

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS
GERAIS – *CAMPUS* BAMBUÍ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Mickael Osvaldo de Oliveira

**ANÁLISE COMPARATIVA ENTRE ALGORITMOS DE
PREDIÇÃO DE PREÇO PARA O BITCOIN**

BambuÍ - MG
2024

MICKAEL OSVALDO DE OLIVEIRA

ANÁLISE COMPARATIVA ENTRE ALGORITMOS DE PREDIÇÃO DE PREÇO PARA O BITCOIN

Trabalho de conclusão de curso apresentado
ao Curso de Bacharelado em Engenharia de
Computação do Instituto Federal de Educa-
ção, Ciência e Tecnologia de Minas Gerais –
Campus Bambuí para obtenção do grau de
Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Ciniro Aparecido
Leite Nametala

Ribeiro, Marcos Roberto.

Classe LaTeX para Trabalhos Acadêmicos de Institutos Federais/
Marcos Roberto Ribeiro. 2017.

55 p. :il.

Orientador: Nome do Orientador.

Co-orientadora: Nome do Co-orientadora.

Monografia de Trabalho de Conclusão de Curso - Instituto
Federal de Educação, Ciência e Tecnologia de Minas Gerais -
Campus Bambuí, Curso Engenharia de Computação, 2017

1. Trabalho de conclusão de curso.	2. Latex.	3.
Monografia.	I. Ribeiro, Marcos Roberto.	II. Título.

Mickael Osvaldo de Oliveira

ANÁLISE COMPARATIVA ENTRE ALGORITMOS DE PREDIÇÃO DE PREÇO PARA O BITCOIN

Trabalho de conclusão de curso apresentado
ao Curso de Bacharelado em Engenharia de
Computação do Instituto Federal de Educa-
ção, Ciência e Tecnologia de Minas Gerais –
Campus Bambuí para obtenção do grau de
Bacharel em Engenharia de Computação.

Aprovado em 15 de Dezembro de 2024 pela banca examinadora:

Prof. Dr. Ciniro Aparecido Leite Nametala – IFMG – *Campus* Bambuí – (Orientador)
Prof. Dr. Geraldo Henrique Alves Pereira – IFMG – *Campus* Bambuí
Prof. Dr. Marcos Roberto Ribeiro – IFMG – *Campus* Bambuí

Bambuí - MG
2024

A minha família e amigos.

Principalmente a minha mãe, que sempre me apoiou e incentivou a seguir em frente.

AGRADECIMENTOS

Agradeço a todos que contribuíram para a realização deste trabalho.

$$B > \frac{1}{n} \sum_{i=1}^n x_i$$

RESUMO

O mercado financeiro atrai a atenção de muitos há séculos, abrangendo ativos variados como arroz, ouro e ações. A flutuação dos preços pode indicar eventos importantes, e prever tais eventos pode determinar o sucesso ou fracasso de um negócio. Diversas técnicas de previsão, como a análise técnica e fundamentalista, surgiram ao longo dos anos. Recentemente, redes neurais artificiais, especialmente as combinadas com células de memória *Long Short-Term Memory* (LSTM), obtiveram sucesso na identificação de padrões em séries temporais antes despercebidos. Paralelamente, mercados voláteis como o de criptomoedas emergiram após o Bitcoin e a *Blockchain* em 2008, oferecendo alternativas para combater a inflação e o controle centralizado. Este estudo visa analisar e comparar algoritmos, em uma base de dados real, que buscam prever o comportamento dos ativos em mercados voláteis, principalmente no contexto do Bitcoin. Foram utilizadas diferentes abordagens baseadas em artigos recentes, no qual se obteve um melhor desempenho em modelos estatísticos no intervalo de quinze minutos.

Palavras-chave: Bitcoin, Redes neurais artificiais, previsão de preço.

ABSTRACT

The financial market has captured attention for centuries, including assets such as rice, gold, and stocks. Price fluctuations can reflect significant events, and the ability to predict these changes often determines the success or failure of businesses. Over the years, techniques like technical and fundamental analysis have been developed. Recently, artificial neural networks, particularly with the integration of Long Short-Term Memory (LSTM) cells, have gained prominence for identifying previously unnoticed patterns in time series data. Simultaneously, volatile markets like cryptocurrencies have emerged, driven by Bitcoin and Blockchain technology introduced in 2008. These markets offer alternatives to counter inflation and centralized control. This research aims to analyze and compare algorithms on a real-world dataset to predict asset behavior in volatile markets, focusing on Bitcoin. Based on recent research, the results reveal that statistical models achieved better performance for data sampled at fifteen-minute intervals.

Keywords: Bitcoin. Neural networks. Price prediction.

LISTA DE FIGURAS

Figura 1 – Representação da Blockchain	17
Figura 2 – Prova de trabalho	18
Figura 3 – Livros lançados por assunto	18
Figura 4 – Camadas de uma MLP	20
Figura 5 – Camadas cíclicas de uma RNN	21
Figura 6 – Diagrama de blocos de uma LSTM	22
Figura 7 – Estrutura de uma GRU	23
Figura 8 – Fluxo de implementação da solução	25
Figura 9 – Técnicas de janelamento	28
Figura 10 – Divisão de dados	28
Figura 11 – Gráfico de convergência	33
Figura 12 – Gráfico de preços reais e previstos - LSTM	34
Figura 13 – Gráfico de preços reais e previstos - BilSTM	34
Figura 14 – Gráfico de preços reais e previstos - GRU	35
Figura 15 – Gráfico de preços reais e previstos - ARIMA	35

LISTA DE TABELAS

Tabela 1 – Exchanges com maior volume e confiabilidade	19
Tabela 2 – Comparação entre estudos na área	24
Tabela 3 – Amostras da base de dados coletada	26
Tabela 4 – Estrutura do modelo baseado em LSTM	29
Tabela 5 – Estrutura do modelo baseado em GRU	29
Tabela 6 – Estrutura do modelo baseado em camadas Bidirecionais	29
Tabela 7 – Configuração dos parâmetros das redes neurais	29
Tabela 8 – Dados da divisão em conjuntos	32
Tabela 9 – Erros em cada modelo	36

LISTA DE ABREVIATURAS E SIGLAS

BTC – *Bitcoin*

FIAT – Moeda fiduciária

ARIMA – *Autoregressive Integrated Moving Average*

ANN – *Artificial Neural Network*

DNN – *Deep Neural Network*

RNN – *Recurrent Neural Network*

LSTM – *Long Short-Term Memory*

BiLSTM – *Bidirectional Long Short-Term Memory*

GRU – *Gated Recurrent Unit*

MAE – *Mean Absolute Error*

MASE – *Mean Absolute Scaled Error*

RMSE – *Root Mean Square Error*

MAPE – *Mean Absolute Percentage Error*

R^2 – *R-squared*

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa	14
1.2	Proposta	15
1.2.1	<i>Objetivo geral</i>	15
1.2.2	<i>Objetivos específicos</i>	15
1.3	Estrutura do documento	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Bitcoin	17
2.2	<i>Autoregressive Integrated Moving Average</i>	19
2.3	Redes neurais artificiais	19
2.3.1	<i>Recurrent Neural Networks</i>	20
2.3.2	<i>Long-Short Term Memory</i>	21
2.3.3	<i>Bidirectional Long Short-Term Memory</i>	21
2.3.4	<i>Gated Recurrent Unit</i>	22
2.4	<i>Séries temporais</i>	22
2.5	Estado da arte	23
3	METODOLOGIA	25
3.1	Solução proposta	25
3.2	Coleta de dados	25
3.3	Pré processamento	26
3.3.1	<i>Validação de completude</i>	26
3.3.2	<i>Normalização</i>	26
3.3.3	<i>Limitações e diferenças entre algoritmos</i>	27
3.3.4	<i>Janelamento</i>	27
3.3.5	<i>Divisão em conjuntos</i>	27
3.4	Implementação	28
3.5	Arquitetura	28
3.6	Configuração	29
3.7	Avaliação	29
3.8	Materiais e Tecnologias	31
4	RESULTADOS E DISCUSSÃO	32
4.1	Validação de completude	32
4.2	Tratamento e pré processamento de dados	32
4.3	Treinamento dos modelo	32
4.4	Visualização de resultados	33

4.4.1	<i>Long Short-Term Memory</i>	33
4.4.2	<i>Bidirectional Long Short-Term Memory</i>	34
4.4.3	<i>Gated Recurrent Unit</i>	34
4.4.4	<i>Auto Regressive Integrated Moving Average</i>	35
4.5	Análise de resultados	35
5	CONCLUSÃO	37
	REFERÊNCIAS	38

1 INTRODUÇÃO

A história da matemática na predição de preços remonta ao trabalho pioneiro *Théorie de la Spéculation* de Bachelier (1900) (COURTAULT *et al.*, 2000). Fundamentado no conceito do movimento Browniano, fenômeno que descreve o deslocamento errático de partículas em um fluido, o autor o utiliza como analogia para o comportamento imprevisível dos preços de ativos no mercado financeiro. Essa ideia deu origem à teoria conhecida como *Random Walk* (Caminhar Aleatório), que posteriormente serviu de base para outras modelagens, como a de Black e Scholes (1973).

Dentre os ramos que compõem a projeção de preços, destacam-se as análises técnicas e fundamentalistas. A análise técnica busca prever os preços futuros baseadas em análises passadas de preço, volume e contratos abertos em opções (PRING, 2002). Por outro lado, a análise fundamentalista se baseia em estimar o valor intrínseco dos ativos, ou seja, encontrar o preço justo por fundamentos do projeto, preços passados, situação atual e oportunidades futuras (AHMED; HASSAN; MABROUK, 2015).

