

---

Curso de Sistemas de Informação  
Universidade Estadual de Mato Grosso do Sul

---

Sistema de suporte à decisão de investimentos em criptoativos  
baseado em sentimento de mercado

Tiago Ajala Mielnik

Prof. Doutor Cleber Valgas Gomes Mira (Orientador)

Dourados - MS

2023



# Sistema de suporte à decisão de investimentos em criptoativos baseado em sentimento de mercado

Tiago Ajala Mielnik

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Tiago Ajala Mielnik e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 07 de novembro de 2023.

Prof. Dr. Cleber Valgas Gomes Mira (Orientador)

---

M575s      Mielnik, Tiago Ajala

Sistema de suporte à decisão de investimentos em criptoativos baseado em sentimento de mercado / Tiago Ajala Mielnik. – Dourados, MS: UEMS, 2023.  
108 p.

Monografia (Graduação) – Sistemas de Informação – Universidade Estadual de Mato Grosso do Sul, 2023.

Orientador: Prof. Dr. Cleber Valgas Gomes Mira.

1. Criptoativos 2. Predição de preços 3. Machine learning I. Mira, Cleber Valgas Gomes II. Título

CDD 23. ed. - 006.31

---

---

Curso de Sistemas de Informação  
Universidade Estadual de Mato Grosso do Sul

---

---

# **Sistema de suporte à decisão de investimentos em criptoativos baseado em sentimento de mercado**

**Tiago Ajala Mielnik**

Novembro de 2023

**Banca Examinadora:**

Prof. Dr. Cleber Valgas Gomes Mira (Orientador)  
Área de Computação - UEMS

Prof. Dr. Ricardo Luís Lachi  
Área de Computação - UEMS

Prof. Dr. Diogo Fernando Trevisan  
Área de Computação - UEMS



# AGRADECIMENTOS

Agradeço a Deus por ter me ajudado a chegar até aqui nesta caminhada de vários anos para concluir minha graduação.

Agradeço aos meus pais e irmãos por sempre me apoiarem nos meus anseios e nesta etapa dos meus estudos.

Agradeço ao meu orientador, professor Dr. Cleber Valgas Gomes Mira, pela disponibilidade em me orientar e apoiar na conclusão deste projeto final de curso.

Agradeço a minha esposa, Gilmara Vieira de Melo, por me incentivar a dar continuidade na conclusão da minha graduação.

Agradeço a todos que contribuíram diretamente ou indiretamente neste período que estive na faculdade.





## RESUMO

O mercado de criptoativos tem apresentado um crescimento relevante nesta última década, o que tem despertado o interesse de muitos investidores, pesquisadores e diversas pessoas nesta classe de ativos. Tem sido um desafio compreender o movimento dos preços e realizar negociações de criptoativos de forma efetiva devido à sua alta volatilidade. Acreditamos que o uso de modelos de *machine learning* com redes neurais tem demonstrado resultados interessantes para a predição de preços de criptoativos. Neste trabalho, propomos uma solução por meio de um sistema de suporte à decisão de investimento com a implementação de uma rede neural LSTM para a predição de preços de criptoativos, a partir do qual, poderemos analisar os movimentos e sentimento de mercado para realizar melhores negociações com base nas informações obtidas.

**Palavras-chave:** Criptoativos, Predição de preços, *Machine Learning*, Investimentos.



## ABSTRACT

The cryptoassets market has shown a relevant growth in the last decade, which has aroused the interest of many investors, researchers and various people in this asset class. It has been challenging to understand price movement and effectively trade cryptocurrencies due to their high volatility. We believe that the use of machine learning models with neural networks has shown interesting results for predicting crypto prices. In this work we propose a solution by using an investment decision support system with the implementation of an LSTM neural network for the prediction of crypto-asset prices, from which we can analyze the movements and market sentiment to make better trades based on the information obtained.

**Keywords:** Cryptoassets, Price prediction, Machine Learning, Investments.



# SUMÁRIO

<b>1</b>	<b>Introdução . . . . .</b>	<b>17</b>
1.1	Justificativa . . . . .	19
1.2	Objetivos . . . . .	19
1.3	Organização do Texto . . . . .	20
<b>2</b>	<b>Revisão Bibliográfica . . . . .</b>	<b>21</b>
2.1	Criptoativos . . . . .	21
2.1.1	Bitcoin . . . . .	23
2.1.2	<i>Blockchain</i> . . . . .	25
2.1.3	Investimentos em criptoativos . . . . .	32
2.2	Predição de preços de criptoativos . . . . .	35
2.3	Estratégias de decisão de investimento . . . . .	38
2.3.1	Machine Learning . . . . .	39
2.3.2	Séries Temporais . . . . .	41
2.3.3	Técnicas de Machine Learning para criptoativos . . . . .	43
2.3.4	Redes Neurais . . . . .	46
2.3.5	Redes Neurais Recorrentes . . . . .	53
2.4	Ferramentas de suporte à decisão de investimento . . . . .	56
2.5	Bibliotecas e <i>frameworks</i> . . . . .	60
<b>3</b>	<b>Implementações e Experimentos . . . . .</b>	<b>63</b>
3.1	Projeto do sistema . . . . .	63
3.2	Implementações . . . . .	64
3.2.1	Realizar predição de preços de criptoativos . . . . .	64
3.2.2	Apresentar criptoativos para investimento . . . . .	66
3.2.3	Analisar o sentimento de mercado do criptoativo a ser investido . . . . .	67
3.2.4	Criar ordem de compra . . . . .	68
3.2.5	Criar ordem de venda . . . . .	69
3.2.6	Criar ordem de <i>stop</i> . . . . .	70
3.2.7	Mostrar o portfólio de criptoativos . . . . .	71
3.2.8	Utilizar o indicador técnico RSI . . . . .	72
3.2.9	Utilizar o indicador técnico SMA20 . . . . .	73
3.2.10	Criar ordem de compra e venda automatizada . . . . .	74
3.2.11	Disponibilizar o histórico das ordens cadastradas . . . . .	77
3.3	Análise dos resultados . . . . .	78
<b>4</b>	<b>Conclusão . . . . .</b>	<b>81</b>

Referências Bibliográficas . . . . .	83
<b>Apêndices</b>	<b>88</b>
APÊNDICE A Testes dos modelos . . . . .	89
APÊNDICE B Modelagem do banco de dados . . . . .	103
APÊNDICE C Implementações do sistema . . . . .	105

# LISTA DE ABREVIATURAS E SIGLAS

ANN	<i>Artificial Neural Network</i>
API	<i>Application Programming Interface</i>
AR	<i>Autoregressive</i>
ARCH	<i>Autoregressive Conditional Heteroskedasticity</i>
ARIMA	<i>Autoregressive Integrated Moving Average</i>
BPTT	<i>Backpropagation Through Time</i>
CCXT	<i>CryptoCurrency Exchange Trading Library</i>
CNN	<i>Convolutional Neural Network</i>
CNTK	<i>Microsoft Cognitive Toolkit</i>
CPU	<i>Central Processing Unit</i>
CSS	<i>Cascading Style Sheets</i>
CVI	<i>Crypto Volatility Index</i>
DAPPS	<i>Decentralized Applications</i>
DEFI	<i>Decentralized Finance</i>
DEX	<i>Decentralized Exchange</i>
DL	<i>Deep learning</i>
GB	<i>Gigabyte</i>
GPU	<i>Graphics Processing Unit</i>
GRU	<i>Gated Recurrent Unit</i>
GRU	<i>Gated Recurrent Unit</i>
HTML	<i>HyperText Markup Language</i>
IA	<i>Inteligência Artificial</i>

ICO	<i>Initial Coin Offering</i>
LSTM	<i>Long Short-term Memory</i>
MA	<i>Moving Average</i>
MAE	<i>Mean absolute error loss</i>
MAPE	<i>Mean absolute percentage error loss</i>
MB	<i>Megabyte</i>
MLP	<i>Multilayer Perceptron</i>
MSE	<i>Mean squared error loss</i>
MSLE	<i>Mean squared log error loss</i>
MVT	<i>Model View Template</i>
NFT	<i>Non-fungible token</i>
PHP	<i>Personal Home Pages</i>
RF	<i>Requisito funcional</i>
RL	<i>Reinforcement learning</i>
RNF	<i>Requisito não-funcional</i>
RNN	<i>Recurrent Neural Network</i>
ROC	<i>Rate of Change</i>
RSI	<i>Relative strength index</i>
TLU	<i>Threshold Logic Unit</i>
VAR	<i>Vector Autoregression</i>
VIX	<i>Volatility Index</i>



# LISTA DE ILUSTRAÇÕES

Figura 1 – Fatores que influenciam no comportamento dos preços de criptoativos. . . . .	18
Figura 2 – Top 10 dos criptoativos com maior capitalização de mercado. . . . .	24
Figura 3 – Valor do preço do Bitcoin em dólares entre 2009 e 2022. . . . .	25
Figura 4 – Tipos de nós da rede <i>peer-to-peer</i> do Bitcoin. . . . .	26
Figura 5 – Arquitetura <i>blockchain</i> . . . . .	27
Figura 6 – Estrutura de uma árvore de Merkle em um bloco de transações. . . . .	28
Figura 7 – Estrutura de um bloco da <i>blockchain</i> . . . . .	29
Figura 8 – Demonstração do funcionamento de uma <i>blockchain</i> . . . . .	31
Figura 9 – Gráfico de velas semanal do <i>token</i> LUNA. . . . .	35
Figura 10 – Linha do tempo de acontecimentos importantes e pesquisas sobre criptoativos e <i>blockchain</i> . . . . .	37
Figura 11 – Distribuição dos tipos de abordagens de negociações de criptoativos apresentadas por Fang et al. . . . .	38
Figura 12 – Tipos de <i>Machine Learning</i> e suas aplicações. . . . .	41
Figura 13 – Comparação entre séries temporais estacionárias e não estacionárias. . . . .	42
Figura 14 – Trabalhos realizados com aplicação de métodos estatísticos no mercado de criptoativos. . . . .	43
Figura 15 – Distribuição entre os métodos estatísticos e métodos de <i>machine learning</i> abordados no trabalho de Fang et al. . . . .	44
Figura 16 – Fluxo da aplicação de <i>machine learning</i> em um modelo para predição de preços de criptoativos. . . . .	45
Figura 17 – Trabalhos realizados com aplicação de métodos de <i>machine learning</i> no mercado de criptoativos. . . . .	45
Figura 18 – Estrutura básica de um neurônio biológico. . . . .	47
Figura 19 – Arquitetura de um neurônio artificial simples conhecido como Perceptron. . . . .	47
Figura 20 – Gráfico de uma função de ativação do tipo função de limiar. . . . .	48
Figura 21 – Gráfico de uma função de ativação do tipo função sigmóide. . . . .	49
Figura 22 – Dados linearmente separáveis e não linearmente separáveis. . . . .	50
Figura 23 – Arquitetura de uma rede neural MLP. . . . .	51
Figura 24 – Estrutura de um neurônio recorrente. . . . .	53
Figura 25 – Funcionamento do algoritmo BPTT. . . . .	54
Figura 26 – Arquitetura de uma célula LSTM. . . . .	56
Figura 27 – Bibliotecas de código aberto para <i>deep learning</i> . . . . .	61
Figura 28 – Funcionamento da arquitetura MVT do <i>Django</i> . . . . .	62
Figura 29 – Apresentação dos criptoativos disponíveis para <i>trade</i> . . . . .	66

Figura 30 – Análise de sentimento do criptoativo ETH. . . . .	68
Figura 31 – Cadastro de ordens de compra. . . . .	69
Figura 32 – Cadastro de ordens de venda. . . . .	70
Figura 33 – Cadastro de ordens de <i>stop</i> . . . . .	71
Figura 34 – Apresentação do portfólio do usuário. . . . .	72
Figura 35 – Visualização do indicador RSI. . . . .	73
Figura 36 – Visualização do indicador SMA20. . . . .	74
Figura 37 – Cadastro de ordem de compra automatizada. . . . .	76
Figura 38 – Listagem de ordens do par de negociação BTC/BUSD. . . . .	78
Figura 39 – Comparação entre os preços reais e previstos do modelo LSTM para os próximos 7 dias. . . . .	79
Figura 40 – Construção do modelo LSTM com a biblioteca <i>TensorFlow</i> . . . . .	92
Figura 41 – Comparação entre o erro de validação e treinamento do modelo LSTM. . . . .	93
Figura 42 – Comparação entre os preços reais e previstos do modelo LSTM. . . . .	93

# 1 INTRODUÇÃO

Os criptoativos são uma classe de ativos que têm se popularizado muito nos últimos anos devido à possibilidade de ganhos com investimentos e também pelas diversas soluções que apresentam. Após o surgimento do Bitcoin em 2009 como meio de pagamento descentralizado, vários criptoativos surgiram com diferentes finalidades ao decorrer dos anos e, atualmente, há cada vez mais projetos que visam resolver algum problema e conquistar investidores. Apesar dos criptoativos trazerem um certo receio para as pessoas que tem pouco conhecimento sobre o assunto, podemos ver que eles têm sido adotados por diversas pessoas e entidades devido aos seus princípios como descentralização e segurança por meio da tecnologia *blockchain* (BURNISKE; TATAR, 2017).

Apesar da tecnologia *blockchain* ter surgido com a criptomoeda Bitcoin, a *blockchain* tem evoluído além do mercado financeiro de criptoativos. A primeira fase foi marcada pelo surgimento do Bitcoin e a descentralização dos meios de pagamento. Já a segunda fase foi marcada pelo surgimento de aplicações descentralizadas e contratos inteligentes. A terceira fase da *blockchain*, na qual nos encontramos atualmente, é a expansão desta tecnologia nos diversos setores da sociedade, para as quais esta tecnologia apresenta soluções para indivíduos, organizações públicas e privadas. Algumas destas soluções são: registro de propriedade intelectual, prova de identidade e existência, armazenamento descentralizado de dados, acesso a serviços governamentais descentralizados, entre outras (SWAN, 2015).

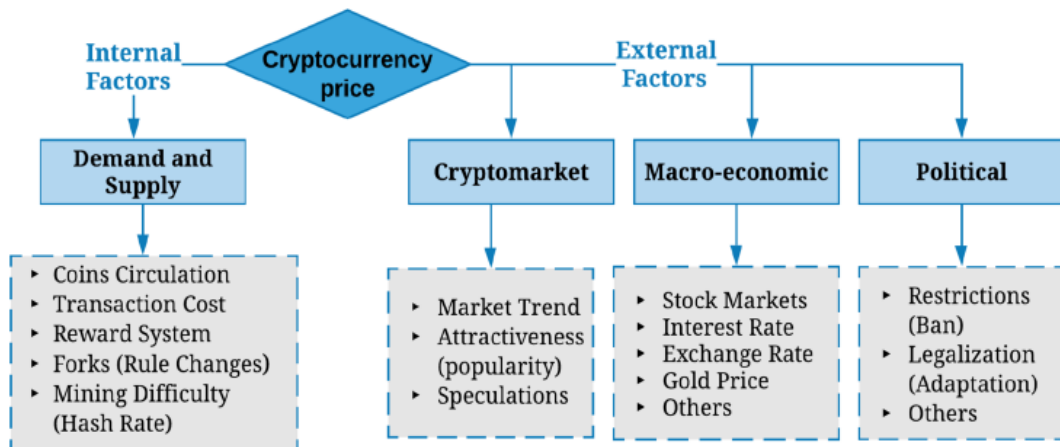
O mercado financeiro de criptoativos tem se mostrado um mercado atraente para a realização de investimentos. Além de ser um mercado que funciona 24 horas por dia e durante os fins de semana, existem várias formas de se obter ganhos, desde investimentos de menor risco a investimentos de maior risco. Muitos pesquisadores, investidores e *traders* tem buscado compreender o movimento dos preços desses ativos tão voláteis que, ao mesmo tempo, oferecem grandes possibilidades de ganhos e também de perdas. Devido a esta alta volatilidade, é importante que os investidores tomem decisões coerentes em suas negociações para que consigam maximizar seus ganhos e minimizar perdas (FANG et al., 2022).

As tomadas de decisões de investimentos em mercados financeiros estão sujeitas a serem afetadas por fatores psicológicos e emocionais dos investidores, e portanto, nem sempre o comportamento dos preços dos ativos correspondem ao seu valor fundamentalista. As notícias são um exemplo de evento que pode impactar o mercado e as emoções dos investidores, tanto positivamente quanto negativamente. A análise de sentimentos é o estudo computacional que visa medir numericamente opiniões e emoções das pessoas, extraídas a partir de dados textuais. Recentemente, algumas das formas que os pesquisadores têm utilizado para extrair sentimentos de mercado são por meio de técnicas de mineração de

dados textuais e processamento de linguagem natural de textos de redes sociais, como *Twitter*, *Reddit* e demais *websites* como o *Google Trends*. Outras formas de analisar o sentimento de mercado são por meio de indicadores técnicos de sentimento como o índice VIX, análise de dados de transações em *blockchains*, entre outras (KRAAIJEVELD; SMEDT, 2020).

A predição de preços de ativos tem sido um problema complexo e desafiador na área de finanças, pois os mercados financeiros são dinâmicos e possuem vários fatores que estão sempre em mudança e que influenciam no comportamento dos preços. A complexidade na predição de preços de criptoativos tem sido maior ainda, visto que esses ativos possuem muita volatilidade (YU; YAN, 2020).

A **Figura 1** mostra alguns dos fatores que influenciam no comportamento dos preços dos criptoativos. Podemos ver que existem os fatores internos como: circulação de moedas, custo de transação, *forks* e dificuldade de mineração. Também, existem os fatores externos como: oferta e demanda, tendência do mercado, popularidade, especulações, situação da macroeconomia, mercado de ações, taxa de juros, políticas governamentais sobre restrições e regulamentações de criptoativos, entre outros (KHEDR et al., 2021).



**Figura 1** – Fatores que influenciam na volatilidade e comportamento dos preços dos criptoativos. Fonte: (KHEDR et al., 2021), p. 5.

Na última década, a inteligência artificial, que é uma área da computação, tem apresentado uma grande evolução. Cada vez mais, pessoas e organizações têm buscado usufruir da inteligência artificial para melhorar seus processos e tomadas de decisões. O uso da IA nos mercados financeiros tem se mostrado interessante para a construção de modelos de *machine learning* para a predição de preços (SABRY et al., 2020).

Segundo Yu e Yan (2020), antes do surgimento de algoritmos de *machine learning* eficientes, os pesquisadores, geralmente, utilizavam métodos estatísticos e econométricos para desenvolverem modelos matemáticos para a predição de séries temporais. Estes modelos convencionais tratam de problemas de dados que apresentam uma relação linear, e por isto, é necessário realizar adaptações no modelo para a aplicação em problemas nos quais os dados não apresentam uma relação linear. Uma maneira interessante de tratar

de problemas de natureza não linear, como é o caso da predição de preços de ativos, é por meio da utilização de redes neurais. As redes neurais possuem vantagens por serem adaptativas e direcionadas a dados. Elas são mais adequadas para lidar com uma grande quantidade de dados que apresentam ruídos.

Dentre as arquiteturas de redes neurais, as redes neurais recorrentes têm sido as mais adequadas para problemas que envolvem séries temporais, como o problema de predição de preços. As redes neurais recorrentes consideram tempo e sequência, portanto, possuem a capacidade de obter *feedback* temporal de eventos anteriores, nos quais contribuem para o estado atual da rede neural. A LSTM (Long Short Term Memory) é um tipo de rede neural recorrente mais robusta que as redes neurais recorrentes tradicionais. Enquanto as redes neurais tradicionais conseguem manter apenas dependências temporais de curto prazo, a arquitetura LSTM possui a vantagem de manter dependências temporais de curto e longo prazo (ACADEMY, 2022).

A utilização de sistemas de suporte à decisão é uma forma interessante dos investidores melhorarem suas tomadas de decisão de investimentos. Os sistemas de suporte à decisão são ferramentas importantes em um ambiente de resolução de problemas pois auxiliam na produtividade e na tomada de decisão dos usuários. Estes sistemas utilizam técnicas computacionais como inteligência artificial e análise de dados do problema para gerar informações relevantes para a tomada de decisão (ARIF, 2013).

Neste trabalho, será implementado um sistema de suporte à decisão de investimentos, cuja principal funcionalidade é a predição de preços de criptoativos. Será construído um modelo de *machine learning* com a aplicação de uma rede neural LSTM para realizar a predição de preços de alguns criptoativos. Tendo realizado a predição de preços, o sistema irá analisar os preços do gráfico de velas e o sentimento de mercado dos criptoativos para auxiliar o investidor na decisão de como realizar os seus investimentos.

## 1.1 Justificativa

Devido à necessidade de investidores de criptoativos tomarem decisões, com base na razão e não pela emoção, este trabalho se justifica através do estudo e aplicação de uma estratégia de investimento automatizada por um sistema de informação.

A utilização de um sistema de informação com uma estratégia bem definida e automatizada, possui a vantagem de evitar que sentimentos humanos de ganância e medo interfiram negativamente nas tomadas de decisões de investimentos.

## 1.2 Objetivos

O principal objetivo deste trabalho é o de implementar um sistema com uma estratégia automatizada para tomada de decisão de investimentos em criptoativos, de forma a obter

melhores oportunidades de investimentos, minimizando altos riscos e proporcionando a possibilidade de melhores retornos financeiros.

Os objetivos específicos são:

1. Estudar o funcionamento das tecnologias dos criptoativos;
2. Analisar o comportamento do mercado de criptoativos por meio de dados técnicos e indicadores de sentimento, coletados na *Internet* com o uso de APIs;
3. Estudar estratégias para a tomada de decisão de investimentos em criptoativos;
4. Desenvolver um sistema implementando um algoritmo de aprendizado de máquina supervisionado, para a estratégia de tomada de decisão;
5. Realizar testes de investimentos nos principais criptoativos para avaliar o desempenho da estratégia aplicada no período de curto e médio prazo.

## 1.3 Organização do Texto

O **Capítulo 1** apresenta a introdução deste trabalho, sua justificativa, seus objetivos e a organização textual.

O **Capítulo 2** apresenta a revisão bibliográfica. Na **Seção 2.1** é abordado o contexto dos criptoativos e suas tecnologias. Na **Seção 2.2** é abordado o problema de predição de preços de criptoativos. Na **Seção 2.3** são apresentadas algumas estratégias de decisão de investimentos com a aplicação de métodos estatísticos e métodos de *machine learning* propostos por diversos autores. Na **Seção 2.4** são apresentadas algumas ferramentas de suporte à decisão de investimentos. Na **Seção 2.5** são apresentadas algumas bibliotecas e *frameworks* a serem utilizados no desenvolvimento do sistema de suporte à decisão de investimentos.

O **Capítulo 3** apresenta as implementações e os experimentos realizados durante o desenvolvimento dos modelos de predição de preços e do sistema *web*. Na **Seção 3.1** são apresentados os requisitos funcionais e não funcionais elaborados durante o projeto do sistema. Na **Seção 3.2** são apresentadas as implementações dos requisitos funcionais do sistema. Na **Seção 3.3** é apresentada a análise dos resultados dos modelos de predição de preços.

O **Capítulo 4** apresenta a conclusão deste trabalho e aponta os desafios encontrados durante a etapa de desenvolvimento e possíveis trabalhos futuros.

Nos apêndices, temos o **Apêndice A** que apresenta os testes dos modelos de predição de preços. O **Apêndice B** apresenta a modelagem do banco de dados do sistema. No **Apêndice C** são apresentados alguns trechos de código em python referentes as tarefas de predição de preços e execução da estratégia automatizada de compra e venda.

