



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO, DE CIÊNCIAS EXATAS E EDUCAÇÃO
DEPARTAMENTO DE ENG. DE CONTROLE, AUTOMAÇÃO E COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Janderson Hillebrecht

Utilizando inteligência artificial para prever preço de moedas digitais

Blumenau
2024

Janderson Hillebrecht

Utilizando inteligência artificial para prever preço de moedas digitais

Trabalho de Conclusão de Curso de Graduação em Engenharia de Controle e Automação do Centro Tecnológico, de Ciências Exatas e Educação da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Carlos Roberto Moratelli, Dr.

Blumenau

2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Hillebrecht, Janderson

Utilizando inteligência artificial para prever preço de moedas digitais / Janderson Hillebrecht ; orientador, Carlos Roberto Moratelli, 2024.

76 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Blumenau, Graduação em Engenharia de Controle e Automação, Blumenau, 2024.

Inclui referências.

1. Engenharia de Controle e Automação. 2. LSTM. 3. Criptomoedas. 4. Inteligência Artificial. I. Moratelli, Carlos Roberto. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Janderson Hillebrecht

Utilizando inteligência artificial para prever preço de moedas digitais

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Engenheiro de Controle e Automação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação.

Blumenau, 09 de Julho de 2024.

Banca Examinadora:

Prof. Carlos Roberto Moratelli, Dr.
Universidade Federal de Santa Catarina

Prof. Mauri Ferrandin, Dr.
Universidade Federal de Santa Catarina

Prof. Ciro André Pitz, Dr.
Universidade Federal de Santa Catarina

Dedico este trabalho a todas as pessoas que me apoiaram até este momento, com especial gratidão à minha família e amigos, que sempre me incentivaram a seguir em frente.

AGRADECIMENTOS

Aos meus pais, Amarildo e Rosi, por terem me dado todo o suporte necessário para seguir com a faculdade e por sempre me proporcionarem o melhor estudo, mesmo com poucos recursos, meu reconhecimento e agradecimento. Ao meu irmão e irmãs, por me incentivarem a entrar no curso de Engenharia de Controle e Automação e não desistir, meu carinho e admiração. Para minha namorada, que em todos os momentos esteve comigo, apoiando, conversando, incentivando e, por vezes, mantendo tudo em ordem por nós, essa conquista é nossa. A jornada acadêmica, marcada por tantos momentos desafiadores, que resultou neste Trabalho de Conclusão de Curso certamente não teria sido tão gratificante sem o apoio incondicional de todos vocês.

A todos os amigos, que tive o prazer de encontrar ao longo dessa caminhada, que compartilharam comigo muitos momentos que nos colocaram à prova, mas que juntos superamos, desenvolvendo e entregando trabalhos nos quais tenho muito orgulho, os levarei comigo, para muito além de apenas colegas de faculdade, mas como verdadeiros amigos que sempre lembrarei com um sorriso no rosto.

Ao meu orientador, Professor Carlos Roberto Moratelli, que desde o começo me instruiu, orientou e dedicou tempo e esforço em tornar essa uma grande entrega, com quem pude aprender muito e zelar ainda mais pela qualidade e excelência do presente trabalho, minha gratidão e admiração.

Agradeço a Universidade Federal de Santa Catarina campus Blumenau, que possibilitou o meu estudo de forma gratuita e com qualidade, que mostrou todo o meu potencial através de uma educação sólida com professores qualificados, que tanto me ensinaram ao longo desses anos, me inspiraram a entregar o meu melhor, todos foram essenciais para o meu desenvolvimento técnico e prático, refletido hoje no meu ambiente de trabalho, onde aplico com muita honra os aprendizados obtidos através desses profissionais.

Esta entrega é uma inebriante vitória, pela qual tenho muito orgulho e tanto a agradecer.

A persistência é o caminho do êxito. (CHAPLIN, 1997)

RESUMO

Nos últimos anos, as moedas virtuais emergiram como um fenômeno disruptivo nos mercados financeiros globais, transformando-se em uma nova fonte de investimento para diversos *traders*. Paralelamente, a inteligência artificial também tem registrado um crescimento significativo em várias áreas do conhecimento, consolidando-se como uma ferramenta valiosa, especialmente no universo das criptomoedas, através de seus modelos sequenciais. Um desses modelos, o Long Short-Term Memory (LSTM), emprega taxas de esquecimento em sua rede neural para aprimorar a predição de séries temporais. Este trabalho visa prever o preço de fechamento de moedas virtuais para um período futuro, utilizando um sistema parametrizável que opera com múltiplas camadas de entrada e uma única saída. Os dados são obtidos diretamente da API da exchange Binance e o modelo é ajustado para qualquer criptomoeda selecionada. A metodologia inclui uma divisão dos dados em conjuntos de treinamento, validação e teste, e seu desempenho é avaliado utilizando métricas como o coeficiente de determinação (r-squared), desvio padrão e análises gráficas. Através de experimentos práticos, mostrou-se que o modelo é capaz de fornecer previsões precisas sobre tendências de alta e queda nos preços de fechamento para períodos futuros.

Palavras-chave: LSTM; Criptomoedas; Inteligência Artificial.

ABSTRACT

In recent years, virtual currencies have emerged as a disruptive phenomenon in global financial markets, becoming a new investment source for various traders. Simultaneously, artificial intelligence has also seen significant growth across various fields of knowledge, establishing itself as a valuable tool, particularly in the cryptocurrency universe, through its sequential models. One such model, the Long Short-Term Memory (LSTM), uses forgetting rates in its neural network to enhance the prediction of time series. This work aims to predict the closing price of virtual currencies for a future period, using a parametrizable system that operates with multiple input layers and a single output. The data is sourced directly from the Binance exchange API, and the model is adjustable for any selected cryptocurrency. The methodology includes dividing the data into training, validation, and testing sets, and performances are evaluated using metrics such as the coefficient of determination (r-squared), standard deviation, and graphical analyses. Practical experiments have demonstrated that the model is capable of providing accurate predictions on the upward and downward trends in the closing prices for future periods.

Keywords: LSTM; Cryptocurrencies; Artificial Intelligence.

LISTA DE FIGURAS

Figura 1 – Principais criptomoedas do mercado ordenados pela maior capitalização de mercado.	17
Figura 2 – Exemplo de uma vela (Candle).	21
Figura 3 – Exemplo gráfico Candle para o Bitcoin.	22
Figura 4 – Exemplo gráfico de linha	23
Figura 5 – Célula piramidal de um neurônio.	28
Figura 6 – Rede feedforward com uma única camada de neurônios.	29
Figura 7 – Rede feedforward com camada oculta.	30
Figura 8 – Modelo não-linear de um neurônio.	31
Figura 9 – Outro modelo não linear de um neurônio; w_{k0} levando em consideração o <i>bias</i> b_k	32
Figura 10 – Exemplo de rede feedforward, desenhada em dois estilos diferentes. . .	35
Figura 11 – Exemplo de uma rede neural convolucional e suas diferentes camadas. .	35
Figura 12 – Exemplo de módulo de repetição em um RNN padrão com uma única ativação.	38
Figura 13 – Módulo de repetição de uma LSTM.	39
Figura 14 – Parâmetros internos.	39
Figura 15 – Linha horizontal da parte superior do modelo LSTM.	40
Figura 16 – Exemplo de porta regularizadora.	40
Figura 17 – Ativação portão do esquecimento.	41
Figura 18 – Ativação porta de entrada.	41
Figura 19 – Operações para a inclusão ou não no contexto.	42
Figura 20 – Operação sigmoide e saída da ativação LSTM.	43
Figura 21 – Resultado da previsão de ações da Apple utilizando LSTM.	43
Figura 22 – Tabela de relações das previsões.	44
Figura 23 – Tabela de relações das previsões.	45
Figura 24 – Desempenho do modelo LSTM.	46
Figura 25 – Fluxograma da implementação da solução	48
Figura 26 – Close versus Open para o Bitcoin de informações diárias.	51
Figura 27 – Reta de regressão através dos dados de Close e Open.	52
Figura 28 – Reta de regressão através dos dados de Close e volume.	53
Figura 29 – Matriz de correlação para as colunas de entrada.	54
Figura 30 – Exemplo de comparativo entre Close, SMA e EMA.	55
Figura 31 – Histórico do preço de fechamento do Bitcoin.	56
Figura 32 – Histórico do preço de fechamento do Bitcoin normalizados entre 0 e 1. .	57
Figura 33 – Exemplo de comparativo entre Close, SMA e EMA para janelas de períodos de 200 dias.	62

Figura 34 – Resultados finais de treino, validação, teste e dia futuro de Close para janelas de períodos de 200 dias.	62
Figura 35 – Exemplo de comparativo entre Close, SMA e EMA para janelas de períodos de 100 dias.	64
Figura 36 – Resultados finais de treino, validação, teste e dia futuro de Close para janelas de períodos de 100 dias.	65
Figura 37 – Exemplo de comparativo entre Close, SMA e EMA para janelas de períodos de 10 dias.	66
Figura 38 – Resultados finais de treino, validação, teste e dia futuro de Close para janelas de períodos de 10 dias.	67
Figura 39 – Inferências para janelas de períodos de 10 dias e novos hiperparâmetros da Tabela 11.	68
Figura 40 – Inferências para do melhor resultado de close do Bitcoin.	68
Figura 41 – Médias SMA e EMA para janelas de períodos de 10 dias do Ethereum.	70
Figura 42 – Inferências para o Ethereum.	70

LISTA DE TABELAS

Tabela 1 – Dados Exemplificados da Exchange Binance.	50
Tabela 2 – Número de neurônios e taxa de esquecimento para cada camada do modelo LSTM.	61
Tabela 3 – Demais hiperparâmetros do modelo LSTM.	61
Tabela 4 – Parâmetros do modelo LSTM.	61
Tabela 5 – Métricas obtidas na primeira tentativa para janelas de períodos de 200 dias.	63
Tabela 6 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para janelas de períodos de 200 dias.	63
Tabela 7 – Métricas obtidas na segunda tentativa para janelas de períodos de 100 dias.	63
Tabela 8 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para janelas de períodos de 100 dias.	64
Tabela 9 – Métricas obtidas na janelas de períodos de 10 dias.	65
Tabela 10 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para janelas de períodos de 10 dias.	65
Tabela 11 – Número de neurônios e taxa de esquecimento para cada camada do modelo LSTM para janelas de períodos de 10 dias.	65
Tabela 12 – Métricas para janelas de períodos de 10 dias e para hiperparâmetros da Tabela 11.	66
Tabela 13 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para janelas de períodos de 10 dias e para hiperparâmetros da Tabela 11.	66
Tabela 14 – Número de neurônios e taxa de esquecimento para cada camada do modelo LSTM para o melhor resultado de close do Bitcoin.	67
Tabela 15 – Demais hiperparâmetros do modelo LSTM para o melhor resultado de close do Bitcoin.	67
Tabela 16 – Parâmetros do modelo LSTM para o melhor resultado de close do Bitcoin.	67
Tabela 17 – Métricas obtidas para o melhor resultado de close do Bitcoin.	69
Tabela 18 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para o melhor resultado de close do Bitcoin.	69
Tabela 19 – Métricas obtidas para o melhor resultado de close do Ethereum.	71
Tabela 20 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para close do Ethereum.	71

LISTA DE ABREVIATURAS E SIGLAS

API	Interface de Programação de Aplicação
CNN	Rede Neural Convulacional
CNNs	Redes Neurais Convolucionais
EMA	Exponential Moving Average
GRU	Gated Recurrent Units
IA	Inteligência Artificial
LSTM	Long Short-Term Memory
MA	Moving Average
RNA	Rede Neural Artificial
RNN	Rede Neural Recorrente
RNNs	Redes Neurais Recorrentes
SMA	Simple Moving Average

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	16
1.1.1	Objetivo Específicos	16
1.1.2	Estrutura do trabalho	16
2	CRIPTOMOEDAS	17
2.1	TIPOS DE CRIPTOMOEDAS	17
2.1.1	Bitcoin	18
2.1.2	Ethereum	18
2.1.3	Tether	18
2.2	BLOCKCHAIN	19
2.3	ANÁLISE TÉCNICA	19
2.4	INDICADORES TÉCNICOS	20
2.5	CANDLESTICKS	20
2.6	GRÁFICOS DE LINHAS	22
2.7	INDICADORES	22
2.7.1	Escolha de indicadores	24
2.7.1.1	<i>Média Móvel</i>	<i>24</i>
2.7.1.2	<i>Média Móvel Exponencial</i>	<i>24</i>
3	INTELIGÊNCIA ARTIFICIAL	26
3.1	APRENDIZADO DE MÁQUINA	26
3.2	REDES NEURAIS ARTIFICIAIS	27
3.3	PERCEPTRON	29
3.3.1	Tipos de Ativação	32
3.4	TIPOS DE REDES NEURAIS	34
3.4.1	Redes Neurais Feedforward	34
3.4.2	Redes Neurais Convolucionais (CNNs)	35
3.4.3	Redes Neurais Recorrentes (RNNs)	36
3.5	OVERFITTING	36
3.6	UNDERFITTING	37
3.7	DROPOUT	37
3.8	LONG SHORT-TERM MEMORY (LSTM)	38
3.8.1	Funcionamento de uma LSTM simples	39
3.9	TRABALHOS RELACIONADOS À LSTM PARA PREDIÇÃO DE PREÇOS	42
4	IMPLEMENTAÇÃO DE UMA REDE NEURAL PARA PREDIÇÃO DE PREÇOS DE MOEDAS DIGITAIS	47
4.1	PROPOSTA E ARQUITETURA DA SOLUÇÃO	47

4.2	DESCRIÇÃO DA IMPLEMENTAÇÃO	49
4.2.1	Tratamento (pré-processamento) e análise dos dados	49
4.2.1.1	<i>API Binance</i>	<i>49</i>
4.2.1.2	<i>Análise de dados</i>	<i>50</i>
4.2.1.2.1	Regressão linear	50
4.2.1.2.2	Matriz de correlação	52
4.2.1.3	<i>Arquivo JSON</i>	<i>52</i>
4.2.2	Cálculo da SMA e EMA	53
4.2.3	Determinação dos hiperparâmetros	54
4.2.4	Determinação do alvo	55
4.2.5	Divisão dos dados	55
4.2.6	Normalização dos dados	56
4.2.7	Construção do modelo LSTM	57
4.2.8	Treinamento do modelo LSTM	59
4.2.9	Inferência com base nos valores de validação e teste	59
4.2.10	Predição para o dia futuro	60
4.2.11	Cálculo r-squared e desvio padrão	60
5	RESULTADOS	61
5.1	TESTES E RESULTADOS PARA BITCOIN	61
5.2	TESTES E RESULTADOS PARA ETHEREUM	69
6	CONCLUSÃO	72
	REFERÊNCIAS	73

1 INTRODUÇÃO

Nos últimos anos, as moedas virtuais emergiram como um fenômeno disruptivo nos mercados financeiros globais. O surgimento do Bitcoin em 2009 marcou o início de uma nova era financeira, caracterizada pela descentralização, transparência e autonomia. Desde então, uma vasta gama de outras criptomoedas surgiram, cada uma possuindo suas próprias características e propósitos (ULRICH, 2014).

Neste contexto, a volatilidade dos preços das moedas virtuais tem sido uma característica comum, apresentando oportunidades lucrativas para investidores e trazendo desafios significativos para os analistas de mercado. A capacidade de prever os movimentos de preços das criptomoedas tornou-se uma tarefa complexa e crucial para aqueles envolvidos no ecossistema financeiros digital (ULRICH, 2014).

A inteligência artificial é uma tecnologia disruptiva que vem crescendo exponencialmente ao longo dos últimos anos e pode ser aplicada a qualquer área do conhecimento e atividade econômica. De acordo com Vidanya (2024), o mercado de inteligência artificial está crescendo exponencialmente, com uma taxa de crescimento anual de 37,7%, onde dentro do contexto das empresas, indica a adoção de soluções de IA (Inteligência Artificial) para ganhar vantagens competitivas.

Segundo Kang, Lee e Lim (2022), a IA tem emergido como uma ferramenta muito poderosa para auxiliar e enfrentar os desafios de prever os preços das moedas virtuais, utilizando em especial as redes neurais recorrentes através do modelo LSTM (Long Short-Term Memory), onde se destacam no campo da análise e previsão de séries temporais, tornando-se uma escolha natural para explorar padrões nos dados de preços das criptomoedas.

Este trabalho propõe a utilização do modelo de rede neural LSTM para prever os preços de moedas virtuais, buscando fornecer *insights* valiosos para investidores e participantes do mercado, combinando técnicas avançadas de aprendizado de máquina com a análise detalhada de parâmetros e hiper parâmetros para assim, criar um sistema capaz de antecipar os movimentos de preço das criptomoedas com precisão e confiabilidade.

A importância deste estudo reside não apenas na sua relevância para os investidores e mercado financeiros digitais, mas também, busca contribuir para o avanço da aplicação da inteligência artificial na previsão de séries temporais. Ao enfrentar os desafios únicos apresentados pelo mercado de criptomoedas, este trabalho visa abrir novas perspectivas para a utilização de técnicas de IA em diversas outras áreas.

Com isso, ao longo deste trabalho, será apresentado um estudo abrangente sobre a aplicação de redes neurais LSTM na previsão de preços de moedas virtuais, buscando trazer um melhor entendimento sobre a aplicação e resultados desse modelo.

1.1 OBJETIVOS

Implementar o modelo de Rede Neural Recorrente LSTM para a previsão dos preços de criptomoedas, empregando dados provenientes de uma API fornecida pela Binance (BINANCE, 2024), com o objetivo de treinar e avaliar a eficiência do modelo utilizando uma variedade de parâmetros.

1.1.1 Objetivo Específicos

- Empregar uma API confiável para adquirir um conjunto de dados históricos robusto, que será utilizado para o treinamento e teste do modelo.
- Implementar técnicas de pré-processamento de dados, especificamente a normalização, para uniformizar a escala dos dados sem comprometer as variações intrínsecas entre eles.
- Desenvolver um código flexível que permita a configuração dinâmica de múltiplos parâmetros essenciais ao treinamento e à avaliação do modelo, garantindo adaptabilidade e personalização.
- Estabelecer um mecanismo de configuração baseado em arquivos JSON, permitindo a modificação fácil e rápida dos parâmetros do modelo, visando uma interação eficiente e descomplicada.
- Integrar a biblioteca TA-LIB (TA-LIB, 2024) ao processo de treinamento, para enriquecer o modelo com indicadores técnicos reconhecidos, melhorando assim a capacidade de análise e previsão do sistema.
- Otimizar os hiperparâmetros do modelo mediante ajustes finos (*tunning*), com o objetivo de maximizar a precisão e eficácia do modelo.
- Conduzir uma análise detalhada dos resultados, empregando gráficos, o coeficiente de determinação (R^2) e o desvio padrão como ferramentas primárias para uma avaliação de desempenho do modelo.

1.1.2 Estrutura do trabalho

Para garantir uma leitura clara e acessível, este documento é estruturado da seguinte maneira: As seções 2 e 3, compõem o referencial teórico. Nesses capítulos, são apresentados os conhecimentos e informações fundamentais que equipam o leitor com uma base sólida para entender em profundidade o conteúdo do trabalho e as discussões que se seguem. O Capítulo 4 descreve o método adotado no projeto, incluindo tanto o planejamento quanto a implementação prática da solução proposta. O Capítulo 5 traz as considerações finais do autor, destacando as principais descobertas e as contribuições do estudo. Por fim, o Capítulo 6 apresenta a conclusão do projeto, resumindo os resultados alcançados e as melhorias futuras.

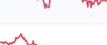
2 CRIPTOMOEDAS

Para prever o comportamento das criptomoedas, é crucial entender seu conceito e origem. Conforme relatado por Mattos, Abouchedid e Silva (2020), em 2008, uma figura misteriosa sob o pseudônimo de Satoshi Nakamoto introduziu ao mundo o Bitcoin, marcando o nascimento das criptomoedas descentralizadas. O propósito era claro: deslocar o controle das moedas, seja estatal ou privado, das instituições financeiras e Bancos Centrais, visando um sistema monetário mais rigoroso e autogerido. A ideologia por trás do Bitcoin reflete as aspirações de grupos anarquistas das décadas de 1980 e 1990, que ve na criptografia uma forma de combater a regulação e vigilância estatais. Assim, o Bitcoin, bem como as demais criptomoedas que vieram a seguir, é transacionado num sistema de pagamentos descentralizado, conhecido como blockchain, e sem a presença de intermediários financeiros.