O *Bitcoin* foi introduzido por Nakamoto (2008), entidade desconhecida sob o pseudônimo de Satoshi Nakamoto. Considerada a primeira moeda digital implementada que transaciona de maneira totalmente descentralizada, assim, sem a necessidade de uma autoridade central como um banco ou governo. Para isso, utiliza uma tecnologia denominada *Blockchain*, que funciona como um livro caixa em uma estrutura de dados distribuída e imutável (HUGHES *et al.*, 2019). Podem ser trocados por outras moedas ou ativos, como o Real, em plataformas de negociação de ativos digitais chamadas *Exchanges* (CHEN; LIN; WU, 2022).

A troca desses ativos gera um histórico de transações como em qualquer mercado financeiro, porém, consideravelmente mais volátil devido a falta de lastro das moedas fiduciárias (FIAT) tradicionais. Tendo em vista o caminhar aleatório do mercado, a análise preditiva não deve ser considerada uma tarefa trivial, mas, pode ser facilitada com o uso de técnicas de aprendizado de máquina. Nos dias atuais, segundo Fang, Su e Yin (2020), os analistas de mercado têm feito uso não somente destes métodos como também de outros modelos computacionais convencionais ou híbridos, que têm por objetivo prever os preços, obtendo sucesso especialmente no contexto das criptomoedas (ATSALAKIS *et al.*, 2019).

1.1 Justificativa

As redes neurais artificiais (RNAs) se tornaram uma peça fundamental na sociedade moderna, por isso, novas técnicas surgem a cada dia para resolver múltiplos problemas, muitas vezes não solucionáveis por humanos (GOODFELLOW; BENGIO; COURVILLE, 2016). Por outro lado, o mercado de ativos baseados em criptografia cresce exponencialmente, sendo muito utilizado como reserva de valor, meio de pagamento e uma forma de diversificação de investimentos (SOUSA *et al.*, 2022).

Então, justifica-se esse trabalho pois essas duas tecnologias convergem para um mundo cada vez mais digital, a união das mesmas pode até mesmo ser intrínseca a muitas criptomoedas. A previsão de preços em mercados financeiros não é recente, entretanto, ativos com alta volatilidade, como o Bitcoin, ainda apresentam potencial para operações lucrativas (TRIPATHI; SHARMA, 2023). Assim, este estudo busca explorar dois temas relevantes atualmente, especialmente sob a perspectiva da engenharia de computação, contribuindo para o avanço do conhecimento em áreas exploradas.

1.2 Proposta

Este estudo tem como proposta analisar e avaliar algoritmos voltados para previsão de preços, espera-se verificar sua lucratividade em operações usando um dos mais famosos criptoativos, o Bitcoin. Os modelos serão avaliados de acordo com suas previsões em um cenário real de variação de preço, estruturados por meio de uma série temporal histórica. A pesquisa utilizou essa mesma base de dados e manteve os métodos de pré-processamento fixos, variando no contexto da aplicação apenas o algoritmo de previsão. Isto será feito de forma a manter as condições de experimentação para todos os algoritmos, garantindo assim que a diferença esteja apenas no funcionamento de cada um. Os dados foram obtidos através dos registros de negociações reais na *Exchange* Binance, que contém informações como o preço, volume e o número de transações realizadas entre janeiro a setembro de 2020. As arquiteturas exploradas incluem a ARIMA, LSTM, BiLSTM e GRU.

1.2.1 *Objetivo geral*

Comparar o desempenho de algoritmos de predição de preço no contexto do Bitcoin.

1.2.2 *Objetivos específicos*

Para atingir o objetivo principal, se fazem necessários os seguintes objetivos específicos:

- Desenvolvimento da estrutura computacional necessária para selecionar, implementar e realizar previsões por meio de ferramentas tecnológicas adequadas (como linguagens de programação, métodos de extração e armazenamento de dados, geradores gráficos e demais ferramentas que se fizerem necessárias);
- Avaliação de RNAs e compará-los, frente aos *Benchmarks* de interesse, a fim de determinar qual tem melhor desempenho;
- Exploração de possíveis variações em métodos conhecidos, visando adaptá-los a um novo cenário;

- Avaliar a rentabilidade desses métodos através de métricas em uma base de dados real.

1.3 Estrutura do documento

Este trabalho está estruturado em seis capítulos, Após a introdução, são apresentados os fundamentos teóricos necessários para a compreensão das principais técnicas empregadas na metodologia. Desse modo, o capítulo 2 aborda conceitos de criptomoedas e séries temporais, fornecendo uma introdução à análise de dados, explorando artigos relacionados à estrutura do mercado e a predição de preços. Além disso, é apresentado o estado da arte em relação as RNAs e suas aplicações no contexto do mercado financeiro.

O Capítulo 3 apresenta a metodologia, organizada de acordo com a sequência de implementação da solução. No capítulo 4, são descritos os resultados e discussões dos experimentos realizados. O capítulo 5 oferece uma conclusão que abrange as principais contribuições, limitações da pesquisa e propostas para trabalhos futuros.

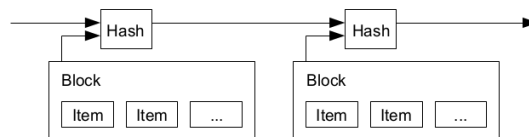
2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo oferece o arcabouço teórico necessário para a compreensão da pesquisa em si. Portanto, serão abordados conceitos fundamentais relacionados a criptomoedas, *blockchain*, RNAs, ARIMA e previsão em séries temporais.

2.1 Bitcoin

A ideia de ativos digitais descentralizados baseados em criptografia, ou criptomoedas, foi marcada por inúmeras tentativas anteriores, contudo, só foi implementada com o advento da *Blockchain* (MOLLING *et al.*, 2020). Referida por Yuan e Wang (2018) como um registro compartilhado distribuído, no qual a verificação, armazenamento, manutenção e transmissão dos dados são baseados na confiança mútua entre as partes, estabelecida por meio de algoritmos matemáticos. Tal estrutura é formada por blocos que contêm um conjunto de transações, cada bloco possui um identificador exclusivo chamado *Hash*, gerado a partir das informações presentes no bloco, incluindo o do anterior. Esse encadeamento de blocos cria uma sequência contínua e cronológica, na qual os blocos são ligados ao anterior formando uma cadeia (VUJIČIĆ; JAGODIĆ; RANĐIĆ, 2018).

Figura 1 – Representação da Blockchain



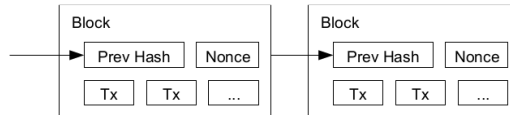
Fonte: NAKAMOTO, 2008.

Segundo Ferdiansyah *et al.* (2019), o primeiro e mais conhecido ativo desse setor se chama Bitcoin, definido como um dinheiro eletrônico negociado diretamente entre pares, sem passar por uma instituição financeira (NAKAMOTO, 2008). Desde então, tem se publicado diversos artigos que buscam explorar essa nova área de mercado, segundo Sousa *et al.* (2022), as tecnologias baseadas em *Blockchain* são mais rápidas, ágeis e seguras que seus pares centralizados. Por isso, esse nicho vem ganhando bastante espaço na mídia, assim, atraindo interesse por parte de investidores individuais e fundos (YUAN; WANG, 2018).

Cada *Bitcoin* é subdividido em 100 milhões de unidades menores chamadas Satoshis, ou simplesmente "Sats". Seu armazenamento é feito em uma carteira digital criptografada de maneira assimétrica *Offline* chamadas *Cold Wallets* ou *Online* em serviços de *Hot Wallets* (DAS; FAUST; LOSS, 2019). A transferência ocorre por prova de trabalho, um validador chamado de minerador é responsável por resolver um problema matemático de dificuldade variável, além disso, serve como nó que confere a operação de outros mineradores e garante a integridade da rede. Quando o problema é resolvido adiciona-se

um bloco de transação na Blockchain. Em troca, o minerador recebe uma recompensa em *Bitcoins* (SRIMAN; GANESH KUMAR; SHAMILI, 2021). A dificuldade da prova de trabalho é ajustada automaticamente pelo *Hashrate*, indicador que mede a quantidade de poder computacional necessário à mineração. Quanto maior o *Hashrate*, mais difícil é encontrar o próximo bloco, regulando o tempo entre a adição de novos blocos a rede, mantendo uma média de 10 minutos.

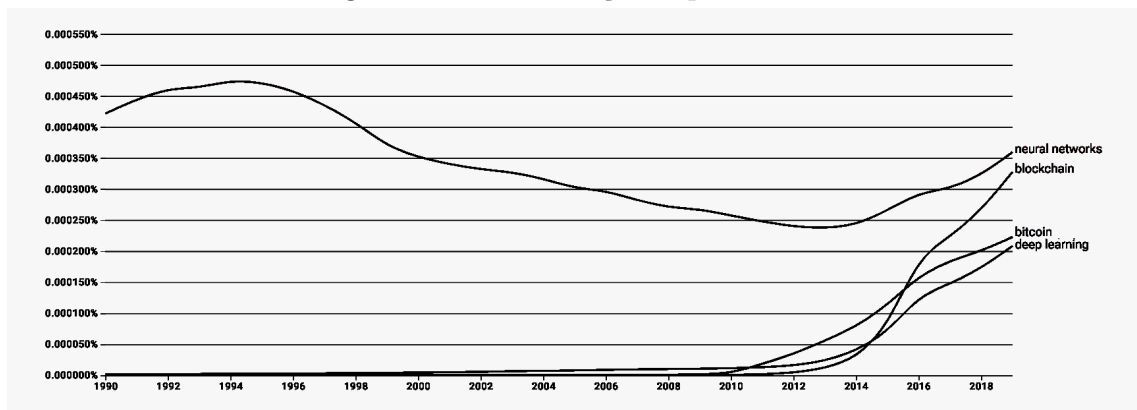
Figura 2 – Prova de trabalho



Fonte: NAKAMOTO, 2008.

