papel de dinheiro, desde *commodities* como o ouro e o sal até representações fiduciárias impressas. Recentemente, as criptomoedas surgiram como uma forma de dinheiro intangível, desafiando concepções tradicionais de moeda. Dado o crescente envolvimento de investidores nas criptomoedas, em especial o Bitcoin, a capacidade de prever seu preço tornou-se crucial para maximizar a lucratividade. Esta pesquisa buscou construir e comparar modelos baseados em análise estatística linear e aprendizado de máquina, visando aprimorar a precisão das previsões do preço do Bitcoin. A hipótese subjacente era que um modelo híbrido, combinando modelos lineares e de machine learning, poderia proporcionar estimativas mais precisas. Para tanto, uma Revisão Sistemática foi conduzida para identificar e analisar os métodos mais utilizados na literatura científica para prever valores do Bitcoin. Foram desenvolvidos, implementados e comparados algoritmos com seis diferentes modelos, sendo dois modelos lineares (ARIMA e SARIMA), dois modelos de aprendizado de máquina (RNN e LSTM) e dois modelos híbridos (RNN+ARIMA e LSTM+ARIMA), a fim de melhorar a acurácia das previsões. Os resultados revelaram que os modelos baseados em machine learning superaram aqueles fundamentados em estatística linear. No entanto, os modelos híbridos destacaram-se, apresentando desempenho superior aos modelos puramente baseados em machine learning, onde o modelo LSTM+ARIMA apresentou uma melhora de 31,48% em relação ao modelo LSTM; e o modelo RNN+ARIMA apresentou uma melhora de 76,09% no desempenho, em relação ao modelo RNN. Dentre todos os modelos implementados o que obteve os melhores resultados foi o RNN+ARIMA, apresentando um desempenho 70% superior em relação ao LSTM+ARIMA. A partir desta conclusão foi realizada uma simulação de investimento, por um período arbitrário de 30 dias, baseada nas previsões do modelo RNN+ARIMA e a mesma obteve um retorno financeiro de 56,12% sobre o capital investido. Concluiu-se que, apesar de modelos puramente lineares não fornecerem resultados satisfatórios, ao utilizar estes modelos juntamente de modelos de aprendizado de máquina, observa-se uma melhoria significativa nas previsões. Esta constatação sugere que a integração de métodos lineares e de machine learning representa uma área promissora para investigações futuras.

Abstract

Money, an intrinsic quality to different goods, varying in degrees over time and place, represents a fundamental concept in modern economics. Throughout history, various assets have played the role of money, from commodities like gold and salt to printed fiat representations. Recently, cryptocurrencies have emerged as a form of intangible money, challenging traditional conceptions of currency. Given the increasing involvement of investors in cryptocurrencies, especially Bitcoin, the ability to predict its price has become crucial for maximizing profitability. This research sought to build and compare models based on linear statistical analysis and machine learning, aiming to enhance the accuracy of Bitcoin price predictions. The underlying hypothesis was that a hybrid model, combining linear and machine learning models, could provide more accurate estimates. To do so, a Systematic Review was conducted to identify and analyze the most used methods in the scientific literature for predicting Bitcoin values. Algorithms with six different models were developed, implemented, and compared, including two linear models (ARIMA and SARIMA), two machine learning models (RNN and LSTM), and two hybrid models (RNN+ARIMA and LSTM+ARIMA), in order to improve prediction accuracy. The results revealed that machine learning-based models outperformed those grounded in linear statistics. However, hybrid models stood out, showing superior performance to purely machine learning-based models, where the LSTM+ARIMA model showed an improvement of 31.48% compared to the LSTM model, and the RNN+ARIMA model showed an improvement of 76.09% in performance compared to the RNN model. Among all implemented models, the one that achieved the best results was the RNN+ARIMA, with a performance 70% superior to the LSTM+ARIMA. Based on this conclusion, an investment simulation was conducted for an arbitrary period of 30 days, based on predictions from the RNN+ARIMA model, and it yielded a financial return of 56.12% on the invested capital. It was concluded that, although purely linear models do not provide satisfactory results, when using these models together with machine learning models, a significant improvement in predictions is observed. This finding suggests that the integration of linear and machine learning methods represents a promising area for future investigations.

Palavras-chave

1. Criptomoedas
2. ARIMA
3. LSTM Long Short-Term Memory
4. RNN Recurrent Neural Network
5. Deep Learning

Lista de Siglas e Abreviaturas

BCB	- Banco Central do Brasil
FED	- Federal Reserve Board (Sistema de Reserva Federal dos Estados Unidos)
BOVESPA	- Bolsa de Valores do Estado de São Paulo
ARIMA	- Autoregressive Integrated Moving Average (Auto-Regressivo Integrado de Médias Móveis)
SARIMA	- Seasonal Autoregressive Integrated Moving Average (Auto-Regressivo Integrado de Médias Móveis com sazinalidade)
RNN	- Recurrent Neural Network (Rede Neural Recorrente)
LSTM	- Long Short-Term Memory (Memória de Longo e Curto Prazo)
AR	- Autoregressive (Auto-Regressivo)
MA	- Moving Average (Médias Móveis)
RNA	- Redes Neurais Artificiais
MLP	- Multilayer Perceptron (Perceptron Multicamadas)
FFNN	- Feed-forward Neural Network (Redes Neurais Feedforward)
SVM	- Support Vector Machine (Máquina de Suporte de Vetores)
HMM	- Hidden Markov Model (Modelo Oculto de Markov)
RS	- Revisão Sistemática
BTC	- Bitcoin
USD	- Dólar Americano
DL	- Deep Learning (Aprendizado Profundo)
MAPE	- Mean Average Percent Error (Erro Absoluto Percentual Médio)
MAE	- Mean Average Error (Erro Absoluto Médio)
RMSE	- Root Mean Square Error (Raíz Quadrada do Erro Médio)

Lista de Figuras

1.1	Perda de poder de compra do Real e Dólar. Fonte: BCB e FED, Elaborado por: F. Ulrich, 2021	18
1.2	Estimativa de Bitcoins minerados por ano Fonte: Próprio Autor	19
1.3	Número de pesquisas por ano Fonte: Próprio Autor	21
1.4	Crescimento do Bitcoin no ano de 2017	22
2.1	Tendência do BitCoin entre 2014 e 2023 Fonte: Próprio Autor	32
2.2	Autocorrelação do BitCoin Fonte: Próprio Autor	33
2.3	Métodos mais utilizados Fonte: Próprio Autor	39
2.4	Gráfico <i>CandleStick</i> do Bitcoin (1 dia) Fonte: TradingView	45
2.5	Modelo de RNN Fonte: [50]	61
2.6	Informação relevante com lacuna longa Fonte: [50]	62
2.7	Informação relevante com lacuna curta Fonte: [50]	63
2.8	RNN com uma camada Fonte: [50]	64
2.9	LSTM com quatro camadas Fonte: [50]	64
2.10	Funcionamento de uma LSTM Fonte: [50]	66
3.1	Dados Históricos do BitCoin Fonte: Próprio Autor	70
3.2	Decomposição da Série Temporal Fonte: Próprio Autor	71
3.3	Autocorrelação da Série do BitCoin Fonte: Próprio Autor	73
3.4	Autocorrelação da Série Diferenciada do BitCoin Fonte: Próprio Autor	74
4.1	Comparação entre as previsões do modelo ARIMA e os valores reais Fonte: Próprio Autor	84

4.2	Comparação entre as previsões do modelo SARIMA e os valores reais Fonte: Próprio Autor	85
4.3	Comparação entre os valores reais e a previsão do modelo RNN Fonte: Próprio Autor	86
4.4	Comparação entre os valores reais e a previsão do modelo RNN+ARIMA Fonte: Próprio Autor	87
4.5	Comparação entre as previsões do modelo LSTM e os valores reais Fonte: Próprio Autor	88
4.6	Comparação entre as previsões do modelo LSTM+ARIMA e os valores reais Fonte: Próprio Autor	89
4.7	Comparação entre os percentuais de retorno Fonte: Próprio Autor	92

Lista de Tabelas

2.1	Resumo dos Métodos Utilizados	38
3.1	Tabela Dickey-Fuller	74
3.2	Comparação de métricas entre os modelos Arima	75
4.1	Comparação de Métricas entre Modelos	89

Lista de Quadros

2.1	Criptomoedas analisadas nos artigos	41
-----	---	----

Sumário

1	Introdução	16
1.1	Contexto	16
1.1.1	O escambo	16
1.1.2	O ouro como moeda	17
1.1.3	A queda do padrão-ouro	17
1.1.4	O artigo Bitcoin	18
1.2	Justificativa e Relevância	20
1.3	Objetivos	22
1.3.1	Objetivos Gerais	22
1.3.2	Objetivos Específicos	22
1.4	Hipótese de Pesquisa	23
1.5	Estrutura da dissertação	23
2	Fundamentação Teórica	24
2.1	O Funcionamento do Bitcoin	24
2.2	Regulação de Criptomoedas no Brasil	26
2.2.1	Leis Regulatórias	26
2.3	Os Mercados Financeiros	28
2.4	Séries Temporais	30
2.4.1	Atributos das Séries Temporais	31
2.5	Revisão Sistemática	33
2.5.1	Justificativa	33

2.5.2	Objetivos	34
2.5.3	Metodologia	34
2.5.4	Extração de Dados	36
2.6	Resultados da Revisão Sistemática	37
2.6.1	Principais Métodos Encontrados	37
2.6.2	Criptomoedas Analisadas	40
2.6.3	Quantidade de observações na série temporal	42
2.6.4	Intervalo entre as observações	43
2.6.5	Variáveis Incluídas na Modelagem	44
2.6.6	Discussão	46
2.6.7	Conclusão da RS	47
2.7	Métodos Preditivos	48
2.7.1	Métodos Qualitativos	48
2.7.2	Métodos Quantitativos	49
2.8	ARIMA	50
2.8.1	Autorregressivo (AR)	51
2.8.2	Média Móvel (MA)	51
2.8.3	Autorregressivo e Média Móvel (ARMA)	52
2.8.4	Autorregressivo Integrado de Média Móvel (ARIMA)	53
2.8.5	Autorregressivo Integrado de Média Móvel com Sazonalidade (SARIMA)	55
2.9	Modelos de Aprendizado de Máquina	57
2.9.1	Algoritmos de Aprendizado de Máquina	57
2.9.2	Deep Learning	59
2.10	Redes Neurais Recorrentes	60
2.10.1	O Problema das Dependências de Longo Prazo	61

2.11	Long Short-Term Memory	63
2.12	Definições Complementares	67
2.12.1	Métricas Preditivas	67
3	Metodologia	69
3.1	Obtenção de Dados	69
3.2	Análise da Série Temporal	70
3.2.1	Decompondo a Série Temporal	70
3.2.2	Analisando a Estacionariedade	72
3.3	Implementação de Modelos ARIMA	74
3.4	Implementação do modelo RNN	76
3.4.1	Código RNN	78
3.5	Implementação do Modelo LSTM	80
3.5.1	Código LSTM	81
4	Resultados e Discussão	83
4.1	Resultados ARIMA	83
4.1.1	ARIMA	83
4.1.2	SARIMA	84
4.2	Resultados RNN	85
4.2.1	RNN	85
4.2.2	RNN+ARIMA	86
4.3	Resultados LSTM	87
4.3.1	LSTM	87
4.3.2	LSTM+ARIMA	88
4.4	Comparações entre os modelos	89
4.5	Simulações de Investimentos	90

5	Conclusões e Trabalhos Futuros	93
5.1	Conclusões	93
5.2	Limitações	95
5.3	Trabalhos Futuros	95
	Referências	97

Capítulo 1

Introdução

Neste capítulo, foi delineado o contexto que envolve esta dissertação, destacando as razões que instigaram a pesquisa sobre a temática abordada, bem como a motivação e a problemática que deram origem a este projeto. A seguir, são expostos os objetivos propostos e a hipótese de pesquisa. Por fim, foi delineado a estrutura organizacional desta dissertação de mestrado.

1.1 Contexto

1.1.1 O escambo

Na trajetória da humanidade, pode-se perceber que o ser humano sempre procura meios de melhorar sua qualidade de vida, isto é, fugir da dor. De acordo com [41], tal fenômeno pode ser percebido na economia. Inicialmente não haviam governos centrais, nem tampouco dinheiro, mas a economia cresce e desenvolve-se onde existe uma sociedade com o mínimo de organização. Os grupos de pessoas e famílias eram nômades, pois se deslocavam para obter alimentos. Quando se descobriu que podia-se plantar e colher alimentos em um mesmo local, criar animais e, a partir deles, obter alimento, roupa e transporte, estes grupos de famílias começaram a se instalar fixamente em determinados territórios. Com isso surgiu o desejo por variedade, ou seja, se um cultivava abóbora e outro cultivava feijão, o que plantava abóbora, eventualmente desejava comer feijão, e o que plantava feijão eventualmente desejava comer abóbora, então os dois, em comum acordo resolviam trocar alguns alimentos. Esta prática de troca direta é denominada escambo [58]. Essa troca direta abrangia não apenas alimentos, mas animais, vestuário, utensílios domésticos, armas, serviços, ferramentas, entre outros.

1.1.2 O ouro como moeda

O surgimento de dificuldades relacionadas ao valor das coisas deu início a uma evolução na economia, conduzindo à adoção espontânea de certos produtos como meios de troca e reserva de valor. Essa escolha era motivada por características como beleza, usabilidade, necessidade ou escassez. Entre esses produtos, o ouro destacou-se ao longo de milênios, sendo empregado por diversos povos e nações como moeda de troca e reserva de valor, consolidando e fortalecendo sua posição no cenário econômico [36].

Na medida em que as pessoas acumulavam ouro e os Estados surgiam, nasce um novo desafio: armazenar e transportar com segurança o ouro. Nesse contexto, os bancos emergiram como instituições seguras que ofereciam serviços de armazenamento do ouro em nome de seus proprietários. O funcionamento inicial dos bancos pode ser exemplificado da seguinte forma: João depositava seu ouro em um banco e, em troca, recebia um documento denominado papel-moeda. Esse papel representava uma quantidade específica de ouro, conferindo a João o direito de resgatá-lo quando desejasse.

No entanto, surgiram transações que simplificaram esse processo. Por exemplo, quando João decidia comprar um cavalo de Marcelo, em vez de retirar fisicamente o ouro do banco e entregá-lo, João simplesmente transferia o papel-moeda a Marcelo. A partir desse momento, Marcelo detinha o direito de resgatar o ouro, embora o ouro físico não fosse essencial. O valor representado pelo ouro podia ser obtido por meio do papel que certificava sua existência nos cofres bancários.

Essas transações, realizadas com papéis-moeda, tornaram-se prática comum, conferindo grande poder aos bancos, uma vez que detinham grande parte da riqueza da população. Se a maioria do ouro não era resgatada, os bancos tinham a oportunidade de emitir mais papéis-moeda do que a quantidade de ouro armazenada, resultando na inflação do papel-moeda, ou seja, na criação de um papel que representava uma quantidade específica de ouro que, na realidade, não existia.

1.1.3 A queda do padrão-ouro

De acordo com [64], neste ponto, os governos dos Estados começaram a intervir e regular a emissão de papel-moeda pelos bancos. Eles estabeleceram bancos centrais, centralizando a prerrogativa de armazenar o ouro e emitir os papéis-moeda. Surgiram, então, as moedas estatais, conhecidas como moedas fiduciárias, baseadas no padrão-ouro. Nesse sistema, os bancos centrais emitiam os papéis-moeda e os entregavam aos bancos,

que por sua vez os repassavam aos proprietários do ouro. Essa estrutura funcionou eficientemente por um período, até que os estados, assim como os bancos antes deles, sucumbiram à tentação de emitir mais papel-moeda do que a quantidade de ouro armazenado. Essa prática inflacionária enfraqueceu progressivamente o padrão-ouro. Gradualmente, os governos abandonaram esse padrão, criando teorias monetárias para justificar o crescente inflacionamento artificial da moeda estatal.

Os Estados Unidos foram o último país a abandonar o padrão ouro, na década de 1970. Hoje o dinheiro segue sendo inflacionado pelos governos, sem nenhum tipo de lastro ou garantia de valor.

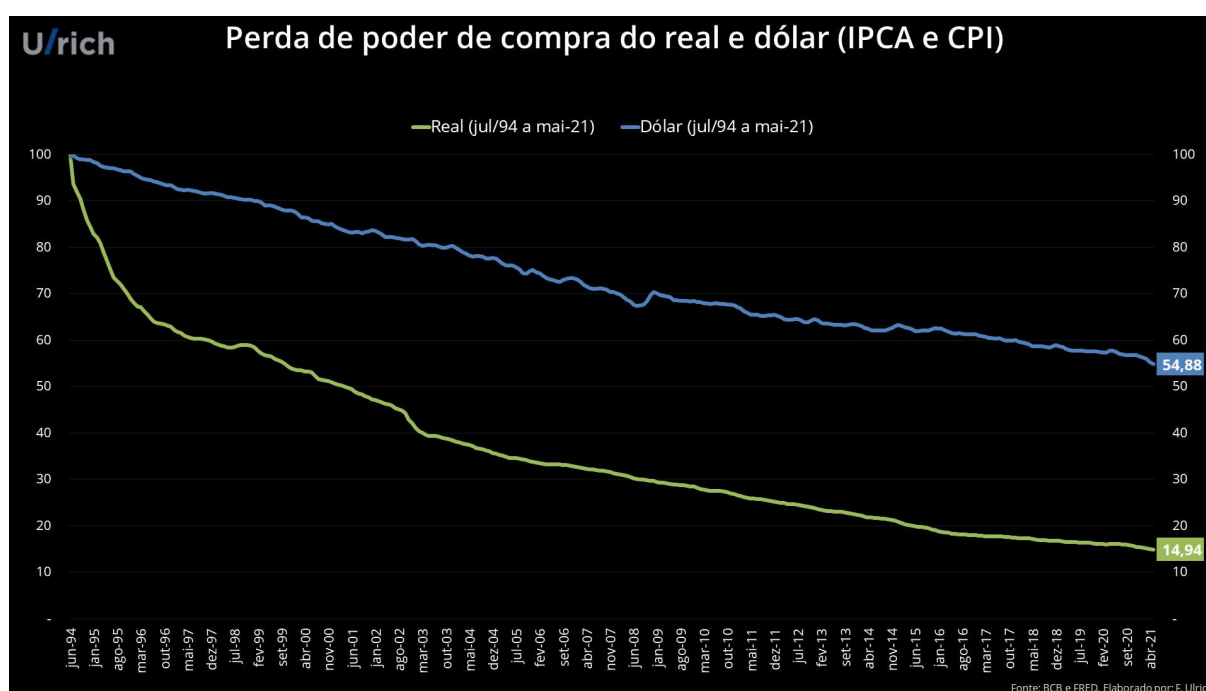


Figura 1.1: Perda de poder de compra do Real e Dólar. Fonte: BCB e FED, Elaborado por: F. Ulrich, 2021

Como pode ser visto na Figura 1.1, a inflação das moedas estatais corrói seu poder de compra, inclusive em moedas consideradas fortes, como o dólar.

1.1.4 O artigo Bitcoin

Foi neste cenário que, em 2008, Satoshi Nakamoto publicou um artigo com o título “Bitcoin: Um Sistema de Dinheiro Eletrônico Peer-to-Peer”.

Baseado em tecnologia criptográfica, [49] propõe um sistema onde uma rede de dispositivos, de maneira descentralizada, cede seu poder computacional para realizar cálculos complexos e validar as transações criptográficas realizadas na rede, tal rede

é denominada *blockchain*. Como recompensa pelo poder computacional que é cedido pelos nós pertencentes a rede, estes recebem Bitcoins, que são criados e validados pela própria *blockchain*. Com o avanço no número de transações, os cálculos para assegurar a criptografia ficam mais complexos, o que demanda um poder computacional cada vez maior por parte dos nós pertencentes à *blockchain* e, conseqüentemente, um gasto com energia elétrica cada vez maior. Diferente das moedas fiduciárias, o Bitcoin não pode ser inflacionado, pois não existe um governo central que detenha o controle do Bitcoin. Seu protocolo está limitado a emissão de apenas 21 milhões de Bitcoins, sendo que a quantidade de Bitcoins criados vêm diminuindo ao longo dos anos. O Bitcoin passou a existir em 2009, atualmente existem 19,61 milhões de Bitcoins¹ e a previsão é que os 21 milhões sejam criados apenas em 2140. Como pode ser visto na Figura 1.2, a criação de novos Bitcoins, reduz-se pela metade, aproximadamente, a cada quatro anos.

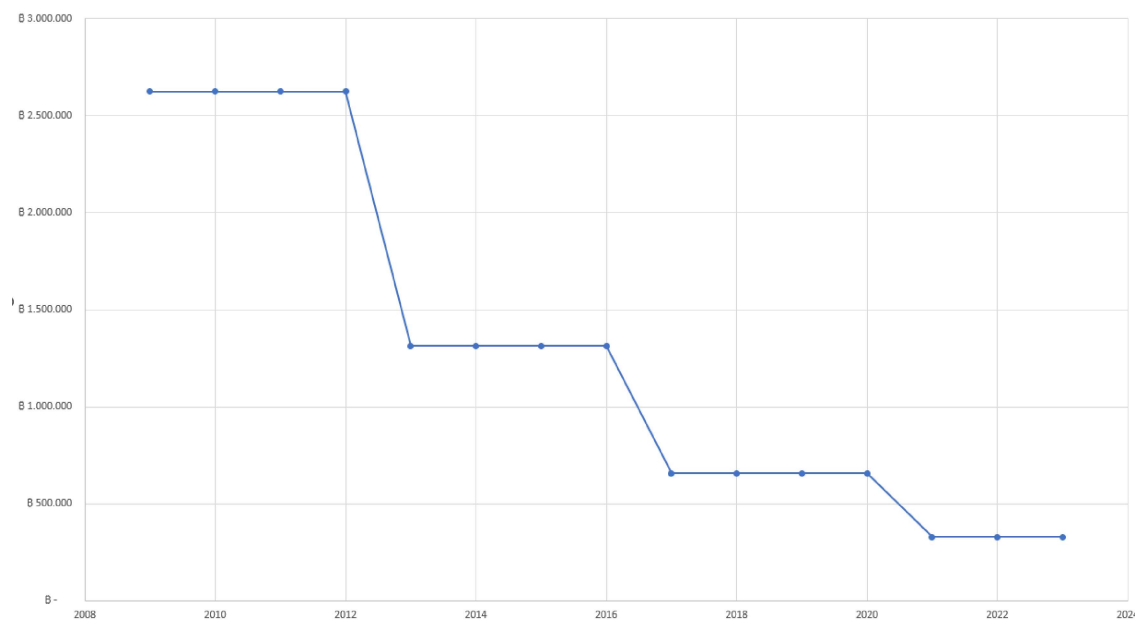


Figura 1.2: Estimativa de Bitcoins minerados por ano Fonte: Próprio Autor

Portanto, o Bitcoin caracteriza-se por ser deflacionário, isto é, as emissões de moedas diminuem ao longo do tempo, ao invés de aumentar, como acontece com as moedas fiduciárias. A *Blockchain* do Bitcoin começou a funcionar em 2009. O primeiro Bitcoin não tinha valor de mercado, mas com o passar do tempo e a adesão de cada vez mais pessoas ao sistema, o mesmo começou a ser transacionado cada vez mais, e teve seu valor crescendo exponencialmente nos últimos anos. A partir da tecnologia da *Blockchain*, surgiram projetos de inúmeras empresas na área da tecnologia e, assim, surgiram outras criptomoedas, como por exemplo o Ethereum, que criou a sua própria

¹<https://https://coinmarketcap.com/pt-br/>

Blockchain, agregando algumas funcionalidades que a *Blockchain* do Bitcoin não oferece. Como o Bitcoin foi a primeira criptomoeda a ser criada, todas as outras criptomoedas que foram criadas posteriormente são denominadas Altcoins (Moedas alternativas). Com o crescimento exponencial deste mercado, a cada dia surgem novas Altcoins e deixam de existir outras tantas. Atualmente existem mais de 12.000 criptomoedas.² Dentre as criptomoedas existentes, algumas se destacam, no entanto o Bitcoin domina aproximadamente 50% do mercado de criptomoedas³. O Ethereum, a segunda maior criptomoeda do mundo, domina aproximadamente 15% do mercado. Os 35% restantes estão divididos entre as outras criptomoedas. Devido à sua grande dominância, as variações de preço do Bitcoin influenciam diretamente no preço de todas as outras criptomoedas.

1.2 Justificativa e Relevância

A interdisciplinaridade neste campo da economia reúne especialistas de diversas áreas, incluindo economistas, estatísticos, investidores, matemáticos, desenvolvedores, entre outros, com o objetivo comum de conduzir pesquisas voltadas para a previsão do comportamento do mercado financeiro.

A necessidade de prever o preço do Bitcoin é impulsionada por diversas questões fundamentais. O principal objetivo é garantir a máxima lucratividade por meio de investimentos bem-sucedidos, proporcionando aos investidores uma vantagem competitiva sobre aqueles que negociam outros ativos financeiros.

Outro fator crucial que influenciou a decisão de realizar esta pesquisa no mercado de criptomoedas é a peculiar distribuição do modelo de dados no qual o Bitcoin opera. Este modelo é caracterizado por oscilações diárias e pela formação de séries temporais complexas. Diante desse cenário desafiador, o mercado de criptomoedas oferece orientações valiosas para o desenvolvimento de modelos lineares e baseados em aprendizado de máquina, visando a previsão de valores futuros.

Desde 2018 a quantidade de pesquisas publicadas sobre o Bitcoin, cresceu exponencialmente, conforme pode ser visto na Figura 1.3, que apresenta o resultado da pesquisa pelo termo "Bitcoin" no Google Acadêmico⁴.