2.1 TIPOS DE CRIPTOMOEDAS

Após o boom do mundo das moedas virtuais, testemunhou-se um aumento expressivo na criação de diversas moedas digitais com finalidades variadas. Neste cenário diversificado, as criptomoedas que se destacaram por sua valorização e estabilidade ao longo dos anos incluem o Bitcoin, Ethereum e Tether. Observa-se através da Figura 1 as principais moedas virtuais do mercado atual, de acordo com o Market Cap.

Figura 1 – Principais criptomoedas do mercado ordenados pela maior capitalização de mercado.

#	Name	Price	1h %	24h %	7d %	Market Cap	Volume(24h)	Circulating Supply	Last 7 Days
1	 Bitcoin BTC	\$64,234.97	-0.14%	-0.55%	-3.13%	\$1,266,410,712,112	\$15,073,795,400 234,644 BTC	19,715,284 BTC	
2	 Ethereum ETH	\$3,497.49	-0.23%	-0.02%	-2.20%	\$427,659,387,319	\$10,319,439,330 2,949,634 ETH	122,276,217 ETH	
3	 Tether USDT	\$0.9996	+0.01%	-0.02%	-0.01%	\$112,892,022,492	\$34,960,329,705 34,977,577,210 USDT	112,941,691,563 USDT	
4	 BNB BNB	\$86.41	-0.09%	-0.61%	-3.50%	\$86,534,012,454	\$1,440,232,763 2,456,082 BNB	147,565,989 BNB	
5	 Solana SOL	\$134.14	-0.30%	-1.77%	-7.54%	\$61,974,168,703	\$1,472,751,399 10,947,681 SOL	462,001,830 SOL	
6	 USDC USDC	\$1.00	-0.00%	-0.01%	-0.02%	\$32,754,747,734	\$3,547,816,870 3,547,785,055 USDC	32,753,231,418 USDC	
7	 XRP XRP	\$0.4856	-0.35%	-0.06%	-0.33%	\$27,008,486,360	\$623,380,797 1,283,529,625 XRP	55,618,185,850 XRP	
8	 Dogecoin DOGE	\$0.1251	+0.17%	-1.73%	-8.74%	\$18,120,905,921	\$396,225,255 3,166,691,598 DOGE	144,815,776,384 DOGE	
9	 Toncoin TON	\$7.29	+0.84%	-1.85%	-9.13%	\$17,915,702,243	\$210,483,361 28,917,713 TON	2,458,203,288 TON	
10	 Cardano ADA	\$0.3855	-0.88%	-1.05%	-6.47%	\$13,776,901,924	\$269,761,545 699,401,249 ADA	35,740,253,204 ADA	

Fonte: Coinmarketcap (2024)

2.1.1 Bitcoin

Antes do surgimento do Bitcoin em 2008, criado por Satoshi Nakamoto, as transações online sempre dependiam de intermediários terceirizados confiáveis, como PayPal ou Mastercard. Esses intermediários registravam os saldos das contas de seus usuários, debitando e creditando valores conforme as transações. Sem esses intermediários, haveria o risco de um dinheiro digital ser usado mais de uma vez, um problema conhecido como “gasto duplo”. O Bitcoin inovou ao apresentar uma solução para o gasto duplo sem a necessidade de um terceiro confiável, por meio da tecnologia blockchain, inaugurando a primeira rede de pagamento global totalmente descentralizada (ULRICH, 2014).

Ainda segundo Ulrich (2014), diferentemente de sistemas tradicionais de pagamento, as transações na rede Bitcoin são denominadas em bitcoins, uma moeda virtual cujo valor não é baseado em ouro ou decretos governamentais, mas na percepção de valor pelos usuários. O preço do Bitcoin é determinado em um mercado aberto, similar às taxas de câmbio entre moedas convencionais, tornando o Bitcoin uma rede de pagamento descentralizada e uma moeda virtual.

2.1.2 Ethereum

A segunda criptomoeda com maior estabilidade mundial, durante a escrita desse trabalho, é o Ethereum, idealizado por Vitalik Buterin em 2013, o Ethereum amplia as capacidades do Bitcoin ao suportar não apenas transações monetárias, mas também a execução de contratos inteligentes e aplicações descentralizadas (DApps). Segundo Antonopoulos e Wood (2018), esta plataforma global permite que desenvolvedores criem e operem aplicações sem intermediários, utilizando um sistema que oferece mais liberdade e flexibilidade do que blockchains tradicionais. Com o Ethereum, Buterin propôs um ambiente onde variados scripts de programação podem ser executados, permitindo uma gama mais ampla de transações e contratos digitais.

Lançado oficialmente em 30 de julho de 2015, o Ethereum introduziu o conceito de “contratos inteligentes”, programas que executam automaticamente os termos de um contrato quando condições predefinidas são cumpridas. Essa inovação promove a automação e transparência em diversas aplicações, transformando a forma como transações e acordos são realizados digitalmente. A criptomoeda associada, Ether, é usada para gerenciar as operações de computação na rede, garantindo que os recursos sejam alocados eficientemente e incentivando o comportamento correto dentro da plataforma.

2.1.3 Tether

Refere-se a um tipo específico de criptomoeda conhecida como stablecoin (moeda estável). A Tether (USDT) é projetada para manter um valor estável, geralmente indexado a uma moeda fiduciária, como o dólar americano (USD), onde para cada USDT em

circulação é garantido por um dólar americano mantido em reserva pela empresa emissora (TETHER, 2024).

Com isso, busca-se manter uma ponte entre o setor financeiro tradicional e o mercado de criptomoedas, facilitando assim, a realização de transações entre grandes empresas e simplificando a conversão de dólares para os investidores individuais (COINEXT, 2024).

2.2 BLOCKCHAIN

O blockchain funciona como um livro-razão digital composto de blocos de dados encadeados cronologicamente e protegidos por criptografia. Cada bloco contém um número de transações, o horário em que foram registradas e um *link* criptográfico para o bloco anterior (CROSBY *et al.*, 2016). Esta estrutura assegura que uma vez adicionada ao blockchain, a informação se torna praticamente imutável, o que é fundamental para a confiança na rede.

A sua segurança deriva do uso de criptografia de chave pública, que assegura que apenas os proprietários dos dados possam alterá-los, enquanto a imutabilidade é garantida pela natureza descentralizada da tecnologia. A descentralização é realizada por meio de uma rede de nós que validam e registram transações independentemente, uma configuração que protege contra ataques e fraudes (NARAYANAN *et al.*, 2016).

Além de criptomoedas, o blockchain tem uma variedade de aplicações em diferentes setores. Na logística, pode melhorar a rastreabilidade e a eficiência da cadeia de suprimentos. No setor financeiro, facilita transações rápidas e seguras, reduzindo a necessidade de intermediários. Além disso, está sendo explorado em áreas como contratos inteligentes, votação digital e gerenciamento de registros médicos (TAPSCOTT, D.; TAPSCOTT, A., 2016).

O blockchain representa uma mudança de paradigma na maneira de gerenciar informações digitais, oferecendo um novo nível de transparência e segurança. À medida que enfrenta seus desafios, pode vir a revolucionar ainda mais indústrias e práticas globais.

2.3 ANÁLISE TÉCNICA

Para analisar a tendência do mercado, emprega-se a análise técnica, que consiste em um conjunto de métodos e ferramentas destinados a identificar padrões futuros com base no comportamento histórico do mercado. Utilizando indicadores específicos, os investidores podem formular estratégias de negociação, prevendo quais ativos estão propensos a valorizar ou desvalorizar (DEBASTIANI, 2008).

Ainda segundo Debastiani (2008), a análise técnica pode ser categorizada em dois grupos principais:

- Análise técnica empírica: Esta abordagem tradicional se baseia em métodos que identificam padrões repetitivos no mercado. Apesar de não haver uma

explicação racional clara ou uma relação de causa e efeito para tais padrões, eles são considerados confiáveis devido à sua consistência e recorrência ao longo do tempo.

- Análise técnica estatística: Esta abordagem mais moderna utiliza modelos matemáticos e estatísticos que são desenvolvidos a partir de dados históricos do mercado, como preços e volumes de negociação.

2.4 INDICADORES TÉCNICOS

Um indicador técnico é um conjunto de dados obtidos pela aplicação de uma fórmula aos preços de um ativo financeiro, que podem incluir qualquer combinação de preços de abertura, máxima, mínima ou fechamento em um determinado período. Alguns indicadores utilizam somente os preços de fechamento, enquanto outros também incorporam dados de volume e contratos em aberto em suas fórmulas. Os preços são inseridos na fórmula, resultando em um dado específico (LEMOS, 2016).

Indicadores técnicos na análise de preços fornecem perspectivas variadas, desde fórmulas simples, como médias móveis, até complexas, como o estocástico.

2.5 CANDLESTICKS

Os gráficos de *candlestick*, cujo termo em inglês se traduz como “candelabro” e cuja forma lembra uma vela, são amplamente reconhecidos como ferramentas essenciais para a análise e previsão de movimentos futuros de preços no mercado financeiro. A proficiência no uso do método *candlestick* é considerada uma habilidade valiosa para *traders* que desejam aprimorar suas capacidades de negociação e proteger seu capital em um ambiente de mercado volátil. Além das sessões de negociação, a análise técnica proporcionada pelos *candlesticks* é uma maneira eficaz de captar as emoções do mercado. Os *traders* profissionais analisam padrões específicos nos gráficos para antecipar futuras variações de preço. Alguns desses padrões são Hanging Man (Enforcado), Dark Cloud Cover (Nuvem Negra) e Hummer (Martelo), que são traduções de expressões japonesas e refletem o sentimento predominante no mercado (HOMMA, 2020).

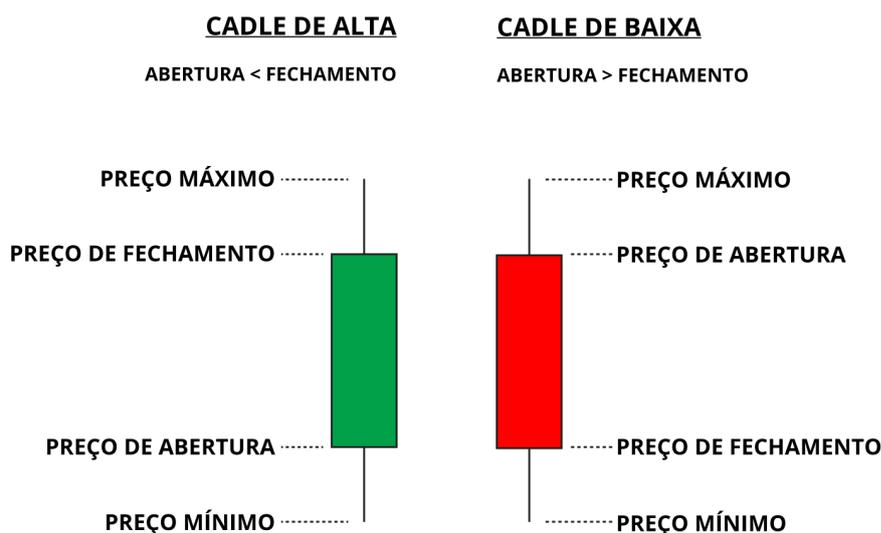
O *candle* (vela), possui quatro preços em cada período, sendo eles:

- Preço de abertura (*Open*): que representa o preço pelo qual foi fechado o primeiro negócio do dia;
- Preço de fechamento (*Close*): que representa o preço pelo qual foi fechado o último negócio do dia;
- Preço mínimo (*Low*): que representa o preço mais baixo pelo qual um negócio foi fechado naquele dia;

- Preço máximo (*High*): que representa o preço mais alto pelo qual um negócio foi fechado naquele dia.

Para compreender a técnica de *Candlestick*, é essencial entender a formação dos corpos dos *Candles*. O corpo é formado pela diferença entre o preço de abertura e de fechamento. Um corpo vermelho indica um fechamento abaixo da abertura, sugerindo domínio vendedor, enquanto um corpo verde indica um fechamento acima da abertura, mostrando domínio comprador. Além do corpo, os *Candles* possuem sombras ou pavios, que são linhas estendidas que representam os preços máximos e mínimos alcançados durante o período. Através da Figura 2, observa-se um exemplo de *Candle*. Assim, é possível construir um gráfico para análise de mercado. A Figura 3 apresenta uma visão ampliada de um gráfico diário dos preços do Bitcoin obtido da plataforma Binance (HOMMA, 2020).

Figura 2 – Exemplo de uma vela (Candle).



Fonte: Do autor (2024).

Observe que o Figura 3 apresenta, em sua maioria, uma trajetória ascendente que indica uma forte valorização do preço do ativo, refletindo uma percepção positiva do mercado. Esta tendência é conhecida como Linha de Tendência de Alta (LTA). Além disso, é possível notar dentro deste período os momentos de maior e menor volatilidade, bem como a intensidade das forças compradoras e vendedoras, os *gaps*, entre outros aspectos relevantes.

Também é possível analisar os gráficos utilizando diversos períodos temporais, onde cada *candle* representa tal período, entre os quais os mais comuns e preferidos pelos investidores incluem:

Figura 3 – Exemplo gráfico Candle para o Bitcoin.



Fonte: Binance (2024)

- Diário: Cada candle mostra o preço de abertura e fechamento do dia;
- Semanal: Cada candle ilustra a abertura e o fechamento ao longo de uma semana;
- Anual: Cada candle ilustra a abertura e o fechamento ao longo de um ano;
- Intraday: Cada candle abrange um período específico, com os mais utilizados sendo de 15, 30, 60 e 120 minutos.

2.6 GRÁFICOS DE LINHAS

Segundo Achelis (2000), o gráfico de linhas é a forma mais básica de representação gráfica com uma linha contínua demonstrando o preço de fechamento do ativo para cada dia. Geralmente, as datas estão posicionadas na parte inferior, enquanto os valores dos preços são mostrados nas laterais do gráfico. A principal vantagem desse tipo de gráfico reside na sua clareza e simplicidade, proporcionando uma compreensão imediata e eficaz das flutuações de preço de um ativo. Usualmente, os gráficos de linhas utilizam os preços de fechamento para traçar os dados. Observa-se uma representação de um gráfico de linha através da Figura 4.

2.7 INDICADORES

No âmbito dos indicadores matemáticos, conforme mencionado por Lemos (2016), identificam-se dois tipos fundamentais de indicadores: os antecedentes e os atrasados. Os indicadores antecedentes antecipam-se aos movimentos de preços, conferindo-lhes uma

Figura 4 – Exemplo gráfico de linha



Fonte: Achelis (2000)

capacidade preditiva, ao passo que os indicadores atrasados funcionam como ferramentas de confirmação, pois acompanham as oscilações dos preços. Em geral, emprega-se o indicador antecedente na ausência de uma tendência claramente definida pelo mercado, enquanto os indicadores atrasados revelam-se mais eficazes durante os períodos em que as tendências estão estabelecidas.

Existem também duas categorias de estruturação de indicadores: aqueles que operam dentro de uma faixa limitada e os que não se limitam a tal faixa. Os primeiros são conhecidos como osciladores, representando o tipo mais comum e amplamente utilizado de indicadores. Os osciladores, por exemplo, variam entre 0 e 100, indicando períodos em que o ativo pode estar sobrecomprado (próximo a 100) ou subvalorizado (próximo a zero). Por outro lado, indicadores não restritos também geram sinais de compra e venda, além de mostrar força ou fraqueza, variando na metodologia empregada.

Os indicadores utilizados na análise técnica fornecem uma fonte valiosa de informações adicionais. Eles facilitam a identificação de impulso, tendências, volatilidade e diversos outros aspectos dos ativos, auxiliando na análise técnica de tendências. É relevante salientar que, embora alguns analistas empreguem um único indicador para produzir sinais de compra e venda, a eficácia desse instrumento é maximizada quando utilizado conjuntamente com a análise dos movimentos de preços, padrões gráficos e outros indicadores.

Os indicadores são extremamente úteis em análises, funcionando como alertas para operações de compra e venda em ações e moedas virtuais. De acordo com Lemos (2016), caso o momentum esteja em declínio, isto pode sinalizar a necessidade de atenção para uma possível ruptura de suporte. Alternativamente, se estiver ocorrendo uma divergência positiva considerável, o indicador pode atuar como um alerta para monitorar um potencial rompimento de resistência.

2.7.1 Escolha de indicadores

Atualmente, há uma vasta gama de indicadores desenvolvidos, fundamentados em diversas teorias, permitindo até, que qualquer indivíduo elabore seu próprio indicador. Conseqüentemente, a seleção de um indicador adequado pode se revelar uma tarefa difícil e desafiadora. Lemos (2016) argumenta que, os indicadores que mais se destacam são aqueles criados há mais tempo, os quais têm demonstrado resistência aos desafios impostos pela evolução tecnológica, tonando-os robustos.

Nesse contexto, é prudente exercer cautela na escolha de um indicador. Ainda segundo Lemos (2016), esforços para empregar mais de cinco indicadores frequentemente se mostram inviáveis. Portanto, é recomendável concentrar-se em no máximo dois ou três indicadores.

2.7.1.1 Média Móvel

Em análises financeiras, a Média Móvel (*Moving Average - MA*) é uma ferramenta estatística que calcula o valor médio dos dados ao longo de um período específico. Por exemplo, uma MA de 5 dias reflete o preço médio dos últimos cinco dias, enquanto uma MA de 20 dias fornece a média dos preços nos últimos vinte dias. Segundo Elder (1993), quando se conecta os valores diários das MAs, cria-se então a linha de média móvel. Sendo assim, a fórmula para a Média Móvel Simples (*Simple Moving Average - SMA*) é dada por

$$SMA(n) = \frac{P(n) + P(n-1) + \dots + P(n-N+1)}{N} \quad (1)$$

Onde P representa o preço de fechamento e N o número de dias na MA, escolhido pelo analista ou *trader*.

O valor da SMA é influenciado por dois fatores principais: os preços que estão sendo medidos e o tamanho da janela de tempo, existindo três tipos principais de médias móveis: Simples, Exponencial e Ponderada. As MAs tradicionais, como a SMA, eram preferidas nos primeiros dias da análise de dados devido à sua facilidade de cálculo. Pioneiros como Donchian e Hurst utilizavam esses métodos em épocas pré-computadorizadas.

2.7.1.2 Média Móvel Exponencial

Uma Média Móvel Exponencial (*Exponential Moving Average - EMA*) é uma ferramenta de acompanhamento de tendências melhor do que uma SMA. Ela fornece maior

peso aos dados mais recentes e responde a mudanças mais rapidamente do que uma MA simples. (ELDER, 1993)

Ainda, de acordo com Elder (1993), a EMA se comporta de acordo com

$$EMA(n) = K \cdot P(n) + (1 - K) \cdot EMA(n - 1) \quad (2)$$

onde n é o índice inteiro que representa o dia atual, e

$$K = \frac{2}{N + 1} \quad (3)$$

sendo N o número de dias na EMA escolhida pelo trader.

A EMA apresenta duas vantagens significativas em relação à SMA. Primeiramente, a EMA confere maior relevância ao preço mais recente, refletindo de forma mais acurada o sentimento atual do mercado. Por exemplo, em uma EMA de 10 dias, o dia mais recente responde por 18% do valor total da média, diferentemente da SMA simples, onde cada dia contribui igualmente. Em segundo lugar, ao contrário da SMA simples, que elimina abruptamente dados antigos, a EMA permite que esses dados desvançam-se gradualmente, oferecendo uma visão mais suave e contínua das tendências de preço ao longo do tempo (ELDER, 1993).

3 INTELIGÊNCIA ARTIFICIAL

De acordo com Russell e Norvig (2010), a inteligência artificial é um dos campos mais recentes que avançam na ciência e engenharia. Seu desenvolvimento começou após a Segunda Guerra Mundial e, atualmente, abrange uma ampla variedade de subcampos. Esses subcampos variam desde áreas mais gerais, como aprendizado e percepção, até áreas mais específicas, como jogos, resolução de teoremas matemáticos, escrita de poesias, direção autônoma de veículos, diagnóstico de doenças, entre outros exemplos. Com isso, a IA possibilita o rápido desenvolvimento de novas tecnologias em grandes passos.

3.1 APRENDIZADO DE MÁQUINA

Dentro da inteligência artificial, destaca-se o aprendizado de máquina, onde, segundo Halbouni *et al.* (2022), são utilizados algoritmos capazes de aprender e se adaptar com base em dados. Esses algoritmos são projetados para gerar resultados a partir do conhecimento adquirido dos dados e exemplos fornecidos.