## 2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, contextualizaremos o mercado financeiro de criptoativos e suas tecnologias e o problema de predição de preços dos mesmos. Abordaremos o assunto, por meio do estudo de trabalhos já propostos por alguns autores. Na **Seção 2.1**, faremos uma análise dos criptoativos, suas tecnologias e formas de sua aplicação em investimentos. Na **Seção 2.2**, trataremos o problema de predição de preços de criptoativos. Na **Seção 2.3**, iremos apresentar as estratégias e técnicas que podem ser aplicadas na decisão de investimentos. A **Seção 2.4**, trará as ferramentas de suporte à decisão nos investimentos de criptoativos. E para finalizar, na **Seção 2.5**, iremos apontar as bibliotecas e *frameworks* que podemos utilizar no desenvolvimento do sistema.

### 2.1 Criptoativos

Com o avanço da tecnologia, os meios de pagamento passaram por muitas mudanças nas negociações financeiras. A principal mudança que vem ocorrendo nas últimas décadas é a digitalização dos meios de pagamento. Os meios de pagamentos digitais têm sido muito importantes para a sociedade, pois têm se tornado mais rápidos e acessíveis para as pessoas, de forma que, pode-se realizar transações financeiras por meio de celulares e computadores de qualquer lugar com acesso à *Internet*.

Atualmente, os meios de pagamentos digitais mais populares no Brasil são os cartões de crédito, débito, pré-pagos, carteiras digitais e PIX. Com o início da pandemia da COVID-19 em 2020, esses meios de pagamentos digitais foram importantes para minimizar o contágio do vírus, pois além de evitar o contato físico com o dinheiro, pôde-se realizar compras *online* de sua própria residência (SANTOS; CAVALCANTI, 2020).

Dentre os meios de pagamentos digitais, atualmente tem crescido a utilização de criptomoedas. Criptomoedas são uma categoria de criptoativos que exercem a função de uma moeda, ou seja, são utilizadas como meio de troca e reserva de valor (BURNISKE; TATAR, 2017).

Os criptoativos são ativos digitais que utilizam a criptografia para manter suas transações seguras e invioláveis (NARAYANAN et al., 2016). A primeira idéia de criação de um criptoativo com características de criptomoeda foi proposta por Wei Dai em 1998. O projeto ficou conhecido como B-Money (ADHAMI; GIUDICI; MARTINAZZI, 2018).

Em 2009, ocorreu o lançamento da primeira criptomoeda conhecida como Bitcoin, e com o tempo foram surgindo vários criptoativos semelhantes e alternativos ao Bitcoin, os *Altcoins*. O primeiro criptoativo alternativo ao Bitcoin foi a criptomoeda *Namecoin* que surgiu em 2011, após um *fork* do projeto do Bitcoin. Apesar de *Altcoins* serem alternativas

ao Bitcoin, estes criptoativos possuem sua própria *blockchain*<sup>1</sup> e seus protocolos (NARAYANAN et al., 2016).

Entre os muitos criptoativos existentes, podemos mencionar os *tokens digitais*, também chamados de “*tokens*”. Os *tokens* servem para representar valor de ativos digitais negociados em plataformas de produtos e serviços de criptoativos. Em comparação com as criptomoedas e *Altcoins*, a grande maioria dos *tokens* não possuem sua própria *blockchain*, pois são criados em outras plataformas descentralizadas (LEWIS, 2018). A plataforma Ethereum é um exemplo de *blockchain* que possui diversos *tokens* lançados em sua rede devido à grande quantidade de *Dapps* que são executados nela.

Os *Dapps* são conhecidos também como “aplicativos descentralizados”. São aplicativos que devem ser *open source* e possuir seus dados e registros de transações criptografados em uma *blockchain* pública e descentralizada. Os *Dapps* possuem *tokens* que são necessários para sua utilização, que geralmente são disponibilizados em um pré-lançamento e também no lançamento dos aplicativos. Os *Dapps* estão sujeitos a mudanças em seu protocolo por meio de *feedbacks* da comunidade, e tais mudanças devem ser aceitas pelo consenso da maioria de seus usuários. A comunidade também pode contribuir com melhorias e receber *tokens* como recompensas (SWAN, 2015).

Entre os diversos criptoativos existentes, existem alguns tipos nos quais podemos classificar (TAPSCOTT; TAPSCOTT, 2016):

- Criptomoedas: utilizadas como meio de pagamento. Exemplos: Bitcoin, Monero e Zcash;
- *Tokens* de plataforma: utilizados em plataformas focadas na execução de aplicativos descentralizados. Estas plataformas possuem *Dapps* em sua rede, onde nas transações envolvendo o uso dos *Dapps*, são cobradas taxas que são pagas por *tokens* da plataforma. Exemplos: Eth, Neo, Aion;
- *Tokens* de utilidade: permitem obter acesso a recursos e benefícios quando utilizados em determinadas plataformas de produtos e serviços. Uma utilidade que podemos mencionar é o uso do *token* Golem para pagamento de poder computacional negociado em sua rede. Alguns outros exemplos: BAT, Chiliz;
- *Tokens* digitais privados: utilizados em plataformas que usam a segurança e confiabilidade da tecnologia *blockchain* para criar e negociar tokens de ações, títulos privados, regulamentações. Exemplo: Polymath, tZero;
- *Tokens* de ativos naturais e commodities: *tokens* de ativos naturais e *commodities* tradicionais dentro de uma *blockchain*. Representam valor do ouro, petróleo, gás natural;

---

<sup>1</sup> Banco de dados distribuído e descentralizado com seus dados organizados em uma cadeia de blocos (MOUGAYAR; BUTERIN, 2016).



- Cripto colecionáveis: *tokens* especiais que representam algo único. Utilizados para representar obras de arte e itens raros. Exemplo: NFTs;
- Criptomoedas fiat e *Stablecoins*: criptomoedas que representam o preço de moedas fiduciárias. São consideradas criptomoedas estáveis por não possuírem a alta volatilidade de outros criptoativos. Exemplos: USDT, BUSD.

### 2.1.1 Bitcoin

O Bitcoin é um criptoativo que surgiu em 2008, em um artigo publicado por um pseudônimo chamado Satoshi Nakamoto, descrito como um sistema de dinheiro eletrônico *peer-to-peer* (NAKAMOTO, 2008). O Bitcoin foi criado a partir de combinações de projetos anteriores, como o HashCash e B-Money para ser um sistema de pagamentos descentralizado. O surgimento do Bitcoin resolveu o problema do gasto duplo que era enfrentado por moedas digitais (ANTONPOULOS, 2014).

Ulrich (2017) descreve o Bitcoin como uma forma de dinheiro digital *peer-to-peer*, seguro e rápido para negociações *online*. Seu valor é determinado livremente pelos indivíduos do mercado, não possuindo uma entidade intermediária nas transações globais, sendo assim, um sistema de pagamentos descentralizado.

Nakamoto (2008) explica em seu artigo que, para resolver o problema do gasto duplo, é necessário um sistema de pagamento eletrônico baseado em criptografia ao invés de ser baseado em confiança pelo intermédio de um terceiro. O surgimento do Bitcoin contribuiu para a solução do problema do gasto duplo e problema dos generais bizantinos.

No problema dos generais bizantinos, existe um grupo de generais e suas tropas separadas em locais diferentes em torno de um castelo. Sem possuir uma comunicação direta entre eles, o objetivo dos generais é realizar um ataque simultâneo com todas as suas tropas e conquistar o castelo. Para isto, é necessário que haja um consenso entre todos os generais no momento de atacar ou recuar. O problema de consenso ocorre, por exemplo, quando é necessário transmitir uma mensagem entre os generais, através de um mensageiro, porém, a mensagem ou não é entregue ou é adulterada, ou algum general é um traidor que não irá ajudar no ataque. Em transações financeiras que envolvem o Bitcoin, não é necessário uma entidade intermediária para estabelecer confiança entre as partes. A confiança é estabelecida através do consenso de uma rede descentralizada de computadores que realizam validações das transações por meio do uso da criptografia e protocolos confiáveis (MAHMOUD; LESCISIN; ALTAEI, 2019).

O problema do gasto duplo, ocorre quando não se pode confirmar que uma quantidade de moeda digital foi gasta apenas uma vez. Desta maneira, indivíduos mal intencionados podem se aproveitar de falhas do sistema e gastar indevidamente uma quantidade de moeda digital. O surgimento da tecnologia *blockchain* apresentada com o Bitcoin resolve este problema, pois as transações financeiras são validadas por protocolos criptográficos e

pelo algoritmo de consenso de uma rede descentralizada de computadores. Portanto, o problema dos generais bizantinos está relacionado com o problema do gasto duplo, pois não é necessário uma entidade intermediária para validar este gasto e estabelecer confiança nas transações (SWAN, 2015).











O Bitcoin foi o primeiro criptativo a ser lançado e, no momento de escrita deste trabalho, é o principal criptativo com maior capitalização de mercado em dólares no *ranking* de criptativos do *website* CoinMarketCap (2022).

Na **Figura 2** podemos ver que o Bitcoin possui mais que o dobro de valor de capitalização de mercado em comparação ao Ethereum que é o segundo colocado. Mesmo o Ethereum possuindo uma quantidade bem maior de *tokens* em circulação, o Bitcoin demonstra ter um valor maior de capitalização de mercado.

O valor de capitalização de mercado é calculado da seguinte forma:

$$MarketCap = CurrentPrice \times CirculatingSupply$$

*CurrentPrice* representa o preço atual e *CirculatingSupply* representa a quantidade de criptativos em circulação.

#	Name	Price	24h %	7d %	▼ Market Cap ⓘ	Volume(24h) ⓘ	Circulating Supply ⓘ
1	 Bitcoin BTC <a href="#">Buy</a>	\$34,205.00	▼ 2.77%	▼ 10.44%	\$651,388,758,141	\$38,251,779,045 1,117,773 BTC	19,034,531 BTC
2	 Ethereum ETH <a href="#">Buy</a>	\$2,532.34	▼ 3.50%	▼ 9.55%	\$306,432,205,215	\$22,058,810,790 8,688,749 ETH	120,700,644 ETH
3	 Tether USDT	\$0.9998	▼ 0.00%	▼ 0.03%	\$83,201,129,898	\$82,472,615,203 82,488,720,139 USDT	83,217,377,095 USDT
4	 BNB BNB <a href="#">Buy</a>	\$356.58	▼ 2.54%	▼ 8.11%	\$58,384,370,432	\$1,974,144,253 5,520,866 BNB	163,276,975 BNB
5	 USD Coin USDC	\$0.9999	▼ 0.01%	▼ 0.01%	\$48,675,790,209	\$5,757,656,624 5,757,794,908 USDC	48,676,959,278 USDC
6	 XRP XRP	\$0.5697	▼ 2.32%	▼ 5.58%	\$27,664,249,004	\$1,608,237,175 2,810,384,352 XRP	48,343,101,197 XRP
7	 Solana SOL <a href="#">Buy</a>	\$75.53	▼ 4.62%	▼ 14.85%	\$25,440,098,374	\$1,527,633,655 20,135,426 SOL	335,320,721 SOL
8	 Cardano ADA	\$0.7469	▼ 2.25%	▼ 4.31%	\$25,354,311,782	\$1,149,849,319 1,533,790,631 ADA	33,820,262,544 ADA
9	 Terra LUNA <a href="#">Buy</a>	\$65.30	▼ 3.18%	▼ 19.67%	\$22,393,866,737	\$5,742,511,794 87,979,995 LUNA	343,092,422 LUNA
10	 TerraUSD UST	\$0.9964	▼ 0.02%	▼ 0.25%	\$18,627,266,840	\$2,413,001,827 2,424,003,530 UST	18,712,194,935 UST

**Figura 2** – Top 10 criptativos com maior capitalização de mercado. Fonte: <<https://coinmarketcap.com>>.

Desde o seu lançamento, o Bitcoin tem se provado um criptativo importante. Além de ser um criptativo descentralizado, seu valor é comparado ao ouro, ou seja, é uma reserva de valor escassa pois é limitado a 21 milhões de unidades. O preço do Bitcoin

é determinado pela oferta e demanda e com o passar dos anos tem se valorizado muito devido aos seus bons fundamentos (ULRICH, 2017).

A **Figura 3** mostra o gráfico em escala logarítmica do preço de mercado do Bitcoin desde o seu lançamento. Podemos perceber que o preço de mercado se manteve numa tendência de alta no decorrer dos anos, onde mesmo possuindo várias quedas em seu valor, tem sempre ultrapassado seus topos máximos anteriores.



**Figura 3** – Valor do preço do Bitcoin em dólares entre 2009 e 2022. Fonte: <<https://www.blockchain.com/pt/charts/market-price>>.

Um exemplo que demonstra a importância do Bitcoin e dos criptoativos é o conflito entre os países Rússia e Ucrânia, no ano de 2022. A Ucrânia tem recebido ajuda de outros países por meio de transferências de Bitcoin e outros criptoativos. Já a Rússia teve sua moeda (rublo russo) muito desvalorizada e os russos tiveram que recorrer às criptomoedas como o Bitcoin para protegerem seu capital da desvalorização e fugir de sanções internacionais (INFANTE; MAZZEI, 2022).

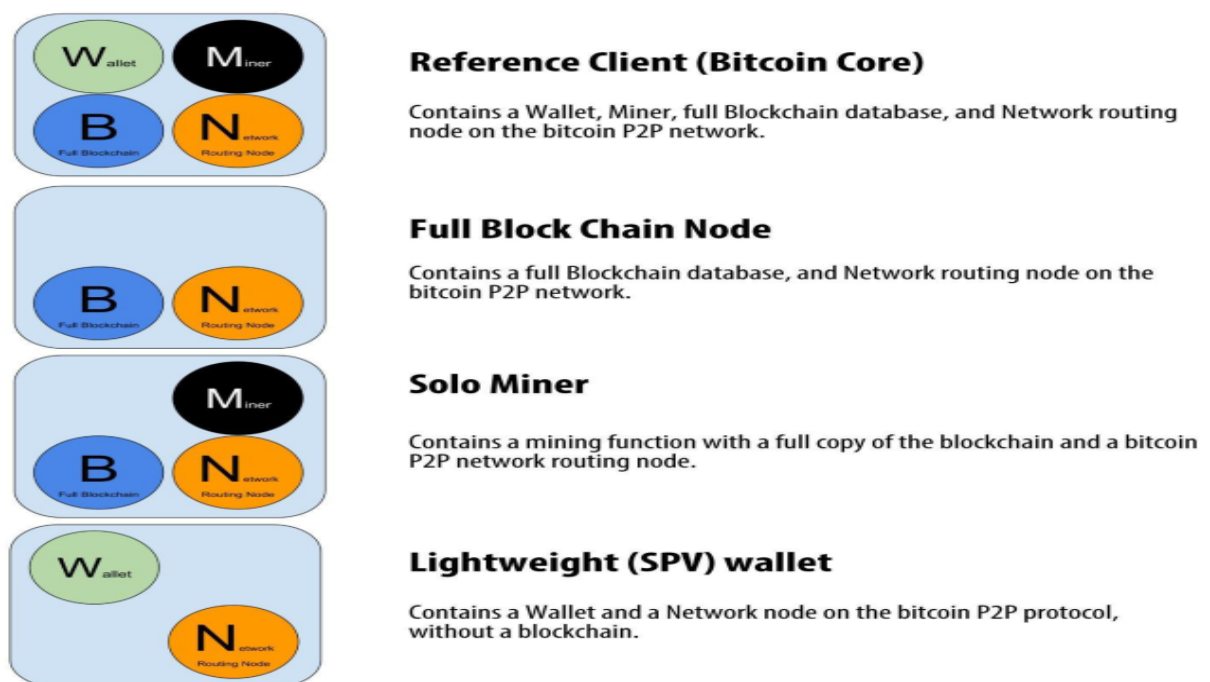
### 2.1.2 Blockchain

*Blockchain* é uma tecnologia que surgiu junto com o Bitcoin em 2008, na apresentação de um artigo publicado por Satoshi Nakamoto (NAKAMOTO, 2008). É uma tecnologia com a finalidade de realizar transações *peer-to-peer* confiáveis por meio da criptografia, de forma que as transações são imutáveis e novas transações são registradas em blocos sequenciais, formando uma cadeia de blocos com todo o histórico validado por uma rede descentralizada de computadores (MOUGAYAR; BUTERIN, 2016).

Na rede *peer-to-peer* do Bitcoin, os participantes da rede são conhecidos como nós. Apesar dos nós serem considerados iguais, eles podem exercer diferentes funções. Um nó da rede pode possuir mais de uma função, as quais são classificadas da seguinte forma (ANTONOPOULOS, 2014):

- Nós completos: mantêm uma cópia completa e atualizada da *blockchain*. Conseguem verificar qualquer transação da rede sem precisar de referências externas de outros nós;
- Mineradores: são nós que competem entre si pela criação e validação de novos blocos de transações a serem adicionados à *blockchain*. Um novo bloco minerado recompensa o minerador com uma quantidade de Bitcoins pela tarefa realizada conhecida como *Proof of Work*;
- Carteiras: são usuários que realizam transações fazendo o gasto da criptomoeda da rede. Nós completos e mineradores também podem realizar esta função;
- Nós roteadores de rede: são nós participantes da rede. Todos os nós exercem tal função, pois propagam e validam transações na rede.

A **Figura 4** exibe alguns tipos de nós da rede do Bitcoin, conforme suas funcionalidades. O círculo verde representa a funcionalidade de carteira, o círculo preto representa a funcionalidade de minerador, o círculo azul representa a funcionalidade de nó com a cópia completa da *blockchain* e o círculo laranja representa a funcionalidade de roteador de rede.

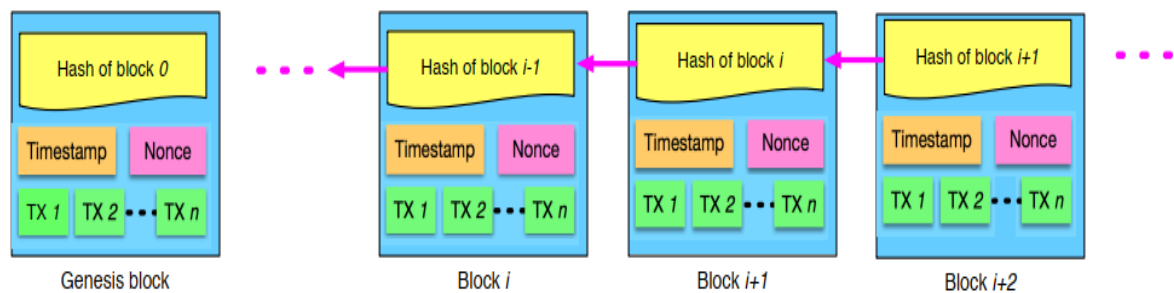


**Figura 4** – Tipos de nós com diferentes funções na rede *peer-to-peer* do Bitcoin. Fonte: Adaptado de (ANTONOPOULOS, 2014), p. 142.

A criptomoeda Bitcoin foi a primeira aplicação prática do uso da *blockchain*. No artigo de Nakamoto não é mencionado o termo “*blockchain*”, pois ele surgiu na comunidade do Bitcoin após o seu lançamento devido a grande quantidade de menções aos termos “*block*” e “*chain*” feitas por Nakamoto em seu artigo.

Antonopoulos (2014) explica que a estrutura de dados da *blockchain* é uma lista ordenada de blocos de transações, onde ocorre apenas a inserção de blocos. Cada bloco é ligado ao anterior por meio de uma referência ao *hash* do bloco anterior, ou seja a referência é feita do final para o início da lista. A *blockchain* pode ser visualizada como uma pilha vertical de blocos, onde o primeiro bloco serve como um suporte e fortalece todos os demais blocos da pilha. Cada bloco da *blockchain* é identificado por um *hash* gerado pelo algoritmo criptográfico **SHA-256**.

Na **Figura 5** podemos ver o bloco *genesis* e a conexão dos demais blocos fazendo uma referência ao bloco anterior, formando a arquitetura da *blockchain*.



**Figura 5** – Arquitetura *blockchain*. Fonte: (ZHENG et al., 2018), p. 4.

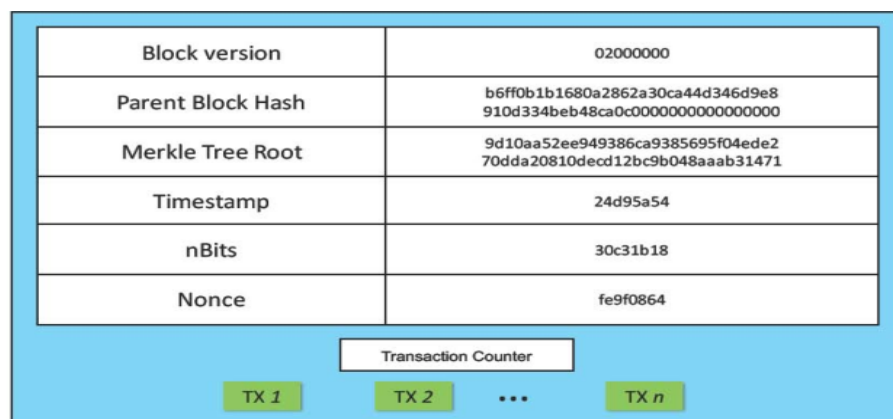
Entrando em um pouco mais de detalhes sobre os blocos da *blockchain*, Zheng et al. (2018) explica que todos eles possuem um *hash* identificador. Além disso, um bloco é composto por um cabeçalho e um corpo de dados. O cabeçalho de um bloco é composto pelos seguintes dados:

- Versão do bloco: exibe qual a sua versão. Esta versão indica quais os conjuntos de regras aplicadas na validação do bloco;
- *Hash* do bloco anterior: indica qual foi o *hash* do bloco anterior. Todos os blocos possuem uma referência ao *hash* do bloco anterior, exceto o primeiro bloco da *blockchain*, conhecido como bloco *genesis*;
- *Hash* raiz da árvore de Merkle: a árvore de Merkle é conhecida como uma árvore binária de *hash*, utilizada para resumir eficientemente e verificar a integridade de dados. É uma árvore binária completa utilizada para resumir todas as transações de um bloco em um único *hash*, conhecido como o *hash* raiz. Nesse tipo de árvore, as folhas da árvore são as transações, e os nós acima são a combinação do *hash* dos filhos. Desta maneira, é possível verificar a existência de uma transação em um bloco de forma eficiente através da raiz da árvore (ANTONPOULOS, 2014).



O corpo de um bloco é composto pelas transações e um contador das mesmas. O número de transações do bloco varia de acordo com o tamanho do bloco e das transações. Atualmente o tamanho médio de um bloco é de 1MB segundo o *website* Blockchain.com (2022).

A **Figura 7** mostra a estrutura de um bloco da *blockchain* composta pelo cabeçalho e o corpo, com os dados sendo representados no formato hexadecimal. Podemos ver que o cabeçalho é constituído pelos campos versão do bloco, *hash* do bloco anterior (*Parent Block Hash*), *hash* da raiz da árvore de Merkle, *timestamp*, *nbits* e *nonce*. Já no corpo do bloco é mostrado o contador de transações e as transações representadas por  $TX_1$ ,  $TX_2$ ,  $TX_n$ .



**Figura 7** – Estrutura de um bloco da *blockchain*. Fonte: (ZHENG et al., 2018), p. 5.

O *Proof of Work* consiste em encontrar um *hash* válido que identifica um bloco. Para isto, é preciso realizar um esforço computacional mínimo exigido pela rede através do campo *nbits*. O bloco também possui um *nonce* que é um número arbitrário gerado aleatoriamente e utilizado no cálculo do *hash* identificador deste bloco. O valor do *nonce* deve ser gerado e testado diversas vezes até que um minerador encontre um valor que satisfaça a prova de trabalho exigida, ou seja, encontre um *hash* identificador do bloco que satisfaça o número de bits zeros exigidos (ANTONOPOULOS, 2014).