A cada quatro anos, o número de *Bitcoins* gerados por bloco é reduzido pela metade, evento conhecido como *Halving*. O fator de redução confere a moeda uma escassez programada, limitada a 21 milhões de unidades, que a torna deflacionária. Ativos baseados em criptografia e a inteligência artificial são parte fundamental da *Web 3.0*, considerada como a terceira geração da internet (ZHU; LI; CHEN, 2024). Recentemente os contratos inteligentes, *DeFi* (Finanças Descentralizadas), *NFTs* (Tokens Não Fungíveis) e a inteligência artificial generativa ensaiam uma nova era de aplicações e serviços (BOOKS, 2024).

Figura 3 – Livros lançados por assunto



Fonte: (BOOKS, 2024)

No contexto de negociação do *Bitcoin* a mais conhecida das *Exchanges* é a Binance, que opera em diversos países e possui um volume de transações diário de bilhões de dólares (CHEN; LIN; WU, 2022). Com uma liquidez alta o *Spread*, diferença entre o preço de compra e venda, é baixo, o que representa o valor bem acurado com o oferecido no mercado. Além disso, existem *Sites* de comparação de preços. os mais populares são o CoinGecko e CoinMarketCap, funcionando como portais que classificam as plataformas, moedas e até a situação atual do mercado através de índices, como visto na tabela 1.

Tabela 1 – Exchanges com maior volume e confiabilidade

Exchange	Volume 24h (BTC)	Score Coingecko	Score CoinMarketCap
Binance (Global)	135.157	10/10	9,9/10
Bybit	44.720	10/10	7.6/10
HTX (Huobi)	34.250	9/10	6.9/10

Fonte: Elaborado pelo autor, 2024.

2.2 *Autoregressive Integrated Moving Average*

O ARIMA (*Autoregressive Integrated Moving Average*) tem suas raízes na econometria e na estatística. Sua história remonta ao trabalho de Box e Jenkins (1970), no qual os autores introduziram uma abordagem sistemática para identificar, estimar e diagnosticar modelos de séries temporais, conhecida como metodologia Box-Jenkins (SHUMWAY; STOFFER, 2017). Combinam três componentes principais: auto-regressão (AR), diferenciação (I) e médias móveis (MA). A componente AR descreve a relação entre uma observação atual e observações passadas, enquanto a componente MA modela a dependência entre uma observação e erros de previsão passados.

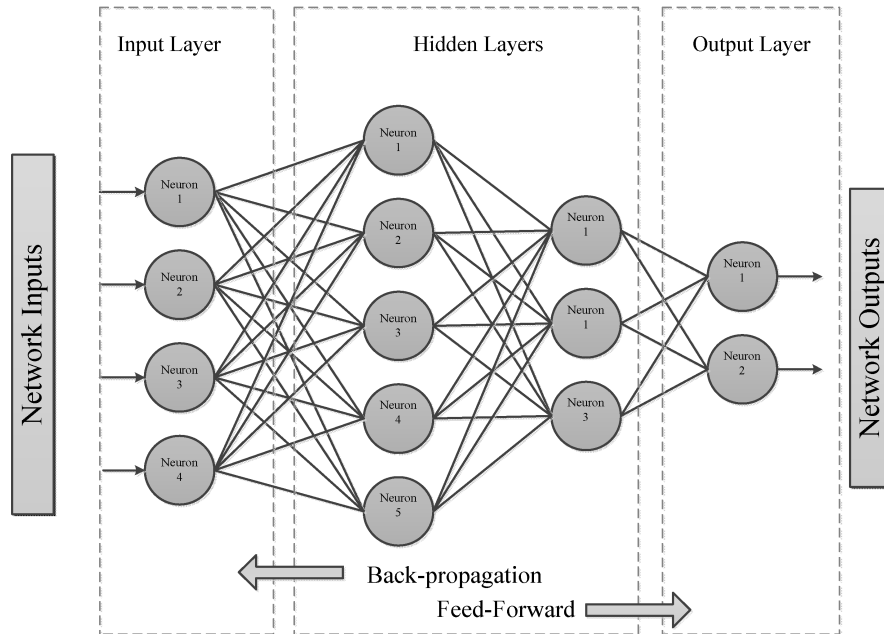
Então, atribui-se a cada uma das componentes do modelos as siglas p , d e q representado o ajuste na série. O parâmetro p é a ordem do modelo AR, d é o grau de diferenciação (I) e q é a ordem do modelo MA. Para ajustar o modelo aos dados, o comum é usar uma análise das funções de autocorrelação total e parcial (ACF e PACF) ou a grade de busca (*grid search*) com validação cruzada para otimizar esses valores, minimizando o erro. Existem outras variantes dos métodos ARMA, como o SARIMA (ARIMA sazonal) e o SARIMAX (ARIMA sazonal com variáveis exógenas).

2.3 Redes neurais artificiais

A teoria das redes neurais artificiais (RNAs), do inglês *Artificial neural networks* (ANNs), surgiu com os estudos realizados por Rosenblatt (1957). Este autor desenvolveu o *Perceptron*, um algoritmo fundamentado nas teorias formuladas por McCulloch e Pitts (1943) e Hebb (1949) que modelavam as conexões dos neurônios no cérebro de animais (GOODFELLOW; BENGIO; COURVILLE, 2016). Apesar de suas vantagens, à época o *Perceptron* era capaz apenas de resolver problemas linearmente separáveis, como destacado por Minsky e Papert (1969). Este fato os impedia de ser utilizados em muitos problemas reais, o que causou uma crise no estudo da área como um todo. Fato que mais tarde foi contornado por Rumelhart, Hinton e Williams (1986), que juntos desenvolveram o algoritmo de *Backpropagation* para treinar a arquitetura denominanda de *Multilayer Perceptron* (MLP).

Após esse início, os algoritmos baseados em RNAs evoluíram de forma a se tornar cada vez mais precisos e especializados em tarefas distintas. Pode-se dar destaque neste contexto aos modelos que podem ser combinados para gerar arquiteturas no que hoje

Figura 4 – Camadas de uma MLP



Fonte: (ABDOLRASOL *et al.*, 2021)

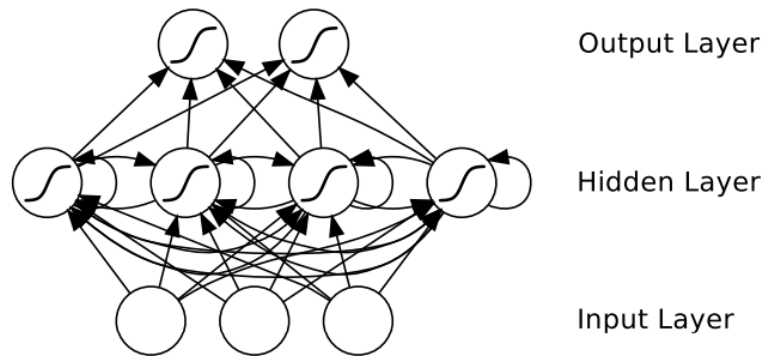
é conhecido como *Deep Neural Networks* (DNNs) ou *Deep Learning* (GOODFELLOW; BENGIO; COURVILLE, 2016). Como se verifica nas revisões bibliográficas mais recentes, tais modelos vêm demonstrando grande capacidade de auxiliar na resolução de problemas de classificação, previsão e análise de sentimentos (HANCOCK; KHOSHGOFTAAR, 2020).

2.3.1 Recurrent Neural Networks

Das arquiteturas apresentadas é possível notar que representam grafos cujas arestas não formam ciclos na mesma fase. Quando tal condição é subvertida e se estabelecem ciclos, como visto na figura 5, obtêm-se redes neurais recorrentes (RNRs), do inglês *Recurrent Neural Networks* (RNNs)(GRAVES, 2012). Embora essa diferença possa parecer trivial, as implicações devem ser consideradas na implementação. Enquanto uma MLP percorre apenas entre vetores de entrada e saída, Graves (2012) destaca que uma RNN, a princípio, consegue mapear todo o histórico de entradas para cada saída. Atuando como uma memória que persiste de resultados passados para execuções futuras. Pode-se então inferir a utilidade desses modelos em dados sequenciais $x^{(1)}, \dots, x^{(\tau)}$ (GOODFELLOW; BENGIO; COURVILLE, 2016).

Armazenar dependências ao longo das iterações é certamente o principal desafio matemático na implementação de redes recorrentes. Segundo (GRAVES, 2012), o problema consiste em que gradientes propagados por muitas execuções tendem a desaparecer (na maioria dos casos) ou explodir (raramente, mas com grande impacto na otimização). Mesmo assumindo que os parâmetros da rede sejam estáveis, a dificuldade surge à medida

Figura 5 – Camadas cíclicas de uma RNN



Fonte: (GRAVES, 2012)

que em que o modelo converge ao passar das épocas e os *Steps*¹ se tornam cada vez menores (GOODFELLOW; BENGIO; COURVILLE, 2016). Uma maneira de lidar com a dispersão dos gradientes é projetar um modelo que opere em múltiplas escalas temporais (EL HIHI; BENGIO, 1995) e realizar o *Clipping*² dos gradientes para evitar as explosões (QIAN *et al.*, 2021).