Segundo Saravanan e Sujatha (2018), existem quatro tipos principais de aprendizado de máquina:

- **Aprendizado Supervisionado:** Este método utiliza o conhecimento adquirido de dados anteriores e presentes, com a ajuda de rótulos, para prever eventos futuros. O processo começa com o treinamento de um conjunto de dados, onde o algoritmo de aprendizado de máquina desenvolve uma função inferida para prever os valores de saída. O sistema compara os resultados obtidos com os resultados reais e ajusta o modelo conforme necessário para melhorar a precisão das previsões.
- **Aprendizado Não Supervisionado:** Utilizado quando os dados de treinamento não são classificados nem rotulados. O sistema analisa os dados para deduzir padrões ocultos sem referência a uma saída correta específica. Ele explora os dados, faz observações e identifica padrões subjacentes, permitindo a descoberta de novas informações e relações dentro dos dados.
- **Aprendizado Semi-Supervisionado:** Combina técnicas de aprendizado supervisionado e não supervisionado, utilizando tanto dados rotulados quanto não rotulados no processo de treinamento. Geralmente, uma pequena quantidade de dados rotulados é usada em conjunto com uma grande quantidade de dados não rotulados. Essa abordagem pode aumentar a precisão do modelo e é particularmente útil quando a obtenção de dados rotulados requer recursos especializados.
- **Aprendizado por Reforço:** Envolve a interação com o ambiente por meio de ações, buscando maximizar recompensas e minimizar erros. Caracteriza-se pela

busca por tentativa e erro e recompensas atrasadas. Esse método permite que sistemas e programas de software identifiquem o comportamento ideal em um contexto específico, aprimorando o desempenho com o tempo.

Essas categorias de aprendizado de máquina formam a base para o desenvolvimento de sistemas inteligentes capazes de melhorar continuamente seu desempenho com base na experiência.

3.2 REDES NEURAIS ARTIFICIAIS

As redes neurais foram concebidas com o objetivo de imitar o funcionamento dos neurônios presentes no cérebro humano. O cérebro humano pode ser considerado um sistema de processamento de informação extremamente complexo, não linear e paralelo. Devido a essas características, ele possui uma habilidade notável para organizar seus componentes estruturais, os neurônios, de modo a executar diversas funções, como reconhecimento de padrões, percepção sensorial e controle motor, de maneira significativamente mais rápida e eficiente do que um computador digital convencional.

Os neurônios, unidades básicas do sistema nervoso, são compostos essencialmente por três partes principais: axônios, dendritos e o corpo celular, como é possível observar a Figura 5. Os axônios atuam como as vias de transmissão dos sinais elétricos, permitindo a comunicação entre diferentes neurônios. Os dendritos funcionam como áreas receptivas, recebendo os sinais de outros neurônios. O corpo celular, ou soma, é a região onde se originam os potenciais de ação, ou impulsos nervosos, que são fundamentais para a transmissão da informação neural (HAYKIN, 2009).

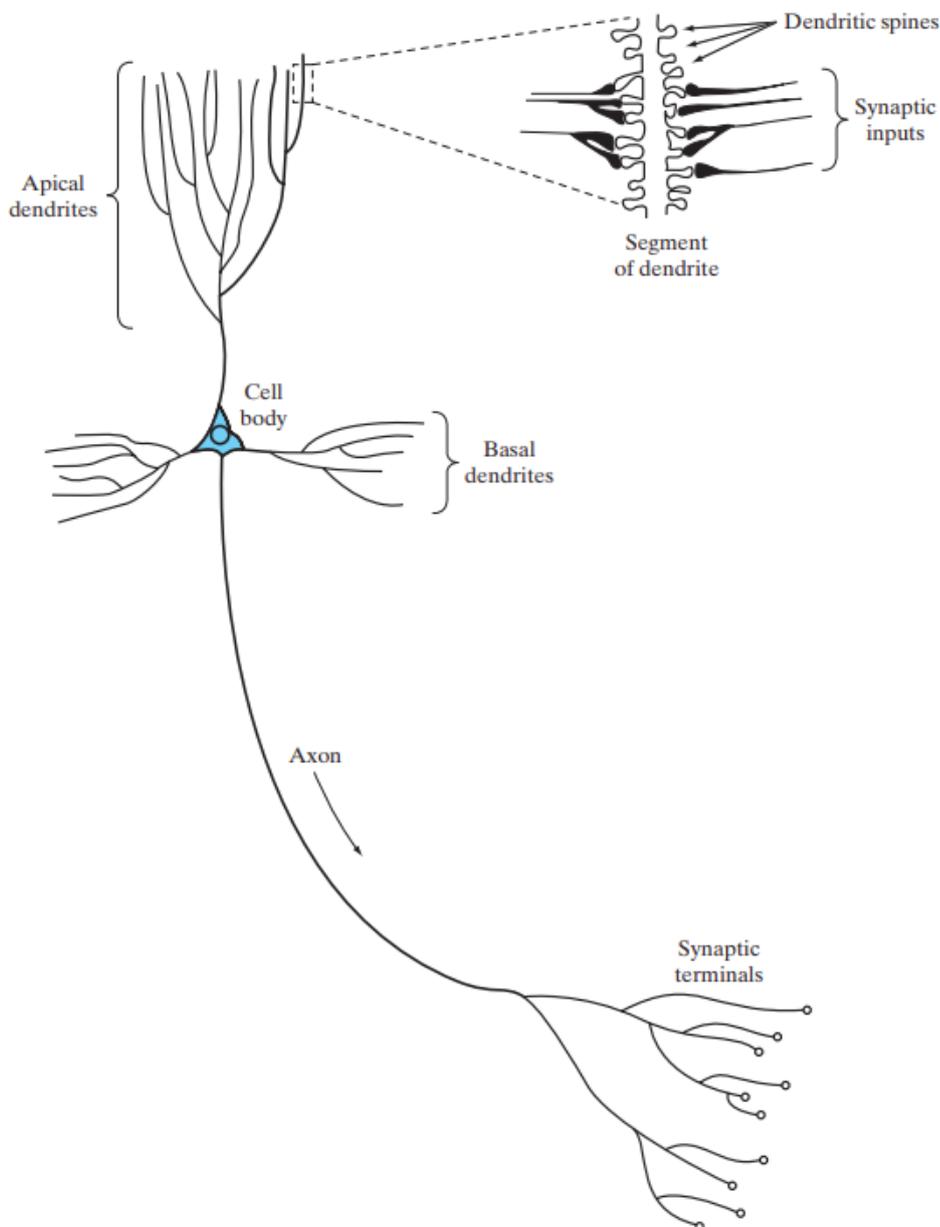
Esta organização estrutural e funcional dos neurônios permite que o cérebro humano execute tarefas complexas com uma eficiência incomparável, servindo de inspiração para o desenvolvimento de redes neurais artificiais que buscam replicar essas capacidades em sistemas computacionais.

Pesquisadores continuam a desenvolver redes neurais artificiais que possam replicar, de maneira semelhante, a capacidade do cérebro humano de resolver problemas com maestria. A estrutura básica de uma RNA (Rede Neural Artificial) pode ser descrita em termos de três tipos de camadas: camada de entrada, camadas ocultas e camada de saída.

Haykin (2009) explica que em uma rede neural em camadas, os neurônios são organizados na forma de camadas. Em sua forma mais simples, tem-se uma camada de entrada de nós fonte que se projetam diretamente em uma camada de saída de neurônios (nós de computação), mas não vice-versa. Em outras palavras, esta rede é estritamente do tipo feedforward. Apresenta-se um exemplo através da Figura 6 para o caso de quatro nós tanto na camada de entrada quanto na camada de saída.

As camadas ocultas ou também chamados de neurônios ocultos, referem-se ao fato de que essa parte da rede neural não será vista diretamente a partir da entrada ou saída

Figura 5 – Célula piramidal de um neurônio.

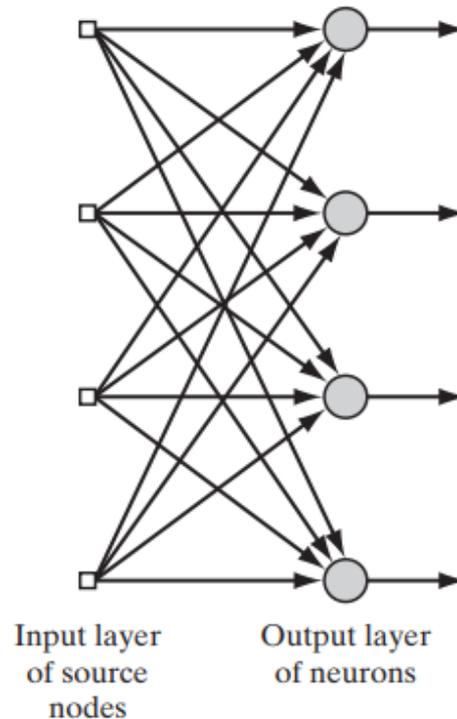


Fonte: Haykin (2009)

da rede. A função dos neurônios ocultos é intervir entre a entrada externa e a saída da rede de uma maneira útil. Ao adicionar uma ou mais camadas ocultas, a rede é capacitada a extrair estatísticas de ordem superior de sua entrada. Em um sentido bastante amplo, a rede adquire uma perspectiva global apesar de sua conectividade local, devido ao conjunto extra de conexões sinápticas e à dimensão extra de interações neurais (CHURCHLAND; SEJNOWSKI, 1992).

Ainda, segundo Haykin (2009), os nós da camada de entrada da rede fornecem elementos respectivos do padrão de ativação (vetor de entrada), que constituem os sinais

Figura 6 – Rede feedforward com uma única camada de neurônios.



Fonte: Haykin (2009)

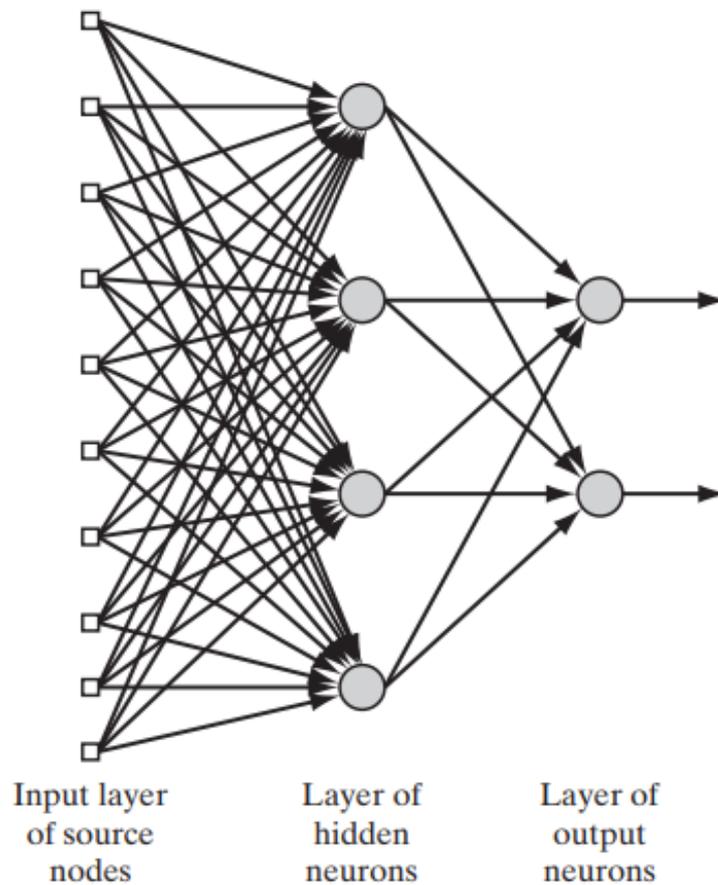
de entrada aplicados aos neurônios na segunda camada (ou seja, primeira camada oculta). Os sinais de saída da segunda camada são usados como entradas para a terceira camada, e assim, repetindo-se ao longo da rede. Tipicamente, os neurônios em cada camada da rede têm como suas entradas os sinais de saída da camada precedente apenas. O conjunto de sinais de saída dos neurônios na camada de saída (final) da rede constitui a resposta geral da rede ao padrão de ativação fornecido pelos nós fonte na camada de entrada (primeira). Observa-se através da Figura 7 um exemplo de uma rede neural feedforward multicamada para o caso de uma única camada oculta.

3.3 PERCEPTRON

Desenvolvido nas décadas de 1950 e 1960 pelo cientista Frank Rosenblatt, e inspirado por trabalhos anteriores de Warren McCulloch e Walter Pitts (NIELSEN, 2018), os perceptrons são considerados pilares fundamentais das redes neurais. Como pioneiro na área, eles geraram grande empolgação sobre as possibilidades da inteligência artificial, sendo a primeira estrutura de neurônios artificiais a serem criadas (AGGARWAL, 2018).

De acordo com Haykin (2009), um neurônio artificial é composto por três conjuntos básicos que formam o modelo neural. O primeiro conjunto é composto de sinapses ou elos de conexão, cada uma caracterizada por um peso ou força própria. Um sinal x_j na

Figura 7 – Rede feedforward com camada oculta.



Fonte: Haykin (2009)

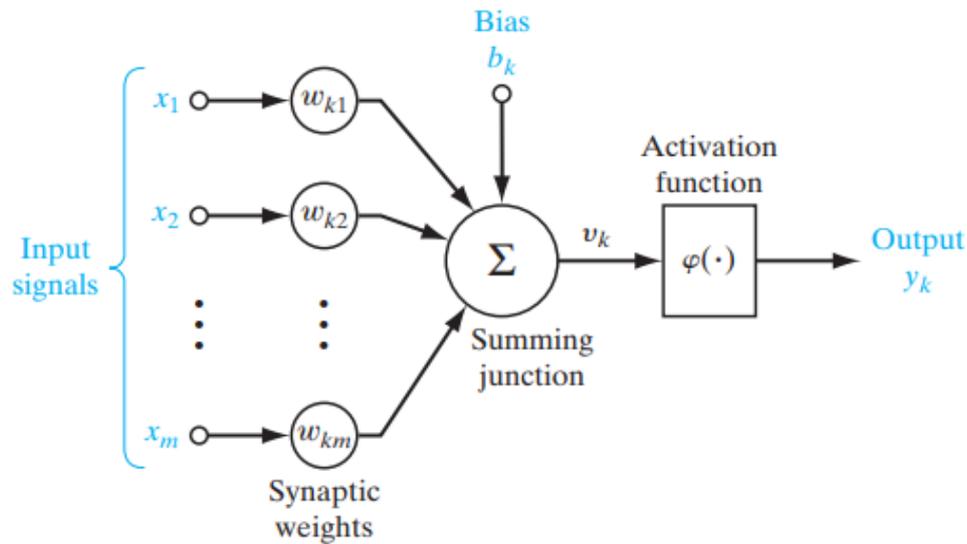
entrada da sinapse j , conectada ao neurônio k , é multiplicado pelo peso sináptico w_{kj} . É importante notar a maneira como são escritos os índices do peso sináptico w_{kj} : o primeiro índice refere-se ao neurônio em questão e o segundo ao terminal de entrada da sinapse à qual o peso se refere. O peso sináptico de um neurônio artificial pode estar em um intervalo que inclui valores negativos bem como positivos.

O segundo conjunto é o somador, utilizado para somar os sinais de entrada ponderados pelas respectivas sinapses do neurônio. As operações realizadas constituem um combinador linear.

O terceiro e último conjunto é a função de ativação, responsável por restringir a amplitude da saída e alterar a sensibilidade de um neurônio. A função de ativação também é referida como função restritiva, pois limita o intervalo permissível de amplitude do sinal de saída a um valor finito. Tipicamente, o intervalo normalizado da amplitude de saída de um neurônio é escrito como o intervalo unitário fechado $[0,1]$ ou, alternativamente, $[-1,1]$.

A Figura 8 ilustra o modelo de um neurônio, que serve como base para o projeto de redes neurais artificiais.

Figura 8 – Modelo não-linear de um neurônio.



Fonte: Haykin (2009)

Além disso, como ilustrado na Figura 8, há um *bias* externo representado por b_k , que tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo ser positivo ou negativo, respectivamente. Com base nisso, formula-se um novo modelo neural, onde matematicamente, um neurônio k é descrito pelo seguinte par de equações:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (4)$$

e

$$y_k = \varphi(u_k + b_k) \quad (5)$$

sendo x_1, x_2, \dots, x_m sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{km}$ pesos sinápticos do neurônio k ; u_k saída do combinador linear devido aos sinais de entrada; b_k sendo o *bias*; $\varphi(\cdot)$ função de ativação; e y_k sinal de saída do neurônio. Ao utilizar o *bias* b_k aplica-se uma transformação afim à saída u_k do combinador linear no modelo, sendo representado por

$$v_k = u_k + b_k \quad (6)$$

O *bias* b_k é um parâmetro externo do neurônio artificial k . Pode-se considerar a sua presença através da equação (5). Alternativamente, consegue-se reformular as equações, gerando um segundo modelo, conforme descrito a seguir:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (7)$$

$$y_k = \varphi(v_k) \quad (8)$$

Adiciona-se uma nova sinapse na equação (7):

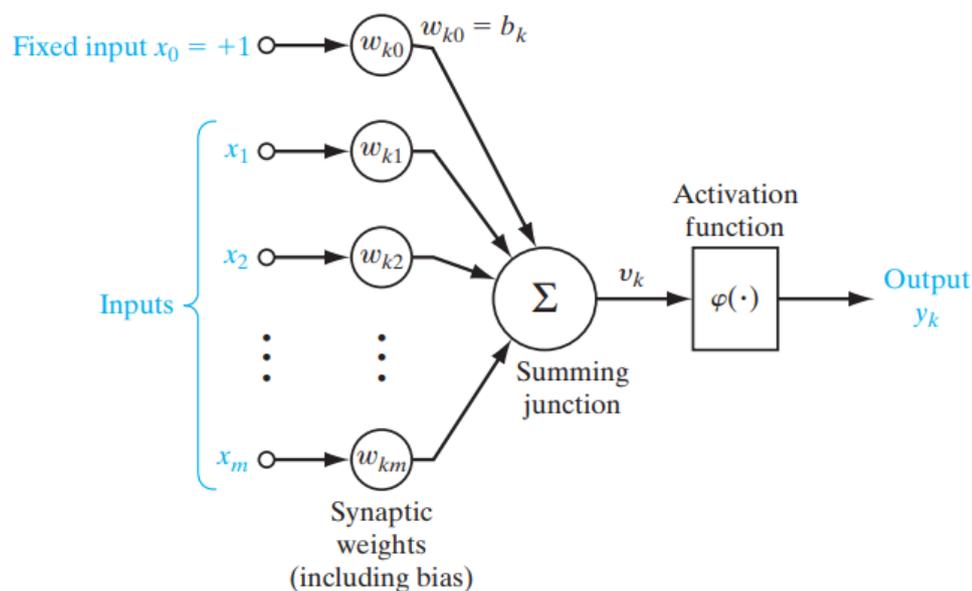
$$x_0 = +1 \quad (9)$$

e o seu peso:

$$w_{k0} = b_k \quad (10)$$

Com isso, reformula-se o modelo neural k , e então, o efeito *bias* é levado em conta de duas maneiras, adicionando-se um novo sinal de entrada fixo em $+1$ e, adicionando-se um novo peso sináptico igual ao *bias* b_k . Representa-se o novo modelo através da Figura 9.

Figura 9 – Outro modelo não linear de um neurônio; w_{k0} levando em consideração o *bias* b_k .



Fonte: Haykin (2009)

Embora os modelos (Figuras 8 e 9) sejam diferentes em sua aparência, os mesmos são matematicamente equivalentes.

3.3.1 Tipos de Ativação

Ainda segundo Haykin (2009), a função de ativação, representada por $\varphi(v)$, define a saída de um neurônio em termos do campo local induzido v . Pode-se identificar dois tipos básicos de funções de ativação:

1. Função de Limiar

Para essa função, tem-se a seguinte expressão:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (11)$$

E então, a saída do neurônio k que emprega essa função é expressa como:

$$y(k) = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases} \quad (12)$$

Sendo v_k o campo local induzido do neurônio, ou seja,

$$v_k = \sum_{j=1}^m w_{kj}x_j + b_k \quad (13)$$

Esse neurônio é conhecido como McCulloch-Pitts, em reconhecimento ao trabalho pioneiro realizado por McCulloch e Pitts (1943), onde a saída de um neurônio assume o valor de 1, se o campo local induzido do neurônio não for negativo, e 0 caso contrário.

2. Função Sigmoide

A função sigmoide é amplamente utilizada como função de ativação em redes neurais artificiais devido à sua capacidade de equilibrar comportamentos linear e não linear de maneira eficaz. Um exemplo de função sigmoide comum é a função logística, definida pela equação:

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (14)$$

onde a representa o parâmetro de inclinação da função sigmoide. Ao ajustar o parâmetro a , diferentes inclinações podem ser obtidas. A função sigmoide mapeia os valores de entrada para um intervalo contínuo entre 0 e 1. Além disso, é importante notar que a função sigmoide é diferenciável, ao contrário da função de limiar.