A resolução do *Proof of Work* é essencial para manter a integridade da *blockchain*. Caso houvesse alguma forma de alterar informações de um bloco, isto implicaria na mudança do *hash* do mesmo e alguém precisaria resolver novamente a prova de trabalho para encontrar um *hash* válido. Como novos blocos são criados utilizando o *hash* do bloco anterior, se o *hash* anterior muda, é necessário refazer toda a prova de trabalho dos blocos seguintes ao bloco modificado para que o restante da cadeia seja válida. Portanto, é muito difícil algum atacante praticar alguma fraude na *blockchain*, pois é muito custoso modificar um bloco, refazer o *Proof of Work* de todos os seguintes, a tempo de não haver a inclusão de um novo bloco na cadeia. Além disso existem vários nós na rede de forma descentralizada que possuem uma cópia da *blockchain*. Um atacante teria que ter controle da maior parte dos nós da rede para conseguir manipular a mineração de novos blocos e obter um consenso na

rede. O *Proof of Work* demora cerca de 10 minutos, pois é mantido por uma média móvel visando um número médio de blocos por hora. Se os blocos são gerados de forma muito rápida, a dificuldade da rede aumenta (NAKAMOTO, 2008).

Nakamoto (2008) explica que pode ocorrer a situação onde dois nós conseguem minerar blocos diferentes simultaneamente e transmitir para os demais nós da rede, ocasionando uma bifurcação provisória da cadeia de blocos principal. Os nós irão trabalhar com o primeiro bloco recebido, mas irão manter o outro ramo salvo. A próxima prova de trabalho irá estender um dos dois ramos, fazendo com que um dos ramos seja maior e, com isso, os nós do ramo menor mudarão para o ramo maior, pois os nós da rede sempre consideram a cadeia de blocos mais longa.

Nakamoto (2008) descreve o funcionamento da rede do Bitcoin com as seguintes atividades:

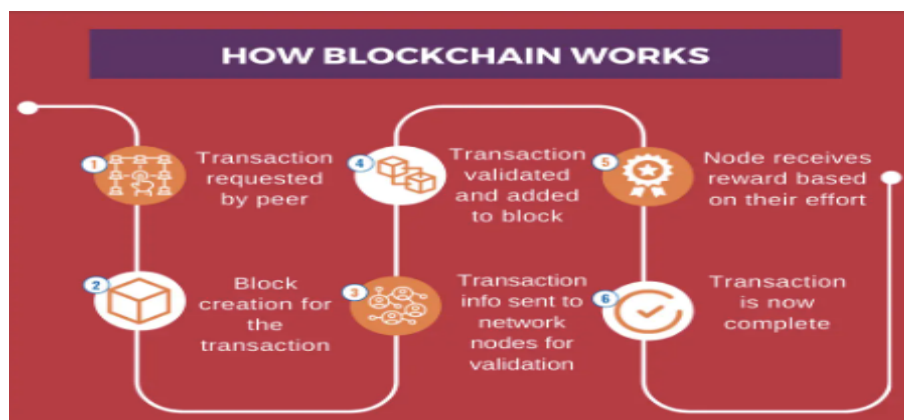
1. Novas transações são transmitidas para todos os nós.
2. Cada nó coleta as transações em um bloco;
3. Cada nó compete com os demais para encontrar o *Proof of Work* do novo bloco a ser adicionado na *blockchain*;
4. Quando um nó conclui o *Proof of Work*, ele é recompensado com uma quantia de Bitcoins e transmite o bloco para todos os demais nós;
5. Os nós aceitam o novo bloco somente se todas as transações do bloco estão válidas e não foi praticado o gasto duplo;
6. Os nós expressam sua aceitação ao bloco, trabalhando na criação do próximo bloco utilizando o *hash* deste último bloco recém minerado.

A **Figura 8** demonstra de forma resumida o funcionamento da *blockchain* conforme os passos mencionados anteriormente.

Apesar da tecnologia *blockchain* ter sido uma revolução no meio digital, existem alguns desafios a serem superados. Yli-Huumo et al. (2016) menciona alguns destes desafios, dos quais podemos destacar:

- Processamento de poucas transações por segundo: a *blockchain* do Bitcoin processa cerca de 7 transações por segundo. Quando a rede está congestionada, a quantidade de transações ainda pode diminuir. Comparada às instituições processadoras de cartões de crédito que processam milhares de transações por segundo, é uma quantidade bem inferior;
- Latência: latência muito alta, pois um bloco de transações da *blockchain* do Bitcoin demora cerca de 10 minutos para ser confirmado pela rede, com a finalidade de manter a segurança contra ataques de gasto duplo;





**Figura 8** – Demonstração do funcionamento de uma *blockchain*. Fonte: Adaptado de <<https://101blockchains.com/blockchain-for-beginners>>.

- Tamanho e largura de banda: Atualmente segundo o *website* Blockchain.com (2022), a *blockchain* do Bitcoin possui um tamanho de cerca de 410GB. Como um bloco é limitado em média a 1MB, o número de transações por bloco é bem limitado. Caso for necessário aumentar o número de transações por bloco, problemas de tamanho e largura de banda devem ser resolvidos na rede;
- Segurança: apesar de confiável, a tecnologia *blockchain* corre risco de um **ataque de 51%**. Nesse tipo de ataque, se uma entidade possuir mais de 50% do controle de poder de mineração da rede, ela será capaz de manipular a *blockchain* pois irá conseguir superar a prova de trabalho dos nós honestos. A descentralização dos nós da rede ajuda a minimizar este risco;
- Gasto excessivo de energia elétrica: devido o grande esforço computacional requerido pelo *Proof of Work*, a mineração utilizando este tipo de consenso desperdiça uma grande quantidade de energia. Se a energia elétrica utilizada não for de fonte renovável, a mineração estará impactando negativamente ao meio ambiente.

Além da *blockchain* do Bitcoin, existem outras *blockchains* que surgiram para suprir algumas necessidades do Bitcoin. A estrutura da rede Ethereum é um exemplo que se propôs a fazer uso de contratos inteligentes dentro de sua *blockchain* e também poder criar *tokens* e aplicações descentralizadas. Além de aplicações financeiras, a tecnologia *blockchain* pode ser utilizada para várias finalidades não-financeiras, como por exemplo em verificação de autenticidade de documentos, prova de existência e integridade de arquivos, armazenamento descentralizado de dados, entre outras (CROSBY et al., 2016).

Como a *blockchain* é considerada um livro-razão público, muitas informações são de fácil acesso. A *blockchain* do Bitcoin é um exemplo onde podemos consultar diversas informações como tamanho da *blockchain*, tempo médio de mineração de um bloco, detalhes das transações dos blocos, dificuldade de mineração da rede, entre outros dados

que podemos utilizar até mesmo para tomada de decisão em investimentos. Podemos consultar estas informações no *website* Blockchain.com (2022).

### 2.1.3 Investimentos em criptoativos

Assim como em investimentos nos mercados financeiros tradicionais, existem perfis de risco no investimento em criptoativos, sendo os criptoativos bem mais arriscados que os ativos tradicionais. Em investimentos de criptoativos é importante que seja feita uma análise fundamentalista e também uma análise técnica do comportamento dos preços.

Na análise fundamentalista de criptoativos, o foco do investimento é mais para o longo prazo. Necessitamos conhecer o projeto do criptoativo e saber quais os seus objetivos e o problema que ele resolve. Isto pode ser feito analisando o *white paper* do criptoativo. Outro fator que auxilia na análise fundamentalista, é identificar se a comunidade e os desenvolvedores são ativos no projeto. A descentralização do criptoativo também é um fator importante a ser considerado. Além disso, a maturidade do criptoativo é um bom critério a ser observado, visto que quanto mais tempo o criptoativo está no mercado, mais ele passa por provas difíceis e se consolida entre os investidores (BURNISKE; TATAR, 2017).

A análise técnica possui foco indiferente em relação ao prazo de investimento. Ela possibilita interpretar o comportamento do mercado por meio da movimentação dos preços dos ativos e da utilização de gráficos e indicadores técnicos, com a finalidade de antecipar o movimento futuro dos preços (LEMOS, 2017).

Os criptoativos oferecem algumas formas de ganhos financeiros. Podemos citar as seguintes formas de ganho (NORMAN, 2017):

- Investimentos de longo prazo (*Buy and Hold*): são investimentos de compra e venda em que o investidor pretende obter um lucro maior, mantendo o criptoativo e aguardando sua valorização em um período de 1 a 2 anos, ou até mais. É importante ter um gerenciamento de risco definido, pois o investidor ficará exposto ao risco por um grande período. Outro fator importante, é a escolha de criptoativos com bons projetos avaliados com a análise fundamentalista, pois no longo prazo eles sempre tendem a valorizar;
- *Trading*: são negociações de compra e venda de criptoativos realizadas em um período de tempo menor, podendo ser dentro de um dia (*day trading*) e por alguns dias (swing trading). Alguns fatores importantes são: analisar o movimento dos preços com o auxílio de gráficos e indicadores técnicos, e utilizar uma gestão de risco para minimizar perdas e manter os ganhos. Neste tipo de investimento, geralmente os investidores não priorizam os fundamentos de um criptoativo, mas buscam oportunidades de negociações para obter ganhos no curto prazo;

- Mineração: forma de ganho em que é preciso investir em equipamentos de bom poder computacional, pois a mineração é uma tarefa que exige a realização do *Proof of Work*. A recompensa pela tarefa realizada, são ganhos de criptoativos e taxas das transações realizadas na rede. Desta maneira, é possível acumular criptoativos obtidos na mineração para posteriormente vendê-los. É uma forma de ganho que possui vantagens e desvantagens. Uma vantagem que podemos mencionar, é a boa recompensa na mineração de blocos na *blockchain* do Bitcoin. Segundo o *website* Blockchain.com (2022), atualmente cada bloco minerado recompensa o minerador em 6,25 Bitcoins, o que corresponde a aproximadamente 130 mil dólares na cotação atual do Bitcoin. Um exemplo de desvantagem, é o alto custo para adquirir equipamentos próprios para mineração, que além disso, necessitam de bom resfriamento e cuidados com manutenção;
- Participação em ICO (*Initial Coin Offering*): forma de ganho em que os investidores financiam um projeto promissor sem garantias de ganhos, mas recebem *tokens* que podem ter uma boa valorização. Caso o projeto não se desenvolva conforme o esperado, ocorre o risco do *token* sofrer uma grande desvalorização e até ocorrer encerramento do projeto. É uma forma de ganho bem arriscada, visto que o projeto do criptoativo está numa fase inicial e pode sofrer dificuldades para se desenvolver. Apesar do alto risco, esta forma de ganho pode oferecer uma grande rentabilidade caso o criptoativo tenha uma boa valorização e o investidor consiga mantê-lo para o longo prazo.

Outras formas de ganhos mencionadas por Guides (2022) são:

- Renda passiva: comparada ao recebimento de dividendos e poupança. O investidor deposita seus criptoativos deixando disponíveis em *Exchanges*, conhecidas como plataformas de câmbio *online* de criptoativos. O investidor pode resgatar seus criptoativos a qualquer momento ou deixar por um período fixado, recebendo pequenas recompensas em criptoativos;
- DEFI (*Decentralized Finance*): ganhos obtidos em plataformas de finanças descentralizadas. Um exemplo de ganho que podemos mencionar é o investidor comprar *tokens* de uma *Exchange* descentralizada (DEX) e disponibiliza-los para a DEX em um período de tempo determinado, onde receberá *tokens* como recompensa. Neste tipo de negociação, o investidor fornece liquidez para a DEX e recebe uma quantia de *tokens* conforme acordo realizado. Um outro exemplo de ganho, são os empréstimos dentro dessas *Exchanges* descentralizadas. O investidor pode emprestar seus criptoativos para outras pessoas (negociação *peer to peer*) e receber juros sobre os criptoativos emprestados, além de poder fazer empréstimos de algum criptoativo e pagar taxas bem atraentes em comparação com as taxas de empréstimos de bancos tradicionais.

Estes acordos realizados entre os usuários, são feitos de maneira automatizada e anônima por meio de contratos inteligentes na *blockchain*;

- *Air Drops*: são *tokens* de algum projeto de criptoativo distribuídos geralmente de forma gratuita. Os *tokens* são distribuídos para indivíduos que atendem algum requisito estabelecido pelos desenvolvedores do projeto. Alguns exemplos são: possuir uma certa quantidade de um determinado criptoativo em carteira e realizar alguma tarefa na comunidade para receber recompensas de *Air Drops*. Apesar de receber *tokens* de forma gratuita, ocorre o risco do investidor sofrer com golpistas. Um exemplo, seria o investidor conectar sua carteira em *websites* e aplicativos desconhecidos para realizar alguma tarefa, e acabar tendo sua carteira comprometida.

É importante ressaltar que os criptoativos e as modalidades de ganhos mencionadas estão sujeitos ao risco de perdas. Apesar de oferecerem oportunidades de grande rentabilidade, também estão sujeitos à grande perdas.

Um exemplo que podemos mencionar é o criptoativo LUNA. Dentro de aproximadamente um ano, no período de Maio/2021 a Abril/2022, o *token* teve uma valorização muito grande de cerca de 2000%. Já no mês de Maio/2022, ocorreu a sua grande desvalorização após uma forte alta que fez o seu preço máximo chegar próximo de 119 dólares por unidade. Após uma falha do algoritmo que mantinha o balanceamento do *token* LUNA e a *stablecoin* UST, a cotação destes criptoativos teve uma forte queda. O desespero dos investidores realizando grandes quantidades de vendas também contribuiu para que o preço do *token* LUNA desvalorizasse quase 100% e chegasse à quase zero (NICOCÉLI, 2022).

Na **Figura 9** podemos ver a movimentação de preço do *token* LUNA, onde cada vela no gráfico representa uma semana. Neste tipo de gráfico, o preço é representado por um valor de abertura, um valor mínimo, um valor máximo e um valor de fechamento. Cada vela na cor verde representa uma semana de alta, onde o preço tem sua abertura iniciada na base preenchida inferior e tem seu fechamento realizado na parte preenchida superior. Uma variação de mínimas e máximas antes do fechamento do preço, acaba deixando linhas nas extremidades da vela. Já as velas vermelhas, cada uma representa uma semana de baixa, onde o preço de abertura foi iniciado na base preenchida superior e o fechamento foi realizado na base preenchida inferior, o que demonstra que o preço de fechamento ficou abaixo do preço de abertura



**Figura 9** – Gráfico de velas semanal do *token* LUNA no período de 2021 a 2022. Fonte: <<https://br.tradingview.com/chart/>>

## 2.2 Predição de preços de criptoativos

Nos últimos anos, têm surgido muitas pesquisas sobre mercado de criptoativos e seus desafios. Isto contribui para que o mercado de criptoativos se desenvolva e se popularize cada vez mais entre as pessoas. Alguns dos desafios enfrentados são problemas relacionados à mineração, segurança, anonimato e privacidade, detecção de crimes/fraudes, negociações automatizadas, predição de volatilidade e predição de preços (SABRY et al., 2020).

A predição de preços de criptoativos tem se tornado importante nos mercados financeiros, devido a este mercado ter crescido e se popularizado entre muitos investidores. Tem sido um grande desafio pesquisadores utilizarem estratégias e técnicas de predição de preços de criptoativos, visto que o mercado de criptoativos é considerado recente e bem mais volátil que os mercados financeiros tradicionais (GUO et al., 2021).

A predição de preços em criptoativos é uma forma de estimar os preços futuros com base no histórico passado dos preços e de alguns fatores que podem influenciar no comportamento destes preços (SABRY et al., 2020).

Fang et al. (2022) explicam que existem diversas abordagens para negociações de criptoativos, as quais podem ser categorizadas como negociações de análise técnica e

negociações de análise fundamental. Essas categorias dependem de dados quantificáveis para serem testados e comparados com os dados históricos de um criptoativo para verificar o seu desempenho. Alguns dos trabalhos realizados até o momento envolvem a utilização da análise técnica, uso de índices de volatilidade dos mercados financeiros, técnicas de *machine learning*, análise de sentimento de mercado, análise de bolhas e análise de dados de *Blockchains* e *Exchanges*.

Dentre alguns trabalhos realizados que usam diferentes abordagens para predição de preços de criptoativos, podemos mencionar os seguintes:

- Análise de sentimento com base em pesquisas realizadas na *web*: Kristoufek (2013) concluiu em seu trabalho, que há uma forte correlação entre o preço do Bitcoin e pesquisas realizadas no *Google Trends* e *Wikipedia*. Para isto, ele utilizou a abordagem de análise de sentimento para extrair sentimentos positivos e negativos dos usuários que realizaram pesquisas sobre o Bitcoin. Foi concluído que a grande quantidade de buscas dos investidores e do público em geral durante o período de alta do Bitcoin, faz com que a demanda pela criptomoeda aumente e com isso os preços tendem a subir. Já a grande quantidade de buscas durante o período de fortes baixas, indica um crescente sentimento negativo sobre o Bitcoin, no qual o interesse em obter a criptomoeda diminui e a incerteza e o medo de grandes perdas aumenta. Esse tipo de comportamento do mercado onde o público em geral está bastante otimista durante a alta do Bitcoin, tem se caracterizado como um comportamento de formação de bolha financeira, onde quando a bolha estoura, o período de grande alta vem seguido de uma forte queda dos preços;
- Bolhas e especulação no mercado de criptoativos: no trabalho de Cheah e Fry (2015), foi concluído que assim como outras classes de ativos, o preço do Bitcoin e demais criptoativos possuem vulnerabilidade à bolhas especulativas, pois as pessoas estão mais interessadas em ter esses criptoativos como ativos de valor, do que utilizá-los como forma de pagamento ou troca. O trabalho explica que a volatilidade do Bitcoin é imprevisível, pois os preços estão sujeitos a grandes variações por causa de sentimentos do mercado. A forma como os investidores fazem a gestão de risco/retorno influencia no comportamento do mercado, pois quando estão muito otimistas, tendem a correr mais riscos. Desta forma, ocorre um grande impacto negativo quando a bolha estoura. Para chegar a estas conclusões, foram utilizados modelos probabilísticos para realização de testes de identificação de bolhas especulativas durante os períodos de grandes altas, seguidas de fortes quedas de preços;
- *Machine Learning*: no trabalho de McNally (2016), foi proposto o uso de aprendizado de máquina supervisionado com a arquitetura LSTM de redes neurais recorrentes para a estimativa da direção dos preços do Bitcoin nos próximos 30 dias. Foi concluído que o uso da LSTM teve melhor desempenho que as redes neurais recorrentes

tradicionais e o modelo estatístico ARIMA, comparados em sua pesquisa. O modelo treinado utilizando a rede neural LSTM, teve melhor desempenho em aprender com dependências temporais de longo prazo em séries temporais. Além disso, foram realizadas comparações de desempenho no treinamento dos modelos e foi concluído que o desempenho do processamento de GPU foi melhor que o processamento de CPU.

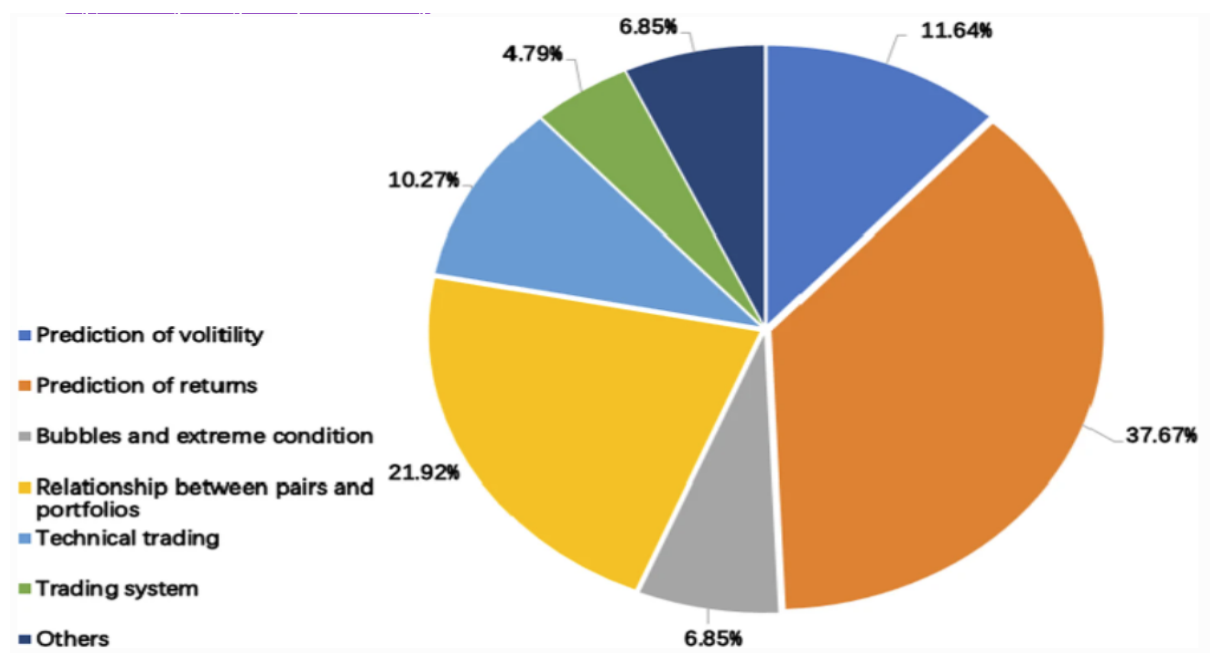
Na **Figura 10**, podemos visualizar uma linha do tempo de acontecimentos históricos importantes e de pesquisas realizadas que visam compreender o movimento de preços do mercado de criptoativos.

Em 2009, ocorreu o lançamento da primeira criptomoeda, o Bitcoin. No ano seguinte em 2010, ocorreu o surgimento da primeira *exchange* de criptomoedas. Em 2013, foram realizadas pesquisas sobre a análise de sentimentos que possuem correlação com o movimento de preços no mercado de criptoativos. Em 2014, houve pesquisas sobre a causalidade de outros mercados financeiros que impactam no mercado de criptoativos. Em 2015, ocorreu o estudo das bolhas especulativas que possuem correlação com os grandes períodos de alta e fortes quedas do Bitcoin. Em 2016, houve pesquisas sobre a aplicação de aprendizado de máquina para predição de preços de criptoativos. Em 2017, foi aplicado o aprendizado de máquina por reforço para gerenciamento de portfólio de criptoativos. Em 2019, foram realizadas também pesquisas sobre otimização de portfólio de criptoativos por meio de combinações de criptoativos em carteira. Em 2020 e 2021, muitos países começaram a cogitar a regulamentação dos criptoativos. Em 2021, a China proibiu a negociação de criptoativos no país, mas em contrapartida, El Salvador adotou o Bitcoin como moeda legal no país (FANG et al., 2022).



**Figura 10** – Linha do tempo de acontecimentos e assuntos pesquisados desde o surgimento do Bitcoin e da tecnologia *blockchain*. Fonte: Adaptado de (FANG et al., 2022), p. 43.

Na **Figura 11**, podemos ver uma distribuição dos tipos de abordagens de diversos trabalhos realizados sobre o estudo do comportamento dos preços de criptoativos. Na distribuição das abordagens, tem-se a predominância da predição de ganhos representando 37,67%, seguida da relação de pares e portfólios de criptoativos com 21,92%, depois predição de volatilidade com 11,64%, depois negociações de análise técnica com 10,27%, depois bolhas e condições extremas com 6,85%, depois outros com 6,85%, e por fim, sistemas de *trading* com 4,79%.



**Figura 11** – Distribuição dos tipos de abordagens apresentadas por Fang et al. em sua pesquisa sobre 146 artigos relacionados à negociações de criptomoedas. Fonte: (FANG et al., 2022), p. 44.

Com base nas pesquisas realizadas por estes vários autores, nas próximas seções utilizaremos algumas abordagens em conjunto para a definição de uma estratégia a ser utilizada nas negociações de criptoativos propostas neste trabalho.