²<https://https://www.coingecko.com/pt>

³<https://https://coinmarketcap.com/pt-br/>

⁴<https://scholar.google.com.br/?hl=pt>

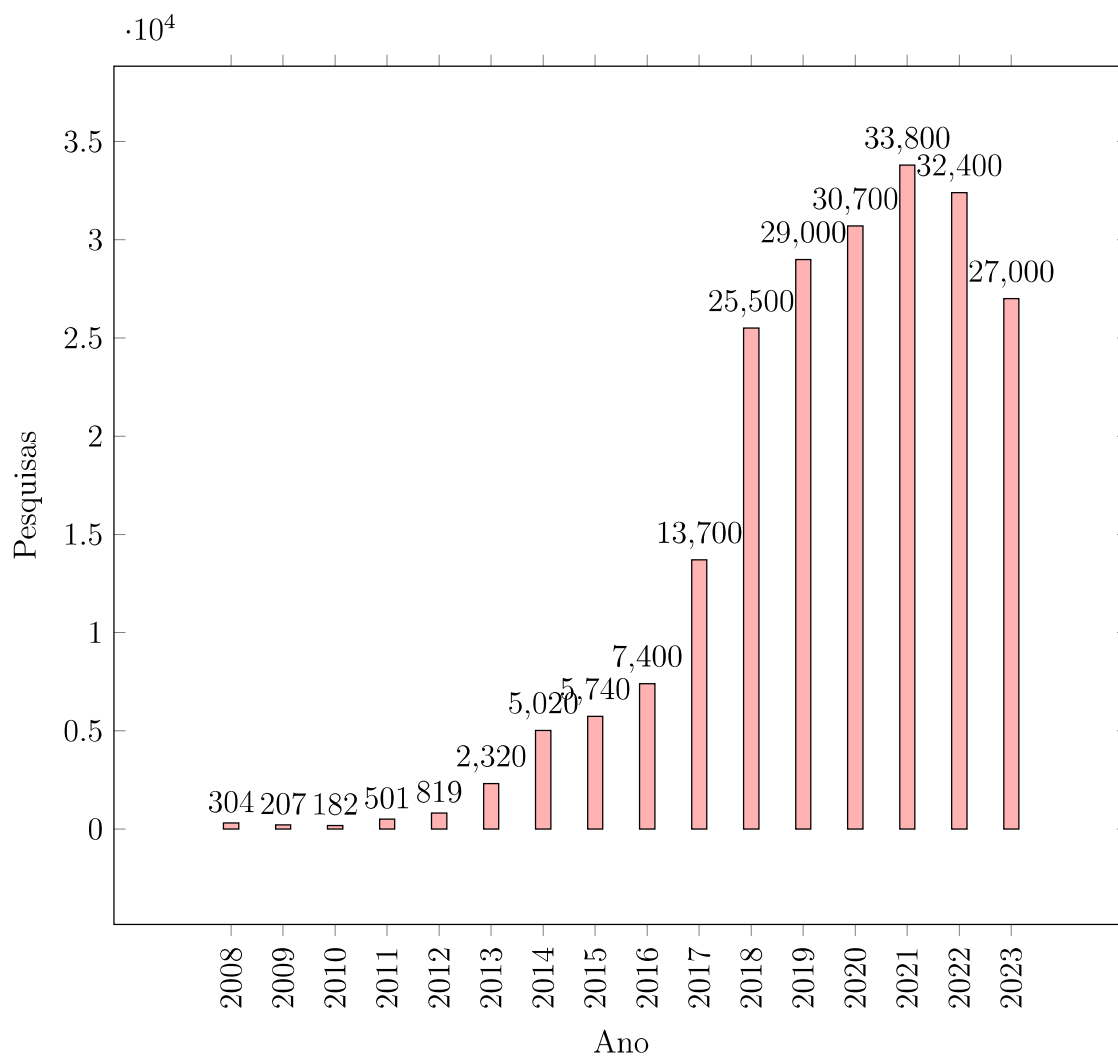


Figura 1.3: Número de pesquisas por ano Fonte: Próprio Autor

Este aumento nas pesquisas se deveu ao fato de que o Bitcoin apresentou uma alta de aproximadamente 2.132% ao longo do ano de 2017, conforme pode ser visto na Figura 1.4



Figura 1.4: Crescimento do Bitcoin no ano de 2017

Fonte: TradingView ⁵

Apesar do crescimento exponencial no número de pesquisas publicadas sobre o Bitcoin desde 2018, os estudos que visam analisar os modelos existentes e propor um modelo híbrido ainda são escassos. Por esse motivo, o presente trabalho tem o potencial de contribuir para a diminuição dessa lacuna na literatura, contribuindo para o avanço da compreensão e previsão do mercado de criptomoedas. A redução dessa lacuna aumentará significativamente a qualidade das previsões, o que se traduzirá em maiores níveis de lucratividade para os investidores.

1.3 Objetivos

1.3.1 Objetivos Gerais

O objetivo geral desta dissertação é desenvolver e comparar diversos tipos de modelos para aprimorar a previsão do preço futuro do Bitcoin.

1.3.2 Objetivos Específicos

1. Fazer uma revisão sistemática sobre modelos estatísticos lineares, abordagens fundamentadas em aprendizado de máquina e, ainda, modelos híbridos, para previsão de criptomoedas, buscando um bom embasamento teórico e compreensão dos mesmos;

2. Desenvolver algoritmos para previsão de valores do Bitcoin utilizando modelos baseados em aprendizado de máquina e modelos estatísticos lineares;
3. Aplicar diferentes tipos de métodos de séries temporais para previsão de criptomoedas, verificando o que melhor se adapta a série do Bitcoin;
4. Comparar os modelos e testar possíveis modelos híbridos, combinando modelos estatísticos lineares com modelos baseados em aprendizado de máquina.

1.4 Hipótese de Pesquisa

Nesta dissertação, a hipótese central perseguida é a seguinte: Acredita-se que um modelo que incorpora tanto aspectos de modelos estatísticos lineares quanto de modelos baseados em aprendizado de máquina no processo de previsão, pode aprimorar a precisão das estimativas do preço do Bitcoin. Para testar a validade dessa hipótese, serão realizadas comparações entre diferentes modelos de previsão de preços.

1.5 Estrutura da dissertação

Este estudo segue uma estrutura composta por cinco capítulos que delineiam a investigação realizada. A descrição do conteúdo dos próximos capítulos é fornecida a seguir.

O segundo capítulo começa com uma breve exposição sobre mercados financeiros e os tipos de análises que usualmente são realizadas sobre este tipo de mercado. Em seguida, é apresentada uma Revisão Sistemática sobre modelos de previsão de Bitcoin.

O terceiro capítulo aborda a metodologia utilizada neste trabalho. No capítulo quatro, realiza-se a Análise de Resultados e Comparação entre os modelos ARIMA, SARIMA, RNN, LSTM e modelos híbridos apresentando gráficos e comparações gerados por simulações computacionais. Na sequência, são realizadas simulações de investimentos, baseadas nas previsões do melhor modelo.

O capítulo cinco apresenta as discussões sobre os modelos considerados no trabalho. Em seguida são apresentadas as conclusões, limitações e sugestões para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo, são apresentados os fundamentos do funcionamento do Bitcoin, um panorama do atual estado de regulação das criptomoedas no Brasil, o funcionamento do mercado financeiro, a definição e a importância das séries temporais, uma revisão sistemática dos métodos preditivos presentes na literatura, e a descrição detalhada dos modelos ARIMA, SARIMA, RNN e LSTM, além das principais métricas utilizadas para avaliação preditiva.

2.1 O Funcionamento do Bitcoin

De acordo com [49], o Bitcoin é uma versão puramente peer-to-peer de dinheiro eletrônico que permite pagamentos on-line sem a necessidade de uma instituição financeira intermediária. Neste contexto, embora as assinaturas digitais ofereçam parte da solução, os benefícios fundamentais são perdidos se ainda for necessário um terceiro confiável para evitar o gasto duplo. A solução de Nakamoto para este problema utiliza uma rede *peer-to-peer* que insere data e hora nas transações por meio de um *hash*, em uma cadeia contínua de prova-de-trabalho, à base de *hash*. Onde um servidor de *timestamp*, distribuído nesta rede *peer-to-peer*, cria um *hash* de um conjunto de dados e o divulga amplamente, de forma semelhante à publicação em um jornal ou post em uma rede social. O *timestamp* atua como uma prova de que os dados precisavam existir naquele momento para serem incluídos no *hash*. Cada *timestamp* incorpora o *hash* do anterior, formando assim uma cadeia, denominada *blockchain*, na qual cada novo *timestamp* fortalece os anteriores.

Para implementar esta *blockchain*, utiliza-se uma prova-de-trabalho semelhante ao Hashcash de Adam Back [7] para garantir a segurança e integridade dos dados. Esta prova-de-trabalho consiste em encontrar um valor cujo *hash*, utilizando o algoritmo

SHA-256, comece com um número específico de bits zero. Este processo requer um esforço computacional exponencialmente crescente conforme aumenta o número de bits zero desejados. Uma vez que o *hash* satisfaz os critérios, o bloco é considerado válido e não pode ser alterado sem refazer o trabalho. A decisão da maioria é determinada pela cadeia mais longa, representando o maior esforço de prova-de-trabalho investido. A dificuldade da prova-de-trabalho é ajustada regularmente para manter uma média constante de blocos gerados por hora, compensando o aumento da velocidade do hardware e a variação no interesse de participação na rede ao longo do tempo.

O funcionamento da *blockchain* segue os seguintes passos:

1. Transmissão de novas transações para todos os nós.
2. Coleta das novas transações em um bloco, por cada nó.
3. Cada nó busca por uma prova-de-trabalho difícil para o bloco,
4. Quando um nó encontra uma prova-de-trabalho, ele transmite o bloco para todos os nós.
5. Os nós aceitam o bloco somente se todas as suas transações são válidas e já não foram gastas.
6. Os nós expressam sua aceitação do bloco, trabalhando na criação do próximo bloco na cadeia, usando o *hash* do bloco aceito como o *hash* anterior.

Os nós consideram a cadeia mais longa como correta e continuam a estendê-la, alterando para o ramo mais longo se necessário. Novas transmissões não precisam alcançar todos os nós imediatamente, mas devem chegar a muitos nós para entrar em um bloco. Os nós também são tolerantes a mensagens abandonadas, solicitando blocos ausentes quando necessário.

A primeira transação de um bloco, por convenção, é uma operação especial que inicia uma nova moeda, recompensando o criador do bloco. Isso incentiva os nós a apoiarem a rede e introduz moedas em circulação sem uma autoridade central. O incentivo pode ser financiado com taxas de transação, que podem eventualmente substituir a criação de novas moedas, tornando o sistema livre de inflação. Este mecanismo incentiva os nós a agirem honestamente, pois um atacante com mais poder de CPU teria mais a ganhar seguindo as regras do que tentando fraudar o sistema.

A *blockchain* do Bitcoin é um sistema de transações eletrônicas descentralizado e confiável. Usando assinaturas digitais para garantir o controle de propriedade. Para evitar o problema do gasto duplo, a *blockchain* utiliza prova de trabalho para registrar transações de forma pública e praticamente imutável, tornando difícil para um atacante alterar o histórico. A rede é robusta, simples e não requer identificação dos nós, que podem entrar e sair livremente. Eles validam transações usando seu poder de CPU, garantindo a integridade do sistema. Este mecanismo de consenso permite aplicar todas as regras e incentivos necessários.

2.2 Regulação de Criptomoedas no Brasil

O mercado de criptomoedas, apesar de ter surgido em 2009, só teve suas primeiras leis regulatórias no Brasil em 2022. Essa lacuna regulatória se deu pela natureza inovadora do Bitcoin e de outras criptomoedas, que não se encaixavam nas definições tradicionais de moedas ou instrumentos financeiros.

2.2.1 Leis Regulatórias

- **Lei nº 14.478/2022 (Marco Legal das Criptomoedas):** Estabelece diretrizes para a prestação de serviços de ativos virtuais e regulamenta as empresas que operam nesse setor. Define o que são ativos virtuais, exige autorização prévia do governo federal para empresas que desejam operar no país, introduz o crime de fraude com criptomoedas, define as responsabilidades das autoridades regulatórias e modifica o Código Penal, a Lei de Crimes contra o Sistema Financeiro Nacional e a Lei de Lavagem de Dinheiro para incluir as prestadoras de serviços de ativos virtuais em suas disposições.
- **Lei nº 14.754/2023 (Lei da Reforma do Imposto de Renda sobre Investimentos no Exterior):** Essa lei, que entrou em vigor em 1º de janeiro de 2024, trata da tributação de criptomoedas detidas no exterior, em *exchanges* sem CNPJ no Brasil.
- Pontos Principais da Lei nº 14.754/2023:
 - Criptomoedas são consideradas ativos virtuais e estão sujeitas à tributação:

- * Ganhos com a venda de criptomoedas no exterior serão tributados como ganho de capital na Declaração de Ajuste Anual (DAA).
 - * A alíquota do IR será de 15% sobre o valor total do ganho.
- É obrigatório declarar à Receita Federal:
- * Quantidade de criptomoedas detidas;
 - * Localização das criptomoedas (em qual corretora ou exchange estão armazenadas);
 - * Valor das criptomoedas em reais no dia 31 de dezembro de cada ano.
- Outras regras:
- * Não há tributação sobre a variação do valor das criptomoedas sem a venda (diferentemente do que acontece com ações, por exemplo).
 - * É possível compensar prejuízos com criptomoedas com ganhos em outras aplicações financeiras.
 - * A Receita Federal ainda está elaborando as normas para regulamentar a tributação de criptomoedas.

Apesar dos avanços, ainda há pontos obscuros na regulamentação de criptomoedas no Brasil, principalmente no que se refere à distinção entre criptomoedas no Brasil e no exterior:

- Criptomoedas no Brasil:
- * A Receita Federal ainda não definiu como a tributação será feita na prática.
 - * Aguarda-se a publicação de novas normas para regulamentar esse ponto.
- Criptomoedas no exterior:
- * Ganhos com venda devem ser declarados na DAA.
 - * Alíquota do IR é de 15%.

É importante ressaltar que a regulação de criptomoedas no Brasil ainda está em desenvolvimento. Novas leis e regulamentações podem ser criadas no futuro, à medida que o mercado de criptomoedas se torna mais maduro e complexo.

2.3 Os Mercados Financeiros

Apesar de ter sido criado como uma moeda digital, o Bitcoin têm sido utilizado, mais como um ativo financeiro do que como uma moeda de troca, propriamente dita.

Os mercados financeiros desempenham um papel crucial na economia e na estrutura da sociedade hodierna. Suas atividades têm uma influência significativa no desenvolvimento econômico global, capturando a atenção de pesquisadores e profissionais de negócios devido às suas aplicações práticas e fundamento teórico [17]. A relevância desse campo de estudo é evidente em exemplos práticos encontrados na literatura. [66] destacam que o mercado financeiro dos EUA desempenha um papel de liderança na economia global, sendo que eventos como a crise financeira de 2008 podem reverberar por vários meses e impactar outros mercados ao redor do mundo.

Um mercado financeiro, abrangendo a negociação de ativos como ações, títulos, moedas e derivativos, é um local onde compradores e vendedores interagem, conferindo valor a esses ativos, os quais podem ser convertidos em moeda fiduciária. O êxito de um investidor em mercados nacionais ou globais depende da qualidade da informação disponível para embasar suas decisões. Dada a sua importância prática, a análise dos movimentos do mercado financeiro é amplamente investigada nas áreas de economia, finanças, engenharia e matemática, impulsionada pelo potencial ganho financeiro [69].

Diferentes estudos sobre os atributos e características do mercado, que exercem influência nos preços dos ativos financeiros, compõem a análise de mercado. O propósito fundamental dessa análise é compreender os comportamentos de mercado para aprimorar o processo de tomada de decisões. Os dois principais enfoques destacados na literatura financeira para a análise do movimento de preços são a Análise Fundamentalista e a Análise Técnica. Apesar de ambas visarem compreender os movimentos de preço e antecipar suas futuras direções, essas abordagens diferem em sua natureza [17].

1. **Análise Fundamentalista:** A análise fundamentalista é caracterizada por considerar fatores internos e externos que impactam o preço de um ativo. Aspectos como a situação econômica da empresa, fatores econômicos e políticos, fluxo de caixa, estratégias de marketing e perspectivas futuras são examinados. Fundamentalistas buscam definir um valor intrínseco do ativo baseando-se em informações como relatórios, balanços e resultados internos da empresa
2. **Análise Técnica:** Por outro lado, a análise técnica se baseia nos princípios da The

Dow Theory [47], onde a previsão de preço se fundamenta no comportamento de valores passados. Assim, ela utiliza o histórico de preços para identificar padrões e indicadores que apontem para movimentos futuros do mercado [65]. Diferentemente da análise fundamentalista, a técnica não leva em conta as características internas e externas da empresa no estudo dos movimentos de preço. A visão dos técnicos é que o preço já incorpora todas as informações fundamentais, e que a história tende a se repetir. Essa abordagem evita a análise de fatores econômicos subjetivos relacionados à economia, política e notícias [17]. Ferramentas como gráficos, indicadores técnicos e modelos são empregadas por técnicos para monitorar a tendência de preço e volume ao longo do tempo.

Este trabalho se baseia na Análise Técnica para prever os valores do Bitcoin.

Nos últimos anos, diversas abordagens têm sido propostas para resolver o desafio da previsão de séries temporais financeiras. A literatura destaca três principais enfoques: modelos estatísticos lineares, modelos baseados em aprendizado de máquina e modelos fundamentados no processo de mineração de texto [17]. Estes últimos exploram o sentimento expresso nos textos de notícias e relatórios financeiros para aprimorar a previsão dos preços.

Na primeira metade deste século, as técnicas estatísticas lineares centradas em modelos Autoregressive (AR), Autoregressive Integrated Moving Average (ARIMA) e Moving Average (MA) eram consideradas o estado da arte para modelar séries temporais e prever tendências no mercado financeiro. Esses métodos pressupõem que os dados seguem uma distribuição normal, são estacionários e lineares [32]. Conhecidos como algoritmos paramétricos, exigem um entendimento de parâmetros matemáticos para o modelo [52].

Nas últimas décadas, houve uma ascensão dos métodos baseados em Aprendizado de Máquina, notadamente em técnicas de regressão e em métodos dependentes do tempo [52]. Devido à natureza não linear das séries temporais financeiras, os modelos tradicionais enfrentam desafios na previsão de séries financeiras. Ao contrário das abordagens lineares, os modelos de Aprendizado de Máquina conseguem capturar relações não lineares e não estacionárias, caracterizando os dados sem a necessidade de conhecimento prévio sobre a distribuição dos mesmos [17].

Dentre as metodologias amplamente empregadas, destacam-se as Redes Neurais Artificiais (RNAs), reconhecidas pela sua adaptabilidade automática, orientação a dados e capacidade de capturar padrões não lineares, prescindindo de suposições estatísticas

sobre os dados, inclusive por meio de modelos híbridos [17]. Outras variantes de RNAs, como a Multilayer Perceptron (MLPs) com algoritmo de retropropagação, foram aplicadas para compreender as relações entre indicadores técnicos, visando a previsão dos preços de abertura e fechamento diários na BOVESPA [40]. [31] adotaram uma Feed-forward Neural Network (FFNN) para desenvolver uma estratégia que aprendesse padrões de velas japonesas. Contudo, apesar da ampla utilização para reconhecimento de padrões, as RNAs apresentam limitações inerentes à natureza das séries temporais financeiras, são elas: operação como uma "caixa preta", taxa de convergência lenta, propensão a estagnar em mínimos locais [18] e propensão ao *overfitting*, o que significa que o modelo aprende não apenas o padrão subjacente nos dados, mas também o ruído e as flutuações aleatórias presentes no conjunto de treinamento. Como resultado, o modelo se torna muito específico para os dados de treinamento e perde sua capacidade de fazer previsões precisas em dados não vistos. Outras abordagens também têm sido aplicadas, como Máquinas de Suporte de Vetores (SVM) e técnicas dos vizinhos mais próximos (kNN-TSPI) [51], e, mais recentemente, Mecanismos de Atenção aplicados a séries temporais [56], [35], [70].

Um denominador comum nessas estratégias é a construção do modelo a partir dos dados históricos da série temporal, onde essas técnicas buscam padrões passados com a expectativa de que se reproduzirão no futuro.

2.4 Séries Temporais

Uma série temporal representa uma coleção de observações (sejam discretas ou contínuas) de uma variável ordenada cronologicamente, denotada por $X_1, X_2, X_3 \dots, X_n$ [43]. Vale ressaltar que a ordem das séries é crucial, e modificar essa ordem altera o significado dos dados.

Outra definição, conforme [52], descreve uma série temporal como uma sequência ordenada Z de observações de tamanho t , onde $Z = (z_1, z_2, z_3 \dots, z_t)$, e cada valor $z_t \in R$ para todos os $t \in [1, m]$. Cada Z_t representa uma observação da série temporal no instante t . Se os valores da série são expressos por uma função matemática $y = f(\text{time})$, a série é chamada de determinística. Quando é adicionado um termo aleatório $E : y = f(\text{time}, E)$, a série é considerada estocástica ou não determinística.

Uma série é considerada estacionária quando se desenvolve aleatoriamente em torno de uma média constante, refletindo um equilíbrio estável. Essa propriedade é essencial, uma vez que muitos métodos pressupõem tal condição para prever valores futuros [52].

[3] afirmam que séries com uma única variável são denominadas *univariadas*, enquanto aquelas com mais de uma variável são chamadas *multivariadas*. Além disso, as séries podem ser contínuas ou discretas. As contínuas podem ser medidas em qualquer momento, enquanto as discretas geralmente têm observações em pontos específicos. Tipicamente, séries temporais com tempo discreto são registradas em intervalos regulares, como milissegundos, horas, dias ou semanas.

2.4.1 Atributos das Séries Temporais

Existem diferentes atributos das séries temporais financeiras, aqui foram destacados alguns dos mais importantes, são a tendência, a sazonalidade e a dependência.

1. **Tendência.** Uma tendência é identificada quando ocorre um aumento ou diminuição de longo prazo nos dados. Essa tendência não necessariamente segue uma trajetória linear e pode ser descrita como uma "mudança de direção", indicando um movimento ascendente ou descendente. Por exemplo, na Figura 2.1, onde foi extraída a tendência do Bitcoin, pode-se observar claramente o crescimento no preço, de 2014 até 2017, com uma rápida ascensão entre 2017 e 2021, seguida de uma queda brusca entre 2021 e 2023. De acordo com [11], pode-se afirmar que há uma tendência se $X_1 < X_2 < \dots < X_n$ em qualquer intervalo de tempo.

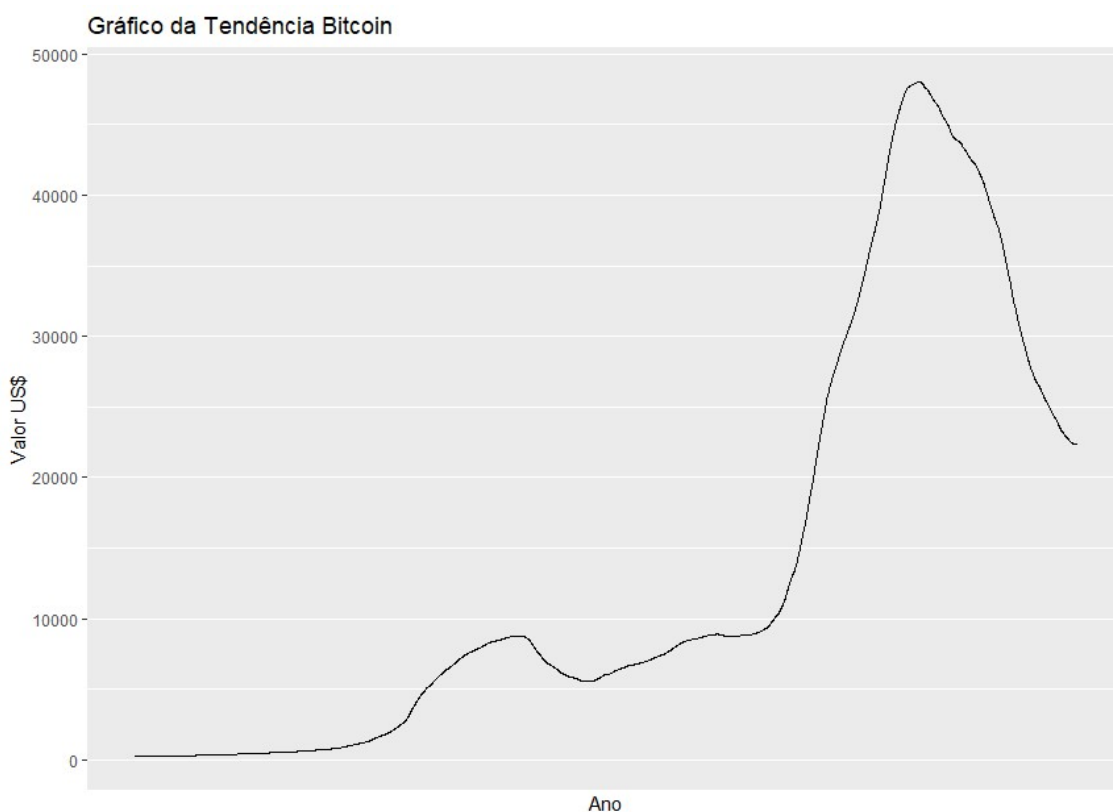


Figura 2.1: Tendência do BitCoin entre 2014 e 2023 Fonte: Próprio Autor

- Sazonalidade.** Refere-se a um padrão recorrente em uma série temporal influenciado por fatores sazonais, como a estação do ano, o dia da semana, as condições climáticas ou práticas tradicionais. Este padrão sazonal ocorre em uma frequência fixa e conhecida. Por exemplo, as vendas de sorvete durante períodos quentes e as vendas de roupas de inverno durante a estação fria. Conforme [44] e [46] indicam, a tendência é identificada quando há um movimento persistente dos dados em uma direção, enquanto a sazonalidade refere-se ao comportamento regular assumido pela série em algum período específico.
- Dependência.** Refere-se inicialmente à correlação com outras variáveis, ou seja, é a medida da extensão de uma relação linear entre duas variáveis. Por outro lado, a autocorrelação avalia a relação linear entre valores defasados da própria série temporal, conforme descrito por [30], existem diversos índices de autocorrelação. Por exemplo, os retornos de uma série temporal y_t são definidos como $r_t = y_t - (y_t - 1)$. A Função de Autocorrelação (ACF) mede a previsibilidade linear da série temporal y_t no tempo t . Por exemplo, r_1 representa a relação entre y_t e $(y_t - 1)$; r_2 representa a relação entre y_t e $y_t - 2$, e assim por diante. A ACF é definida pela Equação 2.1:

$$r_k = \frac{\sum_{t=k+1}^T (y_t - \hat{y})(y_{t-k} - \hat{y})}{\sum_{t=k+1}^T (y_t - \hat{y})^2} \quad (2.1)$$

Onde T é o comprimento da série temporal. Uma representação visual da ACF é apresentada na Figura 2.2.