2.3.2 Long-Short Term Memory

Os modelos baseados em células de memória *Long-Short Term Memory* (LSTM) foram desenvolvidos por Hochreiter e Schmidhuber (1997), fazem parte de uma seleta classe dos modelos sequenciais de RNNs controladas (*gated RNNs*). A ideia é produzir um *loop* entre uma mesma unidade, o que leva a caminhos onde o gradiente pode perdurar por muitas iterações. Novas abordagens introduziram o peso nesse *loop* condicionado ao contexto em vez de fixo, sendo então controlado por outra unidade oculta (GERS; SCHMIDHUBER; CUMMINS, 2000). Assim, a escala pode ser alterada dinamicamente, mesmo que uma LSTM tenha parâmetros fixos, a escala pode mudar com base na sequência de entrada, pois as constantes de tempo são determinadas pelo próprio modelo.

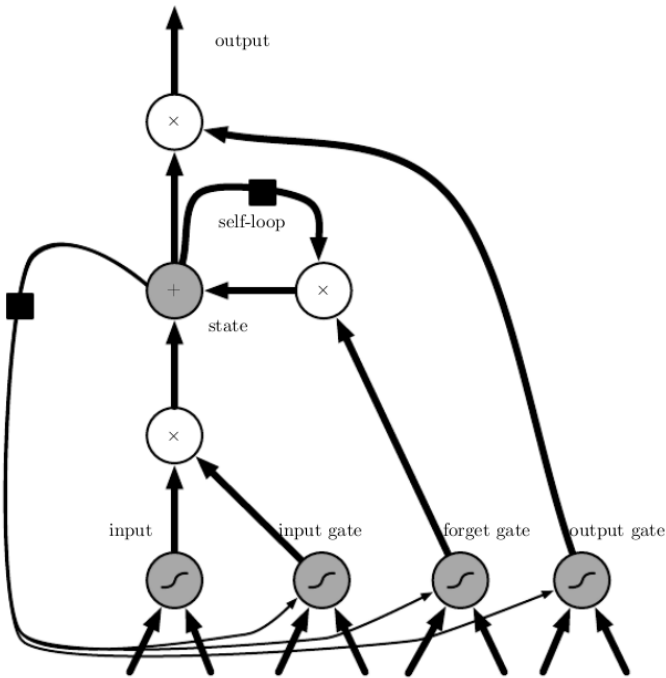
2.3.3 Bidirectional Long Short-Term Memory

A ideia de adicionar camadas em direções opostas nas RNNs foi proposta por Schuster e Paliwal (1997) e denominadas redes bidirecionais. Essa técnica posteriormente foi adaptada nos estudos de Graves e Schmidhuber (2005), que propuseram uma LSTM Bidirecional, ou *Bidirectional Long Short-Term Memory* (biLSTM). Sua estrutura é

¹ O salto no aprendizado (ou *step*) da rede se tornar exponencialmente menor pode ser explicado observando as mínimas ou máximas locais e globais em uma função n-dimensional. Fato demonstrado no livro de Stewart (2012).

² *Clipping* é uma técnica que consiste em limitar a magnitude dos gradientes a um intervalo válido, impedindo que ultrapassem valores predefinidos.

Figura 6 – Diagrama de blocos de uma LSTM



Fonte: (GRAVES, 2012)

formada por duas LSTMs: uma recebendo as entradas à frente enquanto outra recebe na direção oposta. O intuito é que a saída de cada unidade influencie indiretamente em sua contraparte, de forma que os gradientes não dispersem conforme o tempo.

2.3.4 Gated Recurrent Unit

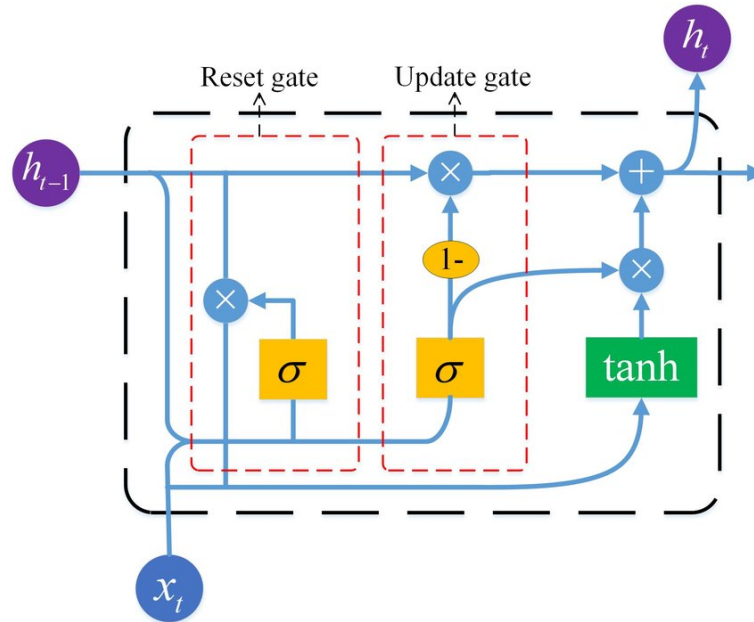
A *Gated Recurrent Unit* (GRU) é um tipo de RNN introduzida por Cho *et al.* (2014). Sua estrutura é semelhante às LSTMs, mas, têm uma arquitetura minimalista e, portanto, são computacionalmente mais eficientes. Utilizam dois tipos principais de portas (*gates*) para controlar o fluxo de informações dentro da unidade: a porta do esquecimento (*Reset gate*) e a porta de atualização (*update gate*).

2.4 Séries temporais

Séries temporais são conjuntos de dados intrinsecamente relacionados ao tempo, sua natureza ordenada e agrupada em intervalos regulares torna a sequência cronológica fundamental (ESLING; AGON, 2012). Alguns desses conjuntos possuem ciclos repetitivos em períodos distintos, chamadas sazonalidades. Possuem inúmeras aplicações, representando vendas, preços de ações e variações nos mais diversos contextos.

O número de amostras e a correlação de eventos são fatores primordiais que definem a complexidade da análise. Nison (2001) destaca que já no século XVIII, os japoneses desenvolveram uma forma de visualização de séries temporais popularmente conhecida como *Candlestick*.

Figura 7 – Estrutura de uma GRU



Fonte: (CHEN *et al.*, 2021)

2.5 Estado da arte

Ferdiansyah *et al.* (2019) obtiveram sucesso em prever preços do Bitcoin para o dia seguinte com modelos LSTM. Autores como Tripathi e Sharma (2023) combinam métodos bayesianos, processamento de sinais e redes neurais a fim de prever o preço desses ativos em múltiplos intervalos de tempo. Estudos propostos por Caux, Bernardini e Viterbo (2020) comparam o desempenho de LSTMs e GRUs na previsão de preços de criptomoedas no gráfico diário com janelamento de 32 dias. Siami Namini e Tavakoli (2019) propuseram uma pesquisa comparativa entre LSTMs e BiLSTMs no contexto de séries temporais como um todo, enquanto Zhang, Cai e Wen (2024) compilaram resultados de modelos para previsão de preço, detecção de bolha e construção de portfólio que podem aumentar a lucratividade no âmbito das criptomoedas.

A variedade de algoritmos empregados e a diversidade em resultados obtidos demonstram a complexidade do problema, assim, reforça a necessidade de se explorar diferentes abordagens. A tabela 2 apresenta uma comparação entre os estudos citados e o presente trabalho.

Tabela 2 – Comparação entre estudos na área

Pesquisadores	Ativo	Entradas	Saídas	Modelos	Métricas
(CAUX; BERNARDINI; VITERBO, 2020)	Bitcoin	Preço	Preço médio	LSTM, GRU	SMAPE
(FERDIANSYAH <i>et al.</i> , 2019)	Bitcoin	Preço	Preço	LSTM	RMSE
(SIAMI NAMINI; TAVAKOLI, 2019)	Ações	Preço	Preço	ARIMA, LSTM, BiLSTM	RMSE
(TRIPATHI; SHARMA, 2023)	Bitcoin	Preço ou indicadores	Preço	DANN, LSTM, BiLSTM, CNN-BiLSTM	RMSE, MAE, MAPE
Autor	Bitcoin	Preço, trocas e volume	Preço	ARIMA, LSTM, BiLSTM e GRU	MSE, MAPE, RMSE, R^2

Fonte: Elaborado pelo autor, 2024.

3 METODOLOGIA

Utilizando como base Gerhardt e Silveira (2009), pode-se dizer que a pesquisa adota uma abordagem quantitativa e experimental, ou seja, visa analisar e contrastar o desempenho de diferentes algoritmos em condições controladas. A natureza aplicada do estudo busca não apenas compreender as nuances de cada algoritmo, mas também oferecer *insights* para a seleção e implementação dos mais eficazes. A metodologia descritiva permite uma análise detalhada dos resultados obtidos, destacando as diferenças significativas entre os modelos avaliados.

3.1 Solução proposta

Figura 8 – Fluxo de implementação da solução



Fonte: Elaborado pelo autor, 2024.