3. Função sinal

As funções de ativação normalmente variam de 0 a 1, mas em alguns casos é desejável que a faixa de valores varie de -1 a +1. Nesses casos, a função de ativação assume uma forma anti-simétrica em relação à origem, ou seja, é uma função ímpar do campo local induzido. Especificamente, a função de limiar pode ser redefinida como:

$$\varphi(v) = \begin{cases} 1 & \text{se } v > 0 \\ 0 & \text{se } v = 0 \\ -1 & \text{se } v < 0 \end{cases} \quad (15)$$

4. Função sigmoide

Para a forma correspondente de uma função sigmoide, a função tangente hiperbólica pode ser utilizada, definida pela equação:

$$\varphi(v) = \tanh(v) \quad (16)$$

Sendo assim, permitir que uma função de ativação sigmoide assumira valores negativos, conforme descrito pela equação (16) pode trazer vantagens analíticas e práticas significativas.

5. Função ReLU

A função de ativação ReLU (*Rectified Linear Unit*) define a saída como o valor de entrada se positivo ou zero se negativo, sendo simples e eficiente. É amplamente usada em redes neurais profundas devido à sua facilidade de cálculo e capacidade de promover esparsidade nas ativações. No entanto, pode sofrer com o problema do gradiente morrendo, onde neurônios ficam inativos e param de aprender. Variações como Leaky ReLU e PReLU foram desenvolvidas para mitigar esse problema.

3.4 TIPOS DE REDES NEURAIAS

Existem 3 tipos básicos de redes neurais artificiais: redes neurais feedforward, convolucionais e recorrentes. Cada uma dessas arquiteturas possui características específicas que as tornam adequadas para diferentes tipos de problemas.

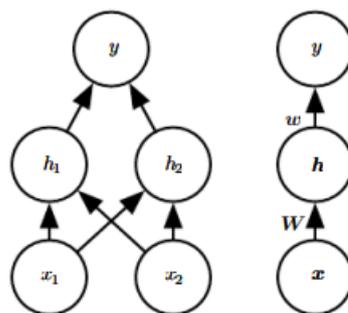
3.4.1 Redes Neurais Feedforward

As redes neurais feedforward, também conhecidas como perceptrons multicamadas, são um dos modelos mais fundamentais de aprendizado profundo. Elas são compostas por uma série de camadas de nós (neurônios), onde cada camada está totalmente conectada à seguinte, mas não há conexões de retorno. Isso significa que a informação flui em uma única direção, da entrada para a saída, sem ciclos ou *feedbacks* (GOODFELLOW; BENGIO, Yoshua; COURVILLE, 2016).

A principal função de uma rede feedforward é aproximar uma função alvo f^* . Em termos de classificação, a rede mapeia uma entrada x para uma saída y . Durante o treinamento, a rede aprende os valores dos parâmetros w , que são ajustados para minimizar

a diferença entre a saída da rede e a saída desejada (GOODFELLOW; BENGIO, Yoshua; COURVILLE, 2016). Esses parâmetros são os pesos das conexões entre os neurônios e os vieses que são ajustados através de métodos de otimização. Apresenta-se um exemplo de duas redes feedforward através da Figura 10.

Figura 10 – Exemplo de rede feedforward, desenhada em dois estilos diferentes.

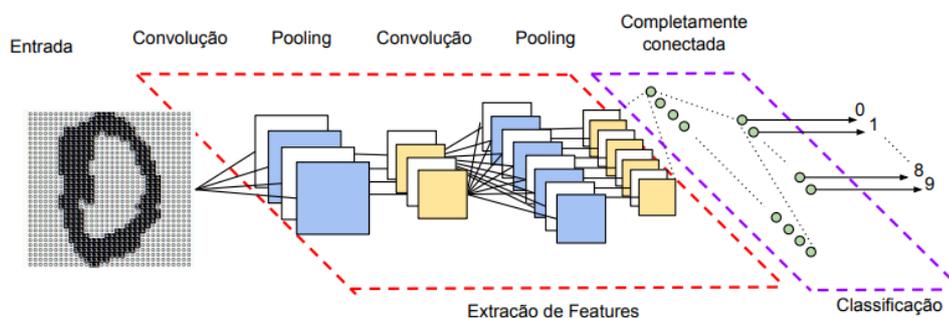


Fonte: Goodfellow, Yoshua Bengio e Courville (2016)

3.4.2 Redes Neurais Convolucionais (CNNs)

As redes neurais convolucionais (CNNs) são uma classe especial de redes feedforward que se destacam no processamento de dados com uma grade topológica, como imagens. As CNNs utilizam camadas convolucionais, que aplicam filtros (ou kernels) a sub-regiões da entrada, permitindo a detecção de padrões locais, como bordas, texturas e formas (LECUN; BENGIO, Y.; HINTON, 2015). A Figura 11 ilustra uma CNN. Este tipo de rede vem sendo amplamente utilizado, principalmente em classificações, detecção e reconhecimento em imagens e vídeos.

Figura 11 – Exemplo de uma rede neural convolucional e suas diferentes camadas.



Fonte: Ana V. Aline C. (2016)

Ainda segundo Lecun *et al.* (1998), cada filtro convolucional é treinado para detectar uma característica específica da imagem, e a combinação dessas características

nas camadas seguintes permite a construção de representações hierárquicas cada vez mais complexas da imagem. As CNNs também fazem uso de camadas de *pooling*, que reduzem a dimensionalidade das representações intermediárias e aumentam a invariância a pequenas translações da imagem.

3.4.3 Redes Neurais Recorrentes (RNNs)

As redes neurais recorrentes (RNNs) são uma classe de redes neurais que permitem conexões de *feedback*. Esses *loops* possibilitam o armazenamento de informações no modelo enquanto ele processa novas entradas. Dessa forma, cria-se uma espécie de memória, tornando-as ideais para tarefas de processamento de dados que envolvem séries temporais (CARVALHO *et al.*, 2018).

Os dados que podem ser inseridos para a aplicação deste modelo podem assumir diversas formas, como palavras, frases ou dados de séries temporais em geral, onde os componentes sequenciais se inter-relacionam com base em regras complexas de semântica e sintaxe. Os neurônios das redes neurais recorrentes (RNNs) são organizados em camadas de entrada, saída e ocultas, conforme discutido em seções anteriores. Nesse modelo, a camada de entrada recebe as informações a serem processadas, enquanto a camada de saída fornece o resultado. Dessa forma, o processamento, análise e previsão dos dados ocorrem na camada oculta.

Na camada oculta, especificamente, os dados de entrada são passados sequencialmente, onde existe um fluxo de trabalho em *loop* ou recorrente. Com isso, a camada oculta pode lembrar e usar entradas anteriores para previsões futuras, funcionando como um componente de memória de curto prazo. Em seguida, utiliza-se a entrada atual e a memória armazenada para prever a próxima sequência.

No entanto, as RNNs tradicionais enfrentam problemas como o desaparecimento e o estouro do gradiente durante o treinamento. Para mitigar esses problemas, variantes como *Long Short-Term Memory* (LSTM) e *Gated Recurrent Units* (GRU) foram desenvolvidas. Essas variantes incluem mecanismos de controle adicionais que regulam o fluxo de informação dentro da rede, permitindo que as RNNs aprendam dependências de longo prazo de forma mais eficaz (HOCHREITER; SCHMIDHUBER, 1997).

3.5 OVERFITTING

Segundo Ying (2019), *overfitting* é um problema comum no aprendizado de máquina supervisionado onde um modelo se ajusta muito bem aos dados de treinamento, mas falha em generalizar para novos dados não vistos, resultando em baixa performance nos conjuntos de teste. Esse fenômeno ocorre devido a várias razões, sendo os principais:

1. Ruído nos Dados de Treinamento: Quando obtém-se um conjunto de treinamento pequeno, não representativo ou com muito ruído. Sendo assim, o modelo tende

a aprender os dados de ruídos em vez dos padrões gerais;

2. Complexidade do Modelo: Modelos com muitas hipóteses (ou parâmetros) podem se ajustar demais aos dados de treinamento, capturando variações que não são representativas dos dados reais;
3. Procedimentos de Múltiplas Comparações: Em algoritmos de indução, a comparação de múltiplos itens baseados em pontuações pode levar à seleção de itens que não melhoram a precisão da classificação.

Diante disso, existem estratégias que são propostas pra reduzir o *overfitting*, sendo:

1. Parada Antecipada: Quando para-se o treinamento antes que o modelo comece a se ajustar aos ruídos;
2. Redução da Rede: Utiliza-se técnicas como poda para remover dados irrelevantes ou menos significativos;
3. Expansão dos Dados de Treinamento: Aumenta-se a quantidade e a qualidade dos dados de treinamento;
4. Regularização: Adiciona-se termos de penalidades à função custo para limitar o impacto de características menos úteis.

3.6 UNDERFITTING

Underfitting ocorre quando um modelo de *machine learning* é demasiadamente simples para capturar as relações subjacentes nos dados de treinamento, resultando em um desempenho pobre tanto nos dados de treinamento quanto nos dados de teste. Esse problema geralmente surge quando o modelo possui pouca capacidade de modelagem, como quando se utiliza um modelo linear para dados não lineares ou quando há poucos parâmetros disponíveis para aprender as complexidades dos dados. Em outras palavras, o modelo não consegue se ajustar adequadamente aos dados, falhando em capturar os padrões presentes (GOODFELLOW; BENGIO, Yoshua; COURVILLE, 2016).

3.7 DROPOUT

Dropout é uma técnica popular e eficaz para prevenir o *overfitting* em redes neurais. A ideia principal do *dropout* é remover aleatoriamente unidades (neurônios) e suas conexões durante o treinamento. Isso ajuda a evitar que os neurônios se adaptem excessivamente, promovendo uma melhor generalização do modelo (CLARK; NIBLETT, 1989).

Segundo o autor Ying (2019), o processo de *dropout* segue uma ordem específica, começando com a remoção aleatória de neurônios. Durante cada interação de treinamento, metade dos neurônios da camada oculta são removidos aleatoriamente ou determina-se a porcentagem da remoção dos mesmos. Em seguida, a rede reduzida é treinada utilizando

métodos de otimização, como a descida do gradiente estocástico. Após o término do treinamento, os neurônios removidos são restaurados, e o processo é repetido até que um conjunto ideal de parâmetros seja encontrado.

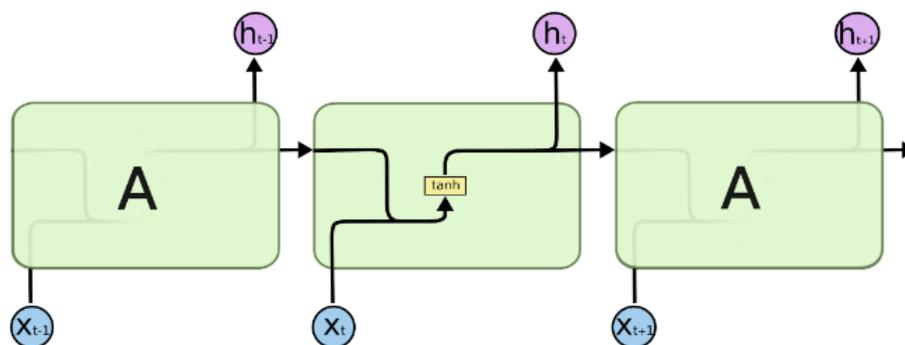
Ao remover temporariamente unidades das redes neurais, o *dropout* simula o efeito de mediação das previsões de várias redes diferentes, reduzindo assim o *overfitting*. Além disso, ao reduzir a quantidade de dados processados, o *dropout* diminui significativamente a quantidade de cálculos, tornando-se uma escolha eficaz para redes grandes ou complexas que exigem muitos dados de treinamento.

3.8 LONG SHORT-TERM MEMORY (LSTM)

As redes de memória de longo e curto prazo, mais conhecidas pela sigla em inglês LSTM (Long Short-Term Memory), foram desenvolvidas por Hochreiter e Schmidhuber em 1997. Essas redes representam uma categoria específica de RNN, projetadas para captar dependências de longa duração.

Segundo Olah (2015), as redes neurais recorrentes possuem a forma de uma cadeia de módulos repetidos de rede neural. Em RNNs padrão, este módulo de repetição terá uma estrutura muito simples como uma única ativação de tanh (Equação (16)). Observa-se um exemplo através da Figura 12, onde x_t é a entrada e h_t sendo a saída.

Figura 12 – Exemplo de módulo de repetição em um RNN padrão com uma única ativação.



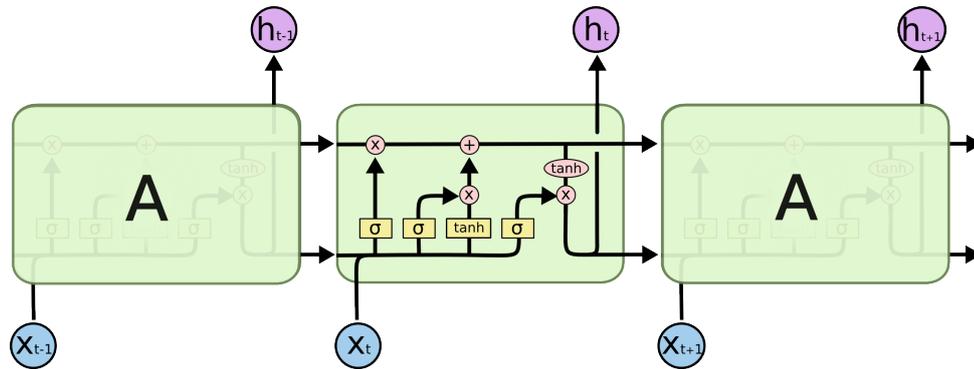
Fonte: Olah (2015)

A rede LSTM se assemelha à estrutura da Figura 12, porém, em vez de existir uma única ativação de rede neural, existem quatro. Observa-se sua estrutura através da Figura 13.

Onde os parâmetros internos do modelo LSTM são apresentados através da Figura 14.

Ainda, de acordo com Olah (2015), cada linha do diagrama na Figura 13 carrega um vetor inteiro, desde a saída de um nó até as entradas de outros. Os círculos em rosa representam operações pontuais, como a adição de vetores, enquanto os retângulos

Figura 13 – Módulo de repetição de uma LSTM.



Fonte: Olah (2015)

Figura 14 – Parâmetros internos.



Fonte: Olah (2015)

amarelos indicam ativações de redes neurais treinadas. A fusão das linhas representa concatenações, enquanto as bifurcações indicam que o conteúdo está sendo copiado e enviado para diferentes destinos.

A ideia principal da LSTM é o estado da célula (Figura 15) que atua como uma esteira transportadora percorrendo toda a cadeia, com apenas algumas pequenas interações lineares. Sendo assim, a informação flui mais facilmente sem ser alterada.

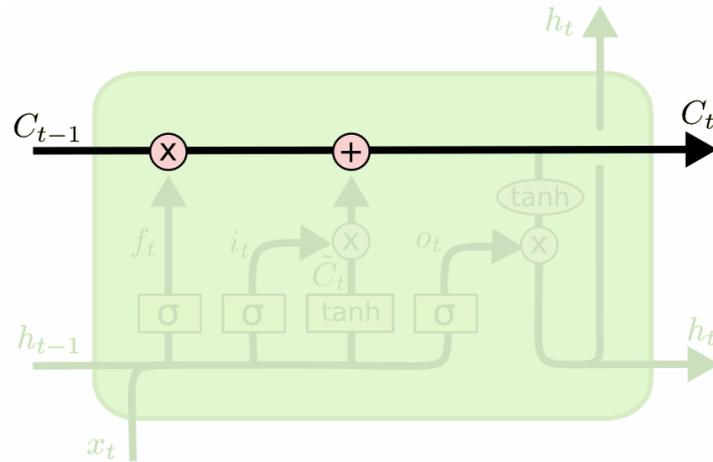
A LSTM pode remover ou adicionar informações ao estado da célula, utilizando estruturas regularizadoras conhecidas como portas ou portões, compostos por uma ativação de rede neural sigmoide e uma operação de multiplicação pontual. Observa-se uma porta de exemplo através da Figura 16.

A função da ativação sigmoide é produzir números entre zero e um, indicando a proporção da informação que será transmitida: um valor de zero significa que nada é transmitido, e um valor de um significa que toda a informação é transmitida. Uma LSTM contém três portas, que servem para proteger e regular o estado da célula.

3.8.1 Funcionamento de uma LSTM simples

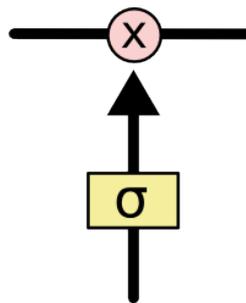
Na etapa inicial, decide-se quais informações serão descartadas do estado da célula, ou seja, se as informações do passado entraram ou não na linha principal também chamada de contexto. Esta decisão é tomada através de uma ativação de rede neural sigmoide

Figura 15 – Linha horizontal da parte superior do modelo LSTM.



Fonte: Olah (2015)

Figura 16 – Exemplo de porta regularizadora.



Fonte: Olah (2015)

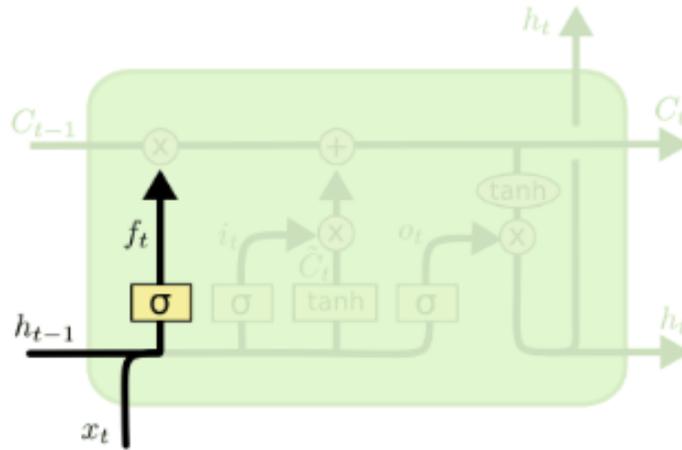
conhecida como ativação do portão do esquecimento, nele, olha-se para h_{t-1} e x_t e então, gera-se um número entre 0 e 1 para cada número no estado da célula C_{t-1} . Nota-se essa função de ativação através da Equação (17) e sua representação na Figura 17.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{17}$$

No próximo passo, decide-se quais novas informações serão armazenadas no estado da célula. Nessa parte, divide-se em duas etapas, sendo uma função de ativação sigmoide chamada de ativação de porta de entrada, onde define-se quais valores serão atualizados e, a segunda, uma ativação de rede neural tanh, onde cria-se um vetor de novos valores candidatos (\tilde{C}_t). Observa-se essa etapa através da Figura 18 e pelas Equações (18) e (19).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{18}$$

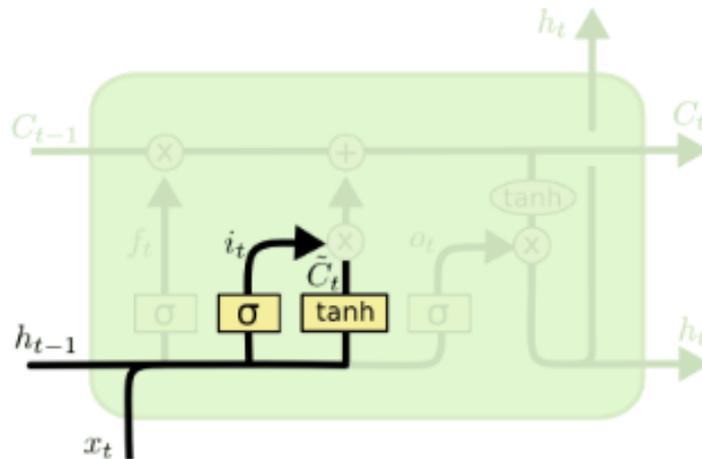
Figura 17 – Ativação portão do esquecimento.



Fonte: Olah (2015)

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{19}$$

Figura 18 – Ativação porta de entrada.