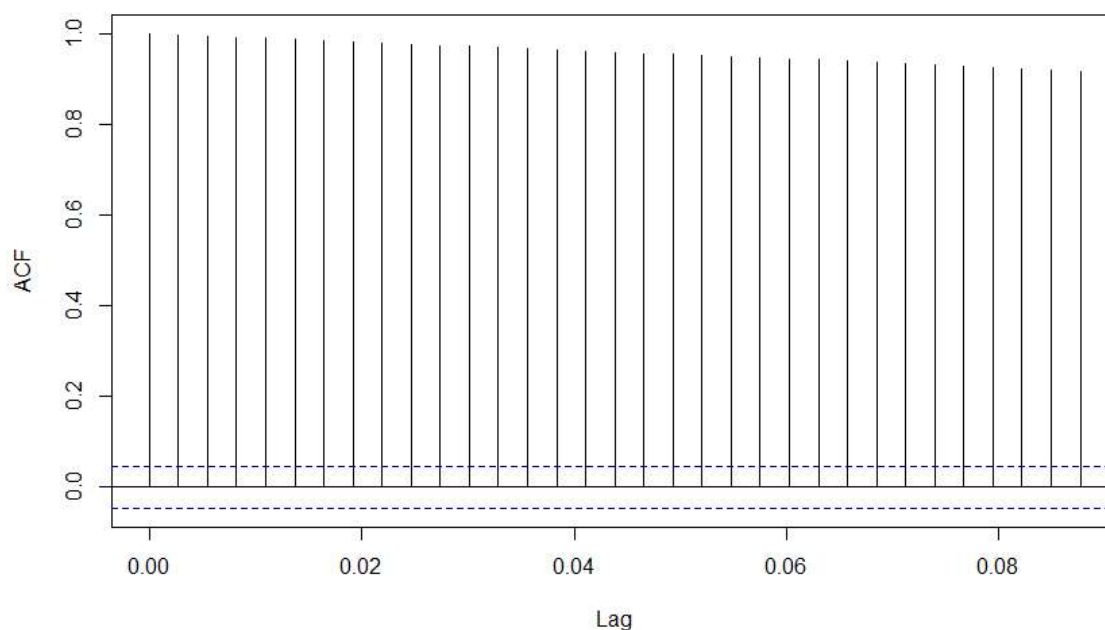


Figura 2.2: Autocorrelação do BitCoin Fonte: Próprio Autor

A previsão de séries temporais, portanto, pressupõe que os dados históricos sejam uma combinação de um formato padrão e de algum erro aleatório. O objetivo é isolar o formato do erro e extrair a dependência, a tendência e a sazonalidade desse padrão [10].

2.5 Revisão Sistemática

2.5.1 Justificativa

Com o exponencial crescimento das bases de dados científicas e o aumento significativo no volume de publicações, torna-se essencial implementar uma estratégia eficaz para sistematizar os resultados desses trabalhos. Nesse sentido, conduziu-se uma Revisão Sistemática, um método de pesquisa explícito e rigoroso, cujo propósito, conforme definido por [12], é identificar o conhecimento científico em uma área específica. Essa identificação é realizada por meio da coleta, combinação e avaliação crítica das diversas abordagens publicadas.

Adicionalmente, de acordo com as diretrizes estabelecidas por [33], a revisão sistemática é categorizada como um estudo secundário. Nesse contexto, é imperativo que o autor apresente de maneira clara e transparente o protocolo de busca e seleção utilizado. Essa prática não apenas fortalece a integridade do estudo, mas também permite que outros pesquisadores interessados auditem a pesquisa, contribuindo para a validação e a robustez do processo de revisão.

2.5.2 Objetivos

Identificar e analisar os modelos e métodos baseados em séries temporais mais utilizados para prever valores de criptomoedas e quais apresentam os melhores resultados.

2.5.3 Metodologia

1. Questões de Pesquisa: A essência da Revisão Sistemática reside em abordar uma ou mais questões fundamentais relacionadas a um determinado tópico, fundamentando-se na análise criteriosa da literatura existente sobre o tema.
2. Formulação da pergunta:
 - a Questão primária: Quais são os métodos e técnicas baseados em séries temporais empregados para prever os valores de criptomoedas que apresentam os melhores resultados?
 - b Intervenção: métodos e técnicas de previsão baseados em séries temporais.
 - c Controle: artigos, teses, dissertações e monografias sobre previsão de preço de criptomoedas utilizando séries temporais disponíveis nas principais bases de dados científicas.
 - d Resultados: análise profunda e abrangente dos modelos e métodos baseados em séries temporais empregados na previsão de valores de criptomoedas; e identificar qual ou quais modelos apresentaram os resultados mais satisfatórios.
 - e Aplicação: pesquisadores da área de previsão de preços de ativos de renda variável; e investidores (*traders* e *holders*) em criptomoedas.
3. Seleção de Fontes: As fontes devem estar acessíveis online, preferencialmente em bases de dados científicas especializadas. Pode-se também incluir trabalhos de outras

origens, como anais de eventos ou consultas a especialistas, contanto que satisfaçam os critérios estabelecidos para uma Revisão Sistemática.

4. Palavras-chave: A partir do tema do trabalho, foram identificados os termos principais e seus equivalentes em inglês. Dessa forma, foram definidas as seguintes palavras-chave: “séries temporais”, previsão, bitcoin, volatilidade; “time series”, prediction, cryptocurrency, forecast, volatility. Essa seleção garante que as palavras-chave sejam abrangentes, específicas e relevantes, aumentando a eficácia da revisão sistemática.
5. Listagem de fontes: Para a obtenção dos trabalhos acadêmicos, foram consideradas duas fontes principais:
 - a Google Acadêmico (<https://scholar.google.com.br/>): uma ferramenta de busca que indexa uma vasta gama de publicações acadêmicas, incluindo artigos, teses, livros, resumos e relatórios técnicos de diversas áreas do conhecimento.
 - b Portal Periódicos da Capes (<http://www.periodicos.capes.gov.br/>): uma base de dados indexada que oferece acesso a um amplo acervo de conteúdos científicos, abrangendo inúmeras disciplinas e áreas de estudo.
6. Serão considerados trabalhos que utilizem séries temporais para prever valores de Bitcoin e/ou outras criptomoedas.
7. Critérios de Inclusão e Exclusão dos trabalhos:

Critérios de Inclusão:

- a Serão incluídos trabalhos publicados e disponíveis integralmente em bases de dados científicas.
- b Serão incluídos trabalhos recentes (publicados a partir de 2017).

Critérios de Exclusão:

- a Serão excluídos trabalhos que não apresentem os resultados no *abstract*, pois ter os resultados no resumo de um trabalho acadêmico é altamente recomendável e, em muitos casos, necessário.

- b Serão excluídos trabalhos que avaliam métodos de previsão de Bitcoin que não sejam baseados em séries temporais, uma vez que o objetivo deste trabalho é avaliar modelos preditivos que utilizem séries temporais.
- c Serão excluídos trabalhos publicados como resumos expandidos ou pôsteres. Esses formatos frequentemente não fornecem a profundidade de pesquisa necessária, podem não passar por uma revisão por pares rigorosa, são limitados em termos de escopo e detalhamento, dificultam a reprodutibilidade dos estudos e geralmente têm menor impacto e visibilidade na literatura científica.
- d Serão excluídos trabalhos que avaliem métodos de previsão de outros ativos financeiros que não sejam criptomoedas, dado que o foco deste trabalho é a previsão de criptomoedas.
- e Para facilitar a leitura e assegurar a consistência na análise, serão excluídos trabalhos que não tenham sido escritos em inglês ou português. Essa exclusão é necessária devido à possível dificuldade de compreensão e interpretação precisa de estudos em outros idiomas, bem como à limitação no acesso a fontes e à padronização dos processos de revisão. Trabalhos em inglês, em particular, também são preferidos devido à sua maior relevância e visibilidade na literatura científica internacional.

2.5.4 Extração de Dados

Após submeter as palavras-chave, conforme definido no item 4, aos motores de busca especificados no item 5, e aplicar os critérios de inclusão e exclusão estabelecidos no item 7, foi identificado inicialmente 19 artigos. Em seguida, foi realizada a leitura completa desses artigos, realizando a extração de dados por meio de um formulário específico, com as seguintes informações:

- I Informações básicas: dados bibliográficos, data de publicação e abstract;
- II Nome do(s) método(s);
- III Criptomoeda(s) analisada(s);
- IV Quantidade de variáveis capturadas na série temporal;
- V Intervalo entre os dados da série temporal;

VI Quais variáveis foram analisadas e incluídas na modelagem.

Ao concluir esta extração de dados, 6 trabalhos foram rejeitados e 13 foram selecionados para comporem a Revisão Sistemática.

2.6 Resultados da Revisão Sistemática

2.6.1 Principais Métodos Encontrados

A partir da revisão sistemática, os artigos podem ser categorizados de acordo com os métodos utilizados. A Tabela 2.1 apresenta quantos e quais os métodos utilizados em cada um dos trabalhos selecionados.

Tabela 2.1: Resumo dos Métodos Utilizados

ID	Título	Nome do(s) Método(s)	Total Métodos
1	Trading bitcoin and online time series prediction	ARIMA, Random Forest, Logistic Regression, LDA, Empirical Conditional	5
2	Regime change and trend prediction for Bitcoin time series data	HMM, RNN, ARMA	3
3	Time series analysis and prediction of bitcoin using long short memory neural network	LSTM	1
4	Predição de séries temporais aplicada ao mercado de criptomoedas	RNN	1
5	Séries temporais para predição de finanças no contexto de criptomoedas	ARIMA, LSTM, SVM	3
6	Utilizando HMM para previsão de preço e estratégia de investimento em criptomoedas BitCoin	HMM, ARIMA, LSTM	3
7	A volatilidade das criptomoedas: um estudo com utilização de modelos GARCH	GARCH	1
8	Previsão de value-at-risk para o mercado de criptomoedas usando modelos EGARCH com regimes markovianos	EGARCH com HMM, EGARCH sem HMM	2
9	Quantificação de risco em finanças: BitCoin sob a avaliação do Value at Risk	GARCH, VAR	2
10	Predicting the direction, maximum, minimum and closing price of daily/Intra-daily bitcoin exchange rate using batch and online machine learning techniques	RNA, RNN, SVM, Ensemble	4
11	Relationship between Bitcoin Exchange Rate and Other Financial Indexes in Time Series	VAR, VECM	2
12	Forecasting cryptocurrency prices time series using machine learning approach	BART, ARIMA, ARFIMA	3
13	Forecasting cryptocurrency markets through the use of time series models	ARIMA	1

O trabalho de [5] destaca-se notavelmente ao comparar uma extensa variedade de métodos, apresentando um total de 5 abordagens distintas. Os autores adotam algoritmos simples, escaláveis e em tempo real, os quais demonstraram um retorno substancial sobre o investimento médio em Bitcoin. Este estudo se destaca por explorar diferentes estratégias, fornecendo uma análise abrangente e fundamentada sobre a eficácia desses métodos no contexto do mercado de criptomoedas.

[37] realiza uma análise comparativa de quatro métodos distintos de previsão de séries temporais. Destaca-se que, embora todos esses métodos estejam fundamentados em princípios de séries temporais, adotam abordagens de aprendizado de máquina para aprimorar a precisão das previsões. Essa integração de técnicas lineares com métodos de aprendizado de máquina representa uma abordagem inovadora e promissora, ampliando as possibilidades de previsão em contextos de séries financeiras.

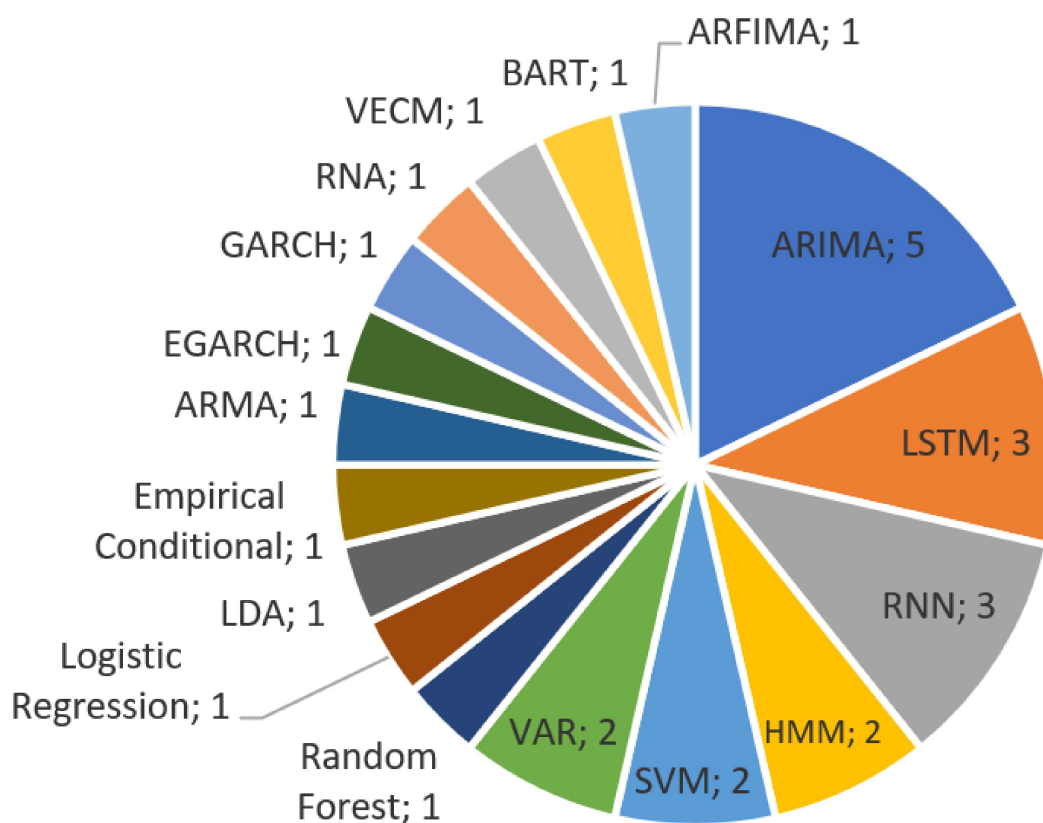


Figura 2.3: Métodos mais utilizados Fonte: Próprio Autor

Conforme pode ser visto na Figura 2.3, o método mais frequentemente empregado nos artigos é o ARIMA, utilizado em cinco instâncias ao longo dos estudos. Além disso, dois artigos apresentam variações desse modelo, a saber, o ARMA e o ARFIMA. Em segundo

lugar, há um empate entre LSTM e RNN, ambos com três ocorrências. Em terceiro lugar, HMM e SVM são mencionados em dois artigos cada. Este panorama evidencia a prevalência e a versatilidade do ARIMA, enquanto destaca o interesse crescente em abordagens baseadas em redes neurais, representadas por LSTM e RNN. O uso ocasional de métodos como HMM e SVM sugere uma diversidade de abordagens para a previsão de séries temporais no contexto de criptomoedas.

De acordo com [5], para a previsão de criptomoedas, métodos clássicos de séries temporais, como os modelos ARIMA produzem previsões ruins e também carecem de interpretação probabilística. A RS indicou que, mesmo o ARIMA não apresentando resultados satisfatórios para previsão de criptomoedas, ele é o mais utilizado. Tal situação ocorre pelo fato de que a série das criptomoedas é uma série paramétrica (com um número finito de observações) e o ARIMA é um modelo paramétrico, além de ser um modelo prático, simples e bastante consolidado na literatura. Por isso o ARIMA é utilizado, comumente nos trabalhos revisados, como parâmetro de comparação, onde na grande maioria das vezes em que o ARIMA é utilizado, também são utilizados outros métodos e comparados os resultados. De acordo com [68], os modelos ARIMA são frequentemente usados em estudos de comparação, desta maneira o ARIMA serve como ponto de partida, onde, caso o modelo comparado seja superior ao ARIMA, merece ser explorado, caso contrário, não é viável.

2.6.2 Criptomoedas Analisadas

Quadro 2.1: Criptomoedas analisadas nos artigos

ID	Título	Criptomoeda(s) Analisada(s)
1	Trading bitcoin and online time series prediction	Bitcoin
2	Regime change and trend prediction for Bitcoin time series data	Bitcoin
3	Time series analysis and prediction of bitcoin using long short term memory neural network	Bitcoin
4	Predição de séries temporais aplicada ao mercado de criptomoedas.	Bitcoin
5	Séries temporais para predição de finanças no contexto de criptomoedas.	Bitcoin
6	Utilizando HMM para previsão de preço e estratégia de investimento em criptomoedas BitCoin	Bitcoin
7	A volatilidade das criptomoedas: um estudo com utilização de modelos GARCH	Bitcoin, Tether, Ethereum, Ripple, EOS
8	Previsão de value-at-risk para o mercado de criptomoedas usando modelos EGARCH com regimes markovianos	Bitcoin, Bitcoin Cash, Ethereum, Ripple, EOS e Stellar
9	Quantificação de risco em finanças: BitCoin sob a avaliação do Value at Risk	Bitcoin
10	Predicting the direction, maximum, minimum and closing price of daily/Intra-daily bitcoin exchange rate using batch and online machine learning techniques	Bitcoin
11	Relationship between Bitcoin Exchange Rate and Other Financial Indexes in Time Series	Bitcoin
12	Forecasting cryptocurrency prices time series using machine learning approach	Bitcoin, Ethereum e Ripple
13	Forecasting cryptocurrency markets through the use of time series models	Ethereum, NEO, HST

De acordo com um dos principais sites de monitoramento de preços de criptomoedas em escala global, o CoinGecko¹, o mercado atual conta com a presença de 12.517 criptomoedas. Entretanto, o Bitcoin destaca-se como a pioneira e a única descentralizada, detendo uma participação superior a 50% no mercado de criptomoedas.

Dada a sua significativa supremacia, as flutuações no preço do Bitcoin exercem uma influência substancial sobre todas as outras criptomoedas. Em virtude desse efeito dominante, o Bitcoin transformou-se em um indicador-chave, agindo como um termômetro para o mercado de criptomoedas. A previsão do valor do Bitcoin torna-se crucial para antecipar os movimentos das principais criptomoedas. Conforme apontado por [22], o Bitcoin atua como um "*driver*", uma força orientadora, cuja tendência é seguida pelas demais criptomoedas.

Essa dinâmica explica, em grande medida, o motivo pelo qual, à exceção de [23], todos os estudos identificados concentram-se na previsão dos valores futuros do Bitcoin, conforme detalhado no Quadro 2.1.

[23] abstêm-se de realizar previsões diretas para o Bitcoin. Contudo, é importante salientar que as criptomoedas analisadas no referido estudo, a saber, Ethereum, NEO e HST, são cotadas em Bitcoin, diferentemente da prática predominante que avalia as criptomoedas em Dólar. Portanto, no contexto deste estudo, as variações no preço do Bitcoin exercem uma influência direta na valoração das criptomoedas examinadas.

2.6.3 Quantidade de observações na série temporal

Segundo [45], modelos de séries temporais são desenvolvidos em dois domínios: temporal e de frequências. No contexto da série em análise, as observações são temporalmente distribuídas, implicando que o pesquisador compila um conjunto de dados com as observações dos valores da criptomoeda ao longo de um período específico, configurando, desse modo, a série temporal. Nos estudos selecionados, não há consenso sobre a quantidade de observações necessárias para modelar séries temporais de criptomoedas, variando de 366 a 7.801.000, com uma média de 602.076 observações. Nota-se que apenas [5] utiliza uma quantidade que atinge a casa dos milhões, devido ao intervalo de 5 segundos entre cada observação, em contraste com a maioria dos artigos que adota um intervalo de 1 dia. Isso resulta em um aumento significativo no número de observações. Considerando a média de observações, excluindo o artigo de [5], têm-se um

¹Disponível em: <https://www.coingecko.com/pt>. Acesso em: 24/11/2023

total de 2.166 observações. Essa análise fornece uma estimativa sobre uma quantidade adequada de observações a serem consideradas na modelagem.

2.6.4 Intervalo entre as observações

Devido ao seu caráter descentralizado, as criptomoedas são transacionadas 24 horas por dia, 7 dias por semana, ao contrário do mercado de ações tradicional que têm seus horários restritos ao funcionamento das bolsas de valores. Neste cenário, plataformas como o TradingView² proporcionam o acompanhamento em tempo real das variações nos preços das criptomoedas por meio de gráficos de velas ou *candlesticks*. Cada vela reflete a variação do preço em um intervalo específico. Uma vela vermelha indica uma queda no preço, enquanto uma vela verde indica um aumento. O intervalo de tempo entre velas pode ser personalizado pelo usuário, variando de 1 segundo a 1 mês.

O intervalo entre as velas é crucial para a classificação dos tipos de investidores em renda variável, pois cada um utiliza prazos distintos para analisar o mercado e tomar decisões de investimento. O *Day Trader*, por exemplo, realiza todas as suas operações no mesmo dia, utilizando intervalos de minutos a horas. Ele busca lucrar com pequenas oscilações de preços, aproveitando a volatilidade do mercado a curto prazo [24]. Existe ainda o *Scalper*, uma variação do *Day Trader*, que executa operações de compra e venda em intervalos extremamente curtos, que podem ser de segundos a minutos, aproveitando movimentos rápidos e breves do mercado, normalmente utilizando intervalos de tempo de segundos a minutos [67]. Já o *Swing Trader* opera com prazos mais longos, de um dia a duas semanas, utilizando intervalos de um dia a uma semana. Ele busca identificar tendências de curto a médio prazo e se posiciona no mercado de acordo com elas [24]. O *Position Trader* opera com prazos ainda mais extensos, de algumas semanas a seis meses, utilizando intervalos entre uma semana e um mês [67]. Ele busca lucrar com tendências de médio a longo prazo. Por fim, o *Buy and Hold* investe em ativos por longos períodos (anos), utilizando intervalos de semanas, meses e até anos [20]. Ele busca o crescimento de capital no longo prazo, acreditando no potencial dos ativos em que investe.

Dada a novidade e volatilidade desse mercado, intervalos mais curtos sofrem mais oscilações, influenciadas por notícias e movimentações de investidores institucionais. Construir um modelo para prever valores em curtos intervalos torna-se ainda mais desafiador. Dentre os trabalhos revisados, apenas 15% avaliaram intervalos inferiores a 1 dia, onde estes caracterizam-se por apresentar um número significativo de observações, os

²<https://br.tradingview.com/>

trabalhos restantes apresentaram dados capturados com intervalos de 1 dia..

2.6.5 Variáveis Incluídas na Modelagem

O conjunto de observações de uma série temporal é essencial para compreender contextos específicos. Conforme destacado por [45], ao analisar uma série de ativos financeiros, a consideração das variáveis de abertura, fechamento, máximo e mínimo diários pode ser benéfica. Esses valores podem contribuir no reconhecimento de tendências, influências e outros atributos essenciais para a compreensão abrangente da série temporal.

De acordo com [45] os dados disponibilizados referentes às criptomoedas são:

1. Data: Dia em que os valores foram coletados;
2. Símbolo: BTCUSD (Sigla da moeda negociada com sua respectiva paridade, neste exemplo verifica-se o valor do Bitcoin em Dólar);
3. Abertura: Valor inicial em dólar;
4. Máxima: Valor máximo atingido em dólar;
5. Mínima: Valor mínimo atingido em dólar;
6. Fechamento: Último valor atingido em dólar;
7. Volume: Quantidade total de Bitcoins comprados e vendidos;
8. Market Cap: Capitalização de mercado é a quantidade total de Bitcoins em circulação multiplicado pelo valor de um único Bitcoin em Dólar.

Conforme mencionado na seção 2.6.4, em gráficos financeiros, os valores são representados por meio do *Candlestick*, conhecido como gráfico de velas. Nesse tipo de gráfico, estão representados os elementos de Abertura, Fechamento, Mínima e Máxima. Onde, para cada período analisado, é criado um retângulo na vertical, denominado vela. Se o valor de Abertura for inferior ao valor de Fechamento, o retângulo é verde, indicando uma valorização do ativo, em que a base do retângulo representa a Abertura e o topo representa o Fechamento. Se o valor de Abertura for superior ao valor de Fechamento, o retângulo é vermelho, indicando uma desvalorização do ativo, em que a base representa o Fechamento e o topo representa a Abertura.