3.2 Coleta de dados

A coleta de dados ou obtenção de *Datasets* envolve inicialmente selecionar locais como APIs, bancos de dados *Online* ou arquivos históricos que forneçam os dados em um formato estruturado, como JSON, CSV ou TOML. Mas, é possível estabelecer um processo de coleta automatizada através de *Crawlers* ou *Web Scrapping* para extrair as informações brutas. Independente da escolha sempre é preciso realizar um projeto de dados, que envolve a definição de quais informações são relevantes e sua correlação, para isso, existem métodos como a análise exploratória de dados (EDA). A tabela 3 apresenta um exemplo de amostras que foram coletadas, contendo informações de volume, que indicam a intensidade de negociações, trocas, que representam a proporção de compras ou vendas, e preço, que é o valor de fechamento do ativo para aquele *Candle* de quinze minutos.

3.3 Pré processamento

No processo de análise de dados, é comum que os dados brutos apresentem inconsistências, como valores ausentes, duplicados ou mal formatados. Essas imperfeições comprometem a eficácia dos modelos, antes de enviar os dados para o treinamento, é crucial assegurar que estejam completos, formatados e escalonados de maneira adequada. A essa etapa dá-se o nome de pré-processamento.

Tabela 3 – Amostras da base de dados coletada

Data (GMT-3)	Volume (USDT)	Trocas (Total)	Preço (USDT)
2020-01-01 00:00:00	1959651,83	2811	7228,5
2020-01-01 00:15:00	1225409,70	1897	7237,15
2020-01-01 00:30:00	1469869,68	2163	7221,27
2020-01-01 00:45:00	1012436,07	1466	7225,01
2020-01-01 01:00:00	1102372,81	1985	7219,09

Fonte: Elaborado pelo autor, 2024.

3.3.1 Validação de completude

Para verificar a completude dos dados, foi utilizada uma função nativa do *Pandas*, garantindo que todas as entradas estejam presentes na base. Embora o conector realize a captura, é necessário uma verificação adicional a cada análise para assegurar a qualidade das amostras. Caso exista valores ausentes, é possível utilizar métodos como a interpolação linear para estimar os valores faltantes e preenchê-los.

$$P_1(x) = y_0 + \frac{(y_1 - y_0)}{x_1 - x_0}(x - x_0) \quad (1)$$

A interpolação linear (equação 1) é baseada na reta que liga dois pontos, ma qual y_0 e y_1 são os valores conhecidos para os pontos x_0 e x_1 entre a variável x , que se deseja calcular o correspondente P .

3.3.2 Normalização

Em um *Dataset* multivariado é comum que os valores estejam em escalas distintas, no caso do *Bitcoin*, o preço e o volume variam em ordens de grandeza superiores ao número absoluto de transações. Isso se deve à própria natureza da dimensão dos dados, por isso, é necessário padronizar as variáveis para que o modelo não seja enviesado. Uma das técnicas de normalização mais famosas ajusta as colunas para um intervalo que esteja entre o máximo e o mínimo valor encontrado, assim, é chamada de *MinMaxScaler*.

$$X_{Scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

Na equação acima, x é o valor original da variável, $\min(x)$ é o valor mínimo encontrado na coluna e $\max(x)$ o valor máximo. Ao final obtemos o equivalente dimensionado da respectiva linha, ajustado em escala de 0 a 1.

3.3.3 Limitações e diferenças entre algoritmos

O processamento de dados em diferentes algoritmos, seja no aprendizado de máquina ou modelagem estatística, requer um dimensionamento de dados contendente com as necessidades e limitações de cada método. Ao utilizar RNAs, é comum que o modelo consiga lidar bem com múltiplas variáveis e forneça um arcabouço robusto de soluções adquiridas durante o treinamento.

Por outro lado, modelos como a regressão linear são tradicionalmente univariados, significando que modelam uma única variável de interesse. Outra característica é que não são treinados no sentido tradicional, em vez disso, ajustam seus parâmetros diretamente aos valores históricos, requerendo dados estacionários. Ou seja, as entradas devem ser ligeiramente adaptadas às necessidades do modelo.

3.3.4 Janelamento

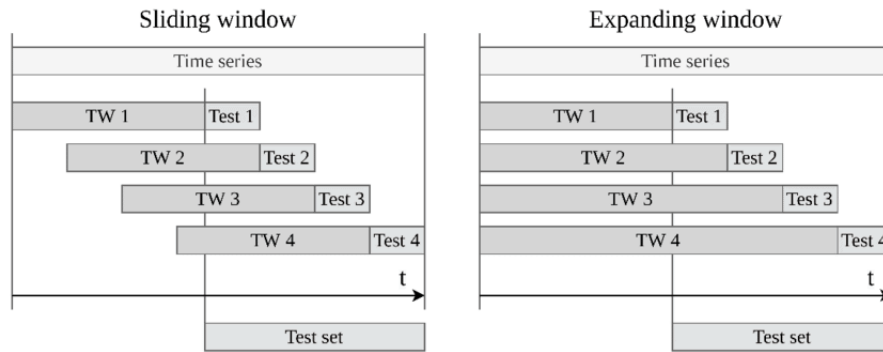
Ao se trabalhar com séries temporais é comum a separação dos dados em segmentos menores a fim de capturar a dinâmica ao longo do tempo. Uma dessas técnicas envolve a criação de janelas deslizantes, na qual um intervalo fixo de observações é utilizado para prever os próximos valores. No contexto de RNNs, essa abordagem é bastante útil, pois permite que o modelo capture padrões locais e evolutivos, aproveitando o poder da memória da RNA para entender sequências.

Entretanto, quando se trata de modelos estatísticos, o processo de segmentação é geralmente feito de maneira incremental. Em vez de usar uma janela deslizante de tamanho fixo, o modelo pode se beneficiar de uma janela expansiva, onde todos os dados disponíveis até um determinado ponto são utilizados para fazer a previsão seguinte. Isso torna justa a comparação, visto que o máximo de informação disponível é utilizada para cada previsão.

3.3.5 Divisão em conjuntos

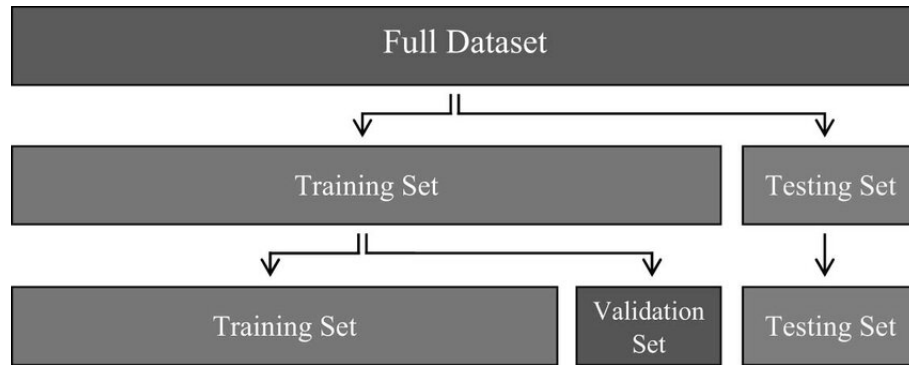
Para avaliar a eficácia dos modelos de aprendizado supervisionado, a base de dados é dividida em três subconjuntos: treinamento, validação e teste. O conjunto de treinamento é usado para ajustar os pesos do modelo, enquanto o de validação serve para otimizar os hiperparâmetros com base na função de perda. Após o limite de épocas o conjunto de teste, que contém dados que o modelo ainda não viu, é utilizado para verificar a capacidade de generalizar seu desempenho. A etapa não é necessária para o ARIMA, que ajusta seus parâmetros diretamente sobre todos os dados.

Figura 9 – Técnicas de janelamento



Fonte: MARKUDOVA *et al.*, 2021.

Figura 10 – Divisão de dados



Fonte: MARKUDOVA *et al.*, 2021.

3.4 Implementação

Todos os modelos foram desenvolvidos utilizando a linguagem de programação *Python* versão 3.10 (RAMALHO, 2021), juntamente com as bibliotecas *Pandas* e *Numpy* (MCKINNEY, 2022). Na implementação de RNNs foram utilizados os *Frameworks* Keras 3.10 e TensorFlow 2.10 (GÉRON, 2021), essas ferramentas podem ser adaptadas para processamento paralelo em GPU, mas, para isso a placa de video deve ter suporte a CUDA, tecnologia de código aberto da NVIDIA. O ARIMA foi feito e otimizado com base no Pmdarima, biblioteca que implementa o modelo de forma eficiente.

3.5 Arquitetura

Para Goodfellow, Bengio e Courville (2016) a arquitetura de uma RNA define sua estrutura geral, incluindo o número de unidades e a forma como estão conectadas entre si. Cada modelo foi implementado de acordo com as especificações definidas para avaliação, respeitando as particularidades de cada um em relação à estrutura e aos parâmetros ajustados. A seguir, são apresentadas as principais características os diferenciam e justificam seu desempenho nas análises.

Tabela 4 – Estrutura do modelo baseado em LSTM

Camada	Tipo	Neurônios	Função de Ativação	Parâmetros
1	LSTM	32	Tangente Hiperbólica	4.000
2	Densa	32	Tangente Hiperbólica	1.056
3	Densa	1	Linear	33

Fonte: Elaborado pelo autor, 2024.

Tabela 5 – Estrutura do modelo baseado em GRU

Camada	Tipo	Neurônios	Função de Ativação	Parâmetros
1	GRU	32	Tangente Hiperbólica	3.552
2	Densa	32	Tangente Hiperbólica	1.056
3	Densa	1	Linear	33

Fonte: Elaborado pelo autor, 2024.