## 2.3 Estratégias de decisão de investimento

Conforme vimos na **Seção 2.2**, existem diversas abordagens que visam compreender o movimento dos preços dos criptoativos. Para estas abordagens, existem diversas técnicas que podem ser aplicadas para desenvolvermos estratégias de negociações de criptoativos. Uma estratégia de negociação de criptoativos é um algoritmo com um conjunto de regras pré-definidas para compra e venda no mercado de criptoativos com a finalidade de obter lucro (FANG et al., 2022).

Com o avanço da tecnologia e a crescente quantidade de dados, o estudo e aplicação da inteligência artificial tem se desenvolvido nos diversos setores da sociedade. Na área de



finanças, muitos pesquisadores têm estudado e utilizado técnicas de *machine learning* em estratégias de predição de preços de diversos ativos, inclusive os criptoativos.

### 2.3.1 Machine Learning

*Machine Learning* é uma sub-área da inteligência artificial que explora o estudo e o desenvolvimento de algoritmos e técnicas para construção de sistemas computacionais baseados no aprendizado de dados para previsões e tomadas de decisões (SWAMYNATHAN, 2019).

Dentre as aplicações de *machine learning*, Alpaydin (2009) menciona os seguintes tipos de aprendizado que podem ser aplicados:

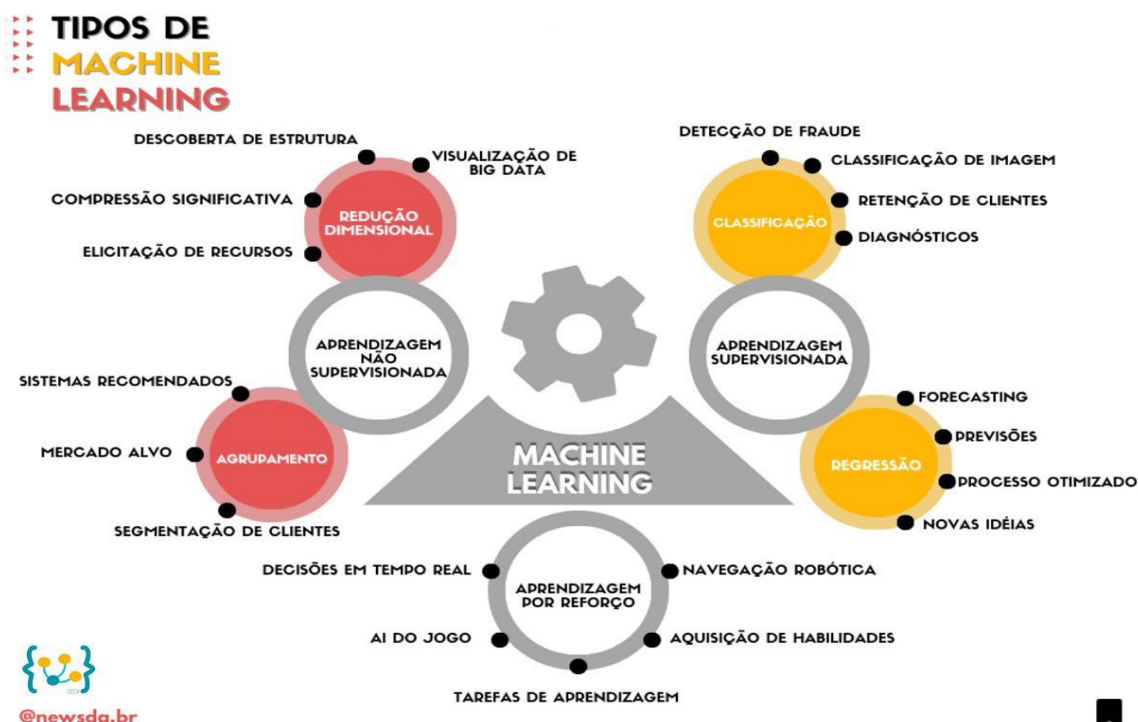
- **Aprendizado supervisionado:** neste tipo de aprendizado, o modelo é treinado utilizando os dados de entrada para gerar uma saída conhecida. O modelo precisa de dados rotulados que são utilizados para o seu treinamento, ou seja, o modelo necessita de alguém para obter os dados rotulados para que consiga prever um resultado esperado. Neste tipo de aprendizado, existem os algoritmos do tipo Regressão, onde a saída do modelo é um valor numérico contínuo. Estes tipos de algoritmos são muito utilizados para prever valores numéricos a partir dos dados de entrada. Existem também os algoritmos do tipo Classificação, onde a saída do modelo é um valor discreto, ou seja, o modelo visa a previsão de categorias ao invés de valores numéricos contínuos;
- **Aprendizado não-supervisionado:** neste tipo de aprendizado, o modelo não possui dados de entrada rotulados, e não temos uma saída conhecida. Portanto, o modelo faz uma reestruturação dos dados para gerar uma saída de objetos agrupados em um padrão encontrado. Neste tipo de aprendizado, existem os algoritmos de *Clustering*, onde o modelo realiza um agrupamento de dados semelhantes e os categoriza conforme padrões encontrados. Também existem os algoritmos de Associação, onde o modelo busca por associações entre os dados, com o objetivo de encontrar elementos que se associam dentro do conjunto de dados;
- **Aprendizado por reforço:** neste tipo de aprendizado, existe um agente inteligente que interage com seu ambiente e recebe recompensas ou penalidades pelas ações realizadas. Após cada ação o estado do agente é atualizado. O objetivo do agente é obter o máximo de pontos de recompensa, de forma que possa aprender com os erros e atingir um desempenho melhor em suas sequências de ações. Podemos dizer que o agente aprende com o *feedback* obtido de suas ações, para que consiga atingir seu objetivo. Neste tipo de aprendizado, podemos mencionar as técnicas *Markov Decision Process* e *Temporal Difference Methods*.

Os algoritmos de *machine learning* são projetados para lidar com um conjunto de dados, e a qualidade destes dados é essencial para o desempenho de aprendizado do modelo. Uma forma de melhorar a qualidade dos dados para treinamento do modelo é realizar um pré-processamento dos dados. Quando o modelo não apresenta um bom desempenho com os dados de treinamento e nem com dados de teste, dizemos que o modelo apresenta o problema de *underfitting*. O modelo pode ter um bom desempenho de aprendizado com os dados de treinamento, porém quando o modelo apresenta um desempenho ruim com dados que não são parte do treinamento, dizemos que o modelo apresenta o problema de *overfitting* (BADILLO et al., 2020).

Com o avanço da IA, muitos pesquisadores têm realizado estudos para desenvolver o aprendizado de máquina de forma semelhante ao aprendizado humano. Dentro da área de *machine learning*, podemos destacar o *Deep Learning*, que é um sub-conjunto de *machine learning* caracterizado pelo aprendizado profundo e representado pelas redes neurais artificiais de múltiplas camadas. O *deep learning* pode ser considerado como uma evolução das redes neurais tradicionais, pois se destaca por possuir uma precisão superior no aprendizado, porém exige maior processamento computacional. O *deep learning* tem se mostrado promissor em soluções para processamento de linguagem natural, visão computacional, entre outras (FANG et al., 2022).

Dentre as arquiteturas de redes neurais, algumas redes neurais que podemos mencionar são: *Recurrent Neural Network* (RNN), *Convolutional Neural Network* (CNN), *Multilayer Perceptron* (MLP), *Gated Recurrent Unit* (GRU) e *Long Short-term Memory* (LSTM). Dentre esses tipos de redes neurais, as LSTM são mais adequadas para o aprendizado com dados sequenciais como em séries temporais (KHEDR et al., 2021).

Na **Figura 12**, podemos ver os tipos de *machine learning* e quais as suas aplicações em diversos tipos de problemas. No aprendizado supervisionado, algumas das aplicações são em detecção de fraudes, classificação de imagens, retenção de clientes, diagnósticos e previsões. Já no aprendizado não-supervisionado, algumas das aplicações são em levantamento de características, visualização de *Big Data*, sistemas de recomendação e *marketing* direcionado. No aprendizado por reforço, temos aplicações em decisões de tempo real, inteligência artificial para *games*, tarefas de aprendizagem, aquisição de habilidades e navegação de robôs.



**Figura 12** – Tipos de *machine learning* e suas aplicações em diversos tipos de problemas. Obs: O termo “Elicitação de recursos” pode ser melhor descrito como “Levantamento de características”. Fonte: <<https://www.facebook.com/photo/?fbid=3082583222000059&set=pb.100057177803083.-2207520000>>

Na **Subseção 2.3.3** iremos apresentar algumas das técnicas de *machine learning* que podemos aplicar no mercado financeiro de criptoativos para a realização de investimentos.

### 2.3.2 Séries Temporais

Séries temporais são um conjunto de observações feitas sequencialmente ao longo de um período de tempo (KIRCHGÄSSNER; WOLTERS; HASSLER, 2012). A análise de séries temporais tem sido importante em diversas áreas da ciência como medicina, meteorologia, astronomia e economia.

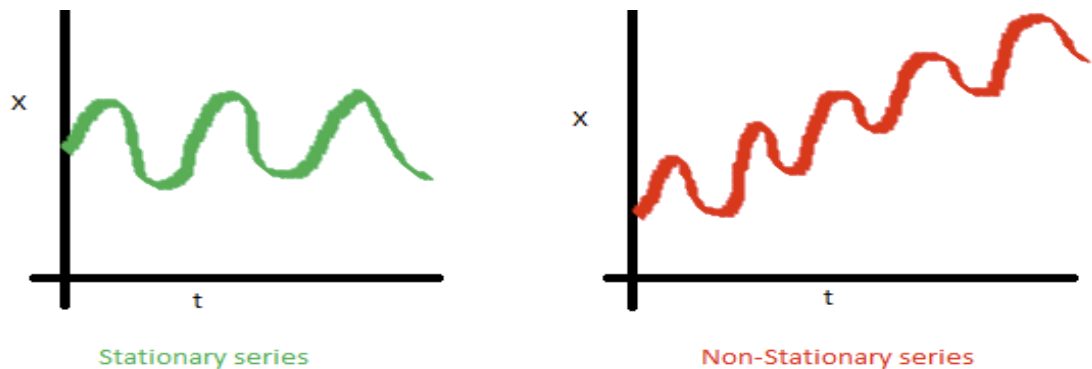
A análise de séries temporais permite extrair informações estatísticas de um conjunto de dados observados em um período de tempo com a finalidade de identificar comportamentos passados e prever comportamentos futuros. Existem diversas abordagens para se realizar análise de séries temporais, desde modelos estatísticos, até a aplicação de técnicas de *machine learning* que têm se destacado recentemente (NIELSEN, 2021).

As séries temporais podem ser de natureza estocástica ou determinística. Séries temporais estocásticas são aquelas que possuem elementos de aleatoriedade em sua composição e não se pode explicar matematicamente o comportamento destas aleatoriedades. Já as séries temporais determinísticas, não possuem elementos de aleatoriedades e podemos

formularizar matematicamente o comportamento dos dados observados. Além disso, as séries temporais podem ser classificadas como (MIGUEL, 2022):

- Estacionárias: são aquelas onde os valores observados variam próximo à uma média ao longo do tempo;
- Não estacionárias: são aquelas onde os valores observados apresentam uma variação crescente ao longo do tempo.

Na **Figura 13** podemos ver uma simples comparação entre séries temporais estacionárias e não estacionárias. Do lado esquerdo, temos um exemplo simples de visualização de uma série estacionária, onde a média dos valores de  $X$  observados, se mantém constante ao longo do tempo  $t$ . Já no lado direito, temos uma série temporal não estacionária, onde a média dos valores de  $X$  variam de forma crescente ao longo do tempo  $t$ .



**Figura 13** – Comparação entre séries temporais estacionárias e não estacionárias. Fonte: <<https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>>

Dentre os modelos estatísticos tradicionais aplicados em séries temporais, podemos mencionar os modelos *Autoregressive* (AR), *Moving Average* (MA), *Autoregressive Integrated Moving Average* (ARIMA), *Vector Autoregression* (VAR), *Autoregressive Conditional Heteroskedasticity* (ARCH) e *Generalized Autoregressive Conditional Heteroskedasticity* (GARCH) (NIELSEN, 2021).

Os mercados financeiros são uma boa área onde podemos realizar análise de séries temporais, pois podemos acompanhar o histórico de preços dos ativos ao longo do tempo e aplicar modelos para analisar o comportamento e previsão dos preços desses ativos.

A **Figura 14** mostra alguns modelos estatísticos aplicados por diversos autores no mercado de criptoativos. Alguns dos modelos como AR, ARIMA, MA, VAR foram aplicados na criptomoeda Bitcoin, utilizando dados de mercado numa frequência diária, com o período de tempo de alguns anos.

Technique	Cryptocurrency	Reference	Data type	Data frequency	Time range
ADF unit-root test and bound testing approach	Bitcoin, ethereum, dash, litecoin, and monero	Sovbetov (2018)	Market data	Weekly data	2010–2018
ARIMA, AR, and MA models	Bitcoin	Roy et al., (2018)	Market data	Daily	July 2013–August 2017
Temporal mixture model	Bitcoin	Guo and Antulov-Fantulin (2018)	Order book data	Hourly volatility	Sep 2015–April 2017
k-chainlets	Bitcoin	Akcora et al., (2018)	Market data	Daily	2009–2018
MA method	Bitcoin	Abu et al., (2019) Bakar	Cash data	Daily	October 1–December 20, 2019
Vector error correction	Bitcoin	Abbatemarco et al., (2018)	Market data	Daily	November 2013–September 2017
Dynamic ordinary least squares method	Bitcoin, ethereum, dash, litecoin, monero, and dash	Bhambhwani et al., (2019)	Market data	Weekly average	August 7, 2015–January 25, 2019
Correlation network and VAR model	Bitcoin	Giudici and Abu-Hashish (2019)	Market data closing price	Daily	May 18, 2016–April 30, 2018
FCVAR model	Bitcoin	Dos Santos Maciel and Ballini (2019)	High and low bitcoin prices	Daily	January 2012–February 2018
MF-DCCA	Bitcoin	Kim et al., (2016)	Daily volume and price index data	Daily	July 17, 2010–May 2, 2018

**Figura 14** – Trabalhos realizados com aplicação de métodos estatísticos no mercado de criptoativos. Fonte: Adaptado de (KHEDR et al., 2021), p. 9.

Entre os métodos estatísticos, o ARIMA é um dos mais populares. No trabalho de Roy, Nanjiba e Chakrabarty (2018), o autor faz uma comparação entre modelos usando os métodos MA, AR e ARIMA para predição de preços do Bitcoin. Os modelos foram treinados com dados do Bitcoin no período de 2013 a 2017. Foi concluído que o modelo utilizando o método ARIMA teve melhor desempenho para previsão dos preços do Bitcoin.

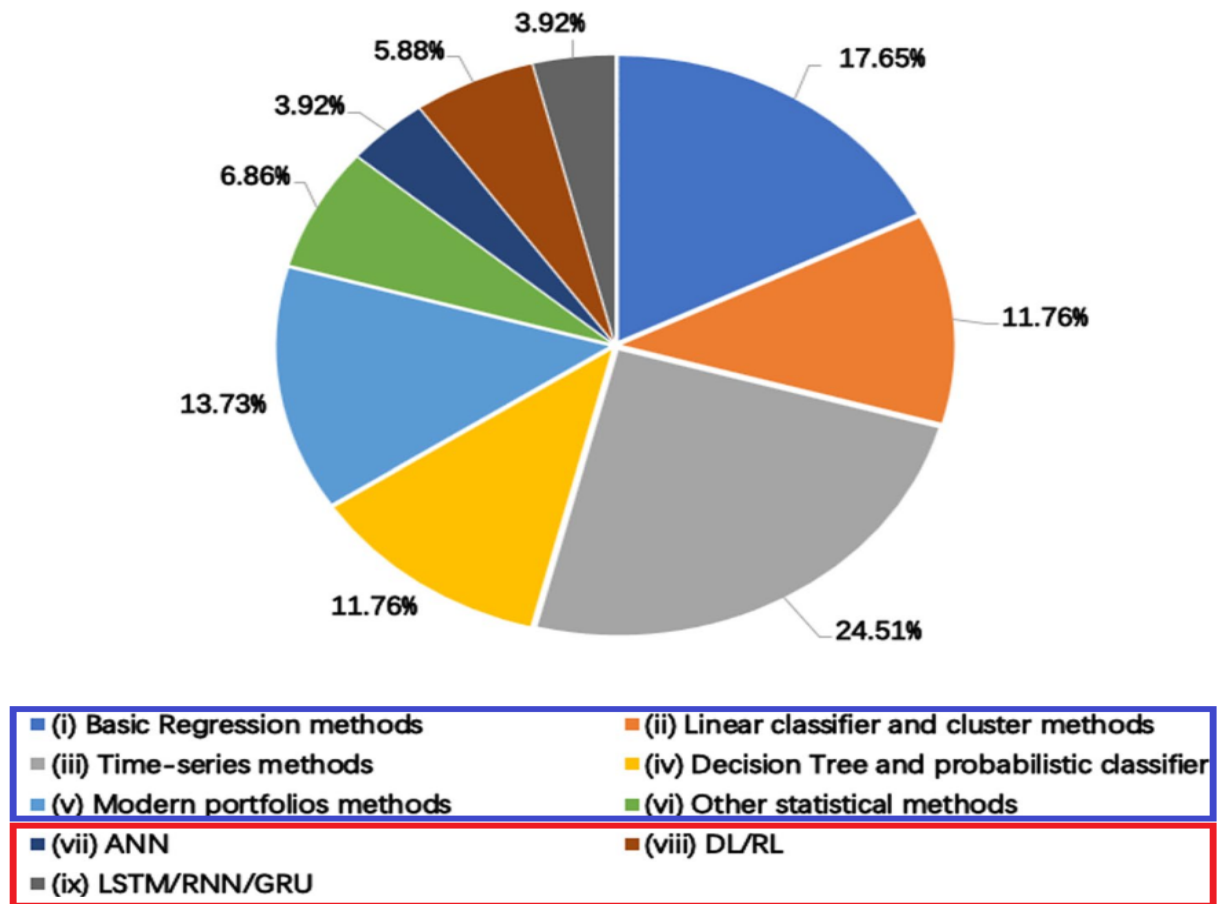
### 2.3.3 Técnicas de Machine Learning para criptoativos

O mercado de criptoativos têm ganhado popularidade nesses últimos anos. Com o avanço da inteligência artificial e *Big Data*, muitos pesquisadores têm realizado o estudo, desenvolvimento e aplicação de técnicas de *machine learning* para realizar negociações no mercado de criptoativos.

Na **Figura 15**, podemos ver uma distribuição entre os métodos estatísticos e os métodos de *machine learning* do trabalho de Fang et al. (2022), aplicados em pesquisas de negociações de criptoativos. O destaque em azul na legenda mostra os métodos estatísticos que representam 86,28%. Já o destaque em vermelho na legenda mostra os métodos de *machine learning* que são representados pelas *Artificial Neural Network* (ANN), LSTM/RNN/GRU e demais métodos de *Deep Learning* (DP) e *Reinforcement Learning* (RL), que juntos, correspondem apenas 13,72%.

Podemos perceber que os métodos estatísticos ainda têm predominância na aplicação de modelos para negociação de criptoativos. Como a IA ainda está em processo de evolução na área de finanças, ainda existem poucas pesquisas sobre a aplicação de *machine learning*

em negociações de criptoativos. Portanto, ainda existe muito conhecimento a ser explorado na área da IA aplicada ao mercado financeiro de criptoativos.



**Figura 15** – Distribuição entre os métodos estatísticos e métodos de *machine learning*.  
Fonte: Adaptado de (FANG et al., 2022), p. 46.

A predição de preços de criptoativos com aplicação de *machine learning*, consiste em dois principais componentes: dados de entrada e função objetivo. Nos dados de entrada, é onde definimos as fontes de dados do modelo, onde podemos utilizar desde dados de análise fundamental até dados de análise técnica. Podemos obter estes dados por meio de indicadores técnicos (preço, volume, variação), indicadores sociais (*Twitter*, *Google Trends*), entre outros. Já a função objetivo é composta pelas métricas e regras necessárias para a otimização do modelo. O objetivo esperado é que o modelo melhore sua precisão e antecipe a direção dos preços para obtermos melhores retornos financeiros nas negociações de criptoativos (FANG et al., 2022).

A **Figura 16** mostra o fluxo da aplicação de *machine learning* em um modelo para predição de preços de criptoativos. A primeira etapa tem a finalidade de reunir e fazer um pré-processamento dos dados para o modelo. A segunda etapa consiste em construir o modelo, definindo sua arquitetura e seus parâmetros de funcionamento. A terceira etapa é responsável pela execução e treinamento do modelo com os dados de entrada. A quarta etapa é quando o modelo realiza a predição dos preços e podemos realizar a avaliação do

resultado. A última etapa consiste em realizar a implantação do modelo. Após implantado o modelo, o mesmo pode fornecer uma contribuição para os seus dados de entrada, a partir dos resultados obtidos ao longo do tempo.



**Figura 16** – Fluxo da aplicação de *machine learning* em um modelo para predição de preços de criptoativos. Fonte: (FANG et al., 2022), p. 11.

No trabalho de Khedr et al. (2021), podemos ver algumas das técnicas de *machine learning* aplicadas em negociações de criptoativos, conforme a **Figura 17**. Estas técnicas incluem métodos de classificação, regressão, *Deep Learning* e *Reinforcement Learning* que foram comparados com outros métodos para a predição de preços de criptoativos. Podemos ver que a maioria destes trabalhos utilizaram principalmente a criptomoeda Bitcoin como referência. Além disso, alguns autores utilizaram abordagens de análise de dados de mercado, análise de transações e análise de sentimentos de mercado.

Technique	Cryptocurrency	Reference	Data type	Data frequency	Time range
Binary autoregressive tree	Bitcoin, ripple, and ethereum	Derbentsev et al., (2019)	Time-series data in log return	Daily	January 1, 2017–January 3, 2019
NN, switching regression model	STORJ	Chakraborty and Roy (2019)	STORJ token prices and related clustering coefficients of the transaction network	Daily	July 2, 2017–April 6, 2018
Linear regression, logistic regression, ANN, SVM	Bitcoin	Greaves and Au (2015)	Bitcoin transactions	Bitcoin price at 15 s intervals	Prior to April 7, 2013
Extreme gradient boosting regression tree model (XGBoost)	ZClassic, ZCash, and bitcoin private	Li et al., (2019)	Sentiment-based on Twitter and trading volume	Hourly data pricing	For a period of 3.5 weeks
Linear regression, polynomial regression, RNN, and LSTM-based analysis	Bitcoin	Mittal et al., (2019)	Social media and web search data	Average price daily	April 9, 2014–January 7, 2019
Bayesian structural time-series approach	Bitcoin	Poyser (2019)	Website data	Daily	January 2013–May 2017
Time-series analysis using bidirectional LSTM	Bitcoin and others	Mohanty et al., (2018)	Market sentiment and social sentiment	Daily data like price, additional 26 features about the blockchain of bitcoin and market, interval data	February 2016–December 2018

**Figura 17** – Trabalhos realizados com aplicação de métodos de *machine learning* no mercado de criptoativos. Fonte: Adaptado de (KHEDR et al., 2021), p. 14.

No trabalho de Aygun e Gunay (2021), foi realizada uma comparação entre os métodos estatísticos *Simple Moving Average* (SMA) e ARIMA e métodos de *machine learning* utilizando redes neurais ANN, CNN e RNN. Os dados de treinamento e testes foram obtidos dos preços do Bitcoin no período de 2013 a 2020. Foi concluído que o método

ARIMA teve melhor precisão que o método SMA. Entre todos os modelos, as redes neurais ANN de três camadas, CNN e RNN obtiveram melhor precisão na previsão dos preços, sendo que a RNN teve melhores resultados.