Dentre os trabalhos revisados, apenas [6] utiliza os dados diários de Abertura, todos os outros utilizam apenas os dados diários de Fechamento.

A Figura 2.4 ilustra, em um gráfico de velas, a variação do Bitcoin em intervalos de tempo de 1 dia, no período de 01 de Outubro de 2023 a 23 de Novembro de 2023.



Figura 2.4: Gráfico *CandleStick* do Bitcoin (1 dia) Fonte: TradingView

2.6.6 Discussão

[54] emprega um modelo da família ARCH em sua modelagem, mas ressalta a existência de pontos controversos e inconclusivos no meio acadêmico sobre esses tipos de modelos. Além disso, aponta a escassez de estudos relacionados à previsão de volatilidade em criptomoedas usando ARCH.

Por outro lado, [23], utiliza exclusivamente o modelo ARIMA para prever os valores das criptomoedas. Entretanto, constata que a variação das estimativas de erro não parece estar intimamente relacionada à maturidade dos mercados, mas depende das características intrínsecas das séries temporais analisadas.

Buscando testar modelos lineares, como o ARIMA, e de Redes neurais, como o BART. [22] compara os modelos BART, ARIMA e ARFIMA, conclui que o algoritmo BART apresentou melhor precisão nas séries temporais de Bitcoin, Ethereum e Ripple. O BART, adaptado dos modelos padrão de árvores de regressão, combina o algoritmo C&RT e modelos autorregressivos como o ARIMA.

O modelo RNN, de acordo com [4], apresenta bom desempenho em períodos de baixa volatilidade, mas falha em períodos de grande variação nos preços do Bitcoin. Já [39], analisando os anos de 2017 e 2018, empregam os modelos EGARCH e MS-EGARCH combinados com diferentes distribuições de probabilidade estatística. Eles observam que os retornos diários das criptomoedas apresentam mudanças de regime em sua volatilidade, indicando que choques nas séries temporais tendem a ter efeitos duradouros.

[62] realizam uma avaliação do VaR (Value at Risk) com base em modelos GARCH, aplicando o modelo GARCH(1,1) e analisando a taxa de rentabilidade, que indica uma expectativa de remuneração diária de 0,098%. Já [34] buscam avaliar a aplicabilidade de Redes Neurais Recorrentes para as séries temporais da oscilação do preço do Bitcoin perante o Euro. Em uma comparação entre RNN, HMM e ARMA, o modelo baseado em RNN apresentou o melhor poder preditivo.

[26] identificam um acréscimo no valor do Bitcoin no início e final dos anos. Os modelos usando Redes Neurais LSTM apresentaram menores erros de predição em períodos de baixa volatilidade, mas não foram eficazes em períodos de alta volatilidade. O modelo ARIMA, apesar da necessidade de redimensionar a série, conseguiu acompanhar o desenvolvimento temporal e prever futuras possibilidades para o valor do Bitcoin. Já o SVM apresentou melhores resultados que o LSTM em períodos de alta volatilidade.

[2] mencionam que a taxa de retornos sobre o BTC varia de acordo com a taxa de

Hash da rede, o volume de transações e o custo por transação.

[5] destaca que os algoritmos baseados em ARIMA e EC apresentaram os piores resultados, embora EC tenha superado ARIMA. Algoritmos de classificação, como Random Forest, Logic Regression e LDA, obtiveram bons resultados, indicando que os padrões de evolução histórica são bem capturados por esses algoritmos.

[63] revelam que, entre os modelos testados, o HMM, utilizando Log Retornos como variável observada na série temporal, obteve os melhores resultados, com um retorno do método proposto de 5x.

2.6.7 Conclusão da RS

Os modelos que se destacaram pelos melhores resultados foram os modelos RNN (Redes Neurais Recorrentes) e LSTM (Memória de Curto Prazo Longa), ambos baseados em Redes Neurais. Os desafios identificados nos estudos estão intrinsecamente ligados à notória volatilidade do Bitcoin. No entanto, uma descoberta significativa foi a presença de dois estados distintos na série temporal do Bitcoin. Em um estado, o ativo exibe um comportamento mais previsível, com menor volatilidade e uma trajetória mais linear. No segundo estado, o Bitcoin revela taxas elevadas de assimetria, tornando a tarefa de previsão notavelmente desafiadora. Embora muitos modelos tenham obtido sucesso durante o primeiro estado, a transição para o segundo estado permanece um ponto crítico, desafiando a capacidade preditiva dos modelos.

Os trabalhos futuros sugerem estratégias para capturar e antecipar estes *outliers* que conduzem ao segundo estado. Entre elas, destacam-se a análise quantitativa de dados *intraday* de alta frequência, a incorporação de modelos que considerem fatores econômicos, políticos e sociais em um contexto global, além do uso de informações provenientes de grandes portais de economia e análise de sentimentos extraídos de redes sociais, com ênfase no Twitter, a plataforma amplamente utilizada por criadores e investidores de criptomoedas.

A revisão sistemática realizada revelou que, embora o modelo ARIMA não apresente os resultados mais precisos, ele se destaca pela praticidade, simplicidade e robustez, sendo amplamente utilizado na literatura como base para comparação com outros modelos. Já o LSTM, conforme indicado por [26] e [2], obteve os melhores resultados nos testes realizados. [34], [4] e [6] demonstram que o modelo RNN também apresenta um desempenho promissor. A análise do item 2.6.2 identificou o Bitcoin como a criptomoeda

mais estudada, detendo mais de 50% do mercado e exercendo forte influência sobre os preços das demais moedas digitais. Com base nos resultados da revisão e na relevância do Bitcoin, decidiu-se implementar neste estudo os modelos ARIMA, RNN e LSTM para a previsão do valor do Bitcoin. A previsão será realizada utilizando dados diários de fechamento de mercado, compreendendo aproximadamente 3.000 observações na série temporal. Essa escolha abrangente permitirá comparar o desempenho dos diferentes modelos e identificar o mais adequado para as características específicas do Bitcoin.

2.7 Métodos Preditivos

Segundo [30], a prática de *forecasting* é uma ocorrência estatística comum em contextos empresariais. Essa prática auxilia na ampliação da informação para decisões relacionadas ao cronograma de produção, demanda de transporte, consumo energético, entre outras. Além disso, fornece orientações cruciais para o planejamento estratégico de longo prazo. A realização de previsões é uma ferramenta essencial para a eficácia e eficiência do planejamento. No entanto, algumas tarefas são mais desafiadoras do que outras, como evidenciado pela dificuldade de prever com precisão as taxas de câmbio, conforme [30] destacam.

Outra perspectiva, conforme [43], define *forecasting* como a antecipação de eventos. Essa prática pode ser classificada como *short-term*, referindo-se à previsão de períodos curtos (horas, dias, semanas); *mid-term*, que abrange um ou dois anos; e *long-term*, estendendo-se por vários anos. Essa atividade desempenha um papel crucial em diversos setores, incluindo indústria, governo, economia, medicina, política e finanças.

A classificação dos métodos de previsão, como detalhado por [30], divide-se em duas categorias principais: métodos qualitativos e métodos quantitativos.

2.7.1 Métodos Qualitativos

Esses métodos entram em cena quando não há dados históricos disponíveis para a previsão, sendo, portanto, designados como previsão por julgamento [30]. Um exemplo prático ocorre quando uma empresa está planejando lançar um novo serviço, para o qual não há dados históricos disponíveis. Nesse caso, a equipe de previsão pode utilizar métodos de previsão por julgamento, reunindo especialistas do setor que possuam conhecimento aprofundado sobre tendências emergentes, comportamento do consumidor e fatores que podem impactar a aceitação do novo serviço.

Conforme indicado por [30], pesquisas nesse campo evidenciam que a precisão da previsão por julgamento aumenta quando o previdente possui: (i) um conhecimento substancial no domínio e (ii) acesso a informações mais recentes e oportunas. Esses métodos também contribuem para aprimorar a previsão quantitativa, ao incorporar informações que não podem ser formalmente modeladas em um contexto puramente estatístico. Entretanto, segundo [30], as previsões estatísticas geralmente superam as previsões baseadas exclusivamente no julgamento.

2.7.2 Métodos Quantitativos

A análise quantitativa, no contexto de previsão, concentra-se exclusivamente no histórico para realizar projeções futuras. Para aplicar essa abordagem, duas condições devem ser satisfeitas: a expectativa de que certos padrões se repetirão no futuro e a existência de uma base histórica de dados. Dentro dessa categoria, diversas técnicas de previsão são empregadas, cada uma com características, vantagens e desvantagens específicas, como será explorado posteriormente.

Conforme destacado por [30], a previsão quantitativa se desdobra em três principais categorias:

1. Modelos Explicativos: Esses modelos pressupõem que a variável a ser prevista possui alguma forma de relação explicativa com uma ou mais variáveis. Essa categoria é útil, pois não requer valores históricos das variáveis a serem previstas. Por exemplo, a Previsão de Vendas (PV) de um produto, pode ser definida por fatores como publicidade, preço, sazonalidade, entre outros. O modelo busca descrever o relacionamento entre essas variáveis para fazer previsões futuras entre outros fatores, com um termo de erro introduzido para permitir variação aleatória e capturar efeitos não incluídos no modelo. O propósito desses modelos é descrever o formato do relacionamento para prever valores futuros.
2. Modelos de Séries Temporais: Utilizando apenas informações da variável a ser prevista, esses modelos não buscam identificar os fatores que afetam seu comportamento.

$$PV_{t+1} = f(V_t, V_t - 1, V_t - 2, \dots, erro)$$

A previsão do próximo período é baseada nos valores passados da variável ou em

erros passados. Modelos comuns nessa categoria incluem ARIMA, *Exponential Smoothing*, entre outros [52].

3. Modelos Mistos: Esses modelos combinam características dos modelos explicativos e de séries temporais. Incorporam elementos dos dois tipos mencionados anteriormente para prever a variável de interesse. Dependendo da disciplina, podem ser conhecidos como Modelos de Regressão Linear Dinâmicos ou modelos lineares, assumindo que a função é linear.

2.8 ARIMA

Os princípios e conceitos estatísticos subjacentes aos modelos de previsão de Box & Jenkins (1970) [13] são amplamente empregados para desenvolver modelos paramétricos destinados a séries temporais *univariadas*. A essência dessa abordagem é derivar uma fórmula que minimize os erros associados. Esse processo envolve uma iteração contínua até se chegar a uma fórmula específica que replica, da maneira mais próxima possível, os padrões da série original, otimizando assim a precisão das previsões. Dessa forma, são aplicados testes estatísticos para assegurar a eficácia do modelo proposto. O conjunto de modelos fundamentais inclui AR (autoregressivo), MA (média móvel) e o combinado ARMA, além da técnica de diferenciação regular (RD). Quando a técnica de diferenciação regular é integrada aos modelos AR e MA, eles são conhecidos como ARIMA, onde o "I" simboliza "integrado", destacando o processo de diferenciação [10]. No contexto deste estudo, foram empregados modelos de previsão *univariados*, especificamente centrados em uma única variável temporal. O uso do ARIMA pode ser considerado importante por algumas razões:

1. Em situações complexas, com um grande número de variáveis, o método *univariado* pode ser a única solução possível para modelagem.
2. Quando há opções de métodos *multivariados*, o método *univariado* serve como referência para avaliar as abordagens mais avançadas.
3. Examinando os modelos *univariados*, é possível obter *insights* sobre tendências de longa duração, variações sazonais e outros aspectos dos dados.
4. Para evitar questões como regressões espúrias ao conduzir análises *multivariadas*, é essencial realizar uma análise *univariada* preliminar.

2.8.1 Autorregressivo (AR)

De acordo com [15] e [14], o método autorregressivo com ordem definida por p , em que p representa a quantidade de parâmetros do modelo, pode ser denotado por $AR(p)$. A equação (2.2) representa a forma clássica do modelo autorregressivo $AR(p)$, onde as observações passadas são ponderadas pelos parâmetros β . A equação (2.3) é uma forma mais compacta usando o operador de defasagem B , representando a mesma relação.

$$Z_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} + \alpha_t \quad (2.2)$$

ou

$$\beta(B)Z_t = \alpha_t \quad (2.3)$$

Onde:

- Z_t é a variável dependente no tempo t , representando a observação atual na série temporal que está sendo modelada.
- $Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$ são as variáveis independentes defasadas (atrasadas) no tempo, onde p é a ordem do modelo autorregressivo. Essas representam as observações passadas que são utilizadas para prever a observação atual Z_t .
- $\beta_1, \beta_2, \dots, \beta_p$ são os parâmetros do modelo que multiplicam as variáveis defasadas. Cada β representa o peso associado à respectiva observação passada.
- α_t é o termo de erro no tempo t , representando a parte não explicada ou aleatória do modelo.
- $\beta(B)$ é um polinômio defasado em B , onde B representa o operador de defasagem. Este polinômio é utilizado para compactar a notação do modelo.

2.8.2 Média Móvel (MA)

De acordo com [15] e [14], modelos de médias móveis (MA) descrevem o valor atual de uma série temporal como uma combinação linear de valores passados do ruído branco. O número de parâmetros em um modelo MA é denotado por q .

A equação 2.4 representa a forma clássica do modelo de médias móveis $MA(q)$, onde os termos de erro defasados são ponderados pelos parâmetros θ . A equação 2.5 é uma forma mais compacta usando o operador de defasagem B , representando a mesma relação.

$$Z_t = \alpha_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q} \quad (2.4)$$

ou

$$Z_t = \theta(B)\alpha_t \quad (2.5)$$

Onde:

- Z_t é a variável dependente no tempo t , representando a observação atual na série temporal que está sendo modelada.
- α_t é o termo de erro no tempo t , que é assumido como um ruído branco, representando a parte não explicada ou aleatória do modelo.
- $\epsilon_{t-1}, \epsilon_{t-2}, \dots, \epsilon_{t-q}$ são os termos de erro defasados (atrasados) no tempo, onde q é a ordem do modelo de médias móveis. Cada termo ϵ representa a componente aleatória associada a uma observação passada.
- $\theta_1, \theta_2, \dots, \theta_q$ são os parâmetros do modelo que multiplicam os termos de erro defasados. Cada θ representa o peso associado à respectiva componente aleatória.;
- $\theta(B)$ É um polinômio defasado em B , onde B representa o operador de defasagem. O polinômio $\theta(B)$ é utilizado para compactar a notação do modelo de médias móveis.

2.8.3 Autorregressivo e Média Móvel (ARMA)

De acordo com [15] e [14], o modelo em questão é caracterizado por uma sequência de equações. A abordagem $ARMA(p, q)$ pode ser visualizada como uma fusão dos modelos $AR(p)$ e $MA(q)$. A equação 2.6 mostra a fusão dos modelos $AR(p)$ e $MA(q)$ no contexto do modelo $ARMA(p, q)$. A equação 2.7 representa a mesma relação usando operadores polinomiais.

$$Z_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} - \alpha_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q} \quad (2.6)$$

ou

$$B_p(L)C_t = \theta_q(L)\epsilon_t \quad (2.7)$$

- Z_t representa a variável dependente no tempo t , que é a observação atual na série temporal que está sendo modelada.
- $Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$ são os valores passados da série temporal, onde p é a ordem do modelo autorregressivo (AR). Cada Y representa uma observação em um tempo anterior.
- $\beta_1, \beta_2, \dots, \beta_p$ são os coeficientes que multiplicam os valores passados da série temporal no componente AR. Cada β representa o peso associado ao valor correspondente de Y no modelo AR.
- α_t é o termo de erro no tempo t , representando a parte não explicada ou aleatória do componente AR.
- $\theta_1\epsilon_{t-1}, \theta_2\epsilon_{t-2}, \dots, \theta_q\epsilon_{t-q}$ são os termos de erro passados multiplicados pelos coeficientes θ no componente de médias móveis (MA). Cada θ representa o peso associado ao termo de erro correspondente no modelo MA.
- ϵ_t é o termo de erro branco no tempo t , que é uma sequência de variáveis aleatórias independentes e identicamente distribuídas.
- $B_p(L)$ representa o operador polinomial relacionado aos termos autorregressivos, onde L é o operador de defasagem.
- $\theta_q(L)$ representa o operador polinomial relacionado aos termos de médias móveis, onde L é o operador de defasagem.

2.8.4 Autorregressivo Integrado de Média Móvel (ARIMA)

Quando encontra-se um processo com tendência que não é estacionário, é possível aplicar uma transformação usando um operador de diferença para alcançar a estacionariedade da série. Assim, no contexto apresentado, um modelo ARIMA(p, d, q) é empregado, sendo um processo que combina características autorregressivas e de médias móveis, com uma integração de ordem d para descrever a série.

De acordo com [15], [43] e [44], quando a série não é estacionária, aplica-se a Equação 2.8.4.

$$Z_t = Y_t - Y_{t-1} = \Delta Y_t \quad (2.8)$$

A Equação representa a primeira diferença da série temporal para torná-la estacionária

Onde:

- Z_t : Representa a série temporal diferenciada. Quando $d = 1$, Z_t é a primeira diferença da série temporal original Y_t . E quando $d > 1$, Z_t representa a diferença de ordem d da série temporal original.
- Y_t é a série temporal original.
- ΔY_t representa a diferença entre dois valores consecutivos da série temporal original. Em outras palavras, é a série das primeiras diferenças, onde $\Delta Y_t = Y_t - Y_{t-1}$.

Ao aplicar d diferenças, a formulação anterior pode ser reajustada conforme pode ser visto na Equação 2.9.

$$Z_t = \Delta^d Y_t \quad (2.9)$$

Onde $\Delta^d Y_t$ indica a diferença de ordem d da série temporal original, obtida aplicando a operação de diferença d vezes.

Portanto, a Equação 2.10 mostra a formulação geral do modelo ARIMA, onde p é a ordem autorregressiva, d é a ordem de diferenciação e q é a ordem da média móvel.

$$Z_t = \beta_1 Z_{t-1} + \beta_2 Z_{t-2} + \dots + \beta_p Z_{t-p} + \dots + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} \quad (2.10)$$

Onde:

- $\beta_1, \beta_2, \dots, \beta_p$ são os coeficientes autorregressivos que multiplicam os valores passados da série temporal no componente AR do modelo ARIMA.
- $\theta_1 \epsilon_{t-1}, \theta_2 \epsilon_{t-2}, \dots, \theta_q \epsilon_{t-q}$ são os termos de médias móveis multiplicados pelos coeficientes θ no componente MA do modelo ARIMA.

- ϵ_t É o termo de erro branco no tempo t , que é uma sequência de variáveis aleatórias independentes e identicamente distribuídas.

Dessa forma, ao inserir a Equação 2.8.4 em 2.10, é obtida a Equação 2.11.

$$\Delta Y_t = \beta_1 \Delta_{t-1} + \beta_2 \Delta_{t-2} + \dots + \beta_p \Delta_{t-p} + \dots + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} \quad (2.11)$$

2.8.5 Autorregressivo Integrado de Média Móvel com Sazonalidade (SARIMA)

De acordo com [30], modelos SARIMA são modelos de séries temporais que combinam as características dos modelos ARIMA e modelos sazonais. Eles são usados para prever séries temporais que apresentam tendências, sazonalidade e ruído.

A fórmula matemática para um modelo SARIMA (Seasonal AutoRegressive Integrated Moving Average) é uma extensão da fórmula do modelo ARIMA, incorporando componentes sazonais. O modelo SARIMA é denotado como SARIMA(p, d, q)(P, D, Q)_s, onde:

- p: Ordem do componente autorregressivo não sazonal.
- d: Ordem do componente de diferenciação não sazonal.
- q: Ordem do componente de médias móveis não sazonal.
- P: Ordem do componente autorregressivo sazonal.
- D: Ordem do componente de diferenciação sazonal.
- Q: Ordem do componente de médias móveis sazonal.
- s: Período sazonal.

Por exemplo, um modelo SARIMA (1,1,1)(1,1,1)₁₂ descreve uma série temporal com uma tendência linear, uma sazonalidade anual e ruído branco.

A fórmula geral de um modelo SARIMA é:

$$\begin{aligned}
& (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - \Phi_1 B^s - \Phi_2 B^{2s} - \dots - \Phi_P B^{Ps}) \\
& (1 - \Delta^d)(1 - \Delta_s^D)Z_t = \\
& (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q)(1 + \Theta_1 B^s + \Theta_2 B^{2s} + \dots + \Theta_Q B^{Qs})\varepsilon_t
\end{aligned} \tag{2.12}$$

Onde:

Z_t é a série temporal observada.

B é o operador defasagem (lag).

Δ é o operador de diferença não sazonal.

Δ_s é o operador de diferença sazonal.

ε_t é o termo de erro branco.

$\phi_1, \phi_2, \dots, \phi_p$ são os parâmetros autorregressivos não sazonais.

$\Phi_1, \Phi_2, \dots, \Phi_P$ são os parâmetros autorregressivos sazonais.

$\theta_1, \theta_2, \dots, \theta_q$ são os parâmetros de médias móveis não sazonais.

$\Theta_1, \Theta_2, \dots, \Theta_Q$ são os parâmetros de médias móveis sazonais.

De acordo com [61], existem várias técnicas que podem ser usadas para ajudar a escolher os parâmetros P, D e Q, incluindo:

- A análise de autocorrelação.
- A análise de autocorrelação parcial.
- A análise de resíduos.
- Estimando os parâmetros do modelo.

Depois de escolher os parâmetros P, D e Q, os parâmetros do modelo são estimados usando o método dos mínimos quadrados ordinários. Este método minimiza a soma dos quadrados dos erros entre os valores reais da série temporal e os valores previstos pelo modelo.

2.9 Modelos de Aprendizado de Máquina

Sendo uma disciplina centrada no avanço de algoritmos capazes de aprender a partir de experiências, o aprendizado de máquina, ou machine learning, destaca-se no desenvolvimento de estratégias para robôs, processamento de linguagem natural, previsão de séries financeiras e diversas outras tarefas. Esta área tem como metas primárias o avanço de técnicas computacionais que possibilitem a simulação do processo de aprendizado e a criação de sistemas aptos a adquirir conhecimento de forma automatizada [42]. Nas seções a seguir, serão expostos os conceitos de AM que serão empregados neste estudo. Os modelos mencionados têm a finalidade de modelar e antecipar variáveis dependentes nas séries temporais abordadas no capítulo anterior.

2.9.1 Algoritmos de Aprendizado de Máquina

Um algoritmo de aprendizado é capaz de internalizar conceitos ou padrões a partir de dados. De acordo com a definição de [42], este tipo de algoritmo aprende da experiência \mathbf{E} em relação a uma classe de tarefas \mathbf{T} e uma medida de desempenho \mathbf{P} , se seu desempenho nas tarefas \mathbf{T} , medido por \mathbf{P} , melhora com a experiência \mathbf{E} .

Aqui, o termo \mathbf{T} refere-se à habilidade de executar uma tarefa específica. Por exemplo, se o objetivo é conceder a um robô a habilidade de varrer, aprender a varrer seria a tarefa em questão. O termo \mathbf{P} representa a medida de desempenho da habilidade \mathbf{T} , enquanto \mathbf{E} indica a experiência, que pode ser um conjunto de dados ou uma coleção de muitos pontos.

A tarefa \mathbf{T} pode assumir vários tipos, como regressão, classificação, agrupamento, entre outros. Algoritmos de aprendizado de máquina podem ser categorizados em diversos tipos, com base em suas características e métodos de aprendizagem. Os 3 principais tipos são: Aprendizado Supervisionado, Aprendizado Semisupervisionado e Aprendizado Não Supervisionado.

Para uma compreensão mais aprofundada, serão abordados os três tipos de aprendizado.

1. **Aprendizado Supervisionado.** A abordagem do Aprendizado Supervisionado é realizada por algoritmos que desenvolvem modelos preditivos. O termo "supervisão" implica a presença de um supervisor que tem conhecimento das saídas (ou "rótulos") esperadas para cada dado, como indicado por [25].

Os algoritmos supervisionados buscam identificar a relação entre atributos (variáveis independentes) e um atributo objetivo (variável dependente). Essa relação, uma vez descoberta, é representada como um modelo, uma estrutura que descreve e explica os fenômenos subjacentes nos dados. Esses modelos têm a capacidade de prever o valor do atributo alvo com base nos valores dos atributos de entrada, conforme apontado por [57].

O aprendizado supervisionado envolve a presença de um supervisor que desempenha o papel de um professor durante o processo de aprendizado. O objetivo é adquirir uma regra geral que otimize o mapeamento das entradas para as saídas por meio de uma função.

No âmbito do aprendizado supervisionado, destacam-se dois tipos principais: classificação e regressão. Os modelos de regressão realizam o mapeamento do espaço de entrada para um valor real, enquanto os modelos de classificação realizam o mapeamento da entrada para um espaço de classes predefinidas.

Desta forma, constata-se que os modelos RNN e LSTM são algoritmos de Aprendizado Supervisionado do tipo regressão.

2. **Aprendizado semissupervisionado.** Esta modalidade surge quando desenvolve-se classificadores para tarefas específicas e se presume a existência de um conjunto de dados com todos os exemplos rotulados.

Na prática, deparamo-nos com situações do mundo real em que há uma quantidade significativa de dados não rotulados, ou é necessário o conhecimento do domínio de aplicação para realizar a rotulação, como em casos de indexação de vídeo, diagnósticos médicos, modelagem de tópicos, entre outras aplicações. Além disso, a rotulação de dados é um processo que consome tempo e pode exigir recursos caros, muitas vezes indisponíveis.

A principal motivação para esse tipo de aprendizado reside na abundância de exemplos não rotulados e na escassez de exemplos rotulados. Diante desses cenários, utiliza-se um conjunto limitado de dados corretamente rotulados para estender esses rótulos a dados que não possuem rótulo.