Tabela 6 – Estrutura do modelo baseado em camadas Bidirecionais

Camada	Tipo	Neurônios	Função de Ativação	Parâmetros
1	Bidirectional	32	Tangente Hiperbólica	9.216
2	Densa	32	Tangente Hiperbólica	2.000
3	Densa	1	Linear	33

Fonte: Elaborado pelo autor, 2024.

3.6 Configuração

Os parâmetros da RNA foram definidos de forma geral e aplicados uniformemente a todos os modelos avaliados. A configuração padrão adotada é descrita detalhadamente na Tabela 7.

Tabela 7 – Configuração dos parâmetros das redes neurais

Parâmetros	Valores
Batch	32
Épocas	500
Otimizador	Nadam ($\eta = 1 \times 10^{-4}$; $\beta_1 = 0,85$; $\beta_2 = 0,989$; $\epsilon = 10^{-6}$)
Fator de decaimento	0,5 em 20 épocas de platô
Função de perda	Huber

Fonte: Elaborado pelo autor, 2024.

3.7 Avaliação

Para avaliar a acurácia dos modelos de previsão, é preciso separar em duas categorias: aprendizado de máquina e estatística. No aprendizado de máquina (supervisionado) é possível validar diretamente na função de perda, que no caso é a Huber, enquanto nos estatísticos o cálculo é inerente à fórmula. A escolha da *loss function* se deve à sua capacidade de lidar com *outliers* e ruídos observados principalmente em intervalos menores

(quinze minutos), Jaiswal e Singh (2023) sugerem que isso a torna mais robusta em relação as outras métricas em séries temporais.

Na comparação final será feita uma combinação de métricas estatísticas que permitem analisar tanto a precisão quanto a robustez das previsões. As métricas escolhidas para essa avaliação são: *Root Mean Squared Error* (RMSE), *Mean Absolute Percentage Error* (MAPE) e o Coeficiente de Determinação (R^2). Cada uma dessas métricas fornece uma perspectiva única sobre a qualidade das previsões e será descrita a seguir.

$$Huber = \begin{cases} \frac{1}{2}a^2, & \text{se } |a| \leq \delta \\ \delta(|a| - \frac{1}{2}\delta), & \text{se } |a| > \delta \end{cases} \quad (3)$$

Na equação acima, a representa a diferença entre o valor real y e o valor predito \hat{y} , e o delta (δ) é um parâmetro que controla o ponto de transição entre os dois regimes de penalização (quadrática e linear). Para erros pequenos, a perda é quadrática, enquanto para erros grandes, a perda se torna linear, o que torna a função robusta a *outliers*.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (4)$$

A principal função dessa métrica é medir a diferença entre os valores observados y e os valores preditos \hat{y} , elevando o erro ao quadrado antes de calcular a média e aplicando a raiz quadrada ao final para que o erro tenha a mesma unidade dos dados originais. Isso significa que o RMSE dá maior peso a erros maiores devido ao efeito do quadrado,

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (5)$$

Média dos erros percentuais absolutos na série, padronizando a análise para que seja possível comparar diferentes escalas ou unidades. Quando a diferença entre y e \hat{y} é dividida por y , obtêm-se o percentual, que é então normalizado pela média. Como o somatório de valores decimais pode ser confuso, alguns autores sugerem multiplicar cada resultado por 100 para facilitar a interpretação.

$$R^2 = 1 - \frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{\sum_{t=1}^n (y_t - \bar{y})^2} \quad (6)$$

O erro R^2 , também chamado de coeficiente de determinação, representa a proporção da variabilidade dos dados que é explicada pelo modelo, variando de 0 a 1, sendo 1 o ajuste perfeito. O y_t define os valores reais, \hat{y}_t os valores preditos e \bar{y} é a média dos valores observados. O numerador contém a soma dos erros ao quadrado entre os valores reais e preditos (erro do modelo), enquanto o denominador é a soma da variabilidade total dos dados

3.8 Materiais e Tecnologias

Para a realização deste trabalho, foram utilizados um processador Intel Core i5-12500H e uma placa de vídeo NVIDIA GeForce RTX 3050 de 4GB com suporte a CUDA 12. Um total de 16GB de memória RAM foram utilizados para armazenar os tensores, parâmetros e janelamento em memória dos dados. O sistema operacional foi o Linux Pop! OS 20.04 e o desenvolvimento foi feito no editor Visual Studio Code, utilizando Python 3.10. As bibliotecas empregadas incluíram Pandas e Numpy para manipulação e computação de dados, e Matplotlib e Seaborn para visualização. Para aprendizado de máquina e RNNs, foram utilizados os frameworks Keras e TensorFlow, e o modelo ARIMA foi implementado com a biblioteca Pmdarima.

4 RESULTADOS E DISCUSSÃO

Os resultados deste capítulo referem-se ao gráfico de preço, volume e transações do Bitcoin, em intervalos de 15 minutos, ao longo de nove meses em 2020 (GMT -3). Foram utilizados dados da Binance para o par BTC/USDT, com uma implementação própria chamada BTools, cobrindo o período de 01/01/2020 a 01/10/2020, representando 271 dias e 26.304 entradas. Esse conjunto de dados foi então armazenado em um arquivo CSV e posteriormente carregado em memória para o treinamento dos modelos supervisionados e ajuste dos modelos estatísticos.

4.1 Validação de completude

O *Dataset* continha 59 valores duplicados que foram complementados por outros 59 entradas faltantes, aos quais foram removidos e preenchidos com o número anterior, respectivamente. Tal fenômeno ocorreu provavelmente devido a manutenções e atualizações da Binance, visto que ao buscar o horário separadamente, a API retornava erro.

4.2 Tratamento e pré processamento de dados

Para o aprendizado supervisionado os dados foram escalonados utilizado o MinMaxScaler, conforme a equação 2 no capítulo anterior. Os conjuntos de treino, teste e validação foram divididos de acordo com a tabela 8 em 60%, 20% e 20% respectivamente. O janelamento utilizado foi de 24 entradas (6 horas) para prever o próximo valor de preço.

No ARIMA foi utilizado o método de auto arima para encontrar os melhores parâmetros p , d e q com base no primeiro janelamento (treino).

Tabela 8 – Dados da divisão em conjuntos

Nome	Início	Fim	Horas	Entradas
Treino	01/01/2020	25/02/2020	1315	5261
Teste	25/02/2020	20/04/2020	1315	5261
Validação	20/04/2020	01/10/2020	3945	15782

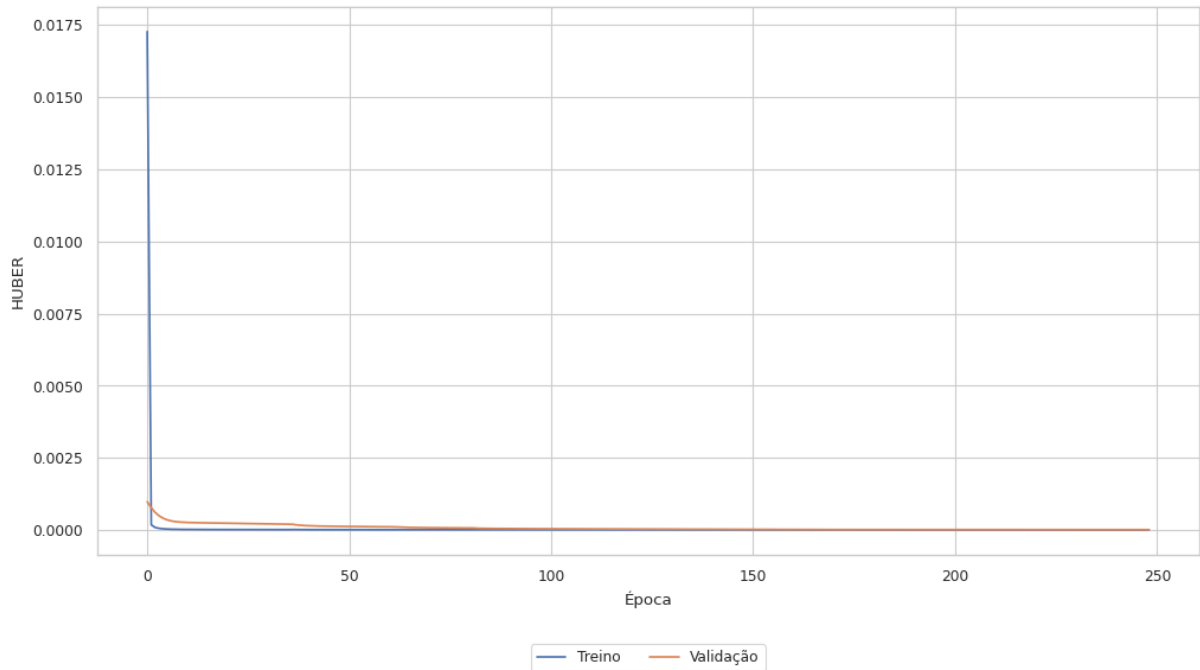
Fonte: Elaborado pelo autor, 2024.

4.3 Treinamento dos modelo

Os modelos supervisionados foram treinados com limite de 500 épocas, limitadas por 30 épocas de paciência, com um *batch size* de 32. Para o *learning rate* a taxa dinâmica foi implementada, com valor inicial de 0.001 e decaimento de 0.5 em platôs de 20 épocas. A função de perda utilizada foi a Huber, que é menos sensível a *outliers*.