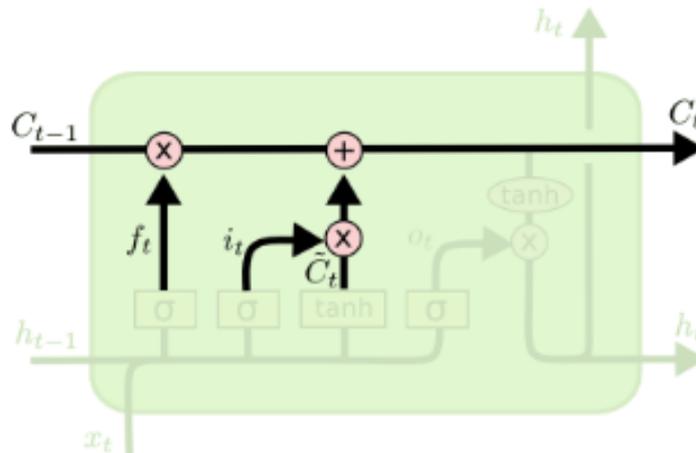


Fonte: Olah (2015)

Logo após, atualiza-se o antigo estado da célula C_{t-1} para o novo estado C_t . Multiplica-se o estado antigo por f_t , esquecendo as informações que antes foram definidas para serem esquecidas, e então, adiciona-se $i_t \cdot \tilde{C}_t$. Com isso, obtém-se os novos valores para o contexto. Observa-se essas representações através da Equação (20) e Figura 19.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{20}$$

Figura 19 – Operações para a inclusão ou não no contexto.



Fonte: Olah (2015)

Por fim, determina-se a saída do modelo, que deriva-se do estado atual da célula, porém em uma forma filtrada. Inicialmente, aplica-se uma ativação sigmoide que seleciona quais partes do estado da célula serão utilizadas na saída. Em seguida, aplica-se o estado da célula através da função \tanh , para atuar como um saturador ajustando os intervalos entre -1 e 1. Com isso, multiplica-se o resultado pela saída da ativação sigmoide, garantindo que apenas as partes selecionadas sejam efetivamente produzidas. Representa-se essa etapa através das Equações (21), (22) e Figura 20.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (21)$$

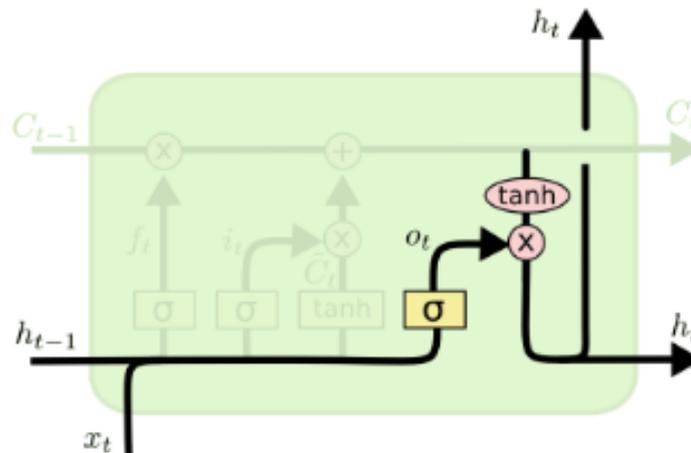
$$h_t = o_t \cdot \tanh(C_t) \quad (22)$$

3.9 TRABALHOS RELACIONADOS À LSTM PARA PREDIÇÃO DE PREÇOS

Nos últimos anos, diversos estudos foram desenvolvidos utilizando redes LSTM para predição de preços. Um exemplo é o trabalho de Fauzan *et al.* (2023), que trata da previsão de movimentos no mercado de ações utilizando a LSTM.

O primeiro passo adotado por Fauzan *et al.* (2023) foi a coleta e o carregamento dos dados de empresas como Apple, Amazon, Google e Microsoft, obtidos do site Yahoo Finance, abrangendo o período de 2013 a 2022. Em seguida, os autores realizaram uma análise inicial por meio do cálculo da MA para melhor compreensão dos dados. Posteriormente, analisaram o retorno diário médio das ações utilizando histogramas e calcularam o risco associado a cada ação.

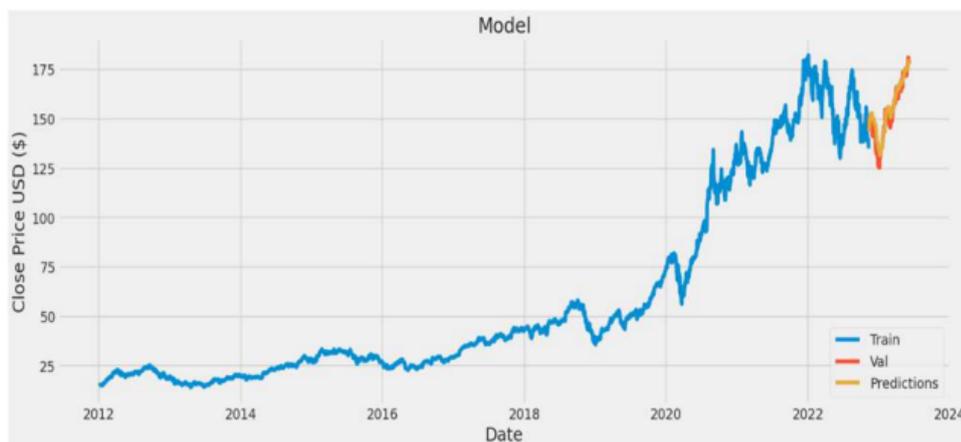
Figura 20 – Operação sigmoide e saída da ativação LSTM.



Fonte: Olah (2015)

Após a análise preliminar dos dados, os autores aplicaram a rede LSTM para prever os preços das ações de cada empresa. Os resultados específicos para a empresa Apple utilizando a rede LSTM podem ser visualizados através da Figura 21.

Figura 21 – Resultado da previsão de ações da Apple utilizando LSTM.

Fonte: Fauzan *et al.* (2023)

Em seguida, foram obtidos os resultados apresentados na Figura 22, que mostram a relação entre o preço de fechamento de um determinado dia e o preço previsto.

Outro exemplo a ser citado é a previsão da estabilidade do mercado de ações das empresas de vacinas COVID-19 utilizando modelos LSTM e de séries temporais dos autores Khan e Baloch (2023), onde os mesmos, buscam identificar as principais marcas no mercado global de vacinas contra COVID-19 e prever a estabilidade dessas ações, ajudando investidores a tomar decisões informadas.

Figura 22 – Tabela de relações das predições.

Company	Date	Close	Predictions	Calculation	
				Accuracy	Error
Apple	2022-11-10	146.8699	143.5350	97.73%	2.27%
	2023-06-02	180.9499	177.6458	98.176%	1.824%
Amazon	2022-11-11	149.6999	144.0239	96.21%	3.79%
	2023-06-15	127.1100	133.8545	94.639%	5.307%
Microsoft	2022-11-18	97.8000	99.5942	98.165%	1.835%
	2023-06-15	125.7900	126.8816	99.133%	0.867%
Google	2022-11-18	97.8000	99.5942	98.165%	1.835%
	2023-06-15	125.7900	126.8816	99.133%	0.867%

Fonte: Fauzan *et al.* (2023)

Os autores utilizaram modelos de séries temporais e redes neurais, incluindo ARIMA, Prophet e LSTM, para prever os preços das ações de três grandes fabricantes de vacinas: BioNTech, Moderna e AstraZeneca. Os dados históricos de vacinação foram obtidos de fontes abertas, e os dados de ações foram extraídos do Yahoo Finance, cobrindo o período de 2019 a 2023, sendo esses dados uma tabela dividida em High, Low, Open, Close, Volume e Adj. Close.

Para a análise dos dados, foram feitos testes de estacionalidade através do método KPSS (Kwiatkowski-Phillips-Schmidt-Shin) para verificar a estacionalidade dos dados. Com isso, os resultados indicaram que os dados não eram estacionários, exigindo uma transformação logarítmica para estabilizar a variância. Após isso, os dados foram divididos em dois conjuntos: 80% dos dados para treinamento e outros 20% para teste, garantindo que as instâncias mais recentes fossem utilizadas para validação.

Com os dados preparados, os autores Khan e Baloch (2023) aplicaram os mesmos aos modelos ARIMA, Prophet e LSTM e assim, obtiveram a previsão. Nota-se através da Figura 23 a predição dos dados para a AstraZeneca utilizando LSTM.

Por fim, os autores utilizaram as métricas MSE, RMSE e MAPE para a avaliação dos modelos, e com isso, obtiveram os resultados para a LSTM que são apresentados através da Figura 24.

Figura 23 – Tabela de relações das predições.



Fonte: Khan e Baloch (2023)

Sendo o MSE (*Mean Squared Error*) uma métrica que calcula a média dos quadrados das diferenças entre os valores reais e os valores previstos por um modelo. Matematicamente, é expresso como a soma das diferenças quadráticas entre y_i (valor real) e \hat{y}_i (valor previsto), dividida pelo número total de observações n . Sendo sua fórmula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (23)$$

Já o MAE (*Mean Absolute Error*) mede a média das diferenças absolutas entre os valores reais e os valores previstos. Essa métrica é menos sensível a *outliers*, pois trata todos os erros de forma linear. Sua equação é exemplificada por:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (24)$$

E MAPE (*Mean Absolute Percentage Error*) sendo a média das diferenças absolutas entre os valores reais e os valores previstos, expressas como uma porcentagem dos valores reais. Onde sua fórmula é apresentada como:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (25)$$

Figura 24 – Desempenho do modelo LSTM.

Metrics	BioNtech	Mordena	AstraZeneca
MSE	26.7	164.4	1.40
RMSE	5.16	12.8	1.18
MAPE	416.9	830.9	94.1

Fonte: Khan e Baloch (2023)

4 IMPLEMENTAÇÃO DE UMA REDE NEURAL PARA PREDIÇÃO DE PREÇOS DE MOEDAS DIGITAIS

Neste capítulo, apresenta-se a implementação de uma rede neural desenvolvida para a predição dos preços das criptomoedas Bitcoin e Ethereum. A análise detalha os métodos utilizados para estimar os preços dessas moedas digitais, com o objetivo de avaliar a eficácia das técnicas avançadas de inteligência artificial aplicadas ao setor financeiro digital.

4.1 PROPOSTA E ARQUITETURA DA SOLUÇÃO

Como discutido em seções anteriores deste trabalho, as moedas virtuais têm apresentado uma trajetória de crescimento constante ao longo dos anos. Este crescimento contínuo tem estimulado o desenvolvimento e a implementação de novas ferramentas tecnológicas destinadas a otimizar as operações no mercado financeiro. Uma solução promissora para alcançar vantagens competitivas na compra e venda de moedas virtuais é o emprego de técnicas avançadas de inteligência artificial, especialmente por meio do uso de RNNs (Redes Neurais Recorrentes) com o modelo LSTM

Com o intuito de prever o preço de moedas virtuais, é necessário inicialmente um conjunto de dados robusto e confiável, sendo eles, “low”, “high”, “open”, “close”, “EMA” e “SMA”. Esses dados, servem como entrada para a rede neural LSTM, onde a quantidade e qualidade dos dados históricos são cruciais para um treinamento eficaz e, por consequência, para uma predição precisa. Seguindo uma prática padrão em aprendizado de máquina, divide-se o conjunto de dados em três partes: treinamento, validação e teste. O conjunto de treinamento, sendo o maior, é utilizado para o modelo aprender e ajustar seus parâmetros. O de validação serve para avaliar o desempenho do modelo e fazer os ajustes necessários, permitindo a identificação e correção de *overfitting*. Por fim, o conjunto de teste é usado para comparar os resultados do modelo após o treinamento e a validação, oferecendo uma avaliação imparcial da sua performance, livre de qualquer viés.

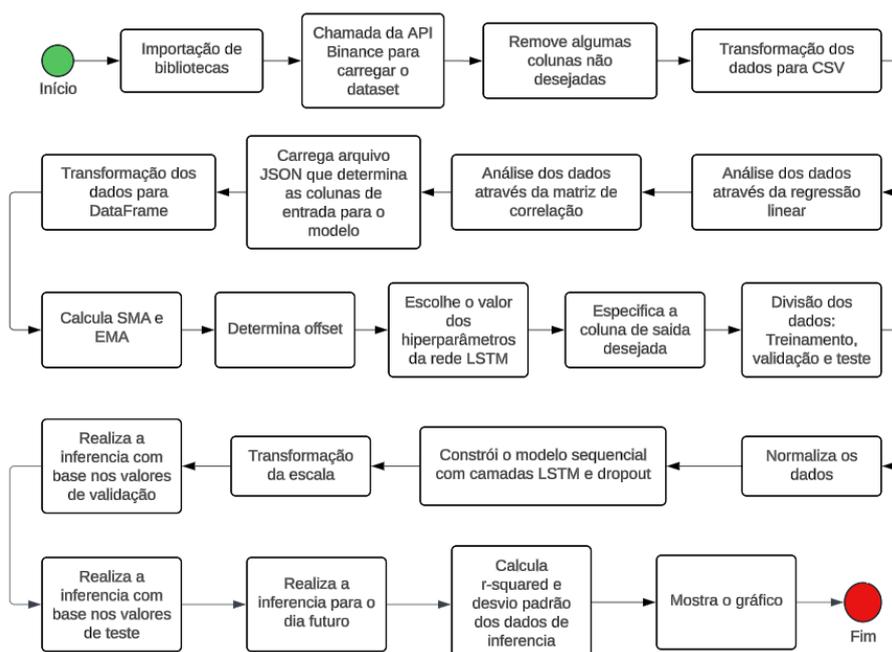
Essa divisão estratégica dos dados é fundamental para minimizar o risco de *overfitting* e garantir que o modelo possa generalizar a partir de dados não vistos anteriormente, o que é crucial para a aplicação prática de modelos de aprendizado de máquina.

Ainda, antes de partir para o modelo, necessita-se realizar a normalização dos dados, visando melhorar o desempenho do treinamento e evitar problemas nos processos de otimização, tais como os gradientes que desaparecem ou explodem. Além disso, a normalização previne o viés nas variáveis, onde os dados podem ser influenciados por valores muito altos ou muito baixos. Um ponto especialmente importante para redes LSTM, que utilizam funções de ativação sigmoide e tanh, é que a normalização assegura que as entradas estejam adequadas às expectativas do modelo, uma vez que se espera dados entre $[0,1]$ ou $[-1,1]$. Com isso, também se obtém consistência entre os dados.

Para aumentar a probabilidade de selecionar as variáveis mais adequadas para um modelo de múltiplas variáveis, é comum utilizar a correlação linear entre as variáveis. Isso é feito por meio de análises gráficas de dispersão e matrizes de correlação. Esses métodos fornecem informações valiosas que auxiliam na escolha mais assertiva das variáveis.

Após a análise e preparação dos dados, implementa-se o modelo LSTM. Para simplificar a manipulação dos dados, parametriza-se as combinações dos dados de entrada e saída que serão inseridas no modelo por meio de um arquivo JSON. Além disso, os pesos dos hiperparâmetros da rede neural são ajustados através de um autoajuste manual. Com o modelo treinado, é feita a previsão para o próximo período utilizando o Desvio Padrão, r-squared e a análise de gráficos como critérios de avaliação do desempenho do modelo, onde sua saída conta com o dado de “Close” (preço de fechamento) previsto para o dia seguinte. Observa-se na Figura 25 um fluxograma que exemplifica a arquitetura utilizada para a solução. Os detalhes desta arquitetura serão apresentados na seção seguinte.

Figura 25 – Fluxograma da implementação da solução



Fonte: Do autor, 2024

Através de todas as etapas, constrói-se um modelo parametrizável, no qual se ajustam apenas os hiperparâmetros, variáveis de entrada e variável de saída via JSON para o modelo, juntamente com a moeda e o período desejados de uma determinada moeda virtual. Com isso, é possível prever o preço de fechamento para o dia seguinte.

4.2 DESCRIÇÃO DA IMPLEMENTAÇÃO

Nesta seção, descreve-se detalhadamente a implementação da arquitetura da solução proposta, destacando as tecnologias empregadas para atingir os objetivos estabelecidos.

4.2.1 Tratamento (pré-processamento) e análise dos dados

Para prever o preço de fechamento das moedas virtuais, é essencial dispor de um conjunto de dados confiável e suficientemente amplo. Adicionalmente, necessita-se realizar uma análise detalhada desses dados a fim de determinar as correlações entre os mesmos. Esse processo visa otimizar a seleção de colunas relevantes para a construção do modelo LSTM.

4.2.1.1 API Binance

Para a seleção dos dados necessários, escolheu-se utilizar os serviços da Binance, uma plataforma de intermediação de compra e venda de ativos digitais, que desempenha um papel semelhante ao de corretoras de valores tradicionais. Reconhecida como uma das exchanges mais populares do mercado, a Binance facilita transações seguras e práticas, funcionando analogamente a uma bolsa de valores para criptoativos. Neste ambiente, são negociadas mais de 100 moedas digitais, destacando-se o Bitcoin como uma das principais. (SENA, 2024)

A Binance se destaca também pela disponibilidade e confiabilidade de seus dados, sendo uma fonte rica para aplicações práticas em criptoeconomia. Adicionalmente, a empresa disponibiliza uma API essencial para o desenvolvimento de aplicações customizadas que se integram às funcionalidades da plataforma. Esta API permite o acesso direto ao ecossistema da Binance, facilitando uma série de operações automatizadas. Para utilizá-la, é necessário possuir uma conta com chaves públicas e privadas, e integrar a API ao código-fonte através dos seguintes parâmetros:

- *Symbol*: Símbolo da criptomoeda de interesse, por exemplo, BTCUSDT para o Bitcoin;
- *Interval*: Intervalo de tempo para a coleta dos dados, por exemplo, 1 dias, 1 semana, e entre outros;
- *StartTime*: Data de início para a extração dos dados;
- *EndTime*: Data final para a coleta dos dados;

Com os parâmetros especificados, os dados são obtidos diretamente da Binance e utilizados para o projeto, conforme demonstrado na Tabela 1. É importante destacar que apenas as entradas referentes ao primeiro e ao último dia dos dados são apresentadas. O conjunto de dados completo abrange 2420 dias.

Tabela 1 – Dados Exemplificados da Exchange Binance.

	Date: 2017-08-17	...	Date: 2024-04-01
Open	4261.48000000	...	71280.00000000
high	4485.39000000	...	71288.23000000
low	4200.74000000	...	68062.86000000
Close	4285.08000000	...	69649.80000000
volume	795.15037700	...	41445.32039000
close time	1503014399999	...	1712015999999
quote asset volume	3454770.05073206	...	2876486469.46552290
number of trades	3427	...	1893438
taker buy base asset volume	616.24854100	...	20211.15024000
taker buy quote asset volume	2678216.40060401	...	1402849944.78228910

Fonte: Do autor, 2024

4.2.1.2 Análise de dados

Conforme discutido em seções anteriores, para elevar a precisão das previsões do modelo, realiza-se uma série de análises e manipulações nas colunas e dados do conjunto de dados. Inicialmente, opta-se por reduzir arbitrariamente a quantidade de dados de entrada que serão empregados no modelo LSTM. Portanto, os dados selecionados para este projeto incluem “Open”, “high”, “low”, “Close” e “volume”, enquanto os demais são descartados.

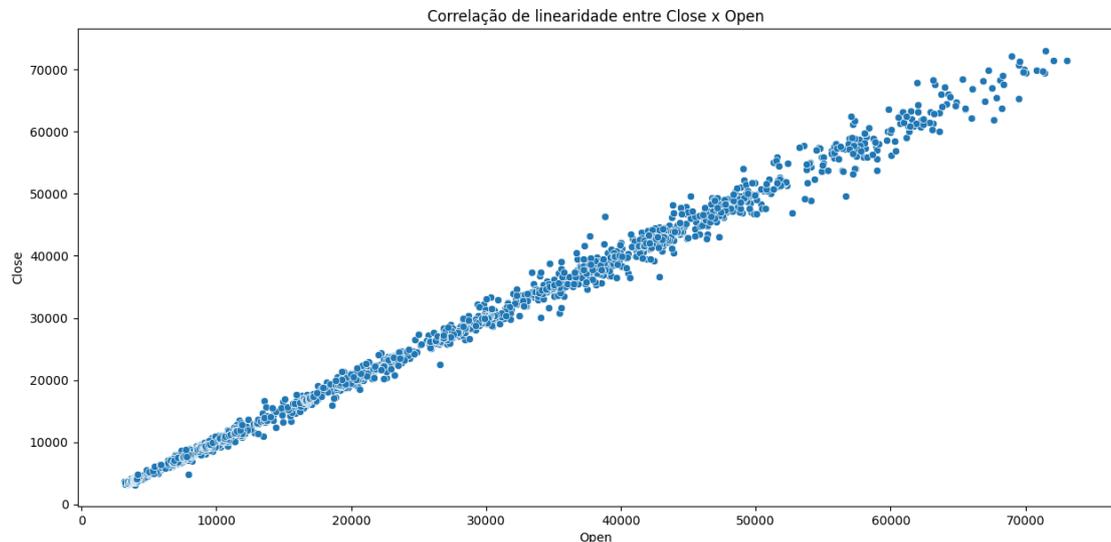
4.2.1.2.1 Regressão linear

Na segunda etapa da análise, procede-se à determinação da correlação entre as colunas selecionadas do conjunto de dados, utilizando a técnica de regressão linear. Esta técnica envolve a seleção de duas variáveis específicas do conjunto de dados, das quais se calcula a regressão linear baseada na dispersão dos dados. Sendo assim, quanto maior for a linearidade entre as variáveis, melhor é a sua relação. Neste contexto, emprega-se a função `regplot` da biblioteca Seaborn (SEABORN, 2024) para analisar a distribuição e a relação linear entre duas variáveis do conjunto de dados. A função `regplot` é projetada para traçar os dados e ajustar um modelo de regressão linear aos pontos observados, facilitando a visualização da tendência entre as variáveis.