Neste trabalho utilizaremos uma estratégia com *machine learning*, por meio de redes neurais. Portanto, na **Subseção 2.3.4** iremos focar no estudo de redes neurais para desenvolvimento de um modelo para predição de preços de criptoativos.

### 2.3.4 Redes Neurais

Na última década, o avanço do poder de processamento dos computadores, programação paralela e a grande quantidade de dados surgindo no meio digital têm contribuído para o avanço do desenvolvimento e a aplicação das redes neurais nos diversos setores da sociedade. Muitas soluções como sistemas de recomendação, reconhecimento facial e carros autônomos, têm ganhado destaque (ACADEMY, 2022).

Rede Neural, também chamada de “Rede Neural Artificial”, é uma modelagem computacional projetada para simular o funcionamento de um cérebro biológico na execução de tarefas. Assim como um cérebro, uma rede neural artificial adquire conhecimento por meio de seu ambiente através de um processo de aprendizagem. As redes neurais possuem capacidade de generalização, ou seja, conseguem aprender e produzir resultados adequados para entradas que não estavam presentes durante o seu treinamento (HAYKIN, 2001).

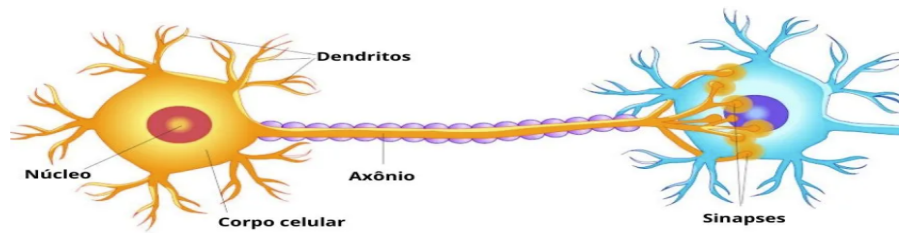
Para compreendermos o funcionamento de uma rede neural, precisamos entender como os neurônios biológicos funcionam em um sistema nervoso. Os neurônios são células do sistema nervoso responsáveis por transmitir impulsos nervosos ao cérebro. A estrutura básica de um neurônio é composto pelos seguintes elementos (PATTERSON; GIBSON, 2017):

- Dendritos: são ramificações a partir do corpo celular que recebem sinais de entrada de impulsos de outros neurônios;
- Corpo celular: região onde se concentra o núcleo do neurônio e de onde os dendritos são ramificados;
- Axônio: é um prolongamento ramificado a partir do corpo celular, de comprimento maior que os dendritos e responsável por conduzir os impulsos nervosos até os demais neurônios;
- Sinapses: são os pontos de junção entre o axônio de um neurônio e os dendritos e o corpo celular de outros neurônios. As conexões entre os neurônios estabelece o processo de aprendizagem no cérebro.

A **Figura 18** mostra a estrutura básica de um neurônio biológico. Os impulsos nervosos são recebidos como entrada através dos dendritos e são direcionados até o corpo do neurônio



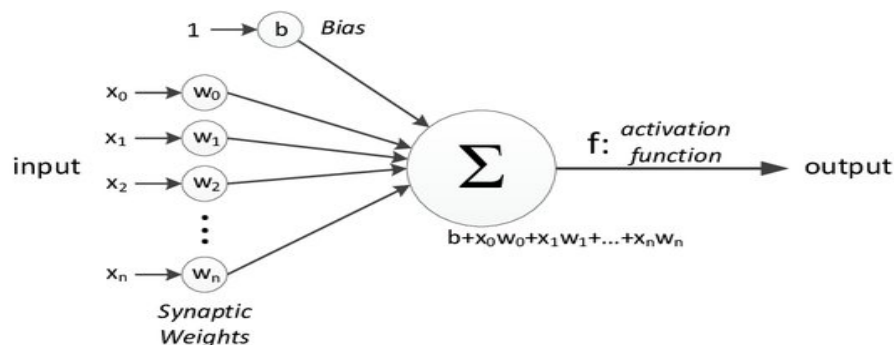
onde são conduzidos pelo axônio que faz a junção com os demais neurônios através das sinapses.



**Figura 18** – Estrutura básica de um neurônio biológico fazendo ligação com o corpo de outro neurônio por meio das conexões sinápticas. Fonte: <<https://www.biologianet.com/anatomia-fisiologia-animal/impulso-nervoso.htm>>

Em uma rede neural artificial, podemos representar um neurônio biológico por um modelo matemático chamado de Perceptron. Segundo Patterson e Gibson (2017), o perceptron é um modelo linear para classificação binária que utiliza um algoritmo de aprendizado supervisionado, sendo uma das mais simples arquiteturas de redes neurais artificiais. Foi desenvolvido em 1957 por Frank Rosenblatt, mas seu conceito básico foi introduzido por McCulloch e Pitts em 1943 através do modelo *Threshold Logic Unit* (TLU) que conseguia aprender operações lógicas AND e OR.

Na **Figura 19** podemos ver a arquitetura do perceptron. As entradas são comparadas aos dendritos e são representadas por  $x_0, x_1, \dots, x_n$ , onde cada entrada possui seu respectivo peso  $w_0, w_1, \dots, w_n$ , que pode ser modificado para melhorar o aprendizado da rede neural. Em seguida é feita a somatória da multiplicação de cada entrada com seu respectivo peso e somada com uma constante  $b = 1$  chamada de *Bias*. O valor da somatória é verificado pela função de ativação, que gera uma saída conforme critérios definidos pela função de ativação.



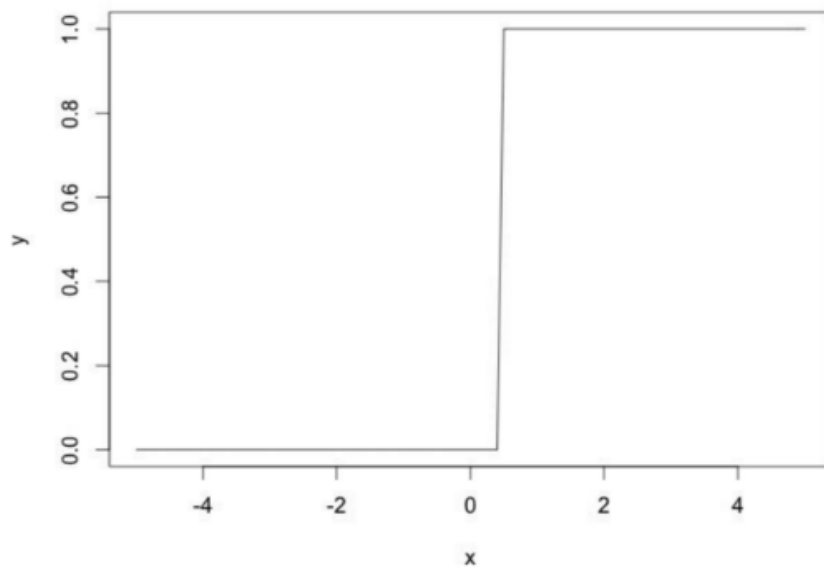
**Figura 19** – Arquitetura de um neurônio artificial simples conhecido como Perceptron. Fonte: <[https://www.researchgate.net/publication/266485234\\_Imperial\\_College\\_Spiking\\_Neural\\_Networks\\_for\\_Human-like\\_Avatar\\_Control\\_in\\_a\\_Simulated\\_Environment](https://www.researchgate.net/publication/266485234_Imperial_College_Spiking_Neural_Networks_for_Human-like_Avatar_Control_in_a_Simulated_Environment)>

Alguns dos tipos básicos de função de ativação são (HAYKIN, 2001):

Função de Limiar: a saída da função é um valor binário podendo ser 1 ou 0 ou os valores 1 ou -1. É geralmente utilizada para classificação binária. Podemos dizer que a saída  $f(x)$  do perceptron é representada pela seguinte equação:

$$f(x) = \begin{cases} 1 & \text{se } \sum_{i=0}^n x_i w_i + b > 0 \\ 0 & \text{senão} \end{cases}$$

A **Figura 20** exibe o comportamento de uma função de ativação do tipo função de limiar.

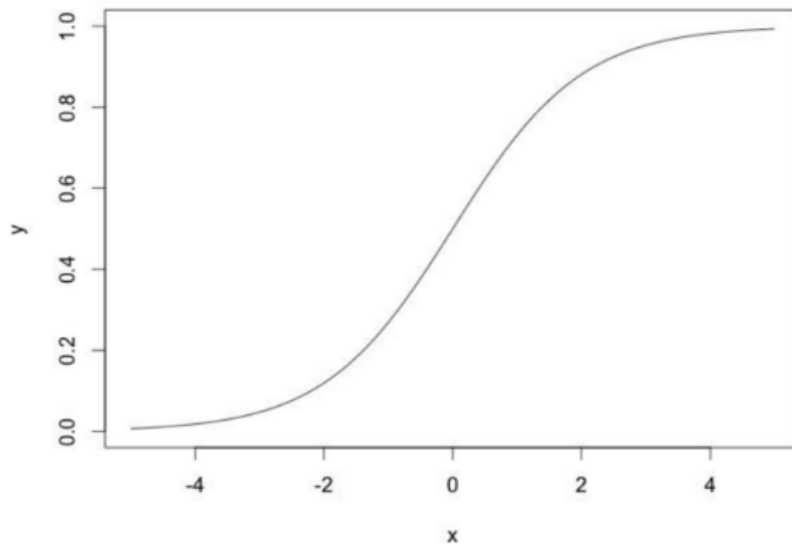


**Figura 20** – Comportamento do gráfico de uma função de ativação do tipo função de limiar. Fonte: <<http://www2.decom.ufop.br/imobilis/redes-neurais-funcoes-de-ativacao/>>

Função Sigmóide: a saída desse tipo de função é um valor contínuo entre 0 e 1, onde sua visualização no gráfico se comporta no formato da letra “S”. É o tipo mais comum utilizado na construção de redes neurais. Uma função sigmóide é representada pela seguinte equação:

$$f(x) = \frac{1}{1 + e^{-x}}$$

A **Figura 21** exibe o comportamento de uma função de ativação do tipo função sigmóide.



**Figura 21** – Comportamento do gráfico de uma função de ativação do tipo função sigmóide. Fonte: <<http://www2.decom.ufop.br/imobilis/redes-neurais-funcoes-de-ativacao/>>

O algoritmo de aprendizado do perceptron consiste em atualizar os pesos das conexões de entrada com o neurônio de saída para produzir um resultado mais preciso. Uma forma de aprendizado simples que podemos aplicar é o aprendizado por correção de erro. Após gerada a saída por meio da validação da função de ativação, é feita uma comparação do resultado com o valor alvo e gerado um valor de erro por meio do cálculo da diferença entre o valor rotulado como alvo e o valor de saída obtido. Em seguida é feita uma alteração no valor dos pesos por meio da aplicação de uma taxa de aprendizado, é iniciada uma nova iteração para produzir um novo resultado, até minimizarmos o valor de erro para próximo de zero. Podemos representar este ajuste dos pesos pela seguinte equação (GÉRON, 2019):

$$w_{ij}(\text{new}) = w_{ij} + \eta(y_j - \widehat{y}_j)x_i$$

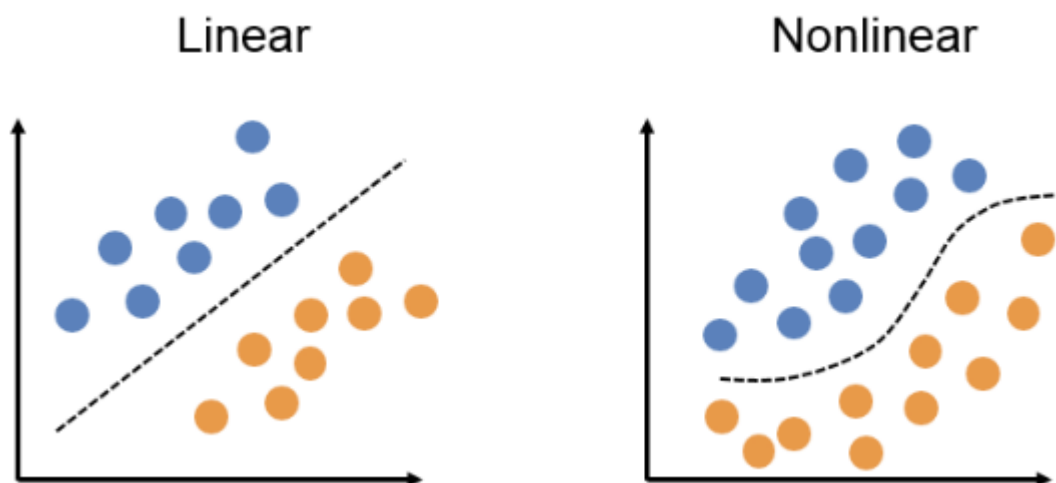
onde:

- $w_{ij}$ : é o peso da conexão entre o i-ésimo valor de entrada e o j-ésimo neurônio de saída.
- $\eta$ : constante de valor positivo que representa a taxa de aprendizado.
- $y_j$ : é a saída alvo na iteração atual do processo de treinamento.
- $\widehat{y}_j$ : é a saída obtida na iteração atual do processo de treinamento.
- $x_i$ : é o valor da i-ésima entrada da iteração atual do processo de treinamento.

O perceptron é considerado uma arquitetura simples de redes neurais, limitado para classificação somente de dados de entrada linearmente separáveis, ou seja, dados que

conseguimos separar em duas classes através de uma reta no hiperplano bidimensional (PATTERSON; GIBSON, 2017).

A **Figura 22** mostra uma simples comparação da classificação de dados linearmente separáveis e não linearmente separáveis. Podemos ver duas classes de dados, onde as saídas do perceptron podem ser 1 para representar os círculos de cor azul, e 0 para representar os círculos de cor amarela. Alguns autores utilizam a classificação binária com os valores 1 e -1 no lugar dos valores 1 e 0 respectivamente.



**Figura 22** – Comparação entre dados linearmente separáveis e não linearmente separáveis.  
Fonte: <<https://www.deeplearningbook.com.br/o-perceptron-parte-2/>>

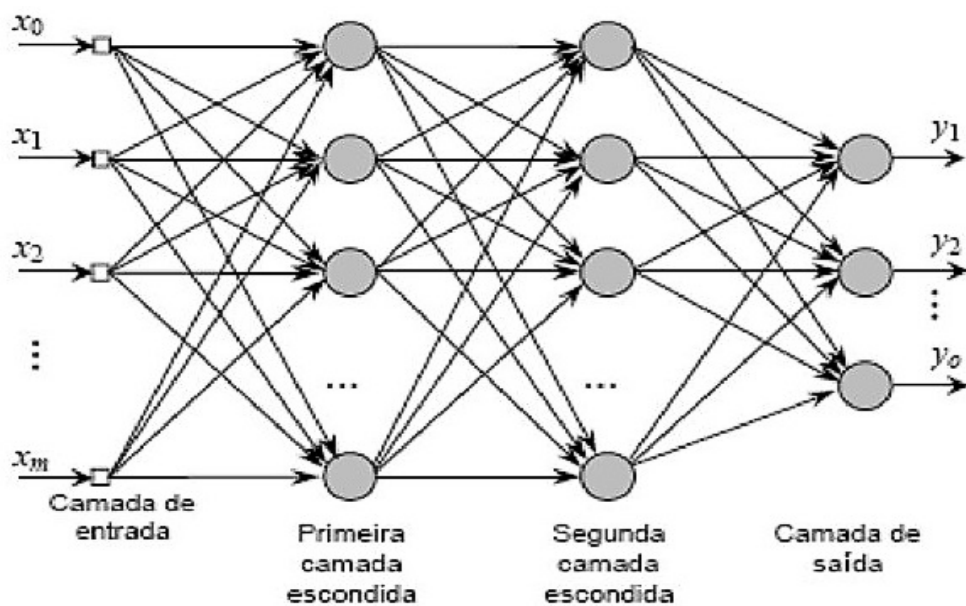
Após o desenvolvimento do perceptron, com o passar dos anos, outras arquiteturas de redes neurais surgiram para suprir suas limitações. A evolução do perceptron foi o desenvolvimento de redes neurais com múltiplas camadas, conhecidas como *Multilayer Perceptron* (MLP). As MLP são redes neurais organizadas em camadas, as quais possuem uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. As MLP possuem característica *Feed-Forward*, ou seja, os dados são processados e seguem em um sentido único em direção a saída, não retornando o fluxo para as camadas anteriores (HAYKIN, 2001).

A **Figura 23** mostra a arquitetura de uma rede neural MLP. Podemos ver que a rede neural é organizada em camadas, onde podemos destacar os seguintes tipos de camadas (PATTERSON; GIBSON, 2017):

- Camada de entrada: é a camada responsável pela entrada de dados, onde os dados de entrada são mapeados em vetores de atributos. A camada de entrada é seguida de uma ou mais camadas ocultas, onde a camada de entrada é totalmente conectada com sua camada seguinte;
- Camada oculta: uma MLP possui uma ou mais camadas ocultas. Esta camada faz o intermédio entre a camada de entrada e a camada de saída, onde extrai dados

relevantes e transmite sinais de saída para a próxima camada oculta ou camada de saída. Ela é responsável por modelar as funções de ativação da rede, onde podemos aplicar funções não lineares que não podíamos aplicar na arquitetura do perceptron;

- Camada de saída: é a camada onde obtemos a resposta após passar pela função de ativação. A saída vai depender se a rede neural está resolvendo um problema de regressão ou classificação. Para um problema de regressão são obtidos valores numéricos contínuos, e para um problema de classificação são obtidos valores discretos que representam classes de dados.



**Figura 23** – Arquitetura de uma rede neural *Multilayer Perceptron*. Fonte: (OLIVEIRA et al., 2010), p. 3.

O algoritmo de aprendizado mais comum das MLP é conhecido como *Backpropagation*, o qual possui este nome devido à sua forma de propagação de aprendizado em uma rede neural de múltiplas camadas. Na primeira etapa, os sinais de entrada se propagam camada por camada em direção à camada de saída até produzir um resultado da rede. Em seguida é feita uma comparação da saída com o resultado alvo, onde é feito o cálculo da diferença entre o valor alvo e o valor de saída para gerar um sinal de erro. Na segunda etapa, este sinal de erro é propagado para trás para ser aplicada uma correção do erro e com isso os pesos das conexões entre os neurônios são ajustados para que a próxima iteração de aprendizado produza um resultado mais próximo do esperado. O ajuste dos pesos entre os neurônios da rede neural consiste em realizar o cálculo das derivadas parciais do erro da rede neural em relação a cada peso que contribui para a rede. Para detalhes sobre o funcionamento do algoritmo *backpropagation*, consultar a obra de Haykin (2001).

No treinamento de redes neurais, as funções de custo são importantes para avaliarmos como está o processo de aprendizado de um modelo. As funções de custo servem como

métricas de avaliação e ajudam a melhorar o processo de treinamento de um modelo de forma que buscamos uma otimização para reduzir seu erro e assim melhorar sua precisão nos resultados esperados. Dadas as seguintes notações:

- $N$ : é o numero de amostras de entrada para a rede neural.
- $Y_i$ : é o  $i$ -ésimo valor alvo esperado que a rede neural atinja.
- $\widehat{Y}_i$ : é o  $i$ -ésimo valor de saída gerado pela rede neural.

Algumas funções de custo que podemos mencionar são (PATTERSON; GIBSON, 2017):

- *Mean squared error loss* (MSE): é uma função para modelos de regressão onde esperamos um valor numérico contínuo. O erro é elevado ao quadrado e é calculada a média sobre o número de amostra de dados. Podemos representar pela seguinte equação:

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - \widehat{Y}_i)^2$$

- *Mean squared log error loss* (MSLE): é uma função parecida com a MSE, onde o erro logarítmico é elevado ao quadrado. É representada pela seguinte equação:

$$MSLE = \frac{1}{N} \sum_{i=1}^N (\log Y_i - \log \widehat{Y}_i)^2$$

- *Mean absolute error loss* (MAE): é uma função para modelos de regressão. É parecida com a MSE, porém é calculado o erro absoluto. É representada pela seguinte equação:

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - \widehat{Y}_i|$$

- *Mean absolute percentage error loss* (MAPE): é uma função para modelos de regressão que calcula o erro percentual absoluto. É representada pela seguinte equação:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i - \widehat{Y}_i|}{Y_i} 100$$

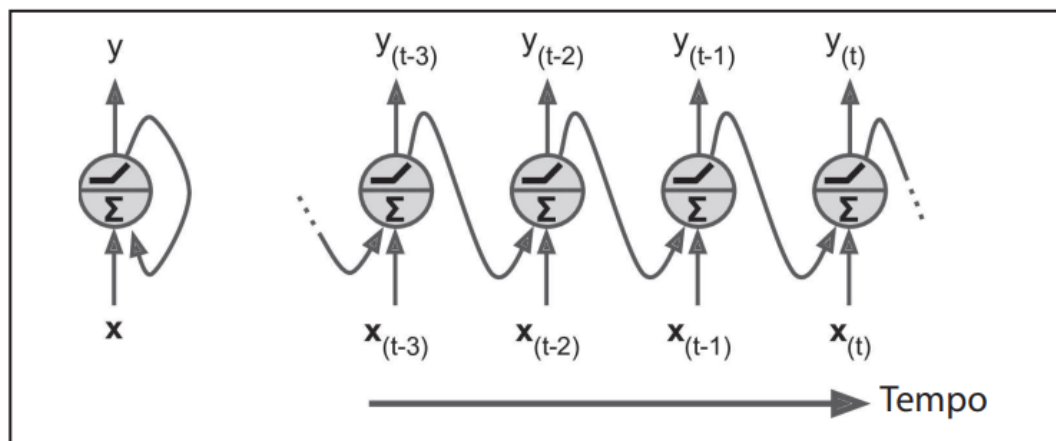
A função MSE é a mais utilizada na maioria dos casos de problemas mais simples. Para casos onde o valor de saída previsto da rede neural possui altas variações em um intervalo numérico, o mais indicado é utilizar as funções MSLE e MAPE.

### 2.3.5 Redes Neurais Recorrentes

*Recurrent Neural Network* (RNN) é uma categoria de redes neurais muito poderosa para aplicação em tarefas que envolvem dados de entrada sequenciais ao longo de um período de tempo. Este tipo de rede neural processa uma sequência de entrada e possui a capacidade de manter um vetor de estados com informações do histórico de todos os elementos anteriores da sequência (LECUN; BENGIO; HINTON, 2015).

As RNNs possuem a característica de memória semelhante ao cérebro humano, onde para resolvermos algum problema atual, podemos utilizar o conhecimento de eventos anteriores para solucionar o problema. Esta correlação de eventos passados que influenciam em eventos recentes é chamada de “dependência de longo prazo”. Os algoritmos das RNNs consideram sequência e tempo e, portanto, podemos utilizar as RNNs para problemas de séries temporais como reconhecer padrões em sequências de textos, prever preços de ativos, prever crescimento populacional, entre outros (ACADEMY, 2022).

A **Figura 24** mostra a estrutura básica de um neurônio recorrente. Ao lado direito, podemos ver o seu comportamento ao decorrer do tempo. A cada etapa do tempo  $t$ , o neurônio recebe os dados de entrada  $X_t$  com seu respectivo peso e o valor de saída do estado anterior  $Y_{t-1}$  que também possui um peso, onde o valor de saída atual  $Y_t$  está correlacionado com o estado anterior do neurônio em tempos anteriores. O elemento da rede neural recorrente que possui esta característica de manter os estados anteriores ao longo do tempo é conhecido como “célula de memória”.