O gráfico de convergência foi basicamente o mesmo para todos os modelos, com uma queda acentuada nas primeiras épocas, seguidas por leves degraus gerados pelo decaimento do *learning rate* e estabilização após 100 épocas, como mostra a figura 11.

Figura 11 – Gráfico de convergência



Fonte: Elaborado pelo autor, 2024.

O modelo ARIMA não pode ser treinado, mas, foi ajustado com os parâmetros ($p = 4$, $d = 1$ e $q = 5$) através do conjunto de treino. Os conjuntos de teste e validação foram concatenados e a cada previsão o valor correto era inserido na janela.

4.4 Visualização de resultados

Esta seção apresenta os resultados dos modelos implementados neste estudo, seja através de gráficos ou detalhando os valores dos erros obtidos para cada abordagem. As métricas de erro foram calculadas para avaliar a precisão das previsões.

4.4.1 Long Short-Term Memory

Os resultados do modelo LSTM são apresentados na Figura 12, no qual é possível observar a previsão do modelo em relação ao preço real do Bitcoin. O modelo apresentou as seguintes métricas de erro: raiz do erro quadrático médio (RMSE) de 193.5493, coeficiente de determinação (R^2) de 0,8990 e erro percentual absoluto médio (MAPE) de 0,0145.

Figura 12 – Gráfico de preços reais e previstos - LSTM

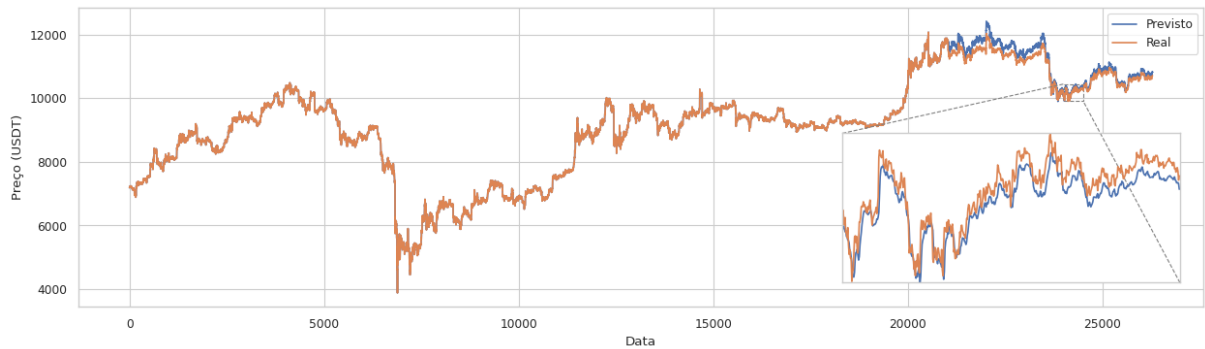


Fonte: Elaborado pelo autor, 2024.

4.4.2 *Bidirectional Long Short-Term Memory*

Os resultados do modelo BiLSTM são apresentados na Figura 13, no qual é possível observar a previsão do modelo em relação ao preço real do Bitcoin. O modelo apresentou as seguintes métricas de erro: raiz do erro quadrático médio (RMSE) de 207,2416, coeficiente de determinação (R^2) de 0,8842 e erro percentual absoluto médio (MAPE) de 0,0153.

Figura 13 – Gráfico de preços reais e previstos - BilSTM



Fonte: Elaborado pelo autor, 2024.

4.4.3 *Gated Recurrent Unit*

As previsões da arquitetura GRU são apresentados na figura 14, dentre os modelos supervisionados implementados, este foi o que apresentou a menor raiz do erro quadrático médio (RMSE) de 86,1828, coeficiente de determinação (R^2) de 0,9799 e erro percentual absoluto médio (MAPE) de 0,0064.

Figura 14 – Gráfico de preços reais e previstos - GRU



Fonte: Elaborado pelo autor, 2024.

4.4.4 *Auto Regressive Integrated Moving Average*

Os resultados das previsões do ARIMA são apresentadas na figura 12, O modelo estatístico se destacou e apresentou a raiz do erro quadrático médio (RMSE) de 28,9699, coeficiente de determinação (R^2) de 0,9989 e erro percentual absoluto médio (MAPE) de 0,0016.

Figura 15 – Gráfico de preços reais e previstos - ARIMA



Fonte: Elaborado pelo autor, 2024.

4.5 Análise de resultados

A tabela 9 apresenta os erros obtidos para cada modelo implementado neste estudo. O modelo ARIMA foi o que apresentou os melhores resultados, fato que pode estar ligado a natureza univariada e janelamento expansivo. No contexto dos modelos supervisionados, o GRU foi o que apresentou o melhor desempenho, seguido pelo LSTM e BiLSTM, respectivamente. Enquanto modelos estatísticos seguem uma abordagem mais simplista, o aprendizado profundo pode ser explorado com outras *Features* além de trocas e volume. No estudo em questão, foram exploradas diversas arquiteturas e hiperparâmetros nos modelos supervisionados, por isso, as taxas de erro foram mais baixas encontradas.

A disparidade entre os modelos ressalta a necessidade de mais estudos para aprimorar as previsões. Grandes volumes de dados exigem mais esforços para otimizar a aplicabilidade de arquiteturas profundas, fato que contribui para o desempenho inferior. Mas, ainda há espaço para alternativas, como a utilização de redes neurais convolucionais ou recorrentes com mecanismos de atenção, além de técnicas de processamento de sinais e processamento de séries temporais.

Tabela 9 – Erros em cada modelo

Posição	Nome	RMSE	R^2	MAPE
1°	ARIMA	28,9699	0,9989	0,0016
2°	GRU	86,1828	0,9799	0,0064
3°	LSTM	193.5493	0,8990	0,0145
4°	BiLSTM	207,2416	0,8842	0,0153

Fonte: Elaborado pelo autor, 2024.

5 CONCLUSÃO

Neste estudo foram exploradas diferentes métodos para a previsão do preço do Bitcoin em intervalos de 15 minutos, comparando métodos estatísticos e RNNs. O *Dataset* continha dados de preço, volume e trocas de janeiro a setembro de 2020, um período de estabilidade no mercado. Desse modo, incluiu uma análise de desempenho de modelos como ARIMA, GRU, LSTM e BiLSTM, aplicados às séries temporais propostas. As arquiteturas supervisionadas e estatísticas foram separadas levando em consideração as limitações de cada uma, como a necessidade de janelamento e a quantidade de *Features*.

O modelo ARIMA, conhecido por sua robustez em séries lineares, obteve os melhores resultados no contexto, superando ligeiramente os modelos de rede neural ao apresentar previsões precisas e consistentes. Porém, o fato das neurais não performarem tão bem quanto seu competidor estatístico não significa que elas não possam ser utilizadas, mas sim que precisam de mais ajustes e engenharia de *Features* para melhorar a previsão.

Como continuidade para este trabalho, sugere-se a investigação de abordagens híbridas que combinem a capacidade preditiva do ARIMA com a sensibilidade temporal das redes neurais. Além disso, é possível expandir os testes para outros períodos e ativos financeiros.

Outras recomendações incluem:

- Realizar ajustes nas configurações das RNNs, como o número de camadas, unidades ocultas, função de ativação e hiperparâmetros
- Identificar arquiteturas ainda mais adaptadas à volatilidade;
- Aplicar a metodologia a outras criptomoedas, intervalos de tempo e ativos financeiros para validar a generalização dos resultados;
- Comparar o desempenho com técnicas mais recentes, considerando diferentes condições de mercado e intervalos temporais.

Portanto, este estudo cumpre os objetivos propostos e contribui para a compreensão do comportamento do mercado de criptomoedas e a aplicação de técnicas de previsão de séries temporais.

REFERÊNCIAS

- ABDOLRASOL, M. G. M. *et al.* Artificial Neural Networks Based Optimization Techniques: A Review. **Electronics**, v. 10, n. 21, 2021. ISSN 2079-9292. DOI: 10.3390/electronics10212689. Disponível em: <https://www.mdpi.com/2079-9292/10/21/2689>.
- AHMED, S.; HASSAN, H.; MABROUK, A. Fundamental Analysis Models in Financial Markets – Review Study. **Procedia Economics and Finance**, v. 30, p. 939–947, 2015. ISSN 2212-5671. DOI: [https://doi.org/10.1016/S2212-5671\(15\)01344-1](https://doi.org/10.1016/S2212-5671(15)01344-1).
- ATSALAKIS, G. S. *et al.* Bitcoin price forecasting with neuro-fuzzy techniques. **European Journal of Operational Research**, v. 276, n. 2, p. 770–780, 2019. ISSN 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2019.01.040>.
- BACHELIER, L. Théorie de la Spéculation. **Annales Scientifiques de L'Ecole Normale Supérieure**, v. 17, p. 21–88, 1900.
- BLACK, F.; SCHOLES, M. The Pricing of Options and Corporate Liabilities. **Journal of Political Economy**, The University of Chicago Press, v. 81, n. 3, p. 637–654, 1973. ISSN 00223808, 1537534X.
- BOOKS, G. **Google Ngram Viewer**. 2024. Acesso em 1 de jul. de 2024. Disponível em: <https://books.google.com/ngrams/>.
- BOX, G. E. P.; JENKINS, G. M. (M. **Time series analysis; forecasting and control**. Holden-Day, 1970.
- CAUX, M. de; BERNARDINI, F.; VITERBO, J. Short-Term Forecasting in Bitcoin Time Series Using LSTM and GRU RNNs. In: ANAIS do VIII Symposium on Knowledge Discovery, Mining and Learning. Porto Alegre, RS, Brasil: SBC, 2020. p. 97–104. DOI: 10.5753/kdmile.2020.11964.
- CHEN, J.; LIN, D.; WU, J. Do cryptocurrency exchanges fake trading volumes? An empirical analysis of wash trading based on data mining. **Physica A: Statistical Mechanics and its Applications**, v. 586, p. 126405, 2022. ISSN 0378-4371. DOI: <https://doi.org/10.1016/j.physa.2021.126405>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378437121006786>.
- CHEN, J. *et al.* Low-Cost and Device-Free Human Activity Recognition Based on Hierarchical Learning Model. **Sensors**, v. 21, p. 2359, mar. 2021. DOI: 10.3390/s21072359.
- CHO, K. *et al.* Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In _____. **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, out. 2014. p. 1724–1734. DOI: 10.3115/v1/D14-1179.
- COURTAULT, J.-M. *et al.* Louis Bachelier on the Centenary of "Théorie de la Spéculation". **Mathematical Finance**, v. 10, p. 339–353, jul. 2000. DOI: 10.1111/1467-9965.00098.