Uma definição mais precisa, conforme mencionado por [16], destaca que a essência do aprendizado semissupervisionado é empregar os exemplos rotulados para adquirir informações sobre o problema e, em seguida, utilizar essas informações para orientar o aprendizado a partir dos exemplos não rotulados.

3. Aprendizado Não Supervisionado. Em tarefas de descrição, o objetivo é elucidar um conjunto de dados para aprimorar sua compreensão. Os algoritmos de Aprendizado de Máquina empregados nesse conjunto prescindem do uso de rótulos. Por exemplo, o *clustering* é uma tarefa que procura agrupar dados semelhantes com base em centroides. Outra tarefa descritiva envolve a identificação de regras de associação que relacionam grupos de atributos, entre outros conjuntos, como mencionado por [25].

Dentro desse domínio de aprendizado sem supervisão, encontram-se também as tarefas de redução de dimensionalidade e aprendizado de representações dos dados, sem a necessidade de rótulos. Esses algoritmos têm a capacidade de segmentar grupos com características semelhantes [1].

Certas abordagens de Aprendizado de Máquina podem ser interpretadas como desafios de otimização. Esse procedimento envolve a definição de um critério de aprendizagem, ou seja, a função a ser otimizada, com o intuito de maximizar ou minimizar uma função objetivo [25]. Há uma variedade de problemas de aprendizado de máquina, abrangendo desde a detecção de anomalias, previsão de séries temporais, tradução automática, classificação, processamento de imagem, detecção de rosto, filtragem de spam, previsão do tempo, entre outros.

2.9.2 Deep Learning

Um termo informal foi introduzido para distinguir as vantagens e desvantagens de dois conceitos principais: arquiteturas "rasas" e "profundas" [8]. Inicialmente, arquiteturas "rasas" com uma ou duas camadas ocultas eram predominantes na modelagem de problemas. Apesar do sucesso de arquiteturas como a *Multilayer Perceptron* (MLP), sua capacidade representativa era insuficiente para abordar uma variedade de problemas reais [21].

O aprendizado conduzido por essas redes profundas é conhecido como *Deep Learning* (DL). Métodos que utilizam DL buscam descobrir uma hipótese, modelo ou função a partir de um conjunto de dados (exemplos) e um método para orientar o aprendizado. Ao final, o DL deve ser capaz de fornecer uma função capaz de receber o conjunto de dados e gerar uma representação adequada. As arquiteturas profundas permitem representar uma ampla gama de funções de forma mais compacta do que as arquiteturas rasas [8].

[55] levantam uma questão interessante sobre o DL: qual é a diferença entre

Aprendizado de Máquina (AM) e DL? Segundo os autores, "a diferença está baseada em como aprende a função $f(\cdot)$. Algoritmos "não-DL"(rasos ou superficiais) buscam diretamente por uma função que, a partir de um conjunto de parâmetros, gere um resultado. No DL, há métodos que aprendem a função $f(\cdot)$ por meio de composições de funções, como exemplificado pela equação (2.13):

$$f(x) = f_L(\dots f_2(f_1(x_1, W_1), W_2) \dots), W_L \quad (2.13)$$

Aqui, f_L representa a primeira camada com índice L que recebe como entrada o conjunto de dados x_1 . Cada função utiliza parâmetros (matriz W_l) para transformar os dados de entrada. A ideia crucial no DL é aprender representações sucessivas dos dados, ou seja, os parâmetros W são aprendidos diretamente dos dados, fazendo combinações de funções que transformam vetores e mapeiam de um espaço para outro até chegar ao resultado final [55]

Desta maneira, RNN e LSTM são considerados arquiteturas de aprendizado profundo.

2.10 Redes Neurais Recorrentes

As Redes Neurais Recorrentes (RNN) constituem uma classe de modelos de redes neurais proposta na década de 80 por [59]. Concebidas para mapear sequências de dados de entrada em sequências de dados de saída, as RNN são particularmente adequadas para modelar dados com dependência sequencial, como texto, genes, fala e para a previsão de séries temporais.

Estas redes foram concebidas com a capacidade de manter informações do passado para utilização no futuro. Nos neurônios ocultos de uma RNN, são estabelecidas conexões associadas a um intervalo de tempo. Essas conexões permitem à RNN reter informações passadas para identificar correlações temporais entre eventos distantes nos dados. Destaca-se pelo seu modelo de memória, que, de certa forma, reflete o funcionamento da memória humana [53].

Em uma RNN, os neurônios ocultos possuem conexões que levam em consideração o tempo e a sequência ordenada temporalmente. Esse modelo é caracterizado pelo tipo de memória que possui, semelhante à memória humana.

Analogamente ao modo como uma pessoa compreende uma palavra em um texto com base nas palavras anteriores, as RNN possuem loops internos e conexões que mantêm essa

informação. A RNN pode ser vista como uma rede com múltiplas cópias de si mesma, transmitindo mensagens ao seu sucessor ao longo do tempo. A Figura 2.5 ilustra uma rede neural (A) no lado esquerdo, com uma entrada x e, a cada passo x_t , produz uma saída h_t . No lado direito, é apresentado o mesmo processo da rede desenrolada. O loop possibilita que as informações persistam, sendo transmitidas a si mesma.

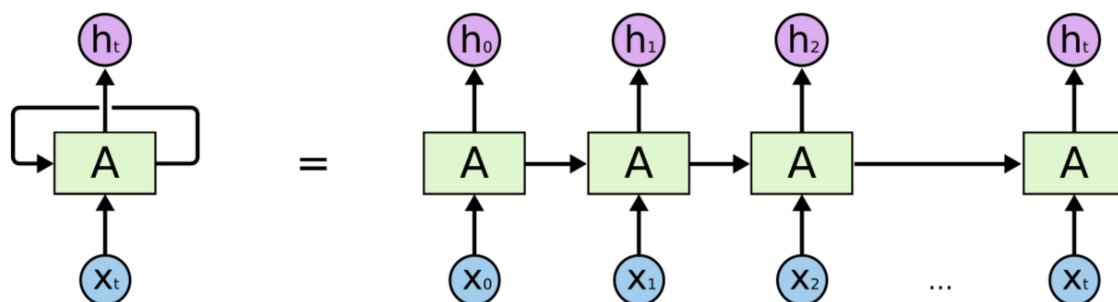


Figura 2.5: Modelo de RNN Fonte: [50]

A capacidade de uma RNN de utilizar informações anteriores para a tarefa atual representa uma de suas vantagens. No entanto, enquanto alguns problemas demandam a verificação de informações recentes para tomar decisões, outros necessitam entender o contexto passado. Uma limitação das RNN é a necessidade de lidar com lacunas consideráveis entre o presente e o passado para aprender essas informações progressas, muitas vezes incapazes de estabelecer essa conexão. O próximo segmento abordará mais detalhadamente essa questão.

2.10.1 O Problema das Dependências de Longo Prazo

Uma das propriedades das RNN é a habilidade de conectar informações prévias a uma tarefa atual. No entanto, elas não conseguem fazer isso de maneira eficiente. Por exemplo um modelo de linguagem em que a previsão da última palavra é feita com base nas palavras anteriores, como em "As árvores estão no parque". Nesse caso específico, o contexto passado não é crucial, e a última palavra pode ser prevista como "parque". Em situações semelhantes, as RNN podem aprender o conceito necessário mesmo quando a lacuna entre a informação relevante e o ponto em que é necessária é pequena. No entanto, há cenários em que é essencial entender mais sobre o contexto, e a lacuna se torna muito grande.

Agora considerando a previsão da última palavra na frase "Eu sou da Itália eu falo italiano". Sugestões recentes de dados indicam que a palavra subsequente

é, provavelmente, um nome de língua. No entanto, se o intuito é especificar qual língua, é essencial ter o contexto da Itália, localizado mais anteriormente. Existe a possibilidade de a lacuna entre a informação crucial e o ponto em que é necessária se tornar consideravelmente extensa. Na Figura 2.6, é observado que a lacuna entre a informação relevante (x_0 e x_1) e o ponto onde é necessária (h_{t+1}) é maior. Enquanto na Figura 2.7 a distancia entre a informação relevante (x_0 e x_1) e o ponto onde é necessária (x_3) está bastante reduzida.

Nessas situações em que a distância é consideravelmente grande, as desvantagens das RNN se tornam evidentes, e na prática, elas enfrentam dificuldades para estabelecer essas dependências de longo alcance.

Embora teoricamente as RNN tenham a capacidade de aprender dependências de longo prazo, na prática, essa habilidade é desafiadora de ser efetivada. Estudiosos como [27] e [9] exploraram esses problemas. [27] identificou que tanto as redes *Feedforward Neural Network* (FFNN) quanto as RNN são difíceis de treinar com o algoritmo *Back Propagation*, devido aos problemas de *vanishing gradient*, quando os sinais de erro que se "propagam para trás no tempo" tendem a desaparecer e *exploding gradient*, quando estes sinais de erro tendem a explodir, em arquiteturas profundas. Embora o modelo da RNN seja simples e poderoso em teoria, ele é altamente sensível a esses problemas.

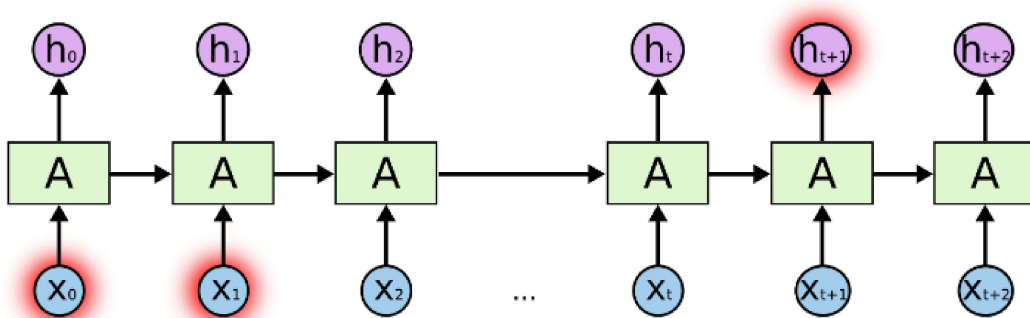


Figura 2.6: Informação relevante com lacuna longa Fonte: [50]

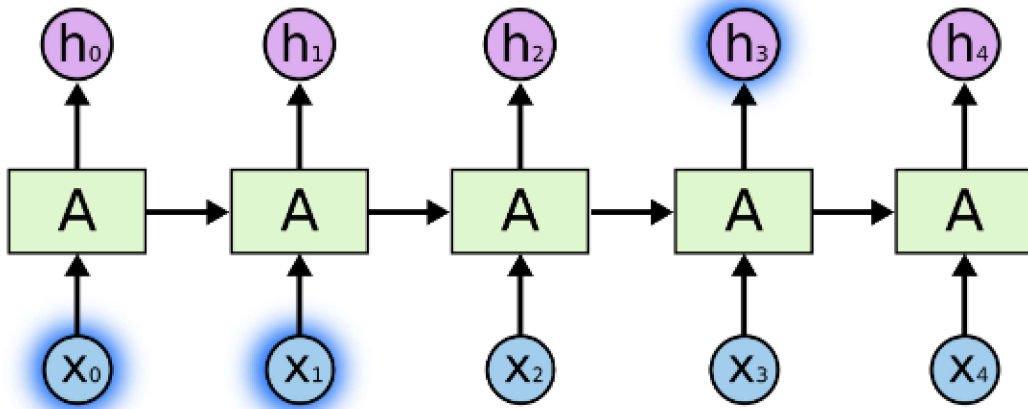


Figura 2.7: Informação relevante com lacuna curta Fonte: [50]

[9] identificaram esses problemas quando:

1. **Exploding gradient** ocorre com um aumento gigantesco na norma do gradiente durante a fase de treinamento. Esses eventos resultam da explosão dos componentes de longo prazo, que podem crescer exponencialmente mais do que os de curto prazo.
2. **Vanishing gradient**, ao contrário do problema anterior, implica que os componentes de longo prazo crescem exponencialmente em direção à norma 0, tornando impossível para o modelo aprender as correlações entre eventos temporalmente distantes.

2.11 Long Short-Term Memory

Nos meados da década de 1990, uma variante das RNN, conhecida como redes Long Short-Term Memory (LSTM), foi apresentada por [27]. As LSTMs destacaram-se por sua habilidade de aprender dependências de longo prazo, ganhando popularidade em vários estudos subsequentes.

Para abordar as questões de *vanishing* e *exploding*, [27] propuseram uma nova arquitetura que mantém um erro constante usando estados internos e unidades especiais chamadas *gates*. A distinção fundamental da LSTM em relação a outras redes, reside no uso de unidades chamadas *Constant Error Carousel* (CEC), as quais empregam uma função de ativação f conectada a si mesma com pesos iguais a 1.0. Devido a isso e à propriedade de que a derivada de uma constante é 1.0, o erro não se propaga através do

CEC, eliminando problemas de *vanish* e *exploding*. Essas CECs são fundamentais para a capacidade da LSTM de memorizar eventos passados, superando as RNN, que falham após 10 passos [60].

A arquitetura da LSTM é semelhante à das RNNs, mas foi especificamente projetada para enfrentar o problema da dependência a longo prazo e reter informações ao longo do tempo. Todas as redes neurais recorrentes têm uma estrutura repetitiva, com a primeira camada sendo uma camada com uma função *tanh* (tangente hiperbólica). A distinção principal com relação a LSTM é que seus módulos são repetitivos, mas, em vez de ter uma única camada, as LSTMs possuem quatro camadas internas com funções *sigmoide*, *sigmoide*, *tanh* e *sigmoide*. Na Figura 2.8, é representada uma RNN com uma camada, enquanto na Figura 2.9, é mostrada a LSTM com suas quatro camadas distintas.

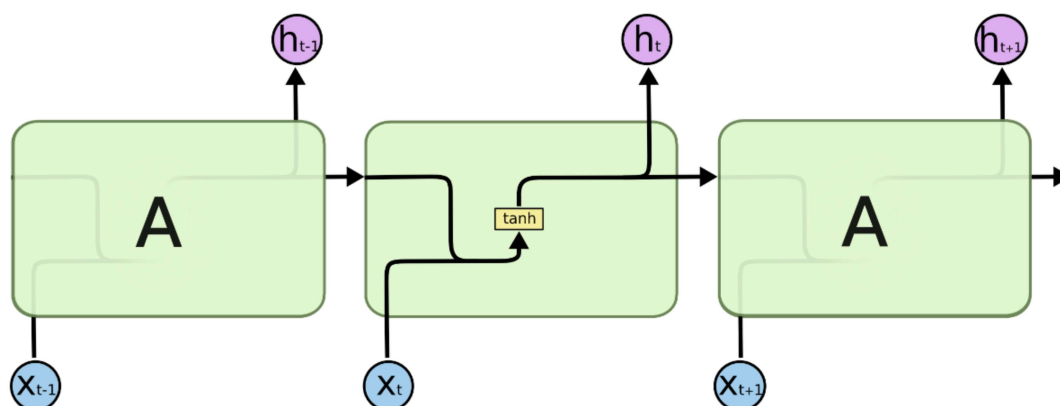


Figura 2.8: RNN com uma camada Fonte: [50]

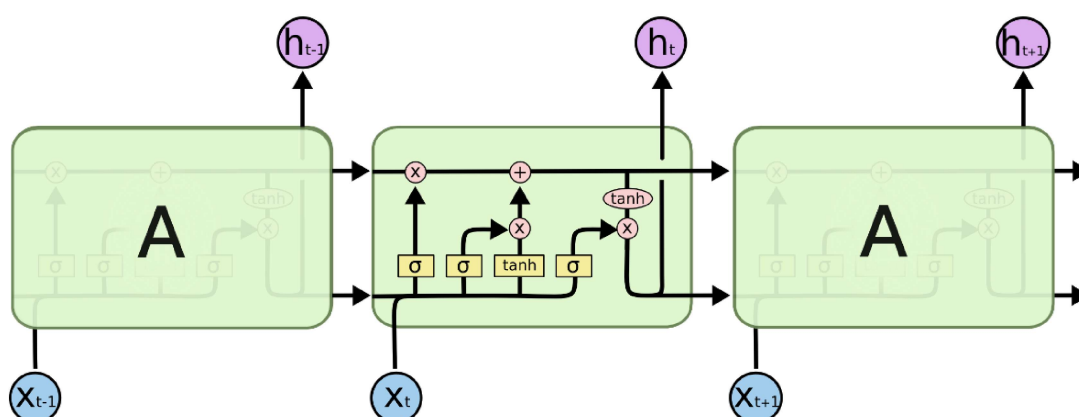


Figura 2.9: LSTM com quatro camadas Fonte: [50]

A arquitetura LSTM compreende uma camada de entrada, uma ou várias camadas ocultas e uma camada de saída. A característica distintiva das LSTMs, que as diferencia das RNNs, reside na camada oculta, especialmente nas células de memória. Cada célula

de memória contém três estruturas de *gates* e uma célula de estado. Os estados da célula são regulados pelos seguintes *gates* [28]:

1. **Forget gate**: Define quais informações serão removidas ou mantidas no estado da célula.
2. **Input gate**: Especifica quais informações serão adicionadas ao estado da célula.
3. **Output gate**: Determina quais informações do estado da célula serão usadas como saída.

Esses *gates* são mecanismos responsáveis por regular o fluxo de informações e são compostos por uma camada sigmoide e uma operação de multiplicação ponto a ponto. A função sigmoide, variando de 0 a 1, indica "não deixar passar nada"(0) ou "deixar passar tudo"(1).

Antes de detalhar os passos, é importante esclarecer a notação utilizada:

1. x_t é o vetor de entrada no passo t .
2. W_f , W_i , W_C são matrizes de pesos.
3. b_f , b_i , b_C são vetores constantes, conhecidos como bias.
4. f_t , i_t , o_t são os valores de ativação para cada *gate*.
5. C_t e \hat{C}_t são vetores para os valores do estado da célula.
6. h_t e h_{t-1} são vetores de saída da LSTM.

O primeiro passo envolve decidir quais informações serão esquecidas do estado da célula, usando a função sigmoide. A Equação (2.14) apresenta o *Forget gate*, que leva em consideração x_t , a saída anterior h_{t-1} , os pesos W_f e o bias b_f . A saída desse *gate* é 1 se o valor do estado anterior C_{t-1} deve ser retido, e 0 se deve ser esquecido completamente.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.14)$$

O próximo passo é decidir quais informações serão atualizadas no estado da célula C_t , em duas etapas. A Equação (2.15) mostra o *input gate*, no qual a função sigmoide

decide quais valores serão atualizados. Em seguida, a Equação (2.16) define a camada *tanh*, gerando um novo candidato \hat{C}_t que será adicionado ao estado.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.15)$$

$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.16)$$

A Figura 2.10(4) ilustra o funcionamento do *output gate* da rede LSTM.

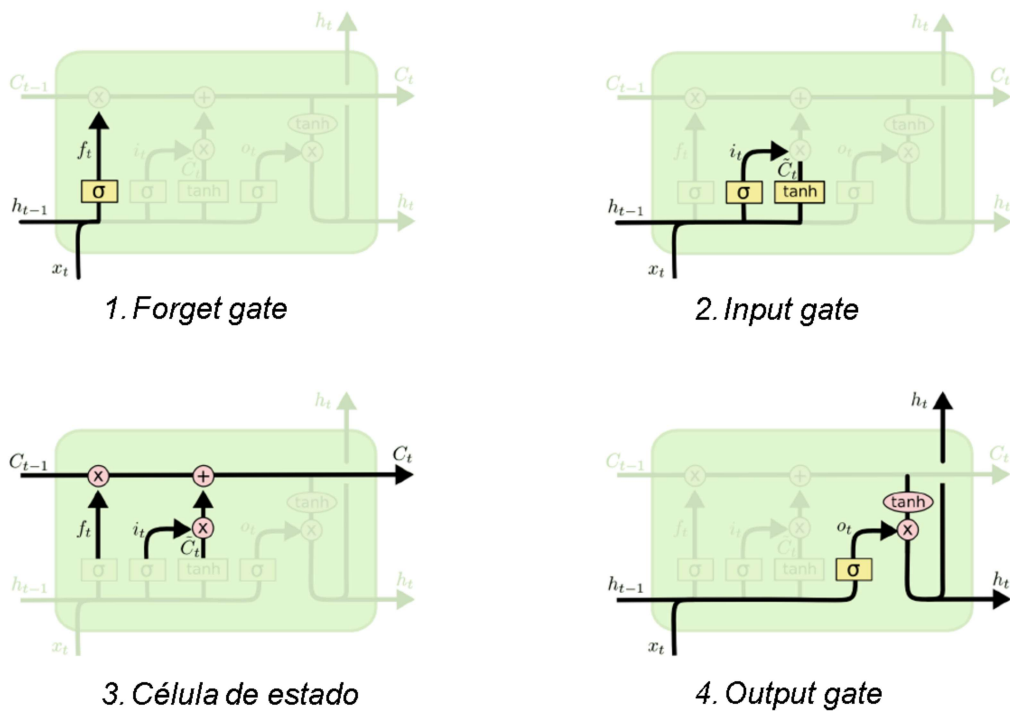


Figura 2.10: Funcionamento de uma LSTM Fonte: [50]

O novo estado C_t é, então, combinado com os valores de i_t . A Equação (2.17) define o novo estado, que é a soma do estado anterior $C_t - 1$ ponderado pelo *forget gate* f_t e do novo candidato de estado da célula $i_t \cdot \hat{C}_t$. As Figuras 2.10(2) e 2.10(3) ilustram esse processo.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \quad (2.17)$$

Finalmente, é necessário obter um valor de saída da rede, realizado pelo *output gate*. A Equação 2.18 define o *output gate*, que resulta da multiplicação do valor da função sigmoide pelo valor do estado da célula C_t , aplicando um filtro $\tanh(C_t)$ no intervalo $[1, 1]$.

O filtro é definido pela Equação 2.19.

$$o_t = \text{sigma}(W_0 \cdot [h_{t-1}, x_t] + b_0) \quad (2.18)$$

$$h_t = o_t * \tanh(C_t) \quad (2.19)$$

2.12 Definições Complementares

Nesta parte, serão abordados conceitos essenciais para avaliar um algoritmo de AM supervisionado. As métricas empregadas incluem aquelas relacionadas à classificação e regressão, que quantificam o desempenho de um preditor. Ademais, serão abordados conceitos essenciais para prevenir que os modelos estejam excessivamente adaptados aos dados, tais como *overfitting* e *underfitting*.

Overfitting manifesta-se quando um classificador procura modelar de maneira excessiva o conjunto de treinamento, demonstrando baixas taxas de erro nesse conjunto, mas elevadas taxas no conjunto de teste [48].

Por outro lado, o **underfitting** ocorre quando um classificador falha em aprender a distribuição dos dados, resultando em altas taxas de erro tanto no conjunto de teste quanto no conjunto de treinamento.

2.12.1 Métricas Preditivas

Para a análise dos resultados alcançados, é essencial avaliar os modelos de previsão de séries temporais. Existem métricas estabelecidas para mensurar a capacidade preditiva, as quais são aplicáveis a problemas de classificação ou tarefas de regressão. De acordo com [25], a avaliação de um algoritmo de aprendizado de máquina supervisionado comumente envolve a análise do desempenho do preditor ao rotular novos objetos não incluídos na base de treinamento. Nesta pesquisa, serão empregadas as seguintes métricas para problemas de regressão:

1. **MAPE (Mean Average Percent Error)**: Calcula a média percentual das diferenças absolutas entre os valores reais e os valores previstos, normalizando essas diferenças pelo valor real. Isso proporciona uma medida de erro em termos percentuais, facilitando a interpretação da precisão do modelo, independentemente

da escala da variável de resposta. Quanto menor o valor do MAPE, melhor é o desempenho do modelo.

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{f}(x_i)|}{|y_i|} \cdot 100$$

2. **MAE (*Mean Average Error*):** Calcula a média das diferenças absolutas entre os valores reais e os valores previstos. Isso fornece uma medida de erro que é fácil de interpretar, pois está na mesma unidade que a variável de resposta. Um MAE menor indica um melhor desempenho do modelo, pois as diferenças entre as previsões e os valores reais são menores.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{f}(x_i)|$$

3. **RMSE (*Root Mean Square Error*):** Calcula a raiz quadrada da média dos quadrados das diferenças entre os valores reais e os valores previstos. Isso penaliza mais os erros maiores do que os erros menores, tornando-se sensível a outliers. Ao contrário do MAE (Erro Médio Absoluto), o RMSE é sensível a outliers porque a elevação ao quadrado amplifica as diferenças maiores. Isso significa que o RMSE pode ser mais influenciado por observações extremas. Portanto, ao escolher entre o MAE e o RMSE, é importante considerar a distribuição dos erros e a sensibilidade a outliers em seu conjunto de dados.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{f}(x_i))^2}$$

Capítulo 3

Metodologia

No presente capítulo, foi discutida a abordagem para o projeto proposto. Utilizando a metodologia "experimental", este estudo conduz desde a criação até a análise dos modelos em um ambiente de simulação, visando testar a hipótese de pesquisa 1.4. Por fim são apresentados e definidos os modelos que serão utilizados neste trabalho.

3.1 Obtenção de Dados

Conforme mencionado previamente, as séries temporais representam observações coletadas de maneira sequenciada ao decorrer do tempo. Para empregar essa abordagem, é preciso ter dados que descrevam adequadamente o problema em questão. Para esta avaliação, foram capturados dados de valores diários do Bitcoin, por um período de 5 anos, compreendendo o período de 01/11/2018 até 31/10/2023. Essas informações foram obtidas através do Yahoo Finanças e englobam dados diários de abertura e encerramento de preços. O processo de coleta foi realizado utilizando-se o software R em conjunto com o pacote *tidyquant*. De acordo com [19], o pacote *tidyquant* no ambiente R representa uma ferramenta para análises quantitativas e financeiras integradas, seguindo os princípios da abordagem *Tidy Data* do *tidyverse*. Este pacote oferece uma interface que combina diversas funcionalidades para a análise de dados financeiros, desde a importação direta de informações de fontes financeiras até a manipulação, visualização e modelagem desses dados. Ao integrar-se com pacotes como *ggplot2*, o *tidyquant* também facilita a criação de visualizações financeiras.