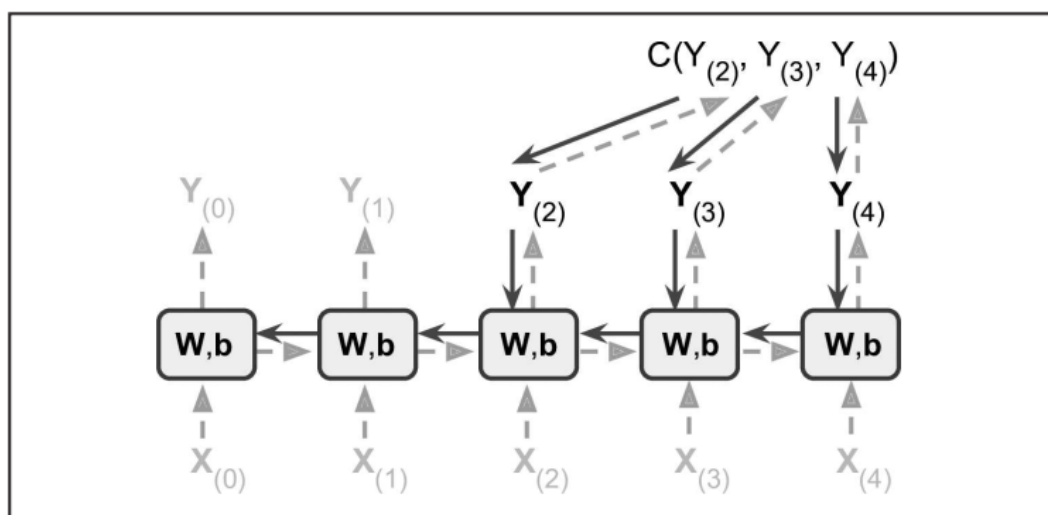
**Figura 24** – Estrutura de um neurônio recorrente. Fonte: (GÉRON, 2019), p. 398.

O algoritmo de aprendizado das redes neurais recorrentes é semelhante ao *backpropagation* das redes neurais tradicionais. O algoritmo é chamado de *Backpropagation Through Time* (BPTT) pois o aprendizado é feito através da propagação do erro e ajustes dos pesos para cada etapa de tempo anterior, visto que o erro atual é a soma dos erros em cada etapa de tempo (GÉRON, 2019).

Diferente das redes neurais tradicionais de múltiplas camadas que usam diferentes parâmetros em cada camada oculta, as RNNs compartilham os mesmos parâmetros na

camada oculta, só que em diferentes etapas de tempo. Desta forma, o estado atual da camada oculta é baseado nos dados de entrada no tempo atual e do estado anterior da camada oculta (ACADEMY, 2022).

A **Figura 25** mostra um exemplo de funcionamento do BPTT. Podemos ver o processo de propagação da rede através das setas pontilhadas, onde para cada etapa do tempo  $t$  é fornecida uma entrada  $X_t$  e gerada uma saída  $Y_t$  que alimenta a entrada do neurônio na próxima etapa de tempo  $t + 1$ . Após obtida a saída da sequência, é feita a validação da saída através de uma função de custo que considera as três últimas saídas neste exemplo. Em seguida vemos as setas sólidas que mostram a retropropagação dos gradientes de erro desta função que são calculados e, com isto os pesos do modelo são atualizados. Para mais detalhes sobre o cálculo dos gradientes de erro, consultar a obra de Goodfellow, Bengio e Courville (2016).



**Figura 25** – Funcionamento do algoritmo BPTT. Fonte: (GÉRON, 2019), p. 408.

Apesar das RNNs possuírem benefícios em comparação com as redes neurais *feed-forward*, assim como nas redes neurais MLP, as RNNs comuns com muitas camadas tendem a possuir o problema de dissipação do gradiente. Este problema ocorre durante a repropagação do gradiente de erro para a atualização dos pesos das conexões entre as camadas da rede neural ou entre as etapas de tempo em RNNs. A aplicação do gradiente que visa minimizar o erro gerado pela função de custo é conhecida como “gradiente descendente”. O problema de dissipação do gradiente ocorre quando o valor calculado do gradiente de erro diminui conforme é repropagado para as camadas iniciais da rede neural ou etapas de tempo iniciais, onde quanto maior o número de camadas ou de etapas de tempo, o cálculo do gradiente vai se aproximando de zero. Desta forma, a rede neural não consegue aprender, pois gera a impressão que está com o seu erro quase totalmente reduzido e, com isto, o processo de treinamento é finalizado (ACADEMY, 2022).

Com o passar dos anos as redes neurais recorrentes foram aprimoradas, de tal forma que foram desenvolvidas novas arquiteturas que se propuseram a resolver o problema de



dissipação do gradiente. Algumas destas arquiteturas mais populares são LSTM e GRU. Neste trabalho, utilizaremos a arquitetura LSTM na construção do modelo para predição de preços de criptoativos e portanto iremos focar somente nesta arquitetura.

*Long Short-term Memory* (LSTM) é uma arquitetura de redes neurais recorrentes proposta por Hochreiter e Schmidhuber (1997) para resolver o problema de dissipação do gradiente enfrentado pelas RNNs tradicionais. Esta arquitetura é formada por células de memória com portões que controlam o fluxo de informações dos estados anteriores e que produzem o estado atual de um neurônio, fazendo com que a rede neural mantenha dependências de longo prazo ao decorrer do tempo.

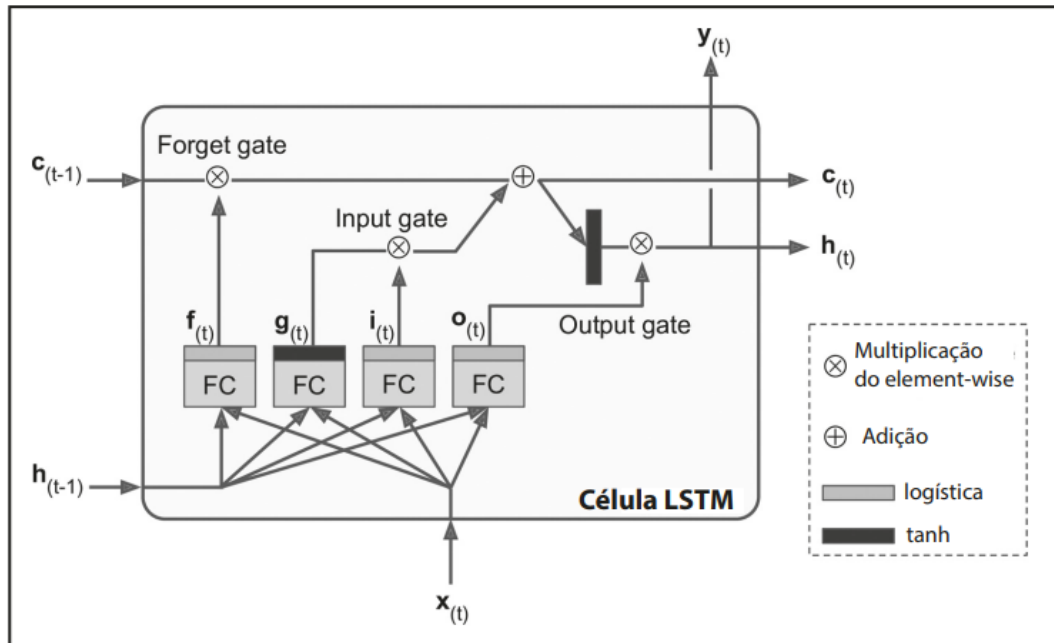
Diferentemente das RNNs tradicionais, as redes LSTM possuem entradas de dois estados recorrentes em uma célula: uma entrada do estado anterior conforme já vimos nas RNNs tradicionais e podemos considerar como o estado anterior de curto prazo, e outra entrada do estado anterior de longo prazo. O fluxo destes estados anteriores e do estado atual são controlados por três portões com as seguintes finalidades (GÉRON, 2019):

- *Forget gate*: controla quais dados do estado de longo prazo devem ser esquecidos;
- *Input gate*: controla quais dados processados da entrada atual e do estado de curto prazo devem ser adicionados ao estado de longo prazo;
- *Output gate*: controla quais dados do estado de longo prazo devem ser obtidos e considerados no estado atual e saída atual da célula.

A **Figura 26** mostra o funcionamento da arquitetura de uma célula LSTM. Podemos ver que a célula possui uma entrada do estado anterior de longo prazo  $c_{(t-1)}$ , uma entrada do estado anterior de curto prazo  $h_{(t-1)}$  e a entrada de dados atual  $x_{(t)}$ . O estado de longo prazo  $c_{(t-1)}$  atravessa a célula passando pelo *forget gate* descartando alguns dados e novos dados selecionados são adicionados ao estado de longo prazo por meio do *input gate*, fazendo com que o estado de longo prazo fique atualizado no tempo atual  $c_{(t)}$ . Além disso, o estado de longo prazo passa por uma função de ativação tangente hiperbólica, onde é filtrado pelo *output gate* que atualiza o estado atual de curto prazo  $h_{(t)}$  e também gera a saída da célula atual  $y_{(t)}$ . Podemos ver também que o estado anterior de curto prazo  $h_{(t-1)}$  e a entrada de dados atual  $x_{(t)}$  são fornecidos para quatro funções de ativação com as seguintes finalidades:

- $f_{(t)}$ : uma função logística do tipo sigmóide que gera saídas entre 0 e 1 e controla o *forget gate*;
- $g_{(t)}$ : uma função tangente hiperbólica que é semelhante a função logística, mas gera saídas entre -1 e 1. É responsável por analisar e fazer uma seleção dos dados de entrada atuais em  $x_{(t)}$  e do estado anterior de curto prazo  $h_{(t-1)}$ . É a principal função de ativação que auxilia no controle do estado atual da célula.

- $i_{(t)}$ : uma função logística que controla o *input gate*, onde este define quais dados selecionados pela função  $g_{(t)}$  irão ser adicionados ao estado de longo prazo.
- $o_{(t)}$ : uma função logística que controla o *output gate*.



**Figura 26** – Arquitetura de uma célula LSTM. Fonte: (GÉRON, 2019), p. 420.

Podemos concluir que uma célula LSTM tem capacidade de aprender a identificar uma entrada importante por meio do *input gate* e armazená-la no estado de longo prazo pelo tempo necessário por meio do *forget gate* e depois conseguir extrair dados importantes do estado de longo prazo por meio do *output gate*.

Com o passar dos anos foram surgindo algumas variações da LSTM, como é o caso da arquitetura GRU introduzida por Cho et al. (2014). A GRU é uma variante menos complexa e simplificada da LSTM.

Atualmente as redes neurais LSTM tem sido muito importantes na área de IA, apresentando soluções para diversos problemas como em tarefas de reconhecimento de fala, reconhecimento de caligrafias, tradução de idiomas, legenda de imagens, previsões em séries temporais, entre outras. Portanto, esta arquitetura de redes neurais é uma das melhores opções para desenvolvermos nosso modelo de predição de preços de criptoativos.

## 2.4 Ferramentas de suporte à decisão de investimento

Negociações e investimentos em mercados financeiros exigem um certo conhecimento sobre o assunto para que as decisões tomadas gerem bons resultados. Existem alguns sistemas e ferramentas de suporte à decisão que podemos utilizar para auxiliar na análise e tomada de decisão ao realizar negociações e investimentos.

Os sistemas de suporte à decisão são ferramentas que auxiliam na produtividade e capacidade de tomada de decisão, pois fornecem melhores *insights* por meio de informações importantes obtidas. Estas ferramentas possuem um conjunto de funcionalidades que utilizam consultas de várias bases de dados, inteligência artificial e apresentação dos dados de forma analítica (ARIF, 2013).

No mercado de criptoativos, existem várias ferramentas que são importantes para quem deseja explorar este mercado e deseja obter uma boa produtividade e experiência. Existem ferramentas para realizar análise gráfica, acompanhar notícias do mercado, gerenciar o portfólio, analisar os movimentos e sentimento do mercado, entre outras. Algumas destas ferramentas que podemos mencionar são (LEARN, 2022):

- *TradingView*: é uma plataforma *online* e *mobile* para análise gráfica de diversos ativos além de criptoativos. Podemos utilizar ferramentas gráficas para fazer análises, usar indicadores técnicos e também fazer simulações de *trading* para testar estratégias. É uma plataforma que pode ser usada gratuitamente, tendo algumas limitações em comparação com a versão paga que, por exemplo, possibilita o uso de mais indicadores técnicos e mais flexibilidade na criação de alertas sobre o comportamento dos preços;
- *Quadency*: é uma ferramenta *online* e *mobile* que integra várias *exchanges* em uma única interface. Possui funcionalidades de automatização de negociações, gerenciamento e análise de portfólio e uma *API* que fornece vários dados obtidos de *exchanges*. Possui acesso gratuito, porém, com limitações em relação às versões *premium* que são pagas por meio de *holding* do seu *token* QUAD;
- *CoinMarketCap*: é uma ferramenta *online* e *mobile* de acesso gratuito que permite obter dados de diversos criptoativos em diversas *exchanges*. Esta ferramenta reúne diversos dados e informações relevantes para análise técnica e fundamentalista como preços históricos, acesso a *white papers*, código fonte, comunidades, entre outros.

Algumas ferramentas interessantes para análise de sentimento do mercado de criptoativos são (CHINAGLIA, 2022):

- *LunarCrush*: é um *website* de inteligência social para criptoativos. Podemos realizar análise dos criptoativos mais comentados nas redes sociais como *Twitter*, *Reddit*, *Youtube*, *Medium*. Ele possui métricas para analisar os sentimentos das redes sociais, volume de negociações, volatilidade, entre outros. O *website* é de acesso gratuito com recursos limitados, mas possui planos *premium* pagos por meio de *holding* do seu *token* LUNR;
- *Santiment*: é uma plataforma *online* que analisa o sentimento de mercado de alguns criptoativos por meio de dados *on-chain*, ou seja, dados obtidos de sua *blockchain*. A

plataforma possui várias métricas e indicadores que medem o humor dos investidores para obter perspectivas das direções dos preços. É uma plataforma de acesso gratuito com limitações, sendo necessário obter algum plano pago para acesso a mais recursos e métricas.

Outras ferramentas encontradas em pesquisas pela *Internet* são:

- *Coin Metrics*: é uma plataforma *online* que obtêm dados *on-chain*<sup>2</sup> para análise de sentimento e comportamento de alguns criptoativos. Também possui métricas e indicadores que auxiliam os investidores a tomarem decisões. A plataforma possui acesso gratuito com limitações, onde é necessário obter um plano pago para obter recursos mais avançados;
- *Crypto Fear & Greed Index*: é um *website* de acesso gratuito que utiliza um índice que mede os sentimentos de mercado do Bitcoin e demais criptoativos por meio de diferentes fatores como volatilidade, volume e momento de mercado, análise de redes sociais como o *Twitter*, dominância do Bitcoin e dados de pesquisas do *Google Trends*. O sentimento do mercado de criptomoedas é medido em três classes: medo, neutro e ganância. É uma ferramenta bastante usada por investidores para identificar momentos adequados de aportes e vendas de criptoativos;
- *CoinGecko*: é um *website* de acesso gratuito parecido com o *CoinMarketCap*. Fornece dados históricos para análise fundamentalista, dados de preço, volume e derivativos de criptoativos. Ele também possui uma *API* que podemos requisitar vários dados relevantes sobre os criptoativos;
- *Coinglass*: é uma ferramenta *online* e *mobile* para análise de dados dos derivativos de criptoativos. Ela fornece várias métricas e gráficos para análise do interesse do público, volatilidade, volume de negociações em *exchanges*, índice de bolha, entre outros. Possui uma *API* que podemos requisitar vários dados relevantes sobre os derivativos de criptoativos.

Um fator importante para a tomada de decisão em negociações de criptoativos é o gerenciamento de risco. Uma das formas importantes de gerenciar o risco é através de ordens de *stop*. Uma ordem de *stop* tem a finalidade de comprar ou vender um ativo quando ele atingir um determinado preço. É uma maneira dos investidores e *traders* gerenciarem o risco de forma que suas emoções não afetem e prejudiquem significativamente seus capitais de investimento e lucros. As formas mais comuns de utilização de ordens de *stop* são (LEMOS, 2017):

---

<sup>2</sup> Oriundo de *blockchains*.

- *Stop loss*: é uma ordem de *stop* com a finalidade de minimizar perdas. É uma forma de sair da operação com uma pequena perda antes que esta aumente e acumule um grande prejuízo que comprometa o capital do investidor e sua estabilidade emocional. Por exemplo, quando se compra um ativo, o *stop loss* é um valor limite do preço que definimos até onde ele pode cair, onde é realizada uma ordem de venda caso o preço chegue ou ultrapasse este limite. Algumas formas de definir este limite do preço são por meio de um valor percentual de perda máxima sobre o valor investido na operação de compra e por meio de pontos estratégicos identificados com a análise técnica;
- *Stop gain*: é uma ordem de *stop* com a finalidade de garantir ganhos. É uma forma de sair da operação com lucro antes que ele diminua ou vire uma perda. Por exemplo: quando se compra um ativo, definimos um valor limite de preço que ele pode subir para realizarmos a venda e assim garantir o lucro. Muitos investidores e traders não colocam este limite por ganância pois acham que o ativo irá valorizar mais que o esperado, onde ocorrem casos em que o preço volta a cair e com isto perdem o lucro que estavam tendo. Algumas formas de definir este limite do preço são por meio de um valor percentual de ganho máximo sobre o valor investido na operação de compra e por meio de pontos estratégicos identificados com a análise técnica;
- *Stop móvel (trailing stop)*: é uma ordem de *stop* dinâmica com a finalidade de manter o lucro em andamento, onde o *stop loss* é reajustado conforme o preço do ativo se movimenta a favor. Por exemplo: quando se compra um ativo no preço de US\$100, não definimos um *stop gain*. Porém é definido um valor de preço em US\$103 que será o gatilho para início do *trailing stop* e um valor ou percentual de ajuste para o *stop loss* que neste caso definimos 5%. Se o preço do ativo subir até o valor de gatilho em US\$103, o *trailing stop* é ativado e o valor do *stop loss* é definido para 5% abaixo do preço atual (US\$97,85), e a partir de então conforme o preço sobe, o preço de *stop loss* também sobe proporcionalmente. O *stop loss* não é ajustado quando o preço do ativo cai e a venda só é realizada quando o *stop loss* é acionado. Se o preço do ativo subir bastante, vai chegar um momento em que o *stop loss* vai estar numa faixa de preço que será considerada como lucro e caso o investidor sofra um *stop loss*, ainda assim estará com algum lucro. Também é interessante criar uma ordem de *stop loss* separada para caso o *trailing stop* não for acionado, o *stop loss* evitará grandes perdas caso o preço começar a cair muito;

Existem algumas ferramentas que são utilizadas em conjunto com a análise técnica para criar estratégias de tomada de decisão de compra e venda de ativos. Algumas destas ferramentas que podemos mencionar são os indicadores de análise técnica que podem ser utilizados para identificar tendências de direção dos preços, reversão de tendências e pontos interessantes de compra e venda. Alguns destes indicadores são: *Rate of Change*

(ROC), *Relative strength index* (RSI), *Gann Box*, retração de Fibonacci, médias móveis, entre outros (FANG et al., 2022).

## 2.5 Bibliotecas e *frameworks*

Neste trabalho, propomos o desenvolvimento de um sistema de predição de preços para realizarmos negociações de criptoativos de forma simulada. Será um sistema *web* com uma interface analítica e, a partir das previsões e sugestões obtidas do módulo de *machine learning*, iremos executar negociações simuladas de compra e venda. O objetivo inicial é conseguir realizar as operações básicas de negociações no mercado à vista de forma a obter lucros e minimizar riscos de grandes perdas. Após o sistema estar com suas principais funcionalidades implementadas, iremos implementar as negociações reais para serem executadas em uma *exchange* e pensaremos em implementar futuras melhorias.

Para implementar o módulo de *machine learning* de uma rede neural LSTM, iremos utilizar a linguagem python, que é uma linguagem de programação de alto nível muito popular para desenvolvimento de aplicações de *machine learning* devido à grande quantidade de bibliotecas e *frameworks* disponíveis. Algumas bibliotecas e *frameworks* que podemos utilizar na construção do nosso modelo de *machine learning* são:

- *TensorFlow*: é uma plataforma de código fonte aberto com bibliotecas e ferramentas para construção de modelos de *machine learning* para computadores, *web* e dispositivos móveis. Possui uma biblioteca bastante popular desenvolvida pelo *Google* e oferece APIs para as linguagens de programação python e C++ e também suporta processamento paralelo com CPUs e GPUs (TENSORFLOW, 2022);
- *Keras*: é uma API de alto nível escrita em python e bastante popular para construção de modelos de *deep learning* de forma ágil e simples. Tornou-se a principal API do *TensorFlow 2* para aproveitar o máximo dos recursos e produtividade do mesmo. É bastante usada em experimentos com modelos de redes neurais convolucionais e recorrentes. É uma API que também pode ser utilizada em outras bibliotecas como *Theano* e *Microsoft Cognitive Toolkit* (CNTK) (CHOLLET, 2017);
- *NumPy*: é uma biblioteca de código fonte aberto para python criada em 2005. É muito utilizada em computação científica pois permite operações matemáticas e operações lógicas com vetores e matrizes multidimensionais. É importante na construção e treinamento de modelos de *machine learning*, pois estes sempre trabalham com dados em *arrays* (NUMPY, 2022);
- *Matplotlib*: é uma biblioteca de código fonte aberto para python para criação de visualizações estáticas, animadas e interativas. É bastante utilizada com a biblioteca

*NumPy* para visualização de dados e análises gráficas de resultados dos treinamentos dos modelos de *machine learning* (MATPLOTLIB, 2022).

As principais bibliotecas que iremos utilizar para construção do modelo de *machine learning* são *TensorFlow*, *Keras* e *NumPy*. *TensorFlow* e *Keras* fazem parte do mesmo ecossistema, muito popular entre a comunidade de desenvolvedores de *machine learning*.

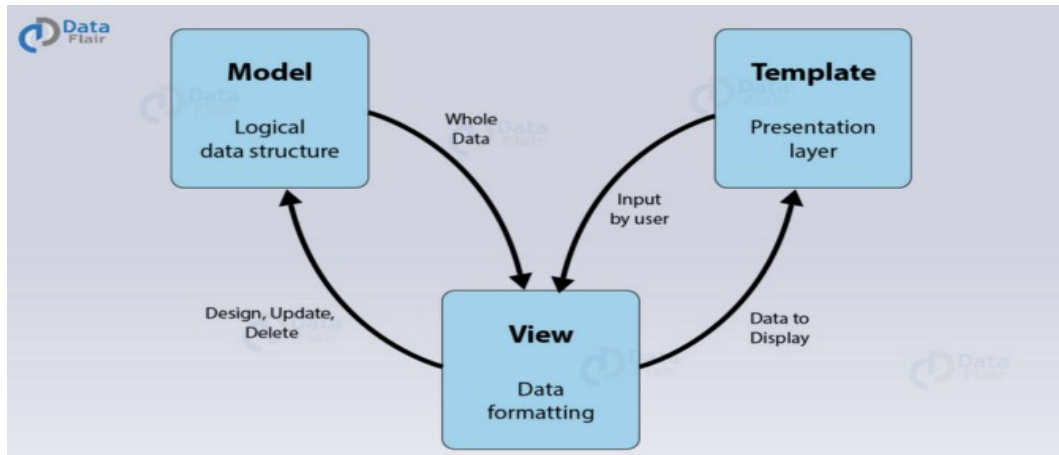
A **Figura 27** mostra algumas outras bibliotecas de código aberto similares ao *TensorFlow*, destacando quais as linguagens suportadas, as plataformas suportadas, os desenvolvedores e o ano de desenvolvimento.