No desenvolvimento de modelos de aprendizado de máquina, o eixo y é comumente utilizado para representar a variável-alvo, enquanto o eixo x é utilizado para a variável cuja relação de linearidade com a variável-alvo é objeto de análise. Observa-se através da Figura 26, um exemplo para a variável-alvo “Close” com a variável de análise “Open” para o Bitcoin no gráfico utilizando informações diárias.

Após a análise inicial, procede-se ao traçado de uma reta de regressão, que representa o melhor ajuste linear aos dados observados. Esta reta é determinada pelo método

Figura 26 – Close versus Open para o Bitcoin de informações diárias.



Fonte: Do autor, 2024

dos mínimos quadrados:

$$y = a + bx \quad (26)$$

onde y é a variável dependente, a o coeficiente linear, b coeficiente angular, x variável independente. com,

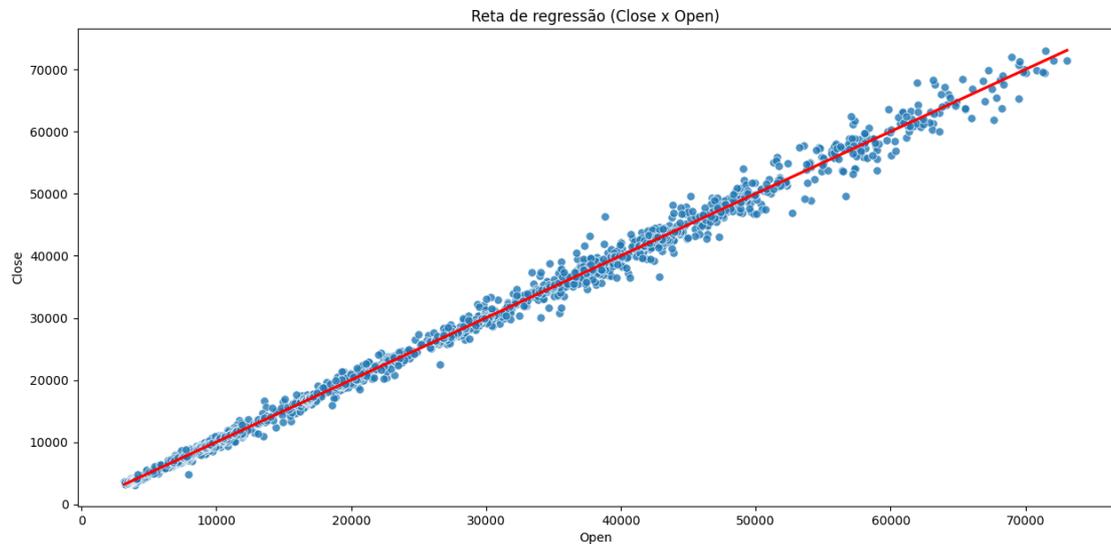
$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (27)$$

$$a = \bar{y} - \beta_1 \bar{x} \quad (28)$$

sendo n o número de observações, x_i valor da variável independente, y_i valor da variável dependente, \bar{x} média dos valores da variável independente e \bar{y} média dos valores da variável dependente. Com isso, calcula-se a linha que minimiza a soma dos quadrados das diferenças entre os valores observados e os valores previstos pela equação linear. Este processo é fundamental para visualizar a tendência central dos dados e avaliar a força e a direção da relação linear entre as variáveis analisadas. Observa-se a reta traçada através da Figura 27.

Também realizou-se uma análise para as variáveis “close” e “volume”, onde constatou-se a ausência de uma correlação linear significativa entre essas colunas, conforme pode ser confirmado na Figura 28. Com isso, após uma análise entre todas as variáveis, notou-se que “volume” não possuiu nenhuma relação com as demais variáveis analisadas, indicando que a mesma não deverá fazer parte do *dataset* do modelo.

Figura 27 – Reta de regressão através dos dados de Close e Open.



Fonte: Do autor, 2024

4.2.1.2.2 Matriz de correlação

Para uma última análise preliminar, empregou-se a biblioteca Pandas para calcular a correlação utilizando o método de Pearson, descrito pela Equação (29). Segundo essa metodologia, valores de correlação próximos de 1 indicam uma forte relação positiva, enquanto valores próximos de -1 indicam uma forte relação negativa entre as variáveis. Neste contexto, x_i e y_i representam os valores observados das variáveis, enquanto \bar{x} e \bar{y} são as respectivas médias aritméticas.

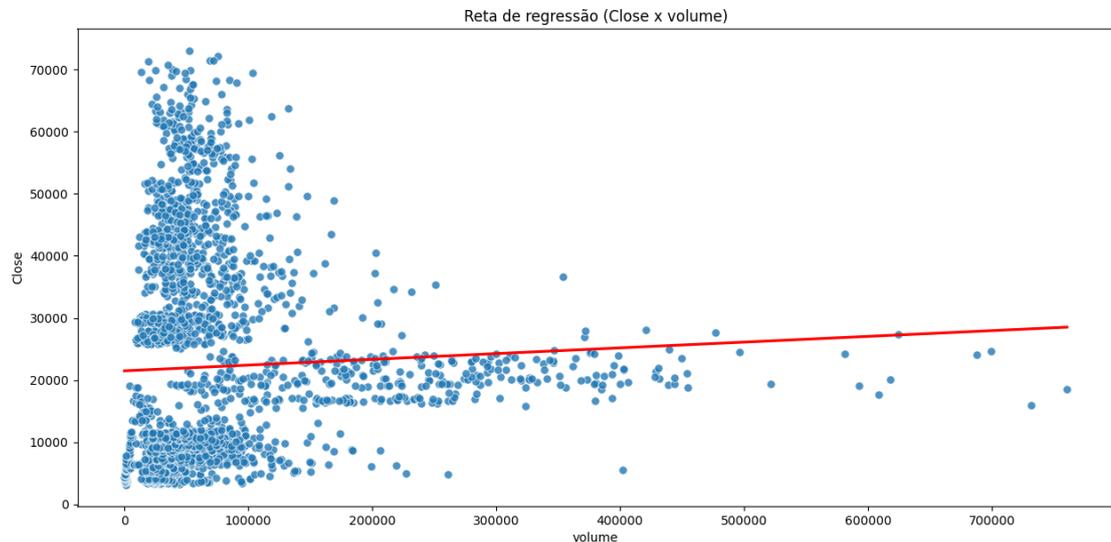
$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (29)$$

Posteriormente, procedeu-se à visualização da matriz de correlação através da Figura 29. Os resultados mostraram uma alta correlação entre a maioria das variáveis analisadas, com a exceção da coluna “volume”. Portanto, concluiu-se que o resultado da seção anterior estava correto, o “volume” não deve ser incorporado ao modelo devido à sua baixa correlação com as demais variáveis.

4.2.1.3 Arquivo JSON

No desenvolvimento do projeto, adotou-se a parametrização por meio de arquivos JSON para a seleção das colunas, pois esses arquivos permitem o armazenamento de informações de maneira estruturada. Dessa forma, especificam-se os parâmetros que serão

Figura 28 – Reta de regressão através dos dados de Close e volume.



Fonte: Do autor, 2024

utilizados no modelo. Este, por sua vez requer no mínimo, três parâmetros essenciais: “Date”, que corresponde à data; “Close”, sendo o valor-alvo; e uma terceira coluna de dados. Um exemplo dessa estrutura pode ser observado em 4.1, na qual são especificados os parâmetros “Date”, “Close”, “High”, “Low”, “Open”, “SMA” e “EMA”.

Exemplo 4.1 – Exemplo da estrutura JSON para o modelo.

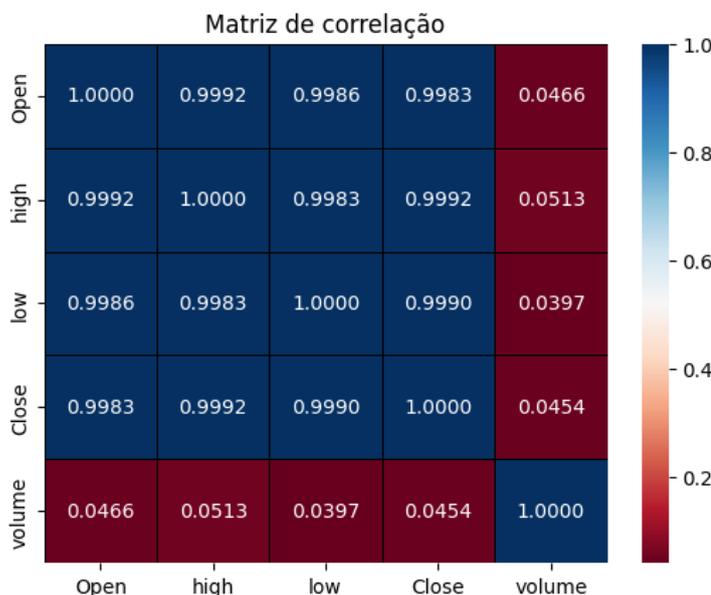
```
1 {  
2   "Date": "Date",  
3   "parameter1": "Close",  
4   "parameter2": "high",  
5   "parameter3": "low",  
6   "parameter4": "Open",  
7   "parameter5": "sma",  
8   "parameter6": "ema"  
9 }
```

Fonte: Do autor, 2024

4.2.2 Cálculo da SMA e EMA

Para enriquecimento do conjunto de dados utilizado no modelo, procedeu-se ao cálculo da SMA (Simple Moving Average) e da EMA (Exponential Moving Average),

Figura 29 – Matriz de correlação para as colunas de entrada.



utilizando a biblioteca TA-Lib (TA-LIB, 2024) em Python. Esta biblioteca é reconhecida por sua ampla aplicação em análises técnicas de mercados financeiros. A TA-Lib oferece uma variedade de ferramentas para a elaboração de indicadores de desempenho, como as médias móveis, o índice de força relativa (RSI), as bandas de Bollinger, entre outros. Tais ferramentas são essenciais para a avaliação de tendências ou padrões nos preços de ações, *commodities*, forex e outros ativos financeiros. A Figura 30 ilustra o cálculo dessas médias utilizando um intervalo de 10 dias.

4.2.3 Determinação dos hiperparâmetros

Para a implementação do modelo LSTM, é fundamental a definição dos hiperparâmetros que serão empregados na rede neural. Os hiperparâmetros selecionados para o modelo incluem:

- Offset: Onde determina-se o número de dias anteriores utilizados para prever o valor atual em uma série temporal. Funciona como uma janela deslizante que define quantos pontos de dados anteriores são considerados em cada previsão;
- Units: Representa-se a quantidade de neurônios em cada camada da rede neural;
- Drop_rate: utiliza-se para definir a taxa de dropout em cada camada da rede. O dropout é uma técnica de regularização usada para evitar o overfitting durante o treinamento da rede, como comentado em seções anteriores;

Figura 30 – Exemplo de comparativo entre Close, SMA e EMA.



Fonte: Do autor, 2024

- `Batch_size`: Onde define-se o número de amostras que serão propagadas pela rede de uma vez;
- `Epochs`: Refere-se ao número de vezes que o conjunto completo de dados é passado através da rede neural durante o treinamento;
- `No_of_outputs`: Especifica-se o número de saídas do modelo LSTM.
- `Target_column`: Especifica-se a coluna do conjunto de dados que será usada como alvo para previsões;
- `Future_days`: Define-se o número de dias no futuro para os quais o modelo tentará fazer previsões.

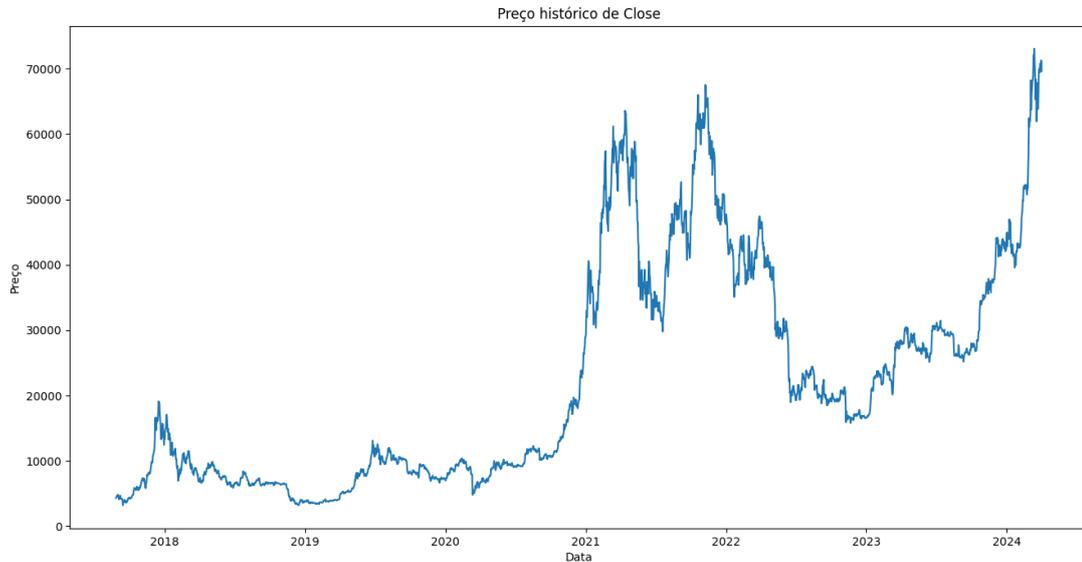
4.2.4 Determinação do alvo

Neste passo, determina-se a coluna alvo, ou seja, a variável cujo valor futuro deseja-se prever. Neste trabalho, decidiu-se prever o preço de fechamento do dia, coluna "Close". A Figura 31 apresenta os dados de "Close" ao longo do tempo, retirados diretamente da Binance, do dia 01/01/2017 até 01/04/2024.

4.2.5 Divisão dos dados

Escolheu-se a divisão do *dataset* em três partes (70% para treinamento, 15% para validação e 15% para teste) a fim de garantir uma avaliação robusta e eficaz do modelo.

Figura 31 – Histórico do preço de fechamento do Bitcoin.



Fonte: Do autor, 2024

- **Treinamento (70%):** Utilizou-se a maior parte dos dados para treinamento, permitindo que o modelo aprenda os padrões e características dos dados. Com isso, assegura-se que o modelo possui uma quantidade significativa de informações para treinar, melhorando sua capacidade de generalização.
- **Validação (15%):** Reserva-se uma fração dos dados para a validação, servindo como ajuste dos hiperparâmetros do modelo e evitando *overfitting*. Ao utilizar 15% dos dados para validação, é possível avaliar o desempenho do modelo durante o treinamento e fazer os ajustes necessários para melhorar sua precisão.
- **Teste (15%):** Utilizam-se os dados de teste para comparação com a resposta da predição, após as fases de treinamento e validação, permitindo uma avaliação imparcial do desempenho do modelo em dados que nunca foram vistos antes.

Com isso, assegura-se que o modelo não só aprenda bem com os dados disponíveis, mas também seja capaz de generalizar e performar adequadamente em novos dados, mantendo um bom equilíbrio entre aprendizado e avaliação.

4.2.6 Normalização dos dados

Após a etapa da divisão de dados do *dataset*, necessita-se normalizar os dados. Para tal tarefa, utilizou-se da função *MinMaxScaler()* da classe *sklearn.preprocessing* da biblioteca *Scikit-learn* (SCIKIT-LEARN, 2024), amplamente utilizada para transformar os dados em um intervalo específico, geralmente entre 0 e 1. O *MinMaxScaler* transforma

cada característica individualmente, de acordo com

$$X_{std} = \frac{(X - X_{min(axis=0)})}{(X_{max(axis=0)} - X_{min(axis=0)})} \quad (30)$$

sendo X valor original dos dados, X_{min} valor mínimo da característica (calculado ao longo de cada coluna) e X_{max} valor máximo da característica (calculado ao longo de cada coluna). E então,

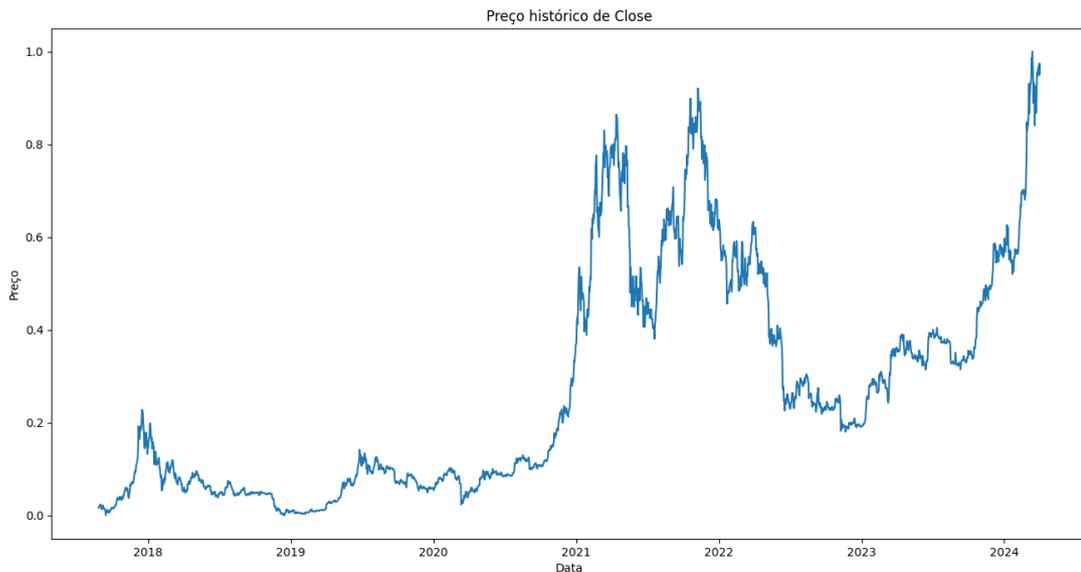
$$X_{scaled} = X_{std} \cdot (max - min) + min \quad (31)$$

onde X_{std} são os dados normalizados obtidos na Equação (30), min é o valor mínimo do novo intervalo desejado e max é o valor máximo do novo intervalo desejado.

Através da Equação (30) normaliza-se os dados para que cada característica esteja no intervalo $[0, 1]$ e então ajusta-se os dados para o intervalo desejado $[min, max]$ através da Equação (31).

Nota-se através da Figura 32, os dados de “Close” normalizados.

Figura 32 – Histórico do preço de fechamento do Bitcoin normalizados entre 0 e 1.



Fonte: Do autor, 2024

4.2.7 Construção do modelo LSTM

Para a construção e treinamento do modelo LSTM, utilizou-se o framework Keras. O modelo foi desenvolvido de forma sequencial, conforme será descrito nesta seção.

Inicialmente, adiciona-se uma camada LSTM com um número específico de neurônios, definido pelo parâmetro “units” na configuração do modelo. Essa camada foi configurada para retornar a sequência completa de saídas, o que é essencial para empilhar múltiplas

camadas LSTM subsequentes. Além disso, define-se a forma da entrada (`input_shape`), que corresponde ao número de passos de tempo e ao número de características das entradas. Após a camada LSTM, inclui-se uma camada *Dropout* para ajudar a prevenir *overfitting*, aplicando uma taxa de *dropout* específica.

Em seguida, adiciona-se múltiplas camadas LSTM de acordo com a quantidade de colunas passados pelo arquivo JSON (Colunas de *dataset* + *sma* e *ema*). Cada uma dessas camadas é seguida por uma camada *Dropout*, com taxas de *dropout* determinadas pelos parâmetros de configuração. Este processo é repetido até a penúltima camada LSTM.

Para a última camada LSTM adiciona-se a mesma sem retornar a sequência completa, apenas a última saída da sequência. Essa camada também é seguida por uma camada *Dropout*.