Para uma primeira análise da série, a Figura 3.1 mostra o gráfico desta Série Temporal.

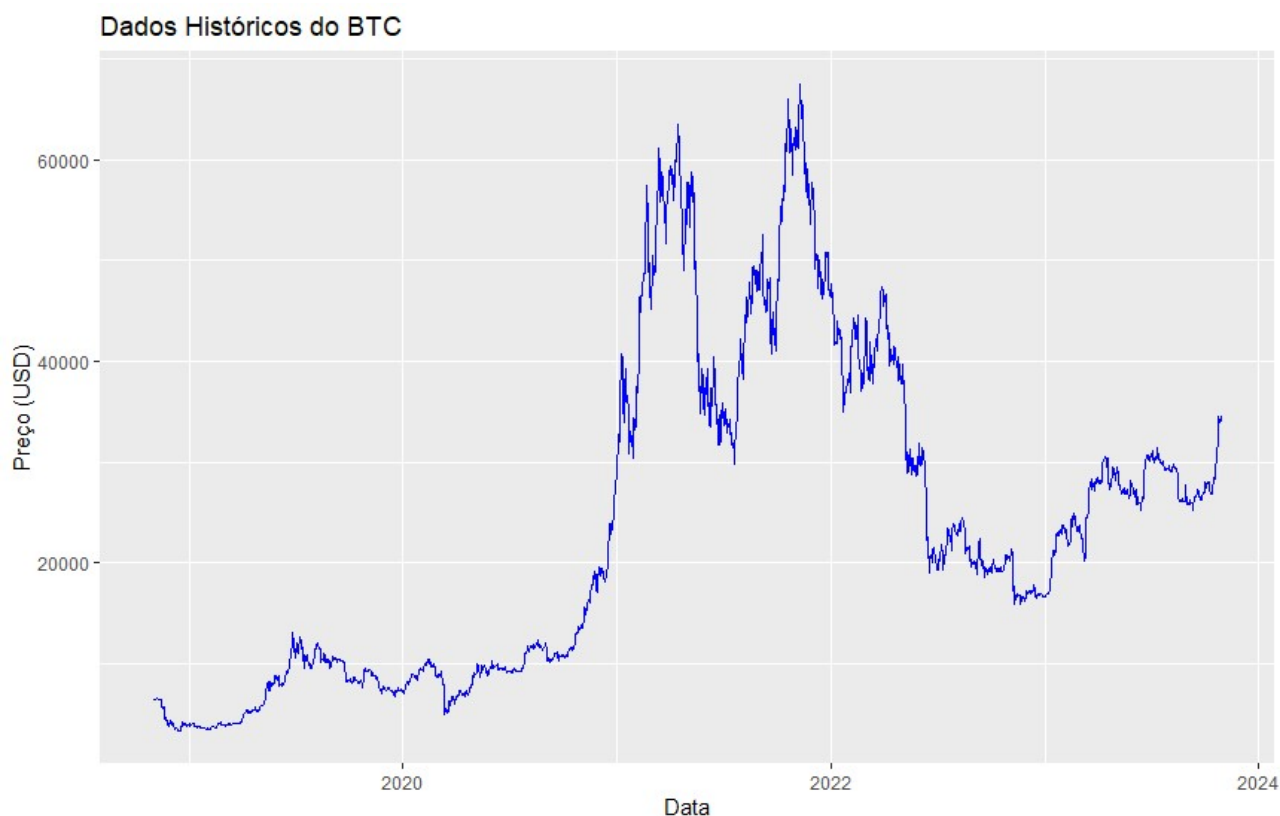


Figura 3.1: Dados Históricos do BitCoin Fonte: Próprio Autor

3.2 Análise da Série Temporal

3.2.1 Decompondo a Série Temporal

A fim de realizar uma primeira análise na Série do BTC, foi utilizada a função *decompose* do R®. Esta função serve para decompor uma série temporal em seus principais componentes, são eles:

1. **Tendência:** Representa a direção geral e o padrão de longo prazo da série temporal. Pode indicar se a série está aumentando, diminuindo ou permanecendo estável ao longo do tempo.
2. **Sazonalidade:** Refere-se a padrões que se repetem em intervalos regulares e previsíveis. Por exemplo, vendas que aumentam todos os anos durante a temporada de férias têm uma componente sazonal anual.
3. **Ruído (ou Resíduos):** Representa a variabilidade aleatória que não pode ser atribuída à tendência ou à sazonalidade. Essencialmente, são as flutuações não explicadas pela tendência ou pela sazonalidade.

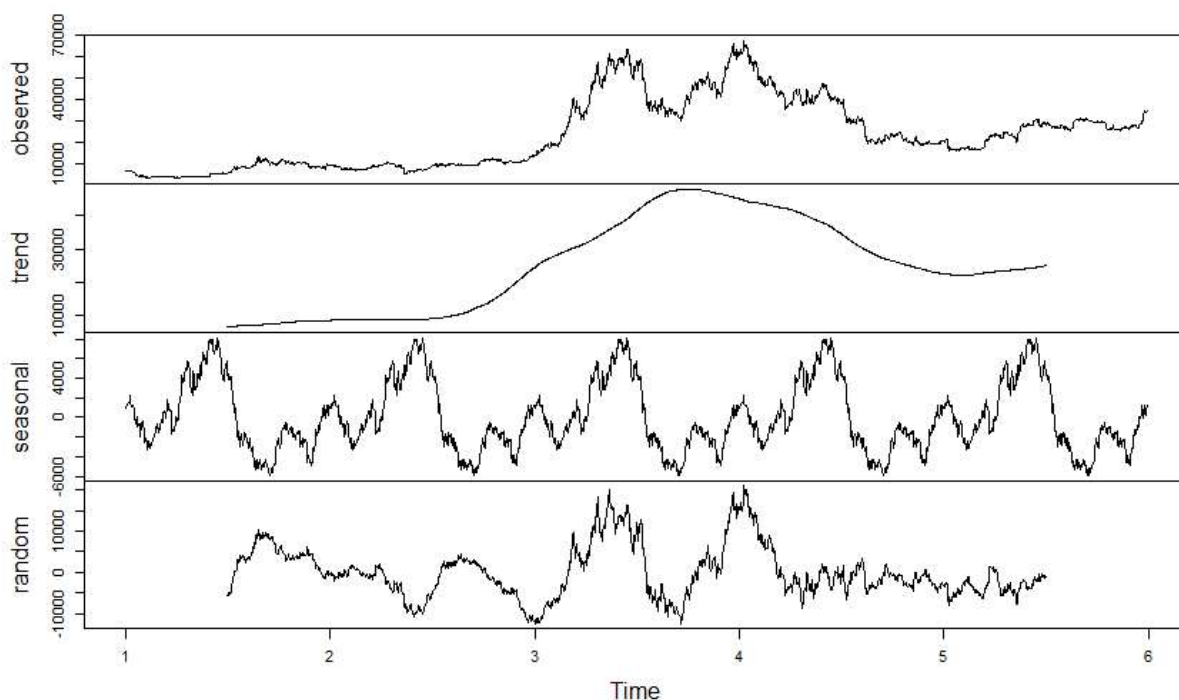


Figura 3.2: Decomposição da Série Temporal Fonte: Próprio Autor

Na Figura 3.2 é apresentado o gráfico resultante da função *decompose*, de onde é possível extrair as seguintes informações:

1. Em relação à **tendência**, a análise revela uma inclinação ascendente no preço do Bitcoin.
2. No que diz respeito à **sazonalidade**, o gráfico revela um padrão anual no comportamento do preço do Bitcoin. Nota-se um aumento expressivo de dezembro a março, seguido por uma queda nos meses subsequentes. É crucial ressaltar que esses movimentos são marcados por uma série de variações, caracterizadas por numerosos picos e quedas. Acredita-se que este aumento nos meses finais e iniciais do ano devem-se a fatores como:
 - No final do ano, muitos investidores fazem ajustes em seus portfólios, o que pode incluir a compra de Bitcoin e outras criptomoedas como parte de estratégias de diversificação e planejamento fiscal.
 - Muitas pessoas recebem bônus ou prêmios de final de ano em seus empregos, o que pode aumentar a quantidade de capital disponível para investimentos, incluindo em criptomoedas.

- Os investidores podem ser influenciados por ciclos de alta anteriores e padrões repetitivos, levando-os a comprar Bitcoin antecipadamente em antecipação de um aumento sazonal.

No entanto a sazonalidade do Bitcoin é um fenômeno complexo e ainda não há uma explicação definitiva para isso, pois o mercado de criptomoedas é altamente volátil e influenciado por muitos fatores imprevisíveis, de modo que padrões históricos podem não se repetir sempre da mesma maneira. As explicações acima são apenas algumas das hipóteses que foram propostas para explicar essa tendência.

3. O *ruído* ou *erro residual*, é relativamente alto, o que significa que os dados são muito variáveis. Isso pode ser causado por vários fatores, como flutuações no mercado, eventos inesperados ou simplesmente por aleatoriedade. Um ruído alto dificulta a previsão de uma série temporal, pois torna mais difícil identificar as tendências e padrões nos dados.

3.2.2 Analisando a Estacionariedade

A série temporal deve ser estacionária, para melhorar sua previsão, ou pode ser transformada para tornar-se estacionária. Uma série estacionária tem propriedades estatísticas, como média e variância, que são constantes ao longo do tempo. Isso significa que a estrutura subjacente da série não muda com o tempo.

Para avaliar a estacionariedade da série, inicialmente realizou-se o teste de Autocorrelação, utilizando a função *acf* do R[®], que gerou o gráfico representado na Figura 3.3.

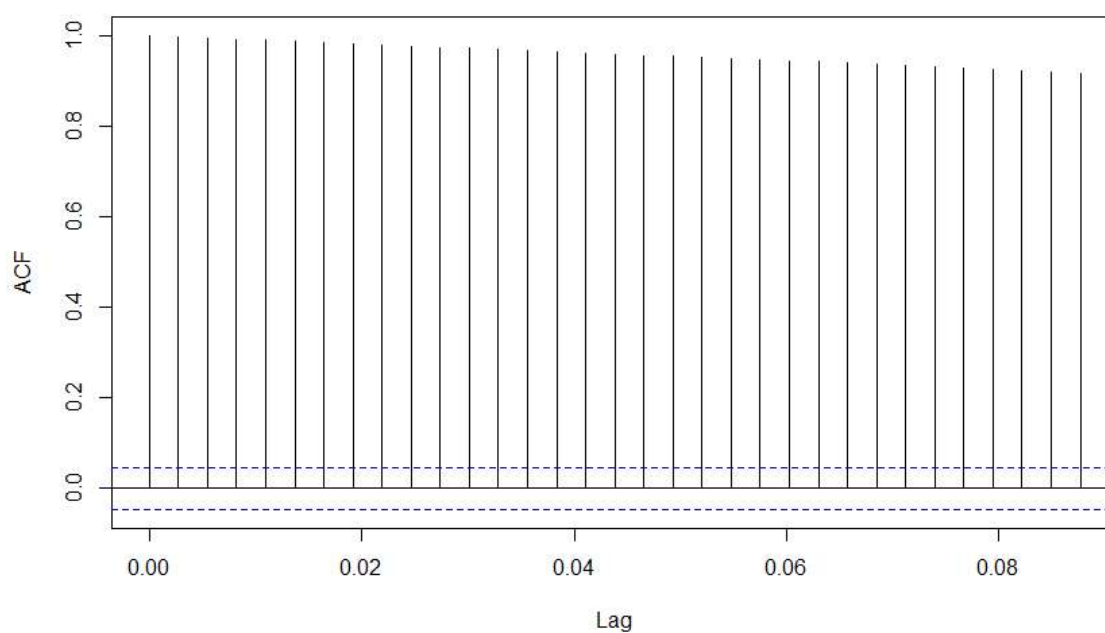


Figura 3.3: Autocorrelação da Série do BitCoin Fonte: Próprio Autor

Na Figura 3.3 é possível verificar que a série apresenta um forte índice de autocorrelação, o que indica que a mesma não é estacionária.

O teste de Dickey-Fuller é um teste estatístico utilizado para verificar se uma série temporal é estacionária ou não. De acordo com [38], o teste de Dickey-Fuller assume que a série temporal tem uma raiz unitária, tornando-a não estacionária. A hipótese alternativa sugere que a série é estacionária. Se o valor-p do teste for inferior ao nível de significância (geralmente 0.05), a série é considerada estacionária. Caso contrário, a série é vista como não estacionária.

Para confirmar os resultados demonstrados no teste de Autocorrelação, utilizou-se a função *adf.test* no ambiente $R^{\text{®}}$, aplicando o Teste de Dickey-Fuller. O valor-p obtido foi de 0.7287, superando o nível de significância de 0.05, confirmando, assim, que a série não é estacionária.

Para garantir a estacionariedade da série, procedeu-se à diferenciação utilizando a função *diff* no ambiente $R^{\text{®}}$. Após essa transformação, o teste de Dickey-Fuller revelou um valor-p de 0.01, confirmando a estacionariedade da série resultante. Conforme pode ser visto na Tabela 3.1.

Foi realizado um teste de Autocorrelação com a série diferenciada, como pode ser visto na Figura 3.4, os resultados confirmaram a estacionariedade da mesma.

Tabela 3.1: Tabela Dickey-Fuller

Teste Dickey-Fuller		
	Antes da diferenciação	Depois da diferenciação
Valor-p	0.7287	0.01

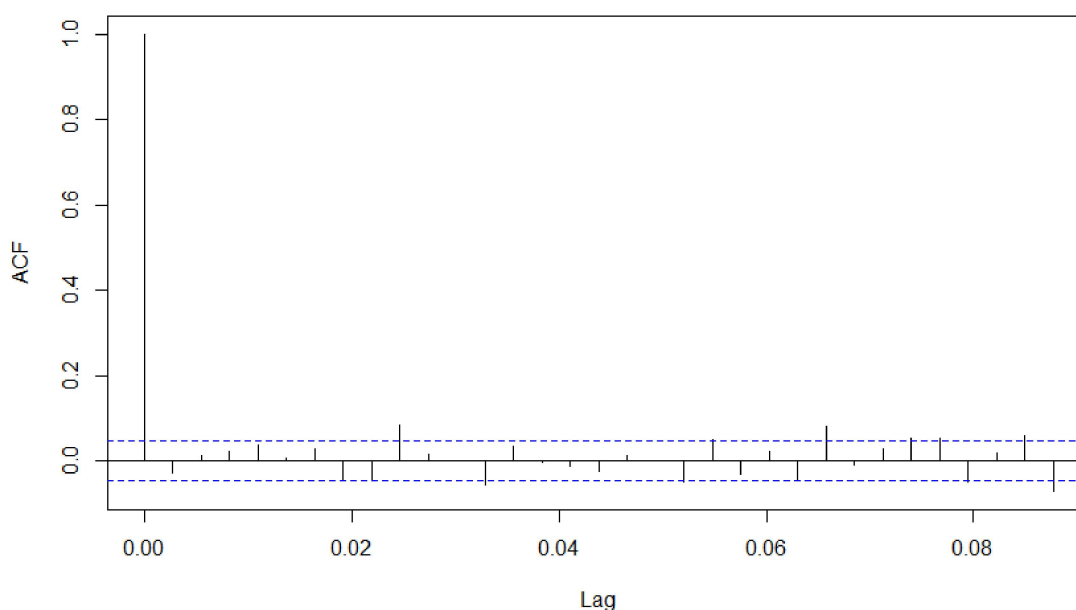


Figura 3.4: Autocorrelação da Série Diferenciada do BitCoin Fonte: Próprio Autor

3.3 Implementação de Modelos ARIMA

Foram realizados testes utilizando o *auto.arima* do R. De acordo com [29], A função ‘*auto.arima()*’ do pacote ‘*forecast*’ no R automatiza o processo de seleção de modelo para séries temporais univariadas usando o método ARIMA (Autoregressive Integrated Moving Average). Ela utiliza algoritmos para explorar uma gama de combinações possíveis de ordens p , d e q para identificar o modelo ARIMA que melhor se ajusta aos seus dados.

Estas são algumas das principais funções realizadas pelo *auto.arima*:

1. Seleção Automática de Modelo: A função tenta identificar automaticamente os melhores parâmetros (ordens p , d e q) para o modelo ARIMA que melhor se ajusta aos dados fornecidos.
2. Diferenciação Automática: Se a série não for estacionária ($d > 0$), a função selecionará o melhor número de diferenças necessárias para tornar a série

estacionária. Ela pode diferenciar a série até um certo limite (por padrão, até duas vezes) e tentará diferentes ordens de diferença.

3. Critérios de Informação: A função usa critérios como o Critério de Informação de Akaike (AIC), o Critério de Informação Bayesiano (BIC) e o Critério de Informação de Akaike corrigido (AICc), para comparar diferentes modelos e selecionar o que tem o melhor ajuste aos dados, levando em consideração a complexidade do modelo.
4. Seleção Sazonal: Se a série exibir sazonalidade, a função também tentará identificar os melhores parâmetros sazonais (P, D, Q) para o modelo SARIMA (ARIMA sazonal).
5. Modelo Final: Após a seleção, a função retorna o melhor modelo ARIMA ou SARIMA identificado para a série.

O melhor modelo selecionado para a série foi o Arima(0,1,0) com um componente de diferenciação. Isso significa que não há termos de autorregressão ($p = 0$) ou termos de média móvel ($q = 0$) no modelo. O termo de diferenciação é $d = 1$, indicando que a série temporal foi diferenciada uma vez para torná-la estacionária. No entanto, conforme visto no item 3.2.1, a série temporal do Bitcoin apresenta sazonalidade anual. Por este motivo, foi gerado o modelo utilizando o Arima(0,1,0), mas também aproximou-se um modelo utilizando o Sarima(0,1,0)(2,2,2)12, como o modelo mais parcimonioso, ou seja, um modelo Arima com componentes sazonais de ordem 2, em seguida, analisadas e comparadas as estatísticas destes modelos através de diversos parâmetros, conforme pode ser visto na tabela 3.2:

Tabela 3.2: Comparação de métricas entre os modelos Arima

Métricas	Arima	Sarima
Q*	735.74	712.55
AIC	30461.99	30285.62
Log likelihood	-15230	-15137.81
RMSE	1023.072	1024.078
MAE	574.7723	593.8914
MAPE	2.366611	2.499565
MASE	0.9994581	1.032704
ACF1	-0.02859788	-0.02189198

Podem ser feitas as seguintes considerações sobre estes resultados:

1. Q* (Teste Ljung-Box): Este teste é usado para verificar a autocorrelação nos resíduos do modelo. Um valor mais baixo é geralmente melhor, indicando menos

autocorrelação nos resíduos. Neste caso, o modelo Sarima tem um valor ligeiramente menor.

2. AIC: É uma medida de qualidade do modelo que penaliza modelos com muitos parâmetros. Um valor menor de AIC indica um modelo melhor em termos de ajuste aos dados. O modelo Sarima tem um AIC mais baixo, o que indica um melhor ajuste.
3. Log likelihood: O log-likelihood mede quão bem um modelo estatístico explica os dados observados. Especificamente, é a probabilidade logarítmica dos dados observados sob o modelo estimado. Quanto maior o log-likelihood, melhor é o ajuste do modelo aos dados. Neste caso o modelo com componentes sazonais apresentou um resultado ligeiramente melhor.
4. RMSE (Erro Quadrático Médio): Este é um indicador da precisão das previsões. Um RMSE mais baixo indica melhor ajuste às observações. Ao contrário dos resultados anteriores, o modelo Arima apresentou um resultado um pouco melhor.
5. MAE (Erro Absoluto Médio): Similar ao RMSE, porém menos sensível a outliers.
6. MAPE (Erro Percentual Absoluto Médio): Este é um indicador de precisão em termos percentuais. Um MAPE mais baixo indica melhores previsões.
7. MASE: Este é o erro de escala média, que compara o erro do modelo com o erro de um modelo de previsão ingênua (por exemplo, uma previsão constante). Um valor próximo de 1 indica que o modelo é tão bom quanto o modelo ingênuo.
8. ACF1: Este é o valor do primeiro coeficiente de autocorrelação dos resíduos. Um valor próximo de zero sugere que os resíduos são não autocorrelacionados, o que é desejável.

Ao considerar todas essas métricas, o modelo Sarima apresenta um desempenho superior ao Arima.

3.4 Implementação do modelo RNN

Foi desenvolvido um protocolo para conduzir as etapas da criação do código em R, para criação do modelo RNN, com as seguintes etapas:

1. Levantamento de Requisitos: Identificar as bibliotecas necessárias e estabelecer os requisitos do ambiente Python
2. Instalação de Bibliotecas: Verificar e instalar as bibliotecas necessárias usando condições (como `'if (!requireNamespace(...))'`)
3. Configuração do Ambiente Python: Configurar o ambiente Python usando `'use_python'`.
4. Obtenção de Dados: Utilizar a biblioteca `'tidyquant'` para obter dados históricos do preço do Bitcoin.
5. Pré-processamento dos Dados: Realizar operações como conversão de tipos, ordenação e remoção de valores ausentes.
6. Visualização dos Dados: Plotar a série temporal dos preços do Bitcoin para visualização.
7. Normalização dos Dados: Criar uma função para normalizar os dados.
8. Criação de Sequências para Entrada e Saída: Implementar uma função para criar sequências de dados a serem usadas no treinamento do modelo.
9. Divisão dos Dados: Dividir os dados normalizados em conjuntos de treinamento e teste.
10. Definição do Modelo RNN: Utilizar a biblioteca `'keras'` para definir um modelo RNN sequencial.
11. Compilação do Modelo: Compilar o modelo definindo a função de perda, otimizador e métricas.
12. Treinamento do Modelo: Treinar o modelo usando os dados de treinamento.
13. Avaliação do Modelo: Avaliar o desempenho do modelo nos dados de teste usando métricas específicas.
14. Desnormalização e Exportação de Resultados: Desnormalizar as previsões e criar um dataframe para exportar os resultados.
15. Validação e Verificação: Verificar se os resultados são consistentes com as expectativas.

16. **Conclusões e Análises Adicionais:** Analisar os resultados, identificar limitações e formular conclusões.

O código foi executado em um ambiente com as seguintes características:

- Sistema Operacional: Windows 10 Pro 2H22
- Processador: Intel(R) Core(TM) i7-4790k CPU @ 4.00GHz
- Memória RAM: 16 Gb
- Tempo de execução médio do código: 23.72 segundos

3.4.1 Código RNN

Foi desenvolvido um código em R, não paralelizado, que vai desde a obtenção de dados históricos do Bitcoin até o treinamento de um modelo de aprendizado de máquina (RNN) para prever os valores futuros do Bitcoin, com os seguintes passos:

1. **Instalação de Bibliotecas:** O código verifica se as bibliotecas *keras*, *tidyquant*, *tidyr*, *tensorflow*, *ggplot2*, *writexl* estão instaladas no ambiente R. Se não estiverem, ele as instala. Isso é feito para garantir que todas as dependências necessárias estejam disponíveis para execução.
2. **Carregamento de Bibliotecas:** Uma vez que todas as bibliotecas estão instaladas, o código as carrega. Bibliotecas são extensões de código pré-escritas que oferecem funcionalidades adicionais. Por exemplo, *keras* e *tensorflow* são bibliotecas populares para aprendizado profundo ou *deep learning*.
3. **Configuração do Python:** O desenvolvimento do código foi inicialmente realizado em R. No entanto, durante o processo, surgiu a necessidade de utilizar pacotes específicos de *machine learning*, o que levou à adoção das bibliotecas *keras* e *tensorflow*, que são nativamente escritas em Python. Consequentemente, é necessário configurar o caminho para a versão específica do Python que será utilizada pelo R. Essa configuração é crucial para que o R possa localizar e utilizar o interpretador Python corretamente.
4. **Obtenção de Dados** O código usa a função *tq_get* do pacote *tidyquant* para buscar os dados históricos do preço do Bitcoin direto de uma API da Yahoo Finance. Esses dados vão de 1^o de novembro de 2018 até 31 de outubro de 2023.

5. **Visualização dos Dados** A biblioteca *ggplot2* é utilizada para criar um gráfico (ou série temporal) dos preços de fechamento do Bitcoin ao longo do tempo. Isso permite visualizar como o preço do Bitcoin flutuou durante o período de dados.
6. **Preprocessamento dos Dados** Antes de treinar um modelo de aprendizado de máquina, os dados geralmente precisam ser pré-processados:
 - A coluna de data é convertida para o formato de data padrão do R.
 - Os dados são ordenados cronologicamente.
 - Quaisquer linhas com dados ausentes são removidas.
 - Apenas a coluna de preço de fechamento é mantida para análise.
 - Os dados são normalizados usando a técnica Min-Max Scaling, o que significa que os valores são ajustados para um intervalo entre 0 e 1
7. **Criação de Sequências para Modelagem:** Para modelar séries temporais com redes neurais recorrentes, os dados são frequentemente divididos em sequências. Aqui, a função *create_sequences* é definida para criar essas sequências, que são basicamente subconjuntos contíguos dos dados que servirão como entradas e saídas para o modelo.
8. **Modelagem com RNN:** O modelo de rede neural recorrente (RNN) é definido usando a função *keras_model_sequential()*. Este modelo contém uma camada RNN e uma camada densa (ou totalmente conectada). O modelo é compilado, especificando a função de perda (MSE, que mede a diferença entre as previsões do modelo e os valores reais), o otimizador (Adam) e as métricas (MAE, que é a média dos erros absolutos).
9. **Treinamento e Avaliação do Modelo:** O modelo é treinado usando os dados de treinamento. Ele faz isso por 50 épocas, ajustando gradualmente os pesos para minimizar a função de perda. Após o treinamento, o modelo é avaliado através das métricas RMSE, MAE e MAPE, usando os dados de teste para ver quão bem ele se saiu em dados não vistos anteriormente.
10. **Desnormalização das Previsões:** Como os dados de preço foram normalizados anteriormente, as previsões do modelo também estão no formato normalizado. A função *desnormalizar* é definida para trazer essas previsões de volta ao formato original, não normalizado.