Biblioteca	API	Plataformas	Iniciado por	Ano
Caffe	Python, C++, Matlab	Linux, macOS, Windows	Y. Jia, UC Berkeley (BVLG)	2013
Deeplearning4j	Java, Scala, Clojure	Linux, macOS, Windows, Android	A. Gibson, J. Patterson	2014
H2O	Python, R	Linux, macOS, Windows	H2O.ai	2014
MXNet	Python, C++, others	Linux, macOS, Windows, iOS, Android	DMLC	2015
TensorFlow	Python, C++	Linux, macOS, Windows, iOS, Android	Google	2015
Theano	Python	Linux, macOS, iOS	Universidade de Montreal	2010
Torch	C++, Lua	Linux, macOS, iOS, Android	R. Collobert, K. Kavukcuoglu, C. Farabet	2002

**Figura 27** – Bibliotecas de código aberto para *Deep Learning* e *Machine Learning*.  
Fonte: (GÉRON, 2019), p. 236.

Para o sistema *web*, também iremos utilizar a linguagem python no *back-end*, um banco de dados com PostgreSQL e alguns *frameworks* e bibliotecas para auxiliar tanto no *back-end* e *front-end*. Algumas bibliotecas e *frameworks* que podemos utilizar são:

- *Django*: é um *framework* de código fonte aberto para desenvolvimento de sistemas para *web* de forma ágil e escalável. O *Django* possui uma arquitetura de *software* conhecida como MVT (*Model View Template*). Nesta arquitetura, existe o *Model* que fornece uma interface com o banco de dados, o *Template* que é responsável pela interface com o usuário e a *View* que é responsável pela implementação da lógica do sistema e faz a ligação entre o *Model* e o *Template* (DJANGO, 2022). A **Figura 28** mostra o funcionamento da arquitetura MVT. Podemos ver que o *Template* é responsável pela interação com o usuário, onde o mesmo fornece uma entrada que é direcionada para a *View* que faz o tratamento desta entrada e faz a comunicação com o banco de dados por meio do *Model*, onde este retorna os dados obtidos pela *View* para serem apresentados ao usuário no *Template*;
- *React*: é uma biblioteca javascript para criação de interfaces de usuário para *web*. Sua aplicação é baseada em componentes, onde são atualizados e renderizados somente os componentes necessários conforme os dados mudam (REACT, 2022);



**Figura 28** – Funcionamento da arquitetura MVT do *Django*. Fonte: <<https://www.educba.com/django-architecture/>>

- *Vue.js*: é um *framework* javascript para criação de interfaces de usuário para *web*. Também é baseado em componentes que utilizam HTML, Css e javascript para criação de páginas *web* dinâmicas (VUE.JS, 2022);
- *jQuery*: é uma biblioteca javascript leve e simples para manipulação de elementos em páginas HTML, eventos e requisição de dados assíncrona no *front-end* (JQUERY, 2023).
- *Cryptocurrency Exchange Trading Library* (CCXT): é uma biblioteca de código fonte aberto para negociação de criptoativos com integração em mais de 100 *exchanges*. É uma biblioteca com suporte para as linguagens de programação javascript, python e PHP. É muito utilizada por desenvolvedores, cientistas de dados e analistas financeiros na construção de algoritmos e modelos para negociações de criptoativos (CCXT, 2022).
- *Celery*: é uma biblioteca python *open source* que permite a comunicação entre diversas aplicações por meio de mensagens e com suporte para agendamento e processamento de grandes quantidades de tarefas em fila de forma distribuída e assíncrona (CELERY, 2023).

Dentre as principais bibliotecas e *frameworks* para o sistema *web*, iremos utilizar o *Django*, *jQuery* e CCXT. Como python é a principal linguagem de programação deste projeto, iremos utilizar o *framework Django*. Já o *jQuery* é uma biblioteca simples para criar interações em páginas HTML de forma simplificada com a linguagem javascript. O uso da biblioteca CCXT é interessante pois oferece uma API que integra negociações de criptoativos em várias *exchanges* sem precisar fazer implementações diferentes para cada *exchange*.



# 3 IMPLEMENTAÇÕES E EXPERIMENTOS

Neste capítulo serão apresentados os experimentos realizados durante o desenvolvimento dos modelos de predição de preços, as tecnologias utilizadas no desenvolvimento do sistema *web* e as suas funcionalidades implementadas.

## 3.1 Projeto do sistema

Antes de realizar as implementações do sistema, foi elaborado um projeto com as especificações dos requisitos funcionais (**RF**) e não-funcionais (**RNF**), e também a modelagem do banco de dados que poderá ser consultada no **Apêndice B**. O projeto foi nomeado como Crypto DSS (*Crypto Decision Support System*). Os seguintes requisitos foram elaborados:

- **RF-01:** Realizar predição de preços de criptoativos;
- **RF-02:** Apresentar criptoativos para investimento;
- **RF-03:** Analisar o sentimento de mercado do criptoativo a ser investido;
- **RF-04:** Criar ordem de compra;
- **RF-05:** Criar ordem de venda;
- **RF-06:** Criar ordem de *stop*;
- **RF-07:** Mostrar o portfólio de criptoativos;
- **RF-08:** Utilizar o indicador técnico: RSI;
- **RF-09:** Utilizar o indicador técnico: SMA20;
- **RF-10:** Criar ordem de compra e venda automatizada;
- **RF-11:** Disponibilizar o histórico das ordens cadastradas;
- **RNF-01:** Fornecer dados de predição de preços por API;
- **RNF-02:** Refazer o treinamento dos modelos a cada 2 horas;
- **RNF-03:** Fornecer interface de usuário de forma responsiva;
- **RNF-04:** Fornecer autenticação de usuário;

- **RNF-05:** Utilizar a linguagem python no *back-end*;
- **RNF-06:** Utilizar banco de dados relacional no *back-end*.

Todos os requisitos, exceto os requisitos RNF-01 e RNF-02, foram implementados. O requisito RNF-01 não foi implementado devido a limitação de tempo. As predições foram implementadas para serem gravadas no banco de dados, porém não estão sendo disponibilizadas para acesso via API. O requisito RNF-02 não foi implementado, pois foi identificado que não seria necessário refazer o treinamento em um curto período de tempo de horas, visto que novos dados para treinamento estarão disponíveis somente após o fechamento diário dos preços e o mais interessante seria refazer o treinamento após dias ou meses de mais dados.

## 3.2 Implementações

A primeira etapa das implementações foi iniciada pelo desenvolvimento dos modelos de predição de preços. Em seguida foram iniciadas as implementações das funcionalidades do sistema, a integração dos modelos de predição de preços com o sistema e os testes das funcionalidades implementadas. As subseções seguintes descrevem os requisitos funcionais implementados.

### 3.2.1 Realizar predição de preços de criptoativos

Neste requisito, foram desenvolvidos modelos de predição de preços com a utilização de diferentes arquiteturas de redes neurais como ANN, RNN e LSTM para que fosse possível fazer uma comparação entre os modelos e identificar qual modelo apresenta melhores resultados para o nosso problema. O desenvolvimento dos modelos foi realizado com a ferramenta JupyterLab, que é de livre uso e permite executar códigos python, tarefas de *machine learning* e *data science* de forma interativa em um navegador *web* (JUPYTER, 2023). As principais bibliotecas da linguagem python utilizadas foram:

- *yfinance*: utilizada para a coleta dos dados históricos dos preços dos criptoativos;
- *numpy*, *pandas* e *scikit-learn*: utilizadas para o pré-processamento dos dados;
- *tensorflow*: utilizada para a construção, treinamento e avaliação dos modelos;
- *matplotlib*: utilizada na criação de gráficos para visualizar o erro e os resultados das predições de preços.

As **Tabelas 1, 2 e 3** apresentam o melhor resultado de cada modelo a partir da execução de 5 treinamentos e testes de predição para o próximo dia, próximos 3 dias e próximos 7 dias respectivamente. Os resultados são apresentados de forma ordenada do

melhor valor da métrica MAPE para o pior valor. Além disso, as tabelas mostram alguns dos parâmetros utilizados por cada modelo como: *Timesteps* (dias anteriores usados como entrada para as previsões) e a quantidade de dias previstos. A **Tabela 2** e a **Tabela 3** não possuem o modelo ANN devido esta arquitetura de rede neural não suportar múltiplos *Timesteps* como os modelos RNN e LSTM. Os dados utilizados no treinamento e avaliação dos modelos foram separados de forma que 90% foram utilizados para o treinamento e 10% foram utilizados para avaliação. Os testes foram realizados de forma padronizada e automatizada com a linguagem python devido a grande quantidade de tempo demandada conforme o aumento de parâmetros combinados. O tempo aproximado de um único treinamento dos modelos testados foi: 19s para o modelo ANN, 23s para o modelo RNN e 30s para o modelo LSTM. Já o tempo aproximado para fazer a previsão de todo o conjunto dos dados de testes foi: 0.80 ms para o modelo ANN, 0.148 ms para o modelo RNN e 0.337 ms para o modelo LSTM. A quantidade de testes de cada modelo e os diferentes parâmetros testados podem ser consultados no **Apêndice A**.

Predição: próximo dia				
Modelo	Criptoativo	Timesteps	Dias seguintes	MAPE
LSTM	BTC-USD	7	1	1.6314
RNN	BTC-USD	7	1	1.7019
ANN	BTC-USD	1	1	1.7598

**Tabela 1** – Melhor resultado de cada modelo nas previsões de preços do Bitcoin para o próximo dia. Os resultados foram obtidos com base na média da métrica MAPE na execução de 5 treinamentos e testes.

Predição: próximos 3 dias				
Modelo	Criptoativo	Timesteps	Dias seguintes	MAPE
LSTM	BTC-USD	7	3	2.8901
RNN	BTC-USD	3	3	2.9406

**Tabela 2** – Melhor resultado dos modelos LSTM e RNN nas previsões de preços do Bitcoin para os próximos 3 dias. Os resultados foram obtidos com base na média da métrica MAPE na execução de 5 treinamentos e testes.

Predição: próximos 7 dias				
Modelo	Criptoativo	Timesteps	Dias seguintes	MAPE
RNN	BTC-USD	3	7	4.7844
LSTM	BTC-USD	7	7	4.8382

**Tabela 3** – Melhor resultado dos modelos LSTM e RNN nas previsões de preços do Bitcoin para os próximos 7 dias. Os resultados foram obtidos com base na média da métrica MAPE na execução de 5 treinamentos e testes.

A configuração do computador utilizado para o treinamento e testes dos modelos foi a seguinte:

- Sistema operacional: Windows 11, 64 bits;
- Processador: AMD Ryzen 5 3600X - 6 núcleos, 3.8 GHz;
- Memória RAM: 32GB, DDR4.

### 3.2.2 Apresentar criptoativos para investimento

Este requisito tem por finalidade, apresentar quais os criptoativos disponíveis para *trade*. Neste trabalho foi decidido que trabalharíamos com os *top 4* criptoativos do *website* CoinMarketCap. Os criptoativos selecionados foram o Bitcoin (BTC), Ethereum (ETH), BNB e XRP. O modelo LSTM foi selecionado para realizar as previsões de preços de todos estes criptoativos. A *stable coin* BUSD também foi adicionada na lista de criptoativos, mas somente para compor o portfólio dos criptoativos em carteira do usuário.

A **Figura 29** exibe a listagem dos criptoativos disponíveis para *trade*. São mostrados o preço atual em tempo real de cada criptoativo e os preços previstos gerados pelos modelos LSTM para o próximo dia, próximos 3 dias e próximos 7 dias. Também são apresentados os botões de ações para acessar a página onde são realizados os *trades* e o botão que acessa os detalhes do criptoativo no *website* CoinMarketCap.

Criptoativos disponíveis para Trade						
#	CRYPTO	PREÇO ATUAL	PRÓXIMO 1D	PRÓXIMOS 3D	PRÓXIMOS 7D	AÇÕES
1	BTC	\$26969.200	\$27143.904	\$27077.175	\$27098.624	 
2	ETH	\$1617.020	\$1622.037	\$1612.944	\$1609.826	 
3	BNB	\$214.200	\$213.755	\$214.899	\$224.296	 
4	XRP	\$0.516	\$0.511	\$0.500	\$0.501	 
5	BUSD	\$1.000	-	-	-	 

**Figura 29** – Página de apresentação dos criptoativos disponíveis para *trade*. Fonte: Elaboração própria.

As principais tecnologias e bibliotecas utilizadas neste requisito foram:

- *Black Dashboard Django*: é um *template* moderno e *open source* baseado no *framework* *Bootstrap 4* e disponível para uso com o *framework* *Django*. O *template* está disponível de forma gratuita e também existe a versão *premium* com mais recursos (TIM, 2023);

- *Websocket*: protocolo que permite estabelecer comunicação em tempo real e bidirecional entre uma aplicação cliente e servidor sem a necessidade de abrir novas conexões ao servidor durante as requisições de dados (MELNIKOV; FETTE, 2011). Os dados de preços em tempo real são coletados da *exchange* Binance sem custos financeiros, por meio de uma conexão *websocket* com a linguagem javascript no *front-end*;
- *jQuery*: biblioteca javascript utilizada para manipulação dos elementos das páginas HTML de forma interativa, fazer requisições dos dados ao servidor *Django* e inseri-los nas páginas *web*.

### 3.2.3 Analisar o sentimento de mercado do criptoativo a ser investido

Neste requisito foi implementada uma forma de analisar o sentimento de mercado dos criptoativos a serem investidos. Para isto, foi preciso utilizar os dados de sentimento do *website* LunarCrush que podem ser acessados por meio de uma API que disponibiliza alguns dados de acesso gratuito e outros de acesso pago. Também pode-se obter os dados por meio de *web scraping*<sup>1</sup>. Outra fonte de dados utilizada para analisar o sentimento de mercado de criptoativos em geral, foi o índice CVI (*Crypto Volatility Index*). Este índice é baseado no índice VIX e indica o nível de volatilidade do mercado de criptoativos (INDEX, 2023). Neste índice, um valor muito alto indica alta volatilidade e alto risco, enquanto um valor baixo, indica pouca volatilidade e menor risco. Após analisar o gráfico histórico do índice em comparação com o gráfico do Bitcoin, foram definidas as seguintes faixas de riscos:

- $CVI > 0$  e  $CVI \leq 80$ : Risco baixo;
- $CVI > 80$  e  $CVI \leq 100$ : Risco médio;
- $CVI > 100$  e  $CVI \leq 120$ : Risco alto;
- $CVI > 120$ : Risco muito alto.

A **Figura 30** exibe o sentimento de mercado do criptoativo selecionado (ETH) ao clicar no símbolo do mesmo. Na parte inferior esquerda são mostrados os níveis de sentimento *bullish* e *bearish* em percentual, conforme os dados do *website* LunarCrush. Na parte inferior direita é mostrado o valor do índice CVI que indica qual o nível de volatilidade e risco do mercado de criptoativos em geral.

As principais tecnologias e bibliotecas utilizadas foram:

- *Python*: linguagem de programação utilizada para a coleta e tratamento dos dados de sentimento por meio das bibliotecas *requests* e *json*;

---

<sup>1</sup> Extração de dados de páginas *web* de forma automatizada.

- *PostgreSQL*: banco de dados utilizado para armazenar os dados de sentimentos a serem utilizados pelo sistema.



**Figura 30** – Análise de sentimento do criptoativo ETH. Fonte: Elaboração própria.

### 3.2.4 Criar ordem de compra

O objetivo deste requisito é permitir que o usuário do sistema possa cadastrar ordens de compra manualmente conforme sua análise realizada. Para isto, foi desenvolvida uma página para a realização de *trades*, que inclui recursos como: analisar o gráfico e indicadores técnicos do criptoativo selecionado para *trade*, cadastrar ordens de compra e venda e acompanhá-las. Neste requisito, logo após o gráfico, é apresentado um formulário com opções de cadastrar ordens do tipo “*Limit*” e “*Market*”. Ordens *Limit* são aquelas que o usuário pode definir um preço limite para que seja realizada uma compra (geralmente um preço limite para comprar abaixo do preço atual). Após cadastrada este tipo de ordem, ela é enviada à *exchange* e aguarda o preço do criptoativo chegar ao preço limite definido para que seja executada automaticamente pela *exchange* e o *trade* seja efetivado. As ordens do tipo *Market* são aquelas que o usuário pretende realizar uma compra no valor atual do preço sem definir um preço limite, e define somente um montante do criptoativo ou um valor a ser investido. Após cadastrada, a ordem é executada pela *exchange* imediatamente. Neste projeto trabalhamos no mercado à vista com a *exchange* Binance, uma das *exchanges* mais conhecidas e com o maior volume de negociações atualmente conforme a listagem no *website* CoinMarketCap (2022).

A **Figura 31** exhibe o formulário para realizar o cadastro de ordens de compra. Podemos ver que tal formulário permite selecionar ordens do tipo “*Limit*” e “*Market*”, um *checkbox* opcional para selecionar uma estratégia de compra automatizada, os campos referentes ao

preço, montante, total e o botão para enviar o formulário. Após realizado o cadastro de uma ordem, é mostrada uma notificação se a ordem foi executada com sucesso ou teve algum erro. Além disso, as ordens do tipo “*Limit*” que estão abertas mas ainda não foram executadas pela *exchange*, são listadas acima do formulário na seção de ordens abertas.

DATA	ID	SYMBOL	TIPO	LADO	STATUS	QTD	PREÇO	OPÇÕES
21/09/2023, 22:23:17	3531827	BTC/BUSD	limit	buy	open	0.01	26000	×

**Figura 31** – Cadastro de ordens de compra. Fonte: Elaboração própria.

As principais tecnologias e bibliotecas utilizadas foram:

- *CCXT*: biblioteca da linguagem python utilizada para fazer os envios e consultas das ordens à *exchange*;
- *jQuery*: biblioteca javascript utilizada para manipular os elementos HTML do formulário e enviá-lo para o servidor *Django*;
- *PostgreSQL*: banco de dados utilizado para armazenar as ordens cadastradas e os *trades* efetivados pela *exchange*.

### 3.2.5 Criar ordem de venda

Este requisito tem o objetivo de permitir o usuário realizar o cadastro de ordens de venda manualmente após ter feito sua análise com o auxílio da predição de preços, do gráfico e alguns indicadores técnicos exibidos na página. O requisito é semelhante ao descrito na **Subseção 3.2.4**. Neste requisito, a diferença está nos tipos de ordens disponíveis para seleção.

A **Figura 32** exibe o formulário de cadastro das ordens de venda. Podemos ver que o formulário é semelhante ao de ordens de compra, mas com 3 tipos de ordem de venda: “*Limit*”, “*Market*” e “*Stop*”. A venda do tipo “*Stop*”, é um tipo de venda diferente que será descrito na **Subseção 3.2.6**.

The screenshot displays a trading application interface. At the top, there are three tabs: 'Ordens abertas' (highlighted in purple), 'Ordens pendentes', and 'Listagem de ordens'. Below the tabs is a table with the following columns: DATA, ID, SYMBOL, TIPO, LADO, STATUS, QTD, PREÇO, and OPÇÕES. The table contains two rows of data:

DATA	ID	SYMBOL	TIPO	LADO	STATUS	QTD	PREÇO	OPÇÕES
21/09/2023, 22:23:17	3531827	BTC/BUSD	limit	buy	open	0.01	26000	×
22/09/2023, 00:29:23	3554751	BTC/BUSD	limit	sell	open	0.01	30000	×

Below the table, there is a 'Buy' button (purple) and a 'Sell' button (orange). Under these buttons are three tabs: 'Limit' (purple), 'Market' (purple), and 'Stop' (purple). Below the tabs is a checkbox labeled 'Estratégia automatizada?'. Below the checkbox are three input fields: 'Preço' (with a '\$' icon), 'Montante' (with a '₹' icon), and 'Total' (with a '₹' icon). At the bottom, there is a 'Sell' button (orange).

**Figura 32** – Cadastro de ordens de venda. Fonte: Elaboração própria.

### 3.2.6 Criar ordem de *stop*

Este requisito é semelhante ao descrito na **Subseção 3.2.5**. Neste requisito, foram implementados os principais tipos de ordens de *stop* para operações de venda, com a finalidade de auxiliar o usuário a fazer uma gestão de risco em suas operações. Foram implementados 3 tipos de ordens de *stop* com os nomes: “*STOP-LOSS-SELL*”, “*STOP-GAIN-SELL*” e “*TRAILING-STOP-GAIN-SELL*”. Estes tipos de ordens de *stop* foram mencionados na **Seção 2.4**.

A **Figura 33** exibe o formulário para realizar o cadastro dos tipos de ordens de *stop*. Podemos ver que após selecionado o tipo de ordem como *stop*, é habilitada a opção de selecionar o tipo de *stop*. Após selecionado o tipo *TRAILING-STOP-GAIN-SELL*, é apresentado um *slider* para definir o valor percentual do *trailing delta*. Em seguida são apresentados os campos para informar o preço limite onde será iniciado o *trailing stop*, o montante a ser vendido e o campo total que assim como nos requisitos de compra e venda, é apenas para mostrar o valor total da operação a ser realizada no preço limite.



**Figura 33** – Cadastro de ordens de *stop*. Fonte: Elaboração própria.

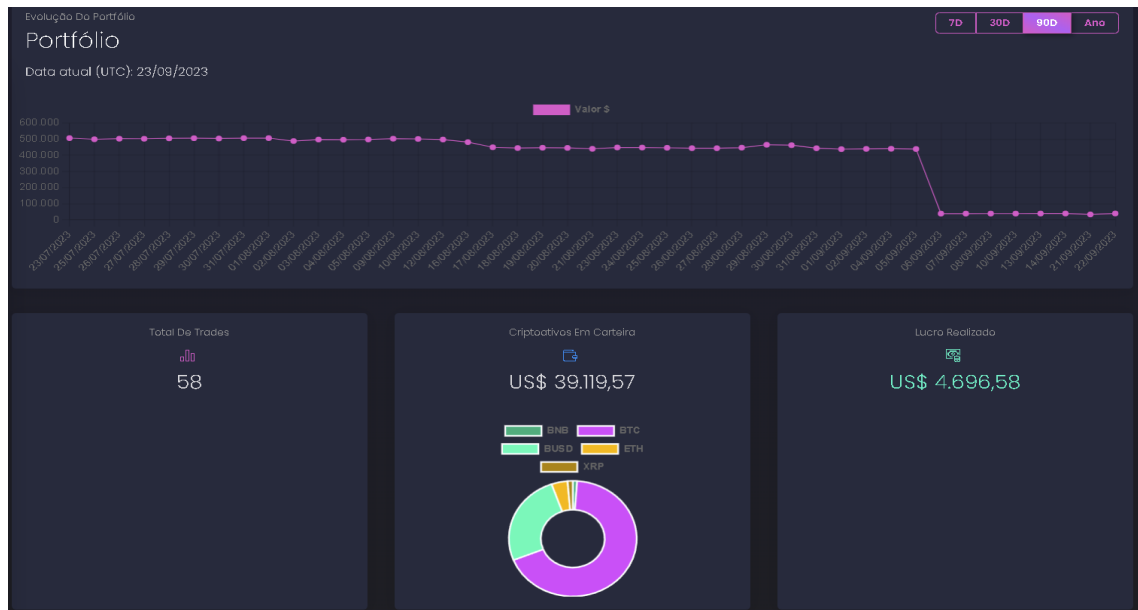
### 3.2.7 Mostrar o portfólio de criptoativos

Este requisito tem o objetivo de mostrar um *dashboard* com informações sobre o portfólio de criptoativos do usuário ao acessar a página *home* do sistema. O *dashboard* exibe algumas informações como: evolução do portfólio do usuário, total de *trades* em um período, criptoativos em carteira e o lucro realizado em um período.

A **Figura 34** exibe o *dashboard* com as informações do portfólio do usuário. Podemos ver a exibição de um gráfico de linhas com a evolução do portfólio do usuário no período de 90 dias, um *card* com o número de *trades* realizados neste mesmo período, um gráfico de rosca com a distribuição dos criptoativos em carteira e o valor do lucro realizado no período (receitas com vendas - despesas com compras).