Por fim, adiciona-se uma camada densa totalmente conectada, com o número de saídas definido pelo parâmetro “`no_of_outputs`”. Esta camada é responsável por produzir a saída final do modelo, transformando as representações aprendidas pela rede em previsões específicas, com isso, determina-se apenas uma saída para o modelo sendo o preço de fechamento (*Close*) atendendo a um problema de regressão que refere-se ao processo de modelar a relação entre a variável dependente (o preço de fechamento) e uma ou mais variáveis independentes (dados de entradas do modelo).

Ainda, cria-se o modelo utilizando a função de perda de erro quadrático médio, adequada para problemas de regressão e o otimizador Adam, conhecido por sua eficiência e adaptabilidade. Também inclui-se a métrica de acurácia para monitorar o desempenho do modelo, conforme comentado nos tópicos a seguir:

- Função de Perda: Com essa função, mede-se a eficácia do modelo ao decorrer da execução do modelo em relação a saída desejada. Durante o treinamento, o objetivo é minimizar essa função de perda. Para isso, utiliza-se a função de perda de erro médio:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (32)$$

sendo n o número total de exemplos no conjunto de dados, y_i o valor real de i -ésima amostra e \hat{y}_i o valor previsto pelo modelo para a i -ésima amostra.

- Otimizador: O otimizador é um algoritmo que ajusta os pesos da rede neural para minimizar a função de perda. Para o modelo, utiliza-se o otimizador Adam (Adaptive Moment Estimation), onde é amplamente utilizado devido à sua eficiência e capacidade de adaptação. Ele combina as vantagens de dois outros métodos de otimização: o método AdaGrad, que mantém uma taxa de aprendizado adaptativa para cada parâmetro, e o método RMSProp, onde ajusta a taxa de aprendizado baseada na média dos quadrados dos gradientes recentes. Essa combinação permite que o Adam seja particularmente eficaz em

cenários de alta dimensão e com grandes volumes de dados (KINGMA; BA, 2014). Onde para este projeto, utilizou-se o *learning rate* padrão de 0,001 para o otimizador Adam.

Com isso, configura-se a forma com que o modelo será treinado, especificando a função de perda a ser minimizada, o algoritmo de otimização a ser utilizado para ajustar os pesos da rede, e as métricas que serão monitoradas para avaliar o desempenho do modelo.

4.2.8 Treinamento do modelo LSTM

Após definir e configurar o modelo LSTM, inicia-se o processo de treinamento, que será utilizado para prever valores futuros. Utiliza-se a função *model.fit()* do Keras, passando os dados de entrada e os valores esperados na saída. Assim, para cada conjunto X aplicado na entrada do modelo, espera-se que a saída seja o mais próximo possível de Y .

Adicionalmente, são fornecidos dados de validação, que incluem as entradas e saídas esperadas para validar o modelo durante o treinamento. Também se especifica o número de épocas e o tamanho do lote (“*batch_size*”). O número de épocas, definido previamente, determina quantas vezes o modelo deve passar por todo o conjunto de dados. O tamanho do lote indica quantas amostras o modelo deve processar antes de atualizar seus pesos.

Ainda, destaca-se que os dados de treinamento são enviesados, pois o modelo obteve acesso ao *dataset* durante o treinamento.

4.2.9 Inferência com base nos valores de validação e teste

Com o modelo treinado, faz-se a inferência para os dados de validação e teste através da função *model.predict()* do Keras. Como os dados de validação reais foram utilizados anteriormente no treinamento do modelo, a etapa de inferência permite um último ajuste, alterando seus pesos internos de acordo com a comparação entre o valor real e o previsto. Assim, os dados de validação se tornam parcialmente enviesados.

Porém, para os dados de teste, que nunca foram utilizados em nenhuma etapa do treinamento, os resultados se tornam dados “puros”, sem qualquer viés, fornecendo uma avaliação mais precisa do desempenho do modelo.

Após a inferência dos dados, é feita a transformação inversa através da função *inverse_transform()* (Equação (33)) da biblioteca *MinMaxScaler*, assim, obtém-se os valores inferidos na escala inicial, e então, consegue-se comparar com os dados reais do *dataset*.

$$X = X_{scaled} \cdot (X_{max} - X_{min}) + X_{min} \quad (33)$$

4.2.10 Predição para o dia futuro

Após a construção e treinamento do modelo, o preço de fechamento para o dia futuro é predito, utilizando os valores de “low”, “high”, “open”, “SMA” e “EMA” dos dias anteriores, que são especificados pelo parâmetro de *offset*. Com isso, obtém-se o valor de fechamento (close) para o dia futuro.

4.2.11 Cálculo r-squared e desvio padrão

A fim de analisar a eficácia dos resultados, utiliza-se o cálculo *r-squared* através da função *r2_score()* do próprio Python. Onde sua fórmula é exemplificada por:

$$r^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (34)$$

sendo, y_i os valores observados, \hat{y}_i os valores preditos, \bar{y}_i a média dos valores observados e n o número total de observações.

Outra métrica utilizada, é o cálculo do desvio padrão, utilizando a biblioteca do *numpy* através da função *numpy.std()* (NUMPY, 2024), onde se passam os dados de fechamento (close) verdadeiros do *dataset* e os dados de fechamento preditos. A sua fórmula é exemplificada através de:

$$\sqrt{\frac{\sum_i |y_i - \bar{y}|^2}{N}} \quad (35)$$

onde y_i são os valores individuais da amostra, \bar{y} a média da amostra e N o número total de observações.

5 RESULTADOS

Após o desenvolvimento do projeto, foram realizados inúmeros testes com o objetivo de identificar os hiperparâmetros mais adequados para a previsão dos preços de fechamento das criptomoedas Bitcoin e Ethereum. Este capítulo apresenta uma análise detalhada dos resultados obtidos por meio dessas tentativas, destacando as estratégias utilizadas e os parâmetros que proporcionaram as previsões mais precisas.

5.1 TESTES E RESULTADOS PARA BITCOIN

Utilizando os dados “low”, “high”, “open”, “SMA” e “EMA” como entradas e “close” como variável alvo para a predição, fixou-se o valor das janelas de períodos em 200 dias para o modelo. Em seguida, variaram-se os valores dos hiperparâmetros, conforme observados nas Tabelas 2, 3 e 4.

Tabela 2 – Número de neurônios e taxa de esquecimento para cada camada do modelo LSTM.

Camadas	Nº de units	Nº de drop_rate
Primeira	32	40%
Segunda	32	40%
Terceira	32	40%
Quarta	32	40%
Quinta	32	40%

Fonte: Do autor, 2024

Tabela 3 – Demais hiperparâmetros do modelo LSTM.

Hiperparâmetro	Valor
Nº de batch_size	5
Nº de epochs	15
Nº_of_outputs	1

Fonte: Do autor, 2024

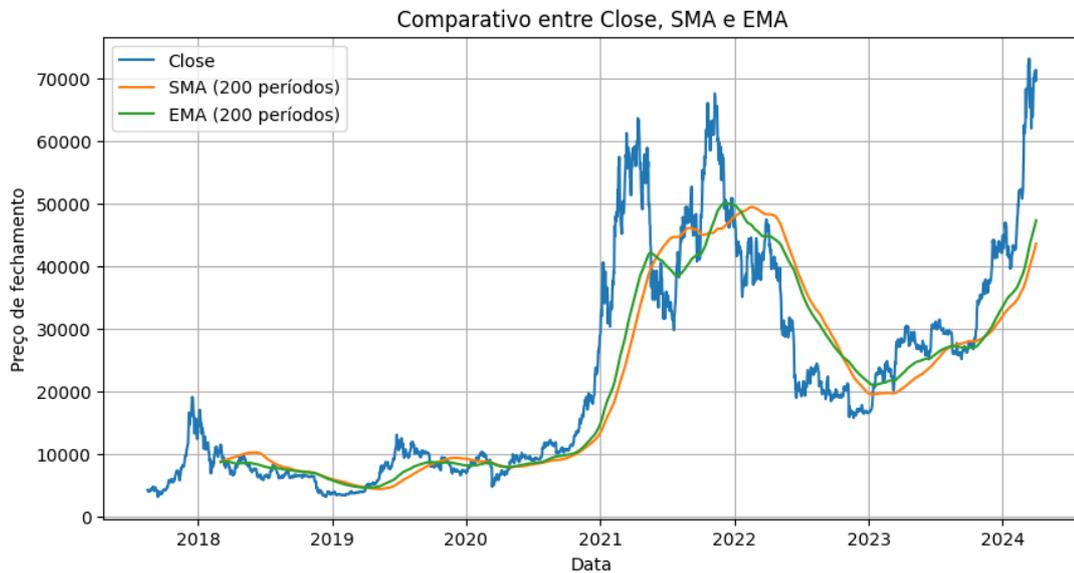
Tabela 4 – Parâmetros do modelo LSTM.

Parâmetro	Valor
target_column	Close
Nº de future_days	1

Fonte: Do autor, 2024

Após a especificação dos parâmetros e hiperparâmetros, as médias SMA e EMA são calculadas utilizando os dados de “Close” para períodos de 200 dias. A representação desses cálculos pode ser visualizada na Figura 33.

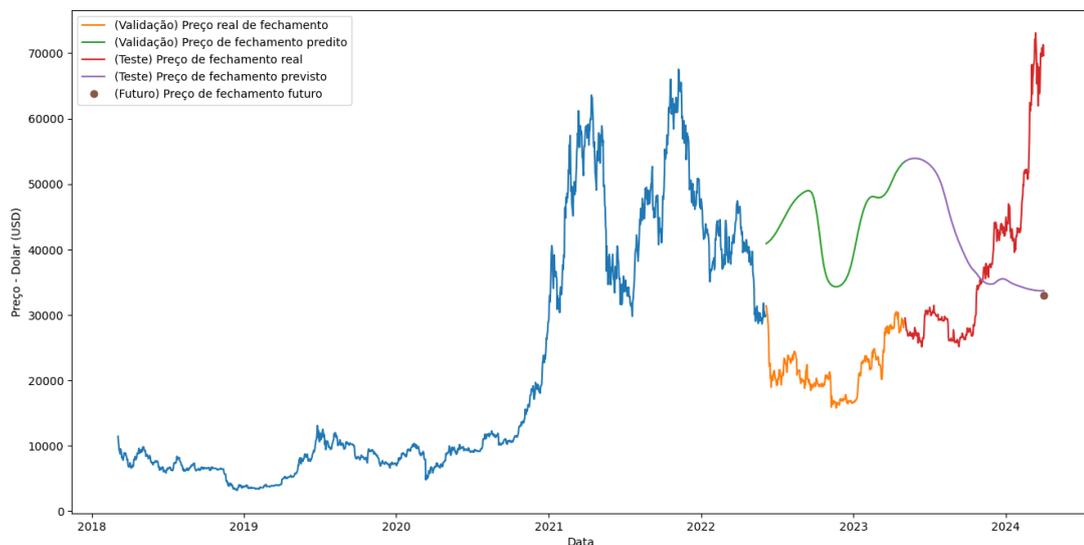
Figura 33 – Exemplo de comparativo entre Close, SMA e EMA para janelas de períodos de 200 dias.



Fonte: Do autor, 2024

E então, obtém-se os resultados das inferências para treino, validação, teste e predição do dia futuro. Através da Figura 34, consegue-se obter uma visualização gráfica dos resultados com uma comparação entre os dados reais, sendo os dados verdadeiros do dataset criado através da API Binance, e os dados de inferência feitos pelo modelo LSTM.

Figura 34 – Resultados finais de treino, validação, teste e dia futuro de Close para janelas de períodos de 200 dias.



Fonte: Do autor, 2024

O gráfico da Figura 34 mostra que o modelo não conseguiu realizar uma boa inferência utilizando os hiperparâmetros das Tabelas 2 e 3. Isso é evidenciado pelos valores do *r-squared* e do desvio padrão apresentados na Tabela 5, esses dados indicam que o modelo está errando mais do que um modelo que simplesmente prevê a média dos valores observados.

Tabela 5 – Métricas obtidas na primeira tentativa para janelas de períodos de 200 dias.

Métrica	Valor
r-squared	-1,2967
Desvio Padrão	18704,1135

Fonte: Do autor, 2024

Além disso, como uma terceira verificação de validação, compara-se a predição do preço de fechamento para o dia futuro previsto com o preço real de fechamento, já disponível durante a escrita desse trabalho na data de 02/04/2024. Essa comparação pode ser observada na Tabela 6.

Tabela 6 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para janelas de períodos de 200 dias.

Dia 02/04/2024	Valor (Dólar - USD)
Verdadeiro	65463,99
Predição	32995,80

Fonte: Do autor, 2024

A partir desses dados, é evidente que os hiperparâmetros não foram escolhidos corretamente. Portanto, realizou-se a etapa de ajuste dos hiperparâmetros, processo no qual se atribuem diferentes valores até obter um melhor resultado. Diversos ensaios foram realizados em busca dessa melhoria. Para o segundo teste, o valor para janelas de períodos foi reduzido para 100, mantendo-se os valores dos parâmetros e hiperparâmetros das Tabelas 2 e 3. Os resultados obtidos podem ser observados nas Figuras 35, 36 e nas Tabelas 7, 8.

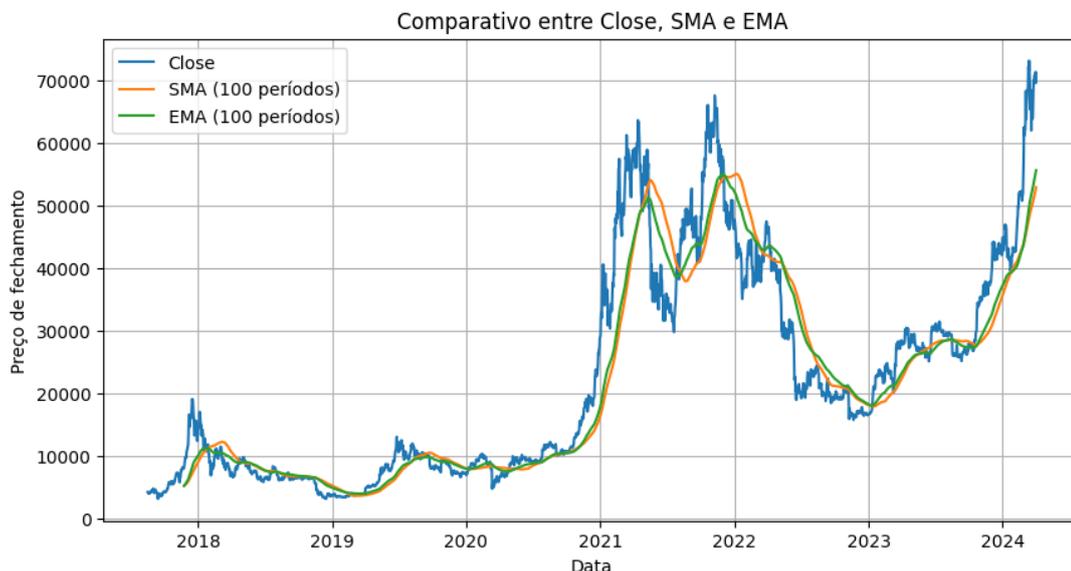
Tabela 7 – Métricas obtidas na segunda tentativa para janelas de períodos de 100 dias.

Métrica	Valor
r-squared	-1,1223
Desvio Padrão	10567,5917

Fonte: Do autor, 2024

Em seguida, foram realizados testes reduzindo ainda mais o valor dos períodos para 50, 25 e 10 dias. Observou-se que, quanto menor o período, mais perto as médias EMA e SMA tendem a ficar do preço de fechamento. Dessa forma, determinou-se que janelas de

Figura 35 – Exemplo de comparativo entre Close, SMA e EMA para janelas de períodos de 100 dias.



Fonte: Do autor, 2024

Tabela 8 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para janelas de períodos de 100 dias.

Dia 02/04/2024	Valor (Dólar - USD)
Verdadeiro	65463,99
Predição	55346,07

Fonte: Do autor, 2024

períodos de 10 dias é o ideal para o modelo. Os dados de EMA e SMA para janelas de 10 dias podem ser observados graficamente na Figura 37.

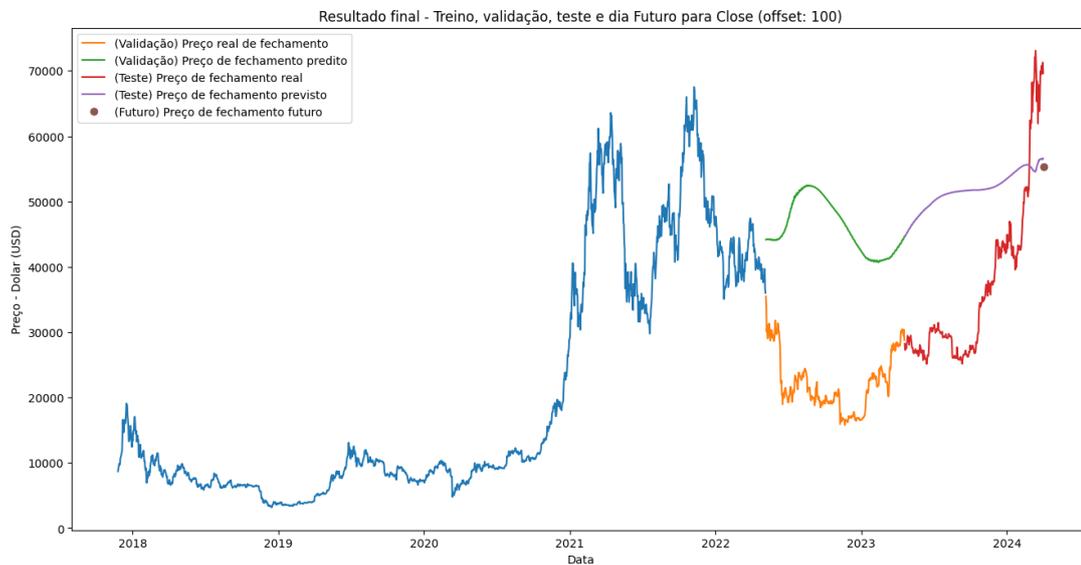
Portanto, com períodos de 10 dias, foram obtidas inferências mais precisas, conforme mostrado na Figura 38.

Além disso, nas Tabelas 9 e 10, pode-se observar as janelas de períodos de 10 dias aplicado aos outros métodos de análise do modelo. No entanto, como mostrado nas Tabelas 9 e 10, o modelo ainda não apresenta métricas satisfatórias, bem como, não possui um valor de fechamento próximo ao real. Para melhorar os resultados, os hiperparâmetros foram ajustados. Com as janelas de períodos de 10 dias determinado, variaram-se os demais hiperparâmetros.

Inicialmente, foram alterados apenas os valores de `units` e `drop_rate`, mantendo os outros parâmetros e hiperparâmetros inalterados, onde os novos hiperparâmetros ajustados são apresentados através da Tabela 11.

Com isso, observa-se as novas inferências realizadas para os dados de treino, validação, teste e predição para o dia futuro através da Figura 39. Dessa forma, as novas

Figura 36 – Resultados finais de treino, validação, teste e dia futuro de Close para janelas de períodos de 100 dias.



Fonte: Do autor, 2024

Tabela 9 – Métricas obtidas na janelas de períodos de 10 dias.

Métrica	Valor
r-squared	0,9129
Desvio Padrão	2980,0792

Fonte: Do autor, 2024

Tabela 10 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para janelas de períodos de 10 dias.

Dia 02/04/2024	Valor (Dólar - USD)
Verdadeiro	65463,99
Predição	58605,04

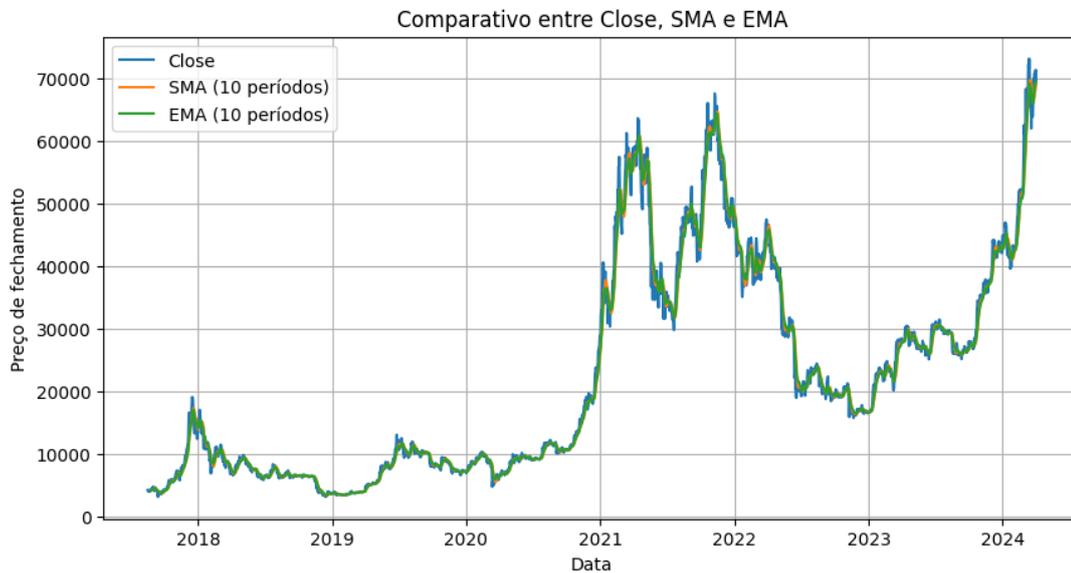
Fonte: Do autor, 2024

Tabela 11 – Número de neurônios e taxa de esquecimento para cada camada do modelo LSTM para janelas de períodos de 10 dias.