DAS, P.; FAUST, S.; LOSS, J. A Formal Treatment of Deterministic Wallets. In: PROCEEDINGS of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London, United Kingdom: Association for Computing Machinery, 2019. (CCS '19), p. 651–668. ISBN 9781450367479. DOI: 10.1145/3319535.3354236.

EL HIHI, S.; BENGIO, Y. Hierarchical recurrent neural networks for long-term dependencies. In: PROCEEDINGS of the 8th International Conference on Neural Information Processing Systems. Denver, Colorado: MIT Press, 1995. (NIPS'95), p. 493–499.

ESLING, P.; AGON, C. Time-series data mining. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 45, n. 1, 2012. ISSN 0360-0300. DOI: 10.1145/2379776.2379788. Disponível em: <https://doi.org/10.1145/2379776.2379788>.

FANG, T.; SU, Z.; YIN, L. Economic fundamentals or investor perceptions? The role of uncertainty in predicting long-term cryptocurrency volatility. **International Review of Financial Analysis**, v. 71, p. 101566, 2020. ISSN 1057-5219. DOI: <https://doi.org/10.1016/j.irfa.2020.101566>.

FERDIANSYAH, F. *et al.* A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market. **2019 International Conference on Electrical Engineering and Computer Science (ICECOS)**, p. 206–210, out. 2019. DOI: 10.1109/ICECOS47637.2019.8984499.

GERHARDT, T.; SILVEIRA, D. **Métodos de pesquisa**. 1. ed.: UFRGS, jan. 2009. p. 120. ISBN 9788538600718.

GÉRON, A. **Mãos A Obra: Aprendizado De Máquina Com Scikit-Learn, Keras e TensorFlow**: Conceitos, Ferramentas e Técnicas para a Construção de Sistemas Inteligentes. 2. ed.: Alta Books, 2021.

GERS, F.; SCHMIDHUBER, J.; CUMMINS, F. Learning to Forget: Continual Prediction with LSTM. **Neural computation**, v. 12, p. 2451–71, out. 2000. DOI: 10.1162/089976600300015015.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016.

GRAVES, A. **Supervised Sequence Labelling with Recurrent Neural Networks**. Springer, 2012. v. 385, p. 1–131. (Studies in Computational Intelligence). ISBN 978-3-642-24796-5. Disponível em: <http://dx.doi.org/10.1007/978-3-642-24797-2>.

GRAVES, A.; SCHMIDHUBER, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. **Neural Networks**, v. 18, n. 5, p. 602–610, 2005. IJCNN 2005. ISSN 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2005.06.042>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0893608005001206>.

HANCOCK, J. T.; KHOSHGOFTAAR, T. M. Survey on categorical data for neural networks. **Journal of Big Data**, v. 7, n. 1, p. 28, 2020. Disponível em: <https://doi.org/10.1186/s40537-020-00305-w>.

HEBB, D. O. **The Organization of Behavior: A Neuropsychological Theory**. New York: Wiley, 1949.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997.

HUGHES, A. *et al.* Beyond Bitcoin: What blockchain and distributed ledger technologies mean for firms. **Business Horizons**, v. 62, fev. 2019. DOI: 10.1016/j.bushor.2019.01.002.

JAISWAL, R.; SINGH, B. A Comparative Study of Loss Functions for Deep Neural Networks in Time Series Analysis. In: nov. 2023. p. 147–163. ISBN 978-981-99-3480-5. DOI: 10.1007/978-981-99-3481-2_12.

MARKUDOVA, D. *et al.* Preventive maintenance for heterogeneous industrial vehicles with incomplete usage data. **Computers in Industry**, v. 130, p. 103468, set. 2021. DOI: 10.1016/j.compind.2021.103468.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, 1943. ISSN 1522-9602. DOI: 10.1007/BF02478259. Disponível em: <https://doi.org/10.1007/BF02478259>.

MCKINNEY, W. **Python Para Análise de Dados**: Tratamento de Dados com Pandas, Numpy e Ipython. 3rd. São Paulo: Novatec Editora, 2022. ISBN 78-8575228418.

MINSKY, M.; PAPERT, S. **Perceptrons: An Introduction to Computational Geometry**. Cambridge, MA, USA: MIT Press, 1969.

MOLLING, G. *et al.* Cryptocurrency: A Mine of Controversies. **Journal of Information Systems and Technology Management**, v. 17, dez. 2020. DOI: 10.4301/s1807-1775202017010.

NAKAMOTO, S. **Bitcoin**: A Peer-to-Peer Electronic Cash System. 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 15 mar. 2024.

NISON, S. **Japanese Candlestick Charting Techniques**: A Contemporary Guide to the Ancient Investment Techniques of the Far East. Prentice Hall Press, 2001.

PRING, M. **Technical Analysis Explained**: The Successful Investor's Guide to Spotting Investment Trends and Turning Points. McGraw Hill LLC, 2002. ISBN 9780071816199.

QIAN, J. *et al.* Understanding gradient clipping in incremental gradient methods. In: PMLR. INTERNATIONAL Conference on Artificial Intelligence and Statistics. 2021. p. 1504–1512.

RAMALHO, L. **Fluent Python**: Clear, Concise, and Effective Programming. 2. ed.: O'Reilly Media, Incorporated, 2021. ISBN 9781492056348.

ROSENBLATT, F. **The perceptron, a perceiving and recognizing automaton**. Ithaca, New York, 1957.

- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533–536, 1986. Disponível em: <https://doi.org/10.1038/323533a0>.
- SCHUSTER, M.; PALIWAL, K. Bidirectional recurrent neural networks. **IEEE Transactions on Signal Processing**, v. 45, n. 11, p. 2673–2681, 1997. DOI: 10.1109/78.650093.
- SHUMWAY, R. H.; STOFFER, D. S. ARIMA Models. In: *TIME Series Analysis and Its Applications: With R Examples*. Cham: Springer International Publishing, 2017. p. 75–163. ISBN 978-3-319-52452-8. DOI: 10.1007/978-3-319-52452-8_3.
- SIAMI NAMINI, S.; TAVAKOLI. The Performance of LSTM and BiLSTM in Forecasting Time Series. In: p. 3285–3292. DOI: 10.1109/BigData47090.2019.9005997.
- SOUSA, A. *et al.* Cryptocurrency adoption: a systematic literature review and bibliometric analysis. **EuroMed Journal of Business**, v. 17, p. 374–390, mai. 2022. DOI: 10.1108/EMJB-01-2022-0003.
- SRIMAN, B.; GANESH KUMAR, S.; SHAMILI, P. Blockchain Technology: Consensus Protocol Proof of Work and Proof of Stake. In: _____. **Intelligent Computing and Applications**. Singapore: Springer Singapore, 2021. p. 395–406. ISBN 978-981-15-5566-4.
- STEWART, J. **Calculus : early transcendentals**. Belmont, Cal.: Brooks/Cole, Cengage Learning, 2012. ISBN 053849790.
- TRIPATHI, B.; SHARMA, R. K. Modeling Bitcoin Prices using Signal Processing Methods, Bayesian Optimization, and Deep Neural Networks. **Computational Economics**, v. 62, n. 4, p. 1919–1945, 2023. ISSN 1572-9974. DOI: 10.1007/s10614-022-10325-8.
- VUJIČIĆ, D.; JAGODIC, D.; RANĐIĆ, S. **Blockchain technology, bitcoin, and Ethereum: A brief overview**. Mar. 2018. p. 1–6. DOI: 10.1109/INFOTEH.2018.8345547.
- YUAN, Y.; WANG, F.-Y. Blockchain and Cryptocurrencies: Model, Techniques, and Applications. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 48, n. 9, p. 1421–1428, 2018. DOI: 10.1109/TSMC.2018.2854904.
- ZHANG, J.; CAI, K.; WEN, J. A survey of deep learning applications in cryptocurrency. **iScience**, v. 27, n. 1, p. 108509, 2024. ISSN 2589-0042. DOI: <https://doi.org/10.1016/j.isci.2023.108509>.
- ZHU, J.; LI, F.; CHEN, J. A survey of blockchain, artificial intelligence, and edge computing for Web 3.0. **Computer Science Review**, v. 54, p. 100667, 2024. ISSN 1574-0137. DOI: <https://doi.org/10.1016/j.cosrev.2024.100667>.