11. **Exportação dos Resultados:** Finalmente, os resultados desnormalizados (previsões) são exportados para um arquivo de planilha eletrônica onde são comparados com os resultados originais.

3.5 Implementação do Modelo LSTM

Foi utilizado o mesmo protocolo que serviu como base para a criação do modelo RNN (3.4), com adaptações para um modelo LSTM:

1. Levantamento de Requisitos: Identificar as bibliotecas necessárias e estabelecer os requisitos do ambiente Python
2. Instalação de Bibliotecas: Verificar e instalar as bibliotecas necessárias usando condições (como `'if (!requireNamespace(...))'`)
3. Configuração do Ambiente Python: Configurar o ambiente Python usando `'use_python'`.
4. Obtenção de Dados: Utilizar a biblioteca `'tidyquant'` para obter dados históricos do preço do Bitcoin.
5. Pré-processamento dos Dados: Realizar operações como conversão de tipos, ordenação e remoção de valores ausentes.
6. Visualização dos Dados: Plotar a série temporal dos preços do Bitcoin para visualização.
7. Normalização dos Dados: Criar uma função para normalizar os dados.
8. Criação de Sequências para Entrada e Saída: Implementar uma função para criar sequências de dados a serem usadas no treinamento do modelo.
9. Divisão dos Dados: Dividir os dados normalizados em conjuntos de treinamento e teste.
10. Definição do Modelo LSTM: Utilizar a biblioteca `'keras'` para definir um modelo LSTM sequencial.
11. Compilação do Modelo: Compilar o modelo definindo a função de perda, otimizador e métricas.

12. Treinamento do Modelo: Treinar o modelo usando os dados de treinamento.
13. Avaliação do Modelo: Avaliar o desempenho do modelo nos dados de teste usando métricas específicas.
14. Desnormalização e Exportação de Resultados: Desnormalizar as previsões e criar um dataframe para exportar os resultados.
15. Validação e Verificação: Verificar se os resultados são consistentes com as expectativas.
16. Exportação do Modelo: Salvar o modelo treinado para uso futuro.
17. Conclusões e Análises Adicionais: Analisar os resultados, identificar limitações e formular conclusões.

O código foi executado em um ambiente com as seguintes características:

- Sistema Operacional: Windows 10 Pro 2H22
- Processador: Intel(R) Core(TM) i7-4790k CPU @ 4.00GHz
- Memória RAM: 16 Gb
- Tempo de execução médio do código: 27.51 segundos

3.5.1 Código LSTM

O código em R, desenvolvido e implementado, destinado à criação, treinamento e avaliação do modelo LSTM, não paralelizado, executa as seguintes tarefas:

1. Instalação e carregamento de pacotes

- Instala os pacotes necessários.
- Carrega as bibliotecas 'keras', 'ggplot2', 'tensorflow', e 'reticulate'.
- Configura o ambiente Python usando 'reticulate'.

2. Preparação dos dados para treinamento

- A coluna de data é convertida para o formato de data padrão do R.

- Os dados são ordenados cronologicamente.
- Quaisquer linhas com dados ausentes são removidas.
- Apenas a coluna de preço de fechamento é mantida para análise.
- Os dados são normalizados usando a técnica Min-Max Scaling, o que significa que os valores são ajustados para um intervalo entre 0 e 1
- Define um tamanho de janela (`window_size`) e cria uma matriz de entrada (`input_data`) e saída (`output_data`) com base nas janelas de dados.
- Divide os dados em conjuntos de treinamento e teste.

3. Construção do modelo LSTM

- Inicializa um modelo sequencial.
- Adiciona uma camada LSTM com 100 unidades, uma camada de dropout para regularização e uma camada densa de saída.

4. Compilação e treinamento do modelo

- Compila o modelo usando o erro quadrático médio (MSE) como função de perda e o otimizador Adam.
- Treina o modelo usando os dados de treinamento.

5. Avaliação do modelo

- Avalia o modelo usando os dados de teste e calcula o erro quadrático médio (MSE).

6. Fazendo previsões

- Faz previsões, dando indicação de momentos para compra e venda, usando o modelo treinado com os dados de teste.

7. Avaliação do desempenho das previsões

- Calcula algumas métricas de erro, são elas: RMSE, MAE e MAPE.
- Plota as previsões contra os valores reais.

Capítulo 4

Resultados e Discussão

A seguir, serão apresentados os resultados preliminares obtidos na análise de dados relacionados ao Bitcoin. Conforme abordado na seção 1.1.4, o Bitcoin foi concebido teoricamente em 2008 e, na prática, começou a circular em 2009. Contudo, a biblioteca *tidyquant* disponibiliza os dados diários do Bitcoin a partir de 17/09/2014, totalizando 3332 observações até 31/10/2023, configurando uma série temporal robusta.

Inicialmente, foram realizados testes utilizando dados abarcando os últimos 9 anos. Contudo, devido às significativas flutuações nos preços do Bitcoin, essas análises não produziram resultados satisfatórios. Visando trabalhar com conjuntos de dados mais atualizados, foram conduzidos testes sucessivos considerando intervalos temporais menores: os últimos 9, 8, 7, 6 e 5 anos. Entre esses testes, aqueles realizados com os dados dos últimos 5 anos demonstraram os menores erros conforme as métricas estabelecidas. Portanto, decidiu-se trabalhar com os dados correspondentes aos últimos 5 anos, abrangendo o período de 01/11/2018 a 31/10/2023.

4.1 Resultados ARIMA

4.1.1 ARIMA

Como destacado na seção 3.3, a função `auto.arima` identificou o ARIMA (0,1,0) como o modelo mais apropriado. No entanto, as projeções geradas por esse modelo revelaram-se estáticas, assemelhando-se a um modelo ingênuo de última observação. O que já era esperado, pois o modelo considerou apenas uma diferenciação, não havendo nenhuma defasagem nos componentes auto-regressivo e média móvel. Essa característica é evidenciada na Figura 4.1, onde se percebe a ausência de dinâmica nas previsões, limitando

a capacidade do modelo em capturar padrões mais complexos ou tendências no conjunto de dados.

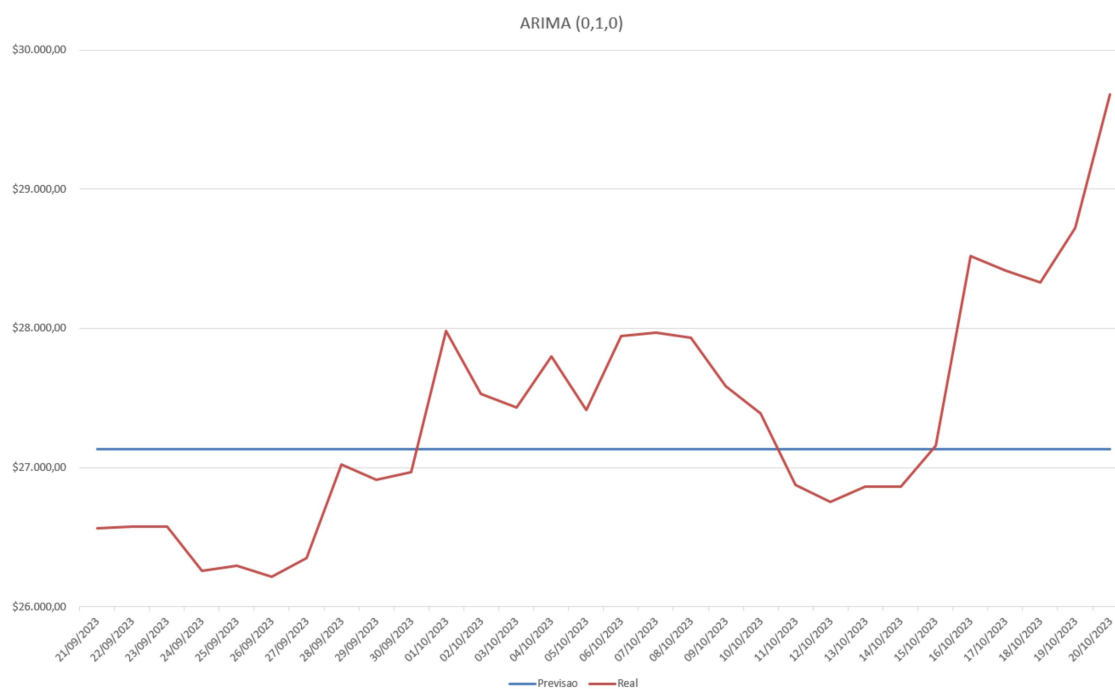


Figura 4.1: Comparação entre as previsões do modelo ARIMA e os valores reais Fonte: Próprio Autor

4.1.2 SARIMA

Conforme evidenciado na seção 3.2.1, a análise da série temporal revelou a presença de sazonalidade, motivando a exploração do modelo SARIMA. Foram realizado oito testes, conduzidos com diferentes configurações para os componentes sazonais, culminando na seleção do modelo SARIMA(0,1,0)(2,2,2)12 como o mais adequado. A Tabela 3.2 documenta que o desempenho do modelo SARIMA superou o modelo ARIMA, enquanto a Figura 4.2 proporciona uma visualização mais tangível desse sucesso. Destaca-se que as projeções do SARIMA exibiram proximidade aos valores reais, evidenciada por uma diferença mínima de 0,2% entre a previsão e o valor real em 13 de Outubro. Entretanto, é importante observar que ocorreu uma divergência mais significativa, atingindo 10% em 20 de Outubro, o que demonstra imprecisões no modelo.

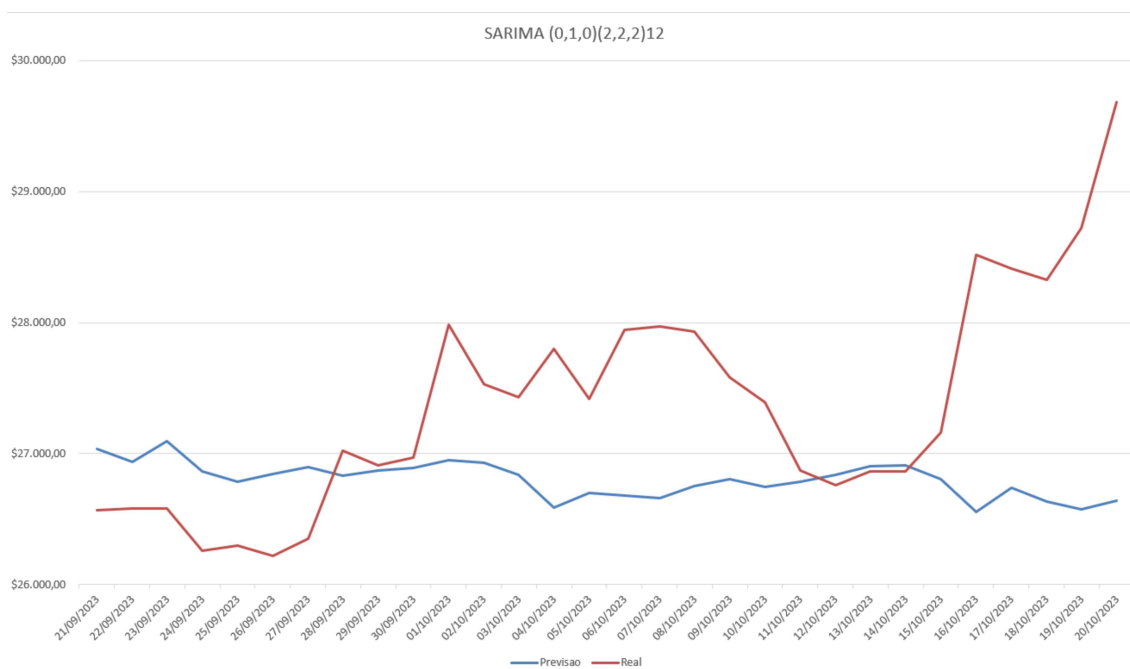


Figura 4.2: Comparação entre as previsões do modelo SARIMA e os valores reais
Fonte: Próprio Autor

4.2 Resultados RNN

4.2.1 RNN

Após a execução do modelo de previsão, conforme detalhado no código fornecido na Seção 3.4, foram obtidos os seguintes resultados:

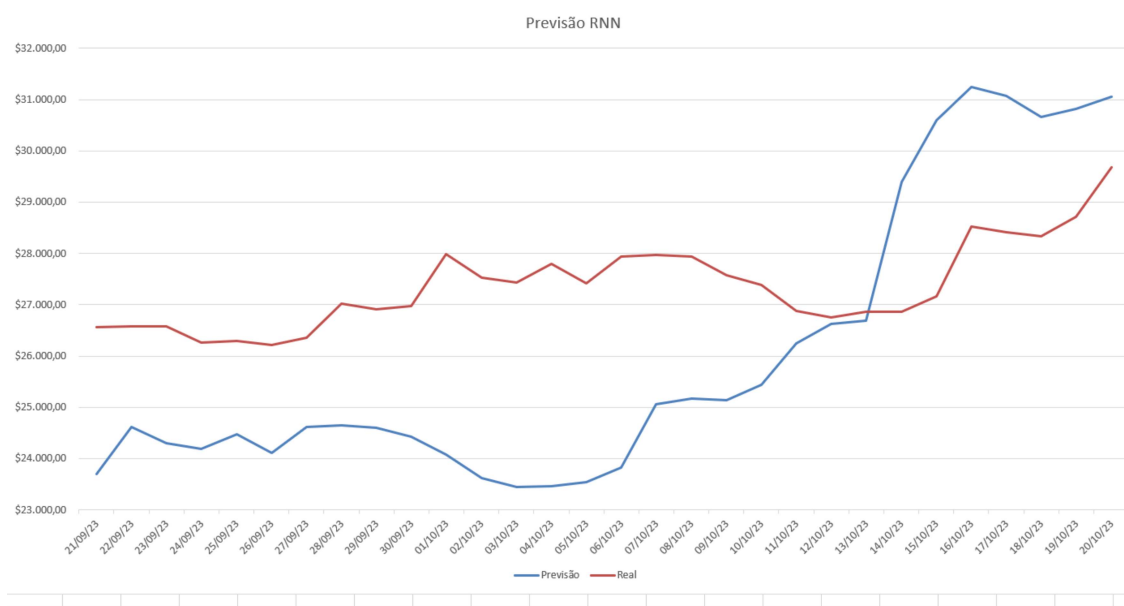


Figura 4.3: Comparação entre os valores reais e a previsão do modelo RNN Fonte: Próprio Autor

Conforme evidenciado na Figura 4.3, a previsão gerada por este modelo demonstrou uma proximidade em relação aos valores reais, alcançando uma diferença inferior a 0,5% no dia 12 de Outubro. Entretanto, é importante observar que, apesar de resultados mais assertivos, ocorreram períodos de discrepância, destacando-se uma diferença aproximada de 20% no dia 04 de Outubro, sinalizando que há imprecisões no modelo.

4.2.2 RNN+ARIMA

Como destacado na seção 3.3, o *auto.arima* identificou o ARIMA(0,1,0) como o modelo mais adequado. Essa configuração implica a ausência de termos de autorregressão ($p = 0$) e termos de média móvel ($q = 0$) no modelo. O termo de diferenciação é $d = 1$, indicando que a série temporal foi diferenciada uma vez para torná-la estacionária. Com base nesse modelo, a decisão foi tomada de avaliar o desempenho do modelo RNN utilizando a série temporal gerada pelo modelo ARIMA(0,1,0).

Conforme evidenciado na Figura 4.4, a previsão gerada por este modelo demonstrou uma proximidade significativa em relação aos valores reais, com resultados melhores que o RNN. A diferença entre as previsões e os valores reais foi inferior a 0,1% no dia 12 de Outubro, atingindo sua diferença máxima, de aproximadamente 6%, no dia 14 de Outubro.

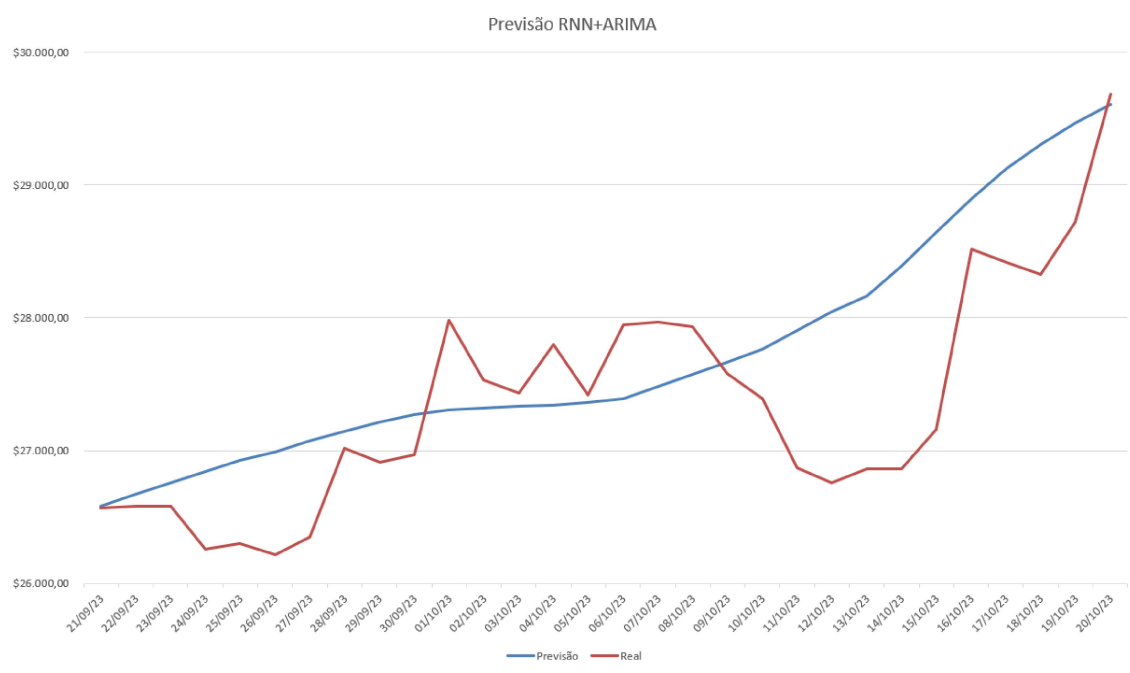


Figura 4.4: Comparação entre os valores reais e a previsão do modelo RNN+ARIMA
Fonte: Próprio Autor

4.3 Resultados LSTM

4.3.1 LSTM

Após a implementação do modelo de previsão, conforme descrito no código fornecido na seção 3.5.1, os resultados a seguir foram obtidos:

Conforme ilustrado no gráfico 4.5, observa-se que a previsão do modelo LSTM demonstrou precisão em alguns períodos, alcançando uma diferença inferior a 2% no dia 08 de outubro. Entretanto, assim como no modelo RNN, em determinados momentos, foram identificados períodos de imprecisão, com divergências superiores a 20%, como no dia 24 de setembro, demonstrando que existem discrepâncias no modelo.

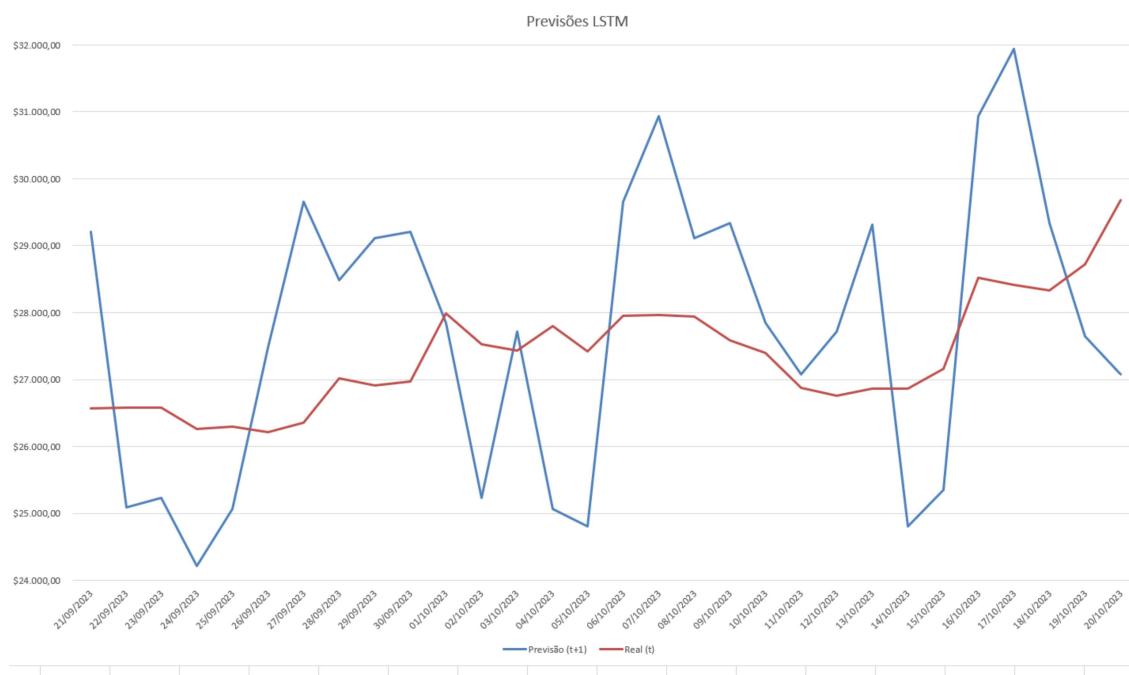


Figura 4.5: Comparação entre as previsões do modelo LSTM e os valores reais Fonte: Próprio Autor

4.3.2 LSTM+ARIMA

Assim como anteriormente realizado com o modelo RNN+ARIMA, aplicou-se o modelo ARIMA(0,1,0) à série temporal do Bitcoin. Posteriormente, o modelo LSTM foi executado com a série temporal resultante, e os resultados dessa abordagem são apresentados na Figura 4.6.

Conforme ilustrado na Figura 4.6, a previsão gerada por este modelo demonstrou uma maior proximidade com os valores reais. Registrando uma diferença mínima inferior a 0,3% em 18 de Outubro. A maior discrepância observada foi de aproximadamente 5% no dia 14 de Outubro. O que demonstra uma melhoria na capacidade preditiva do LSTM.

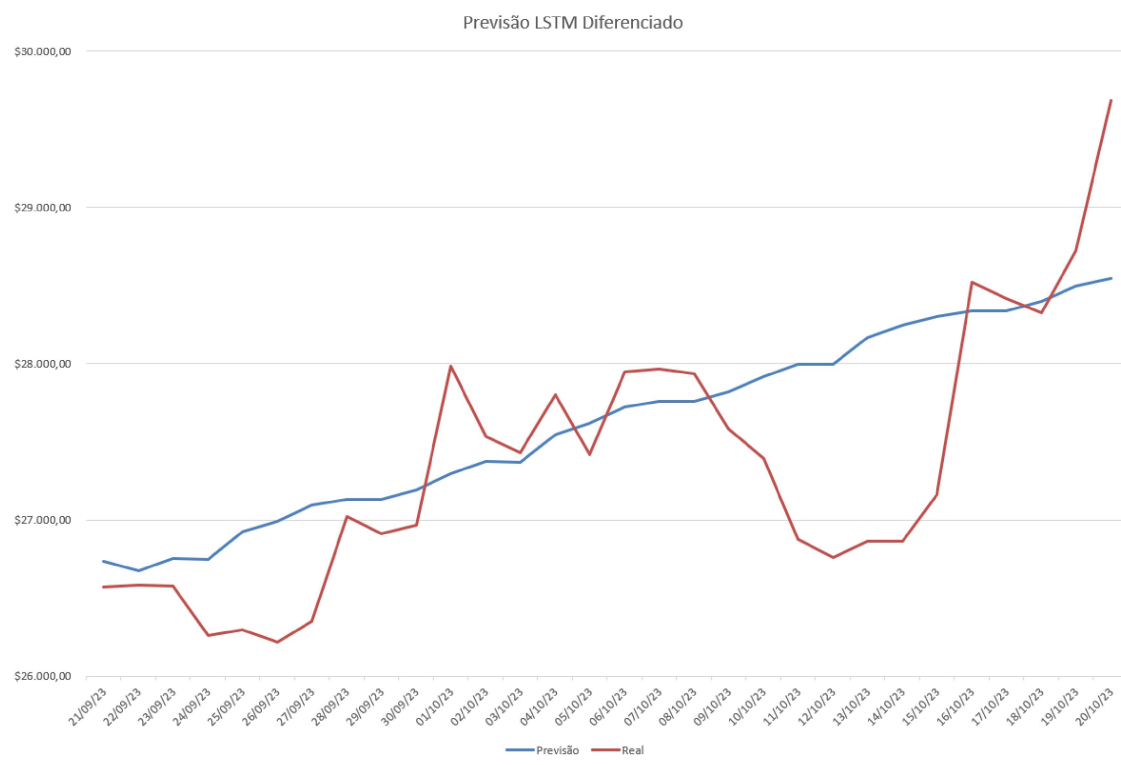


Figura 4.6: Comparação entre as previsões do modelo LSTM+ARIMA e os valores reais
 Fonte: Próprio Autor

4.4 Comparações entre os modelos

As métricas de desempenho, RMSE, MAE e MAPE, foram computadas para todos os modelos implementados. Diante destas métricas foi possível constatar que o modelo com melhor desempenho foi o RNN+ARIMA. Os resultados detalhados estão disponíveis na Tabela 4.1.