Camadas	Nº de units	Nº de drop_rate
Primeira	50	20%
Segunda	50	20%
Terceira	50	20%
Quarta	50	20%
Quinta	50	20%

Fonte: Do autor, 2024

Figura 37 – Exemplo de comparativo entre Close, SMA e EMA para janelas de períodos de 10 dias.



Fonte: Do autor, 2024

métricas de validação são apresentadas nas Tabelas 12 e 13.

Tabela 12 – Métricas para janelas de períodos de 10 dias e para hiperparâmetros da Tabela 11.

Métrica	Valor
r-squared	0,9510
Desvio Padrão	2613,4967

Fonte: Do autor, 2024

Tabela 13 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para janelas de períodos de 10 dias e para hiperparâmetros da Tabela 11.

Dia 02/04/2024	Valor (Dólar - USD)
Verdadeiro	65463,99
Predição	60625,69

Fonte: Do autor, 2024

Após a realização dos testes iniciais para identificar os melhores hiperparâmetros para o preço de fechamento do Bitcoin em 02/04/2024, as métricas ainda não apresentavam resultados satisfatórios. Com isso, foram realizados diversos ajustes finos, após os quais as métricas mais eficazes foram alcançadas utilizando os parâmetros e hiperparâmetros detalhados nas Tabelas 14, 15 e 16.

Figura 38 – Resultados finais de treino, validação, teste e dia futuro de Close para janelas de períodos de 10 dias.



Fonte: Do autor, 2024

Tabela 14 – Número de neurônios e taxa de esquecimento para cada camada do modelo LSTM para o melhor resultado de close do Bitcoin.

Camadas	Nº de units	Nº de drop_rate
Primeira	150	10%
Segunda	150	10%
Terceira	150	10%
Quarta	150	10%
Quinta	150	10%

Fonte: Do autor, 2024

Tabela 15 – Demais hiperparâmetros do modelo LSTM para o melhor resultado de close do Bitcoin.

Hiperparâmetro	Valor
Nº de batch_size	5
Nº de epochs	150
Nº_of_outputs	1

Fonte: Do autor, 2024

Tabela 16 – Parâmetros do modelo LSTM para o melhor resultado de close do Bitcoin.

Parâmetro	Valor
target_column	Close
Nº de future_days	1

Fonte: Do autor, 2024

Figura 39 – Inferências para janelas de períodos de 10 dias e novos hiperparâmetros da Tabela 11.



Fonte: Do autor, 2024

Empregando os parâmetros descritos nas Tabelas 14, 15 e 16, conseguimos ajustar o modelo de forma mais precisa aos dados. Os preços de fechamento inferidos aproximaram-se significativamente dos valores reais. Essa análise pode ser visualizada através da Figura 40.

Figura 40 – Inferências para do melhor resultado de close do Bitcoin.



Fonte: Do autor, 2024

Ademais, a qualidade da inferência é confirmada pelos resultados apresentados na Tabela 17, onde o *r-squared* alcançou um valor de 0,9886 e um desvio padrão de 1294,1737. Esses resultados indicam uma pequena variação entre os dados inferidos e os dados reais do modelo.

Tabela 17 – Métricas obtidas para o melhor resultado de close do Bitcoin.

Métrica	Valor
r-squared	0,9886
Desvio Padrão	1294,1737

Fonte: Do autor, 2024

Em conclusão, a Tabela 18 apresenta a previsão para o dia 02/04/2024, evidenciando que o preço predito está muito próximo do preço real, com uma diferença de apenas 83,41 USD ou 0,1274%.

Tabela 18 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para o melhor resultado de close do Bitcoin.

Dia 02/04/2024	Valor (Dólar - USD)
Verdadeiro	65463,99
Predição	65380,58

Fonte: Do autor, 2024

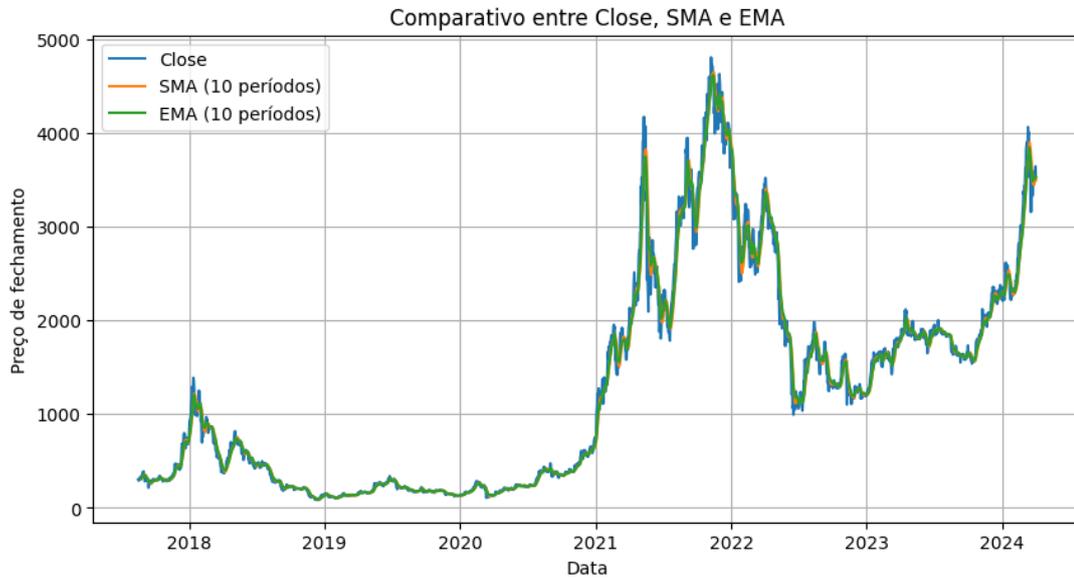
5.2 TESTES E RESULTADOS PARA ETHEREUM

Para o Ethereum, os procedimentos detalhados na Seção 5.1 referentes ao Bitcoin foram replicados. O projeto foi desenvolvido de forma totalmente parametrizável, permitindo a inserção de ETHUSD como o símbolo para a negociação de Ethereum em relação ao Tether (lastreado ao dólar americano). Conforme mencionado anteriormente, identificou-se que uma menor janela de período é mais eficaz, portanto, o período foi mantido em 10. A partir disso, as médias SMA e EMA foram calculadas e seus resultados podem ser visualizados na Figura 41.

Após um processo similar de ajuste dos hiperparâmetros, chegou-se a conclusão que os parâmetros e hiperparâmetros apresentados nas Tabelas 14, 15 e 16 também foram os mais eficazes entre todos os testes realizados. A Figura 42 ilustra a inferência e predição para os dados de treino, validação, teste e predição do close para o dia futuro do Ethereum.

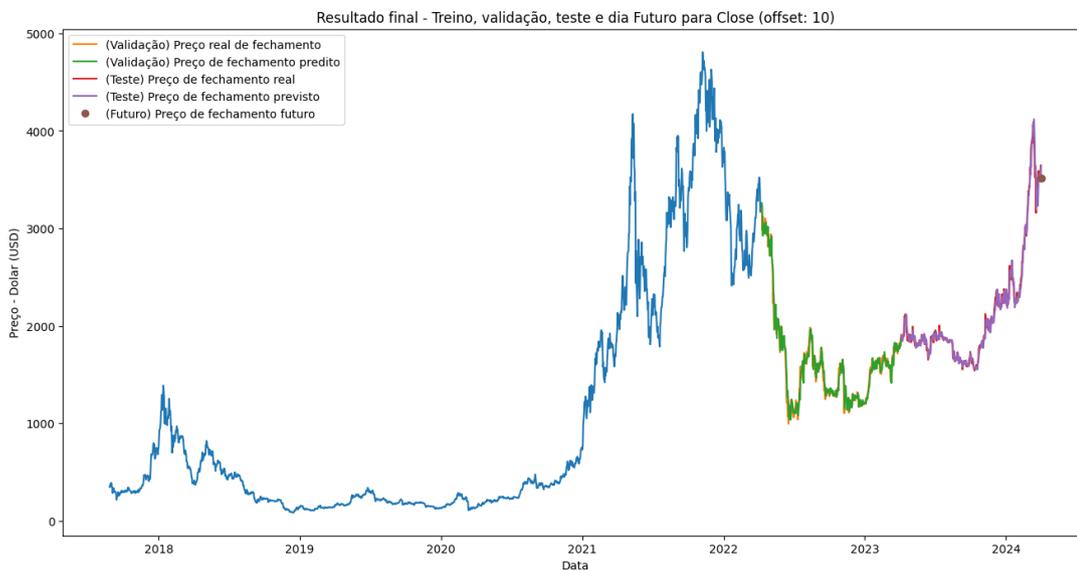
Com base nas métricas apresentadas na Tabela 19, nota-se que o modelo se adaptou de maneira mais eficiente aos dados do Ethereum do que ao Bitcoin. Uma possível explicação para essa superioridade pode ser a menor variância nos preços de fechamento do Ethereum em comparação ao Bitcoin. Assim, o modelo demonstrou um desempenho ligeiramente superior no segundo caso, alcançando um *r-squared* de 0,9863 e um desvio padrão de 66,3934.

Figura 41 – Médias SMA e EMA para janelas de períodos de 10 dias do Ethereum.



Fonte: Do autor, 2024

Figura 42 – Inferências para o Ethereum.



Fonte: Do autor, 2024

No entanto, ao analisar a diferença de 219,11 USD ou 6,68% entre o valor de fechamento previsto e o real para o dia futuro, conforme ilustrado na Tabela 20, percebe-se que, embora o modelo tenha se ajustado melhor aos dados do Ethereum, aquele que utiliza os dados do Bitcoin demonstra maior precisão na previsão do preço de fechamento futuro para este projeto.

Tabela 19 – Métricas obtidas para o melhor resultado de close do Ethereum.

Métrica	Valor
r-squared	0,9886
Desvio Padrão	66,3934

Fonte: Do autor, 2024

Tabela 20 – Comparação do preço de fechamento para o dia 02/04/2024 futuro predito e verdadeiro para close do Ethereum.

Dia 02/04/2024	Valor (Dólar - USD)
Verdadeiro	3278.96
Predição	3498.07

Fonte: Do autor, 2024

6 CONCLUSÃO

Neste estudo, foi apresentado a implementação de uma rede neural utilizando o modelo LSTM para a predição de preços de moedas virtuais, empregando um sistema completamente parametrizável que integra múltiplas camadas de entrada e uma única saída. Os dados utilizados tanto para entrada quanto para saída foram extraídos por meio da API da Binance, e complementados com cálculos das médias EMA e SMA efetuados através da biblioteca TA-LIB, elementos esses incorporados ao modelo de entrada. O objetivo primordial foi avaliar a eficácia do modelo LSTM configurado com múltiplas entradas na predição de preços de fechamento das moedas virtuais, e simultaneamente desenvolver uma estrutura que pudesse ser facilmente adaptada para a inclusão de novas criptomoedas, sem a necessidade de modificações extensivas.

A consecução desse objetivo apresentou desafios não triviais, decorrentes principalmente da necessidade de compreender profundamente as teorias das redes neurais LSTM e as dinâmicas do mercado de criptomoedas. Contudo, conforme exposto nos capítulos anteriores, o desenvolvimento foi bem-sucedido, resultando em um código robusto que utiliza metodologias eficientes para o teste, treinamento e validação do modelo, além de empregar métricas rigorosas para sua avaliação. O modelo LSTM demonstrou uma eficácia razoável na predição dos preços das criptomoedas, alcançando previsões que se aproximaram consideravelmente dos valores reais e oferecendo indicações preliminares sobre as tendências de aumento ou diminuição dos preços para os dias subsequentes.

Para futuras ampliações e melhorias deste trabalho, propõe-se a incorporação de camadas adicionais ao modelo para aumentar sua capacidade analítica e robustez. Ademais, sugere-se uma avaliação detalhada do impacto das médias SMA e EMA na matriz de correlação dos dados de entrada. Uma melhoria promissora seria a implementação de um algoritmo que ajuste automaticamente os parâmetros e hiperparâmetros do modelo, fazendo uso de uma gama mais ampla de métricas de avaliação. Isso não apenas aprimoraria a precisão das predições, mas também otimizaria o processo de adaptação do modelo a novas criptomoedas, facilitando a expansão contínua do projeto. Tais avanços contribuiriam significativamente para a evolução do campo de análise preditiva no contexto das moedas digitais.

REFERÊNCIAS

ACHELIS, Steven B. **Technical Analysis from A to Z**. [S.l.]: Mc Graw-Hill, 2000.

AGGARWAL, Charu C. **Neural Networks and Deep Learning: A Textbook**. Cham: Springer, 2018. P. 497. ISBN 978-3-319-94462-3.

ANA V. ALINE C., Cristina V. Um estudo sobre Redes Neurais Convolucionais e sua aplicação em detecção de pedestres. *In*: CAPPABIANCO, F. A. M.; FARIA, F. A.; ALMEIDA, J.; KÖRTING, T. S. (Ed.). **Electronic Proceedings of the 29th Conference on Graphics, Patterns and Images (SIBGRAP'16)**. São José dos Campos, SP, Brazil: [s.n.], out. 2016.

ANTONOPOULOS, A.M.; WOOD, G. **Mastering Ethereum: Building Smart Contracts and DApps**. [S.l.]: O'Reilly, 2018. ISBN 9781491971949.

BINANCE. **Binance - Bitcoin Exchange**. [S.l.: s.n.]. Disponível em: <https://www.binance.com/pt-BR>. Acesso em: 9 jun. 2024.

CARVALHO, Hanna V. F. de; CARVALHO, Eduardo C.; ARRUDA, Helder; IMPERATRIZ-FONSECA, Vera; SOUZA, Paulo de; PESSIN, Gustavo. Detecção de Anomalias em Comportamento de Abelhas Utilizando Redes Neurais Recorrentes. *In*: ANAIS do IX Workshop de Computação Aplicada a Gestão do Meio Ambiente e Recursos Naturais. Natal: SBC, 2018.

CHURCHLAND, Patricia S.; SEJNOWSKI, Terrence J. **The computational brain**. [S.l.]: Massachusetts Institute of Technology, 1992.

CLARK, Peter; NIBLETT, Tim. The CN2 Induction Algorithm. **Machine Learning**, v. 3, n. 4, p. 261–283, 1989. ISSN 1573-0565.

COINEXT. **O que é Tether Dólar (USDT) e como funciona essa criptomoeda?** Disponível em: <https://coinext.com.br/criptomoedas/tether-usd>. Acesso em: 22 jun. 2024.

COINMARKETCAP. **Today's Cryptocurrency Prices by Market Cap**. [S.l.: s.n.]. Disponível em: <https://coinmarketcap.com/>. Acesso em: 22 jun. 2024.

- CROSBY, Michael; NACHIAPPAN, Pradan; PATTANAYAK, Pradan; VERMA, Sanjeev; KALYANARAMAN, Vignesh. **BlockChain Technology: Beyond Bitcoin**. São Paulo: Saraiva Educação, 2016. ISBN 9788502637832.
- DEBASTIANI, Carlos Alberto. **Análise Técnica de Ações**. [S.l.]: Novatec, 2008.
- ELDER, Alexandre. **Trading for a Living: Psychology, Trading Tactics, Money Management**. New York: John Wiley & Sons, Inc., 1993.
- FAUZAN, Achmad; SUSANANGGREAINY, Maria; NATHANIEL, Nicholas; KURNIAWAN, Afdhal. Predicting Stock Market Movements Using Long Short-Term Memory (LSTM). *In*: 2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS). [S.l.: s.n.], 2023. P. 144–147.
- GOODFELLOW, Ian J.; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2016.
- HALBOUNI, Asmaa; GUNAWAN, Teddy Surya; HABAEBI, Mohamed Hadi; HALBOUNI, Murad; KARTIWI, Mira; AHMAD, Robiah. Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. **IEEE Access**, v. 10, p. 19572–19585, 2022.
- HAYKIN, Simon S. **Neural networks and learning machines**. Third. Upper Saddle River, NJ: Pearson Education, 2009.
- HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997.
- HOMMA, M. **The Candle Stick Trading Bible**. [S.l.]: Independently Published, 2020.
- KANG, Chuen Yik; LEE, Chin Poo; LIM, Kian Ming. Cryptocurrency Price Prediction with Convolutional Neural Network and Stacked Gated Recurrent Unit. **Data**, v. 7, n. 11, 2022. ISSN 2306-5729.
- KHAN, Rahemeen; BALOCH, Safia. Forecasting the Stability of COVID-19 Vaccine Companies Stock Market using LSTM and Time-series Models. *In*: 2023 International Conference on Communication, Computing and Digital Systems (C-CODE). [S.l.: s.n.], 2023. P. 1–6.

KINGMA, Diederik P; BA, Jimmy. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

LECUN, Yann; BENGIO, Y.; HINTON, Geoffrey. Deep Learning. **Nature**, v. 521, p. 436–44, mai. 2015.

LEMOES, Flávio. **Análise Técnica dos Mercados Financeiros**. São Paulo: Saraiva Educação, 2016. ISBN 9788502637832.

TA-LIB. **TA-Lib - Technical Analysis Library**. Disponível em: <https://ta-lib.org/>. Acesso em: 7 jul. 2024.

MATTOS, Olívia Bullio; ABOUCHEDID, Saulo; SILVA, Laís Araújo e. As criptomoedas e os novos desafios ao sistema monetário: uma abordagem pós-keynesiana. **Economia e Sociedade**, Instituto de Economia da Universidade Estadual de Campinas, Publicações, v. 29, n. 3, p. 761–778, set. 2020. ISSN 0104-0618.

NARAYANAN, Arvind; BONNEAU, Joseph; FELTEN, Edward; MILLER, Andrew; GOLDFEDER, Steven. **Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction**. USA: Princeton University Press, 2016. ISBN 0691171696.

NIELSEN, Michael A. **Neural Networks and Deep Learning**. [S.l.]: Determination Press, 2018. Disponível em: <http://neuralnetworksanddeeplearning.com/>.

NUMPY. **numpy.std**. Disponível em: <https://numpy.org/doc/stable/reference/generated/numpy.std.html>. Acesso em: 7 jul. 2024.

OLAH, Christopher. **Understanding LSTM Networks**. 2015. Disponível em: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acesso em: 9 jun. 2024.

RESSELL, Stuart; NORVIG, Peter. **Artificial Intelligence**. [S.l.]: Pearson Education, 2010.

SARAVANAN, R.; SUJATHA, Pothula. A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification. *In*: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). [S.l.: s.n.], 2018. P. 945–949.

SCIKIT-LEARN. **scikit-learn**. Disponível em: <https://scikit-learn.org/stable/>. Acesso em: 7 jul. 2024.

SEABORN. **seaborn: statistical data visualization**. Disponível em: <https://seaborn.pydata.org/>. Acesso em: 7 jul. 2024.

SENA, Eduardo. **O que é Binance? Saiba como funciona a exchange**. 2024. Disponível em: <https://www.nomadglobal.com/conteudos/o-que-e-binance>. Acesso em: 16 jun. 2024.

TAPSCOTT, Don; TAPSCOTT, Alex. **Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World**. [S.l.]: Portfolio, 2016. ISBN 1101980133.

TETHER. **Why use Tether?** Disponível em: <https://tether.to/en/why-tether>. Acesso em: 22 jun. 2024.

ULRICH, Fernando. **Bitcoin: A Moeda na Era Digital**. [S.l.]: Instituto Ludwig von Mises Brasil, 2014.

VIDANYA, Bruna. **Estatísticas de Inteligência Artificial: O Futuro da Tecnologia em Números**. Acesso em: abril 2024. Hostinger Tutoriais. 2024. Disponível em: <https://www.hostinger.com.br/tutoriais/estatisticas-inteligencia-artificial>. Acesso em: 23 fev. 2024.

YING, Xue. An Overview of Overfitting and its Solutions. **Journal of Physics: Conference Series**, v. 1168, 2019.