As principais tecnologias e bibliotecas utilizadas foram:

- *CCXT*: biblioteca da linguagem python utilizada para consultar o balanço de criptoativos do usuário na *exchange* Binance;
- *Chart.js*: biblioteca *open source* da linguagem javascript, utilizada para a construção dos gráficos de linha e rosca utilizados no *dashboard* (CHART.JS, 2023);
- *PostgreSQL*: banco de dados utilizado para armazenar diariamente o valor de portfólio do usuário conforme os criptoativos disponíveis no sistema e presentes na carteira do usuário;
- *jQuery*: biblioteca javascript utilizada para requisitar os dados de valor do portfólio do usuário no servidor *Django* e apresentar os dados no *front-end*.



**Figura 34** – Apresentação do portfólio do usuário. Fonte: Elaboração própria.

### 3.2.8 Utilizar o indicador técnico RSI

Este requisito foi implementado com a finalidade de auxiliar o usuário a fazer suas análises com o indicador técnico RSI, bastante conhecido nos mercados financeiros. Além disso, o indicador foi selecionado para fazer parte de uma estratégia automatizada descrita na **Subseção 3.2.10**.

O indicador RSI (*Relative Strength Index*), nos permite identificar se o preço atual de um ativo está sobrecomprado (acima do normal) ou sobrevendido (abaixo do normal) e portanto está próximo de ocorrer uma reversão/correção do preço. Ele compara a variação média de preços dos períodos de alta com a variação média dos períodos de baixa. O indicador apresenta um valor em uma escala de 0 a 100 com base em um período de 14 fechamentos do preço (Exemplo: 14 dias se estiver analisando o preço no gráfico diário, 14 horas se estiver analisando no gráfico de 1 hora). Quando o RSI se aproxima de 100, significa que a média das baixas é próxima de zero e que, nos últimos 14 fechamentos do preço, eles têm sido positivos seguidamente. Já para o RSI próximo de 0, significa que a média das altas é próxima de zero e que, nos últimos 14 fechamentos do preço, eles têm sido negativos seguidamente. Por padrão, o indicador considera um valor menor do que 30 como ativo sobrevendido. Já um valor maior do que 70 como ativo sobrecomprado (LEMOS, 2017). O RSI é calculado da seguinte forma:

$$RSI = 100 - (100 / (1 + RS))$$

onde:  $RS = MG / MP$

$MG$  = soma absoluta das variações positivas do período / 14

$MP$  = soma absoluta das variações negativas do período / 14

A **Figura 35** exibe o indicador RSI na página de *trade* do sistema. Após selecionado o criptoativo desejado, é carregado o gráfico do criptoativo e algumas informações dos indicadores de sentimento e técnicos. O *card* destacado em vermelho mostra o indicador RSI no *timeframe* de 1 hora, que será utilizado pela estratégia automatizada. Já na parte inferior destacada no gráfico, o indicador RSI é mostrado no formato de um gráfico de linhas no mesmo *timeframe* do gráfico principal, com o indicador dentro de um intervalo de valores, sendo que quando ultrapassa a linha tracejada superior, o preço do criptoativo está sobrecomprado e, quando ultrapassa a linha tracejada inferior, o preço está sobrevendido.



**Figura 35** – Visualização do indicador RSI na página de *trade* do sistema. Fonte: Elaboração própria.

As principais tecnologias e bibliotecas utilizadas foram:

- *Advanced Real-Time Chart Widget*: *widget* javascript que possibilita sua integração em *websites* e disponibilizado gratuitamente pela plataforma TradingView. Neste requisito, o *widget* foi personalizado e utilizado para mostrar o gráfico dos preços dos criptoativos e alguns indicadores técnicos;
- *python-tradingview-ta*: biblioteca utilizada no *back-end* do sistema para coletar dados dos indicadores RSI e SMA20, que são extraídos da plataforma TradingView para serem utilizados na estratégia automatizada. Isto foi necessário, pois o *widget* apenas integra ao *front-end* e não disponibiliza os dados utilizados pelo mesmo.

### 3.2.9 Utilizar o indicador técnico SMA20

Este requisito tem o objetivo de auxiliar o usuário a fazer suas análises com o indicador técnico SMA de 20 períodos. O SMA (*Simple Moving Average*) é um indicador técnico que calcula a média dos  $n$  últimos fechamentos do preço para identificar a tendência atual

de um ativo. Quando o preço atual está acima da média móvel e continua subindo, o ativo está em uma tendência de alta. Se o preço atual está abaixo da média móvel e continua caindo, o ativo está em uma tendência de baixa. Quando o preço atual está oscilando abaixo ou acima da média móvel, mas não está assumindo uma direção (alta ou baixa), o ativo está com uma tendência neutra. Existem vários tipos de médias móveis de diferentes períodos que são utilizadas conforme o tipo de estratégia do investidor. A fórmula para cálculo do indicador SMA é dada por (LEMOS, 2017):

$$SMA = (PF1 + PF2 + PF3 + \dots + PF_n) / n$$

onde:

$PF$  = preço de fechamento

$n$  = quantidade de fechamentos do período

A **Figura 36** exibe a utilização do indicador SMA20 na página de *trade* do sistema. Na parte superior da página é mostrado o indicador no *timeframe* de 1 hora. Já no gráfico dos preços que está em um *timeframe* diário, o indicador é apontado por uma seta e apresenta o formato de uma linha amarela.



**Figura 36** – Visualização do indicador SMA20 na página de *trade* do sistema. Fonte: Elaboração própria.

As principais tecnologias e bibliotecas utilizadas foram as mesmas apresentadas na Subseção 3.2.8.

### 3.2.10 Criar ordem de compra e venda automatizada

Este requisito tem a finalidade de demonstrar o uso da predição de preços em combinação com os indicadores de sentimento e os indicadores técnicos RSI e SMA20 na implementação

de uma estratégia automatizada para as operações de compra e venda. A estratégia não foi testada para saber se apresenta bons retornos em investimentos, mas será avaliada com mais atenção em trabalhos futuros. A estratégia foi pensada com o objetivo de realizar operações de compra e venda mais conservadoras, utilizando a tendência de alta do preço como a principal base das operações. Portanto, espera-se que as operações de compra e venda sejam feitas quando a predição de preços e a análise técnica com indicadores estiverem indicando uma alta do preço.

Para uma ordem de compra ou venda ser executada automaticamente, precisará atender as seguintes condições definidas:

- A média ponderada das predições para o próximo dia, 3 dias e 7 dias deve ser maior que o preço atual e apresentar um percentual de alta de no mínimo 1%. Os pesos maiores serão para os dias mais próximos do dia atual (Exemplo: próximo dia tem peso 3, próximos 3 dias tem peso 2 e os próximos 7 dias tem peso 1). Esta abordagem foi escolhida devido as predições dos modelos serem mais precisas para os dias mais próximos ao dia atual;
- O índice de sentimento do criptoativo precisa indicar que o sentimento atual não está *bearish*, ou seja, o criptoativo está sendo bem visto nas mídias sociais;
- O índice CVI precisa indicar no máximo um risco médio ( $CVI \leq 100$ ). Ele indica que o mercado de criptoativos não está em um momento muito volátil;
- O indicador SMA20 precisa indicar que o criptoativo está em uma tendência de alta no gráfico diário (preço atual acima da média móvel de 20 períodos). Portanto, enquanto o preço não estiver abaixo da média móvel, ele tende a continuar subindo;
- O indicador RSI no *timeframe* de 1 hora precisa indicar que o preço está em um momento adequado para compra ( $RSI < 30$ ) ou em um momento adequado para venda ( $RSI > 70$ ). Para ordens de compra, a idéia é fazer a compra quando o preço estiver recuando para uma faixa de preço mais barata, e esperamos que ocorra um novo impulso de alta do preço e a tendência de alta continue. Para ordens de venda, a idéia é vender quando o preço estiver sobrevalorizado, ou seja, já teve uma boa valorização e o preço está próximo de uma forte correção;
- Caso definido um preço limite para compra ou venda automatizada, o preço atual do criptoativo deverá estar dentro de tal limite definido pelo usuário. O objetivo é definir um limite de preço até onde a estratégia poderá operar caso o investidor não esteja confiante que a mesma será adequada em uma outra faixa de preços.

Visto que a estratégia possui várias condições a serem satisfeitas, ela pode ser considerada como conservadora. Porém como há várias condições, ela tende a ter poucas execuções

em comparação com outra estratégia com menos condições, podendo até apresentar uma boa assertividade, mas pode não apresentar uma boa rentabilidade. Esta estratégia foi elaborada inicialmente com o objetivo de colocar em prática os assuntos estudados neste trabalho e, portanto, não é uma recomendação para compra ou venda.

A **Figura 37** exibe o formulário de cadastro de uma ordem de compra do tipo “*Limit*” com uma estratégia automatizada. Podemos ver que para utilizar uma estratégia automatizada, é preciso marcar o *checkbox* e selecionar a estratégia disponível. Em seguida, para ordens do tipo “*Limit*”, é preciso definir um preço limite até onde a estratégia poderá ser executada (preço atual  $\geq$  preço limite). Depois de informar o montante e confirmar a compra, a ordem ficará com o *status* “*waiting*” e aparecerá na seção de ordens pendentes. Isto significa que a ordem está aguardando todas as condições da estratégia serem satisfeitas para ser executada na *exchange*.

DATA	ID LOCAL	SYMBOL	TIPO	LADO	STATUS	QTD	PREÇO LIMITE	OPÇÕES
29/08/2023, 21:34:00	411	BTC/BUSD	SELL-MARKET-STRATEGY	SELL	waiting	0.010000	-	×
25/09/2023, 21:02:32	417	BTC/BUSD	BUY-LIMIT-STRATEGY	BUY	waiting	0.010000	25000.00	×

Se for uma operação "Limit", a estratégia de compra será executada somente se o preço atual for  $\geq$  que o preço limit. Se for uma operação "Market", a estratégia de compra será executada independente da faixa do preço atual.

☒ Estratégia automatizada?

Rsi + SMA20 + Sentiment

\$ Preço

₪ Montante

₹ Total

Buy

**Figura 37** – Cadastro de ordem de compra automatizada. Fonte: Elaboração própria.

As principais tecnologias e bibliotecas utilizadas foram:

- *python-binance*: biblioteca utilizada no *back-end* para coletar os preços dos criptoativos em tempo real por meio de conexões *websocket* com a *exchange* Binance;
- *python-tradingview-ta*: biblioteca utilizada no *back-end* do sistema para coletar dados dos indicadores RSI e SMA20, extraídos da plataforma TradingView;
- *Celery*: biblioteca python utilizada para o processamento de tarefas em fila de forma distribuída e assíncrona. Esta ferramenta permitiu criar diversas tarefas distribuídas para serem agendadas e processadas em segundo plano pelas *threads* do servidor, melhorando a responsividade do sistema e permitindo diferentes aplicações comunicarem entre si de forma assíncrona;

- *RabbitMQ*: *software open source* que faz o intermédio entre diferentes aplicações distribuídas por meio de mensagens. Ele foi integrado com o *Celery* com a finalidade de receber as mensagens de tarefas do *Celery* e distribuí-las para as filas de tarefas do servidor;
- *Redis*: banco de dados *NoSQL* de armazenamento em memória. É uma ferramenta bastante usada como *cache* por muitos desenvolvedores. Neste projeto, o *redis* foi utilizado principalmente para armazenar os dados de tarefas em *cache*, possibilitando um rápido acesso aos dados compartilhados por diferentes implementações do sistema;
- *Docker*: ferramenta *open source* utilizada para o desenvolvimento e gerenciamento de aplicações em *containers* de forma isolada e escalável. Neste projeto, esta ferramenta possibilitou instalar e gerenciar de forma simples, diferentes *softwares* do *back-end*.

### 3.2.11 Disponibilizar o histórico das ordens cadastradas

Este requisito tem o objetivo de listar o histórico de ordens feitas pelo usuário. Ao acessar a página de *trade* do sistema e selecionar o criptoativo desejado, o usuário poderá acessar a opção de listar todas as ordens, sendo redirecionado para a página de listagem de ordens.

A **Figura 38** exibe a página de listagem de ordens do criptoativo Bitcoin negociado na paridade com o dólar (BTC/BUSD). Podemos ver que são listadas todos os tipos de ordens com os diferentes *status*. A listagem na tabela permite ordenar pela coluna desejada, escolher a quantidade de registros por página e também buscar por alguma informação específica na barra de pesquisa. Para consultar as ordens de outros pares de negociação, basta clicar no *link* de voltar da página e refazer o processo.

As principais tecnologias e bibliotecas utilizadas foram:

- *jQuery*: biblioteca javascript utilizada para requisitar no *back-end*, todas as ordens cadastradas pelo usuário;
- *DataTables*: *plugin* da biblioteca *jQuery* para a construção de tabelas interativas e dinâmicas em páginas HTML.

DATA	ID LOCAL	SYMBOL	TIPO	LADO	STATUS	QTD	QTD PREENCH.	QTD REST.	PREÇO	TOTAL APROX.	ESTRATÉGIA
25/09/2023 21:02:32	417	BTC/BUSD	BUY-LIMIT-STRATEGY	BUY	waiting	0.010000	None	None	\$	\$	Rsi + SMA20 + Sentiment
22/09/2023 00:29:22	414	BTC/BUSD	SELL-LIMIT	SELL	closed	0.010000	0.010000	0.000000	\$30000,0000	\$300,0000	
21/09/2023 22:23:17	413	BTC/BUSD	BUY-LIMIT	BUY	closed	0.010000	0.010000	0.000000	\$26000,0000	\$260,0000	
29/08/2023 21:34:00	411	BTC/BUSD	SELL-MARKET-STRATEGY	SELL	waiting	0.010000	None	None	\$	\$	Rsi + SMA20 + Sentiment
03/08/2023 01:19:00	396	BTC/BUSD	STOP-LOSS-SELL	STOP	closed	0.100000	0.100000	0.000000	\$29000,0000	\$2900,0594	
27/07/2023 19:46:11	338	BTC/BUSD	BUY-LIMIT	BUY	canceled	0.100000	0.000000	0.100000	\$29000,0000	\$2900,0000	
11/07/2023 19:18:13	326	BTC/BUSD	BUY-MARKET-STRATEGY	BUY	closed	0.010000	0.010000	0.000000	\$30610,0100	\$306,1001	Rsi + SMA20 + Sentiment
08/07/2023 21:13:29	324	BTC/BUSD	SELL-LIMIT-STRATEGY	SELL	closed	0.010000	0.010000	0.000000	\$30559,3100	\$305,5931	Rsi + SMA20 + Sentiment
08/07/2023 20:58:44	323	BTC/BUSD	SELL-MARKET-STRATEGY	SELL	closed	0.010000	0.010000	0.000000	\$30348,0000	\$303,4800	Rsi + SMA20 + Sentiment

**Figura 38** – Listagem de ordens do par de negociação BTC/BUSD. Fonte: Elaboração própria.

### 3.3 Análise dos resultados

Conforme os resultados dos modelos de predição de preços apresentados nas **Tabelas 1, 2 e 3**, os modelos de redes neurais recorrentes obtiveram os melhores resultados pela métrica MAPE. O modelo com a arquitetura LSTM foi o que apresentou mais vezes os melhores resultados.

Os resultados analisados também indicam que os modelos apresentam um crescimento no erro conforme o aumento de dias seguintes a serem previstos. Ao utilizar previsões de etapa única ou de múltiplas etapas à frente, foi identificado que os modelos capturam padrões dos movimentos de preços anteriores e tenta aplicá-los nas previsões seguintes, fazendo com que as previsões apresentem um certo “*delay*” em relação aos movimentos de preços reais. Nas previsões de múltiplas etapas à frente, isto fica mais evidente.

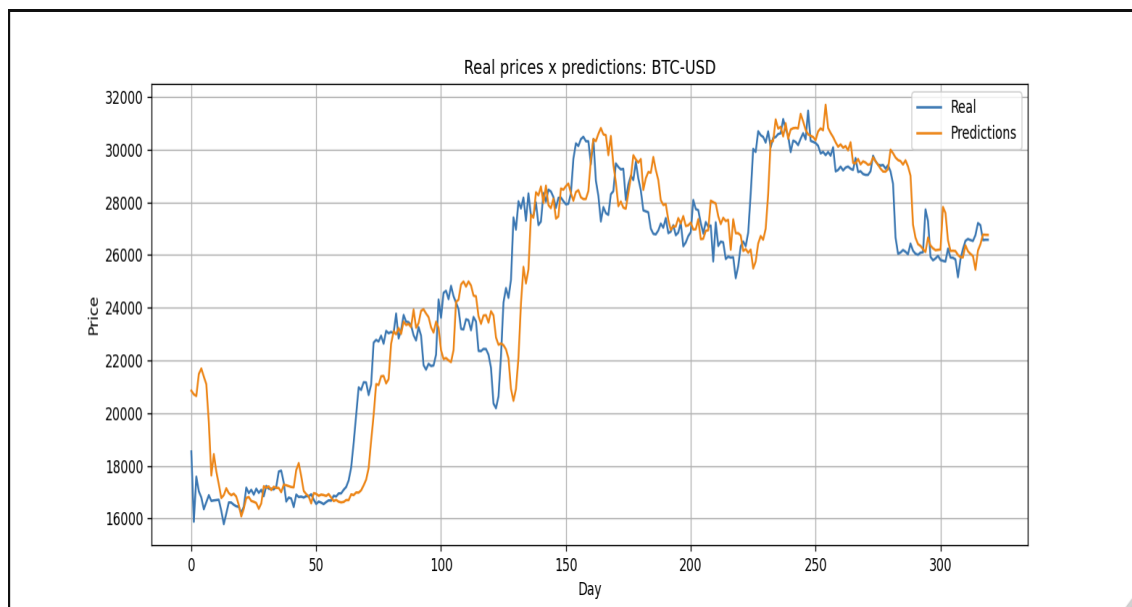
Segundo Brownlee (2023), esse atraso entre valores previstos e reais, é comum em modelos de previsões de séries temporais que fazem uma previsão de persistência. A previsão de persistência possui a característica de usar a saída observada na etapa de tempo anterior como entrada para a próxima previsão. Se um modelo apresenta uma previsão de persistência, significa que o modelo precisa de mais ajustes, o modelo não se adaptou ao conjunto de dados ou o problema de série temporal não é previsível.

Outro detalhe observado sobre os modelos de previsão de múltiplas etapas, são as previsões que estão sempre atualizando a cada dia. Como o modelo faz novas previsões todos os dias, as previsões feitas nos dias anteriores para os próximos  $n$  dias, podem sofrer alterações devido o modelo utilizar dados mais recentes a cada dia que se passa. Desta



forma, as previsões mais recentes dos modelos acabam tendo mais importância do que as previsões anteriores.

A **Figura 39** exibe a comparação entre os preços reais e previstos do modelo LSTM para os próximos 7 dias. Podemos ver que os preços previstos acompanham os movimentos dos preços reais, porém somente após alguns dias depois.



**Figura 39** — Comparação entre os preços reais e previstos para o Bitcoin nos próximos 7 dias com o modelo LSTM. Resultado obtido com um MAPE de 4.9166 no período de testes de previsões entre 14/10/2022 a 07/09/2023. Fonte: Elaboração própria.



## 4 CONCLUSÃO

Neste trabalho, foi apresentado o problema de predição de preços de criptoativos e proposto o uso de um modelo de *Machine Learning* para auxiliar as tomadas de decisão do investidor de criptoativos.

O produto gerado por este trabalho foi o desenvolvimento de um sistema de suporte à decisão de investimentos direcionado por dados, sentimento de mercado e modelos de predição de preços. O sistema Crypto DSS foi projetado inicialmente para fazer negociações simuladas, mas continuará sendo aprimorado com o objetivo de auxiliar o investidor a melhorar suas tomadas de decisão e proporcionar a possibilidade de obter melhores retornos financeiros no mundo real.

Os modelos de predição de preços utilizados pelo sistema foram desenvolvidos com a arquitetura de redes neurais LSTM. Durante o projeto foram desenvolvidos e testados modelos com as arquiteturas ANN, RNN e LSTM. O motivo de ter selecionada a arquitetura LSTM, foi devido esta ter apresentado os melhores resultados dentre os analisados. Outras funcionalidades implementadas no sistema que contribuem para a tomada de decisão do investidor são: análise de sentimento dos criptoativos, análise de volatilidade do mercado, uso de indicadores de análise técnica e gerenciamento de risco com ordens de *stop*.

As implementações do sistema foram desafiadoras, pois foi envolvido o estudo e prática de diversos assuntos como: desenvolvimento de modelos de predição de preços, integração do sistema com diferentes fontes de dados, comunicação de dados em tempo real, execução de tarefas de forma assíncrona e distribuída. No desenvolvimento dos modelos, o maior desafio foi realizar o pré-processamento dos dados antes e depois do processo de treinamento e teste. Nas implementações das funcionalidades do sistema, o principal desafio foi realizar a integração das diferentes funcionalidades implementadas para funcionarem de forma assíncrona e responsiva, principalmente ao utilizar a estratégia automatizada de compra e venda.

Ao analisar os resultados obtidos e funcionalidades implementadas, foi realizado um levantamento de possíveis trabalhos futuros. Para os modelos de predição de preços, pode-se realizar o estudo de arquiteturas mais complexas que envolvem a combinação de camadas de redes neurais LSTM com camadas bidirecionais, convolucionais, entre outras. Também seria interessante fazer um estudo direcionado para outras abordagens de previsões de séries temporais de múltiplas etapas, como por exemplo, um modelo autoregressivo para tentar melhorar os modelos de previsão de múltiplas etapas. Outra forma de tentar melhorar os modelos, seria fazer um estudo e experimentos com adição de novas *features*, como por exemplo, índices de sentimento. Para o sistema de suporte à decisão, existe a possibilidade de trabalhar com o mercado de contratos futuros, que possibilita ganhos com

a alta e a queda dos preços dos criptoativos. Também seria interessante implementar e testar outras estratégias automatizadas, pois a predição de preços alinhada com a análise técnica dos preços, pode oferecer melhores oportunidades de negociações bem sucedidas.

# REFERÊNCIAS BIBLIOGRÁFICAS

ACADEMY, D. S. *Deep Learning Book*. 2022. <<http://deeplearningbook.com.br/>>. Accessed: 21 July 2022.

ADHAMI, S.; GIUDICI, G.; MARTINAZZI, S. Why do businesses go crypto? an empirical analysis of initial coin offerings. *Journal of Economics and Business*, Elsevier, v. 100, p. 64–75, 2018.

ALI, P. J. M. et al. Data normalization and standardization: a technical report. *Mach Learn Tech Rep*, v. 1, n. 1, p. 1–6, 2014.

ALPAYDIN, E. *Introduction to Machine Learning, second edition*. [S.l.]: MIT Press, 2009. (Adaptive Computation and Machine Learning series). ISBN 9780262303262.

ANTONOPOULOS, A. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. [S.l.]: O'Reilly Media, 2014. ISBN 9781491902646.

ARIF, F. A decision support framework for infrastructure maintenance investment decision-making. *FIU Electronic Theses and Dissertations*, 2013.

AYGUN, B.; GUNAY, E. K. Comparison of statistical and machine learning algorithms for forecasting daily bitcoin returns. *Avrupa Bilim ve Teknoloji Dergisi*, n. 21, p. 444–454, 2021.

BADILLO, S. et al. An introduction to machine learning. *Clinical pharmacology & therapeutics*, Wiley Online Library, v. 107, n. 4, p. 871–885, 2020.

BLOCKCHAIN.COM. *Gráficos de blockchain*. 2022. <<https://www.blockchain.com/charts>>. Accessed: 05 June 2022.

BROWNLEE, J. *Why is my forecasted time series one step behind the actual time series?* 2023. <<https://machinelearningmastery.com/faq/single-faq/why-is-my-forecasted-time-series-right-behind-the-actual-time-series/>>. Accessed: 20 Aug. 2023.

BURNISKE, C.; TATAR, J. *Cryptoassets: The Innovative Investor's Guide to Bitcoin and Beyond*. [S.l.]: McGraw-