Tabela 4.1: Comparação de Métricas entre Modelos

Métricas	ARIMA	SARIMA	LSTM	LSTM+ARIMA	RNN	RNN+ARIMA
RMSE	1023.072	1024.078	0.054	0.037	0.046	0.011
MAE	574.772	593.891	0.042	0.024	0.027	0.007
MAPE	2.367%	2.500%	9.725%	4.782%	4.796%	1.857%

RMSE

O RMSE, uma métrica que expressa a média das distâncias entre os valores previstos e os valores reais, é um indicador de desempenho de modelos preditivos. Quanto menor o valor do RMSE, melhor é a precisão do modelo.

Analisando os resultados, o modelo híbrido RNN+ARIMA destacou-se como o

mais eficaz, apresentando uma melhora de 76,09% em relação ao ao modelo RNN. Seguido de perto pelos modelos LSTM+ARIMA e RNN que apresentaram desempenhos bastante similares. O LSTM registrou o desempenho mais modesto entre os modelos de Aprendizado de Máquina avaliados. É possível observar também que o modelo híbrido LSTM+ARIMA apresentou um resultado 31,48% melhor em relação ao modelo LSTM. Conforme esperado, os modelos ARIMA e SARIMA demonstraram os resultados menos favoráveis em comparação aos outros modelos testados.

MAE

O MAE, uma métrica adicional que calcula a distância média entre os valores previstos e os valores reais, é mais um indicador de desempenho de modelos preditivos. Quanto menor o valor do MAE, mais preciso é o modelo.

Analisando os resultados em termos de MAE, é possível observar um padrão semelhante ao RMSE. Novamente, o RNN+ARIMA se destacou como o modelo de melhor desempenho, apresentando uma melhora de 74,07% em relação ao modelo RNN, seguido pelos modelos RNN e LSTM+ARIMA, sendo o LSTM o menos eficaz entre os modelos baseados em Aprendizado de Máquina. Os modelos ARIMA e SARIMA mantiveram-se como os menos favoráveis em termos de desempenho.

MAPE

O MAPE, uma medida que avalia a proporção média dos erros em relação aos valores reais, oferece insights sobre o desempenho do modelo. Quanto menor o valor do MAPE, maior é a precisão do modelo.

Analisando os resultados em termos de MAPE, foram observadas diferenças notáveis em comparação com os resultados do RMSE e MAE. O modelo RNN+ARIMA manteve-se como o líder em desempenho, seguido pelos modelos ARIMA e SARIMA em segundo e terceiro lugares, respectivamente. Em seguida, verifica-se os modelos LSTM+ARIMA e RNN+ARIMA com resultados bastante similares, enquanto o modelo LSTM ficou em último lugar em termos de desempenho. Essa análise abrangente destaca a importância de avaliar métricas variadas para uma compreensão holística do desempenho do modelo.

4.5 Simulações de Investimentos

Diante da constatação de que o modelo RNN+ARIMA apresentou os melhores resultados, o próximo passo consistiu na realização de simulações de operações de compra

e venda de Bitcoin, por um período arbitrário de 30 dias, com base nas previsões do mesmo.

As simulações seguiram os passos abaixo:

1. Inicialmente, considerou-se um investimento de US\$ 1.000,00.
2. No primeiro dia, avalia-se se a previsão para o fechamento do Bitcoin no dia seguinte ($t + 1$) era maior que o fechamento do dia atual (t). Se verdadeiro, indicando uma valorização prevista, realiza-se a compra do Bitcoin com o valor total investido; caso contrário, nenhuma ação é tomada.
3. No dia seguinte, caso tenham sido comprados Bitcoins no dia anterior, procede-se com a venda dos Bitcoins comprados, pelo valor de fechamento do mercado, independentemente de ter atingido ou não o valor previsto.
4. Imediatamente após a conclusão do passo 3, repete-se o passo 2 e assim sucessivamente. Este processo se repete continuamente ao longo de um mês.
5. Ao final do mês, calcula-se a diferença entre o valor em carteira e o valor inicialmente investido, determinando lucro ou prejuízo. Em seguida, é calculado o percentual de lucratividade ou perda.

Uma exemplificação desse processo pode ser apresentada da seguinte forma:

No dia 21 de Outubro, o preço de fechamento foi US\$ 26.000,00, e a previsão para o fechamento do dia 22 de Outubro era US\$ 26.500,00, indicando uma valorização. Nesse caso, efetuava-se a compra de US\$ 1.000,00 em Bitcoin, equivalente a aproximadamente 0,03846 Bitcoins. No dia 22 de Outubro, o valor de fechamento do Bitcoin foi US\$ 26.650,00, e os Bitcoins foram vendidos por esse valor, resultando em US\$ 1.025. Nesse momento, ao verificar a previsão para o dia 23 de Outubro (US\$ 26.700,00), também indicando uma valorização, utilizavam-se os US\$ 1.025 para adquirir novamente os 0,03846 Bitcoins (na prática, essa compra e venda não seria necessária). No dia 23 de Outubro, o valor de fechamento do Bitcoin foi US\$ 26.200,00, e os Bitcoins foram vendidos por esse valor, convertendo-se em US\$ 1007,69. Observando que a previsão para o dia 23 de Outubro era de US\$ 26.100,00, indicando uma desvalorização, nenhuma ação era realizada, mantendo-se os US\$ 1007,69 na conta.

A fim de comparar a eficácia do modelo e da estratégia de investimento adotada nas simulações, procedeu-se à comparação com um modelo ingênuo. Nesse modelo, realiza-se

a compra de US\$ 1.000,00 em Bitcoin no primeiro dia das simulações e, após 30 dias, efetua-se a venda pelo valor de mercado. Não são realizadas verificações diárias, apenas uma operação de compra e outra de venda após 30 dias.

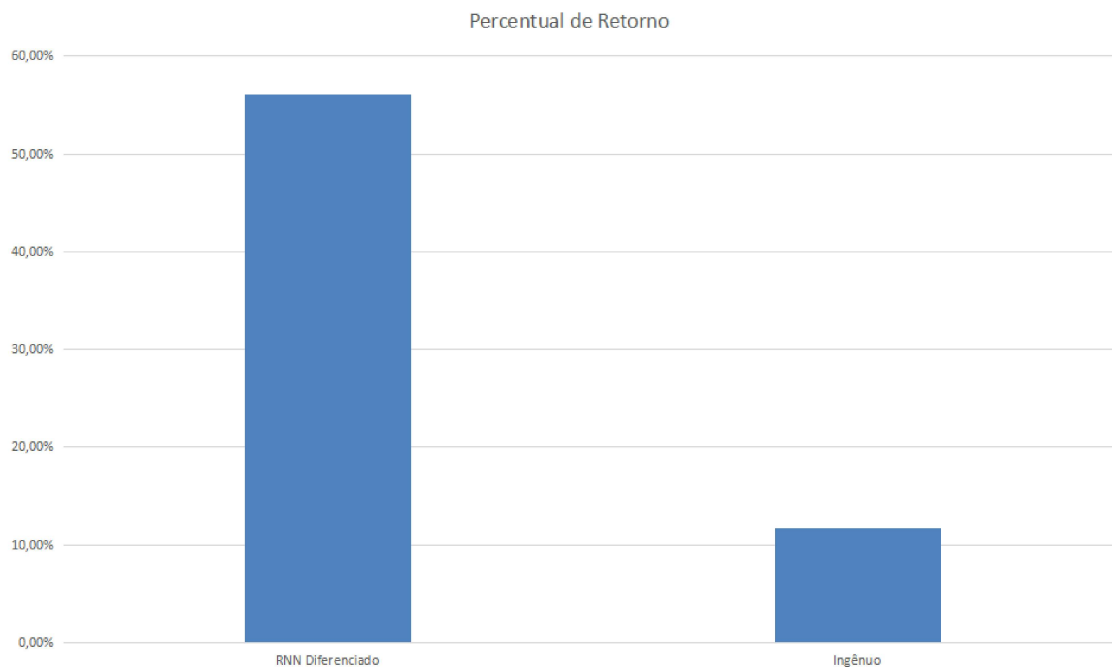


Figura 4.7: Comparação entre os percentuais de retorno Fonte: Próprio Autor

Como evidenciado na Figura 4.7, o modelo RNN+ARIMA apresentou um desempenho superior ao modelo ingênuo, onde este obteve um lucro de 11,7%, enquanto aquele registrou um retorno financeiro de 56,12%.

Capítulo 5

Conclusões e Trabalhos Futuros

Neste capítulo apresenta-se as conclusões, limitações e trabalhos futuros desta pesquisa.

5.1 Conclusões

A presente dissertação propôs-se a construir modelos de previsão de séries temporais, empregando abordagens estatísticas lineares, modelos fundamentados em aprendizado de máquinas, e híbridos. A condução de testes exaustivos e a comparação entre esses modelos visaram aprimorar a capacidade de previsão do preço do ativo digital Bitcoin.

Para a realização deste trabalho foram necessárias várias etapas, em cada foram encontrados problemas e desafios a serem resolvidos. O primeiro foi identificar quais seriam os métodos a serem utilizados na modelagem. Para tal tarefa realizou-se uma criteriosa Revisão Sistemática com o objetivo de analisar os modelos e métodos baseados em séries temporais utilizados para prever valores de criptomoedas e quais apresentam os melhores resultados.

Outro desafio foi identificar qual o intervalo de dados na série do Bitcoin seria utilizado na modelagem deste trabalho. A coleta foi feita sobre uma grande quantidade de dados e os mesmos foram testados nos modelos, o que nos permitiu desenvolver modelos mais robustos e que apresentem melhores resultados.

Com os dados coletados, iniciou-se uma análise minuciosa dos mesmos. Nesta análise foi identificado que trata-se de uma série com as seguintes características:

1. Tendência ascendente,

2. Sazonalidade presente,
3. Ruído alto,
4. Não estacionária, porém torna-se estacionária após uma diferenciação.

Ao realizar as comparações entre os resultados dos modelos, utilizando as métricas RMSE, MAE e MAPE, foi constatado que, em todas as métricas, o modelo RNN+ARIMA apresentou os melhores resultados. Os modelos RNN e LSTM+ARIMA apresentaram resultados muito semelhantes entre si. E o modelo LSTM não obteve um desempenho tão bom quanto os outros modelos baseados em Aprendizado de Máquina.

Acredita-se que, apesar de o LSTM ser considerado uma evolução do RNN, o RNN apresentou um melhor desempenho porque no contexto do preço do Bitcoin, as flutuações podem ser influenciadas mais fortemente por eventos recentes e padrões de curto prazo. Os LSTMs foram projetados para capturar dependências temporais de longo prazo, permitindo que o modelo memorize informações relevantes por períodos mais extensos. No entanto, essa capacidade de lembrar dependências de longo prazo também pode tornar o LSTM mais suscetível ao *overfitting*, especialmente quando há grande quantidade de ruído nos dados, como no caso do Bitcoin. Esta hipótese se confirma ao observar que o desempenho do modelo LSTM+ARIMA melhora consideravelmente em relação ao LSTM, pois quando os dados são diferenciados, os ruídos diminuem e o desempenho do modelo melhora.

Além disso, é importante observar que, apesar de apresentarem os piores desempenhos nas métricas RMSE e MAE, os modelos Arima e Sarima apresentaram, na métrica MAPE, um desempenho superior aos modelos LSTM, LSTM+ARIMA e RNN+ARIMA, o que demonstra, que um modelo mais simples pode contribuir na previsão do preço do Bitcoin e ajudar a embasar decisões de investimentos.

O teste de Simulações de Investimentos corroborou com os resultados das métricas onde o modelo RNN+ARIMA apresentou um retorno financeiro consideravelmente superior a um modelo ingênuo.

A análise dos resultados evidencia a superioridade dos modelos de redes neurais, LSTM e RNN, indicando sua maior adequação na previsão do preço do Bitcoin em termos de retorno financeiro. Em contrapartida, os resultados obtidos nos modelos ARIMA e SARIMA sugerem que essas abordagens possuem uma eficácia limitada nesse contexto específico.

Além disso, vale ressaltar que os modelos híbridos (RNN+ARIMA e LSTM+ARIMA) apresentaram resultados superiores às abordagens exclusivamente baseadas em Aprendizado de Máquina, o que reforça a premissa subjacente de que a combinação de diferentes métodos, como a integração de elementos estatísticos lineares, pode potencializar a precisão das previsões de preços do Bitcoin.

É crucial salientar que a previsão do preço do Bitcoin é uma tarefa intrinsecamente complexa e desafiadora. Não existe um modelo perfeito para essa empreitada, e os resultados de qualquer modelo podem variar consideravelmente com base no conjunto de dados utilizado e nas condições de mercado.

O objetivo geral deste trabalho foi atingido, ao construir e comparar diferentes modelos de previsão de valores de Bitcoin, verificando que os modelos híbridos, isto é, incorporando tanto aspectos de modelos lineares quanto de modelos baseados em Aprendizado de Máquina apresentaram um aprimoramento na precisão das estimativas do preço do Bitcoin.

5.2 Limitações

Esta pesquisa aborda modelos destinados a prever valores do Bitcoin, destacando o uso de modelos estatísticos lineares e aprendizado de máquina. A série temporal dos valores do Bitcoin capturada, refere-se a valores diários do ativo, porém estes dados podem ser capturados em intervalos menores, como por exemplo por hora e, até mesmo, por minuto. Caso os modelos fossem testados com dados capturados em intervalos diferentes, os resultados poderiam ser diferentes.

Além disso, existem fatores externos como notícias, movimentos macroeconômicos, políticos e sociais que podem causar grandes variações no valor do Bitcoin no curto prazo. Estes fatores não foram considerados nesta pesquisa, pois precisariam ser tratados por um modelo baseado em Análise de Sentimentos, o que foge ao propósito deste trabalho.

5.3 Trabalhos Futuros

A abordagem proposta permitiu encontrar modelos com métricas de erros bastante reduzidas. Além de obter um ROI (Retorno Sobre o Investimento), superior a 50% em 30 dias. No entanto, ainda há limitações nos experimentos durante o desenvolvimento desta pesquisa que são diretrizes para iniciar trabalhos futuros. São elas:

- Os dados capturados referem-se a um período específico da série temporal, o que não significa que, pegando outro intervalo de dados, os resultados serão iguais.
- Os dados foram capturados com intervalos de 1 dia entre cada observação, porém, acredita-se que, se os dados forem capturados em intervalos diferentes, como dados horários ou até mesmo semanais, a série apresentará características diferentes da série estudada neste trabalho, principalmente por questões sazonais.
- Dado seu caráter volátil e alta sensibilidade a eventos como notícias, movimentos políticos e econômicos globais, o mercado das criptomoedas pode se beneficiar de um modelo baseado em mineração de textos. Ao considerar uma análise de sentimentos sobre notícias mundiais, tal abordagem pode aprimorar a precisão das previsões e pode utilizar técnicas como Redes Neurais Fuzzy, Teoria dos Conjuntos Aproximados e Sistemas Baseados em Regras.
- Durante a simulação de investimentos não foram consideradas as taxas cobradas pelas *Exchanges* (Corretoras de criptomoedas), pois estas taxas variam de acordo com a *Exchange* e, em alguns casos elas suprimem esta taxa para determinadas datas, períodos e clientes. Desta maneira, um trabalho que considere as taxas das corretoras pode apresentar diferentes *insights* e perspectivas.
- A aplicação da metodologia desenvolvida neste estudo a outras criptomoedas com menor liquidez pode gerar resultados e *insights* distintos.

Referências

- [1] Aprendizado profundo. M. Press, Ed.
- [2] ADEGBORUWA, T. I., ADESHINA, S. A., BOUKAR, M. M. Time series analysis and prediction of bitcoin using long short term memory neural network. Em *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)* (2019), IEEE, p. 1–5.
- [3] ADHIKARI, R., AGRAWAL, R. K. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613* (2013).
- [4] ALMEIDA, C. L. Predição de séries temporais aplicada ao mercado de criptomoedas.
- [5] AMJAD, M., SHAH, D. Trading bitcoin and online time series prediction. Em *NIPS 2016 time series workshop* (2017), PMLR, p. 1–15.
- [6] ARGUELLES, D. C. M. Predicting the direction, maximum, minimum and closing price of daily/intra-daily bitcoin exchange rate using batch and online machine learning techniques.
- [7] BACK, A., OTHERS. Hashcash-a denial of service counter-measure.
- [8] BENGIO, Y., LECUN, Y., OTHERS. Scaling learning algorithms towards ai. *Large-scale kernel machines 34*, 5 (2007), 1–41.
- [9] BENGIO, Y., SIMARD, P., FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks 5*, 2 (1994), 157–166.
- [10] BERTOLO, L. Técnicas de previsão de box-jenkins-arima. *Catanduva: IMES* (2009).
- [11] BHATTACHARYYA, G. K. 5 tests of randomness against trend or serial correlations. *Handbook of Statistics 4* (1984), 89–111.
- [12] BIOLCHINI, J., MIAN, P. G., NATALI, A. C. C., TRAVASSOS, G. H. Systematic review in software engineering. *System engineering and computer science department COPPE/UFRJ, Technical Report ES 679*, 05 (2005), 45.
- [13] BOX, G. E., JENKINS, G. M., REINSEL, G. C., LJUNG, G. M. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [14] BROCKWELL, P. J., DAVIS, R. A. *Time series: theory and methods*. Springer science & business media, 1991.

- [15] BROCKWELL, P. J., DAVIS, R. A. *Introduction to time series and forecasting*. Springer, 2002.
- [16] BRUCE, R. F. A bayesian approach to semi-supervised learning. Em *NLPRS* (2001), p. 57–64.
- [17] CAVALCANTE, R. C., BRASILEIRO, R. C., SOUZA, V. L., NOBREGA, J. P., OLIVEIRA, A. L. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications* 55 (2016), 194–211.
- [18] CHEN, Y., HAO, Y. A feature weighted support vector machine and k-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications* 80 (2017), 340–355.
- [19] DANCHO, M. Tidy quantitative financial analysis. [s.l.].
- [20] DE PAULA, A., TRICHES, D. O mercado acionário e o desempenho dos ativos financeiros no brasil com emprego da análise técnica. *Perspectiva Econômica* 6, 1 (2010).
- [21] DENG, L., YU, D., OTHERS. Deep learning: methods and applications. *Foundations and trends® in signal processing* 7, 3–4 (2014), 197–387.
- [22] DERBENTSEV, V., DATSENKO, N., STEPANENKO, O., BEZKOROVAINYI, V. Forecasting cryptocurrency prices time series using machine learning approach. Em *SHS Web of Conferences* (2019), vol. 65, EDP Sciences, p. 02001.
- [23] DESEV, K., KABAIVANOV, S., DESEVN, D. Forecasting cryptocurrency markets through the use of time series models. *Business and Economic Horizons* 15, 2 (2019), 242–253.
- [24] ELDER, A. Aprenda a operar no mercado de ações. *Rio de Janeiro: Editora Campus* (2006).
- [25] FACELI, K., LORENA, A. C., GAMA, J., ALMEIDA, T. A. D., CARVALHO, A. C. P. D. L. F. D. Inteligência artificial: uma abordagem de aprendizado de máquina.
- [26] GARCIA, V. D. S. F. Séries temporais para predição de finanças no contexto de criptomoedas.
- [27] HOCHREITER, S. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München* 91, 1 (1991), 31.
- [28] HOCHREITER, S., SCHMIDHUBER, J. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [29] HUNDMAN, R. Forecasting functions for time series and linear models.
- [30] HYNDMAN, R. J., ATHANASOPOULOS, G. *Forecasting: principles and practice*. OTexts, 2018.
- [31] JASEMI, M., KIMIAGARI, A. M., MEMARIANI, A. A modern neural network model to do stock market timing on the basis of the ancient investment technique of japanese candlestick. *Expert Systems with Applications* 38, 4 (2011), 3884–3890.

- [32] JOTHIMANI, D., SHANKAR, R., YADAV, S. S. Discrete wavelet transform-based prediction of stock index: A study on national stock exchange fifty index. *arXiv preprint arXiv:1605.07278* (2016).
- [33] KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University 33*, 2004 (2004), 1–26.
- [34] KODAMA, O., PICHL, L., KAIZOJI, T. Regime change and trend prediction for bitcoin time series data. Em *CBU International Conference Proceedings* (2017), vol. 5, p. 384–388.
- [35] LAI, G., CHANG, W.-C., YANG, Y., LIU, H. Modeling long-and short-term temporal patterns with deep neural networks. Em *The 41st international ACM SIGIR conference on research & development in information retrieval* (2018), p. 95–104.
- [36] LIPSEY, R. G. *An introduction to positive economics*. Weidenfeld and Nicolson, 1968.
- [37] MALLQUI, D. C., FERNANDES, R. A. Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques. *Applied Soft Computing 75* (2019), 596–606.
- [38] MARGARIDO, M. A., DE MEDEIROS JUNIOR, H. Teste para mais de uma raiz unitária: uso do software sas® na elaboração de uma rotina para o teste dickey-pantula. *Pesquisa & Debate 17*, 1 (29) (2006).
- [39] MARSCHNER, P. F., CERETTA, P. S. Previsão de value-at-risk para o mercado de criptomoedas usando modelos egarch com regimes markovianos. *Brazilian Review of Finance 18*, 3 (2020), 80–107.
- [40] MARTINEZ, L. C., DA HORA, D. N., 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. Em *2009 International Joint Conference on Neural Networks* (2009), IEEE, p. 2006–2013.
- [41] MENGER, C., OTHERS. *Principios de economía política*. Bubok, 2012.
- [42] MITCHELL, "machine learning", new york, ny, usa: Mcgraw-hill, inc.
- [43] MONTGOMERY, D. C., JENNINGS, C. L., KULAHCI, M. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.
- [44] MORETTIN, P. A. *Modelos para previsão de séries temporais*, vol. 1. Instituto de Matemática Pura e Aplicada, 1981.
- [45] MORETTIN, P. A. *Econometria financeira: um curso em séries temporais financeiras*. Editora Blucher, 2017.
- [46] MORETTIN, P. A., TOLOI, C. M. D. C. Análise de séries temporais.
- [47] MURPHY, J. J. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin, 1999.

- [48] MURPHY, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [49] NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* (2008), 21260.
- [50] OLAH, C. Understanding lstm networks.
- [51] PARMEZAN, A. R. S., BATISTA, G. E. A study of the use of complexity measures in the similarity search process adopted by knn algorithm for time series prediction. Em *2015 IEEE 14th international conference on machine learning and applications (ICMLA)* (2015), IEEE, p. 45–51.
- [52] PARMEZAN, A. R. S., SOUZA, V. M., BATISTA, G. E. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information sciences 484* (2019), 302–337.
- [53] PASCANU, R., MIKOLOV, T., BENGIO, Y. On the difficulty of training recurrent neural networks. Em *International conference on machine learning* (2013), Pmlr, p. 1310–1318.
- [54] PIZZETTI, F. V. A volatilidade das criptomoedas: um estudo com utilização de modelos garch.
- [55] PONTI, M. A., DA COSTA, G. B. P. Como funciona o deep learning. *arXiv preprint arXiv:1806.07908* (2018).
- [56] QIN, Y., SONG, D., CHEN, H., CHENG, W., JIANG, G., COTTRELL, G. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971* (2017).
- [57] ROKACH, L., MAIMON, O., SHMUELI, E. *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*. Springer Nature, 2023.
- [58] ROTHBARD, M. N. *O que o governo fez com o nosso dinheiro?* Instituto Ludwig Von Mises, 2013.
- [59] RUMELHART, D. E., MCCLELLAND, J. L., PDP RESEARCH GROUP, C. *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1: Foundations*. MIT press, 1986.
- [60] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural networks 61* (2015), 85–117.
- [61] SHUMWAY, R. H., STOFFER, D. S., STOFFER, D. S. *Time series analysis and its applications*, vol. 3. Springer, 2000.
- [62] SILVA, A. R. S. Quantificação de risco em finanças: Bitcoin sob a avaliação do value at risk.
- [63] SILVA, L. P. Q., DE ARAÚJO, A. P., COTA, D. O., COTA, G. O., ROCHA, A. A. D. A. Utilizando hmm para previsão de preço e estratégia de investimento em criptomoedas bitcoin. Em *Anais do IV Workshop em Blockchain: Teoria, Tecnologias e Aplicações* (2021), SBC, p. 134–147.

-
- [64] SZABO, N. Shelling out: the origins of money. *Satoshi Nakamoto Institute 1* (2002).
- [65] TEIXEIRA, L. A., DE OLIVEIRA, A. L. I. A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert systems with applications* 37, 10 (2010), 6885–6890.
- [66] VRIGAZOVA, B., PAVLOVA, T., BOGDANOVA, B. Wavelet-based prediction of stock market returns during 2008 financial crisis.
- [67] WAGNER, D. Sector trading strategies. *John Wiley Sons* (2016).
- [68] WANG, Q., SONG, X., LI, R. A novel hybridization of nonlinear grey model and linear arima residual correction for forecasting us shale oil production. *Energy* 165 (2018), 1320–1331.
- [69] YOO, P. D., KIM, M. H., JAN, T. Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. Em *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)* (2005), vol. 2, IEEE, p. 835–841.
- [70] ZHANG, X., LIANG, X., ZHIYULI, A., ZHANG, S., XU, R., WU, B. At-lstm: An attention-based lstm model for financial time series prediction. Em *IOP Conference Series: Materials Science and Engineering* (2019), vol. 569, IOP Publishing, p. 052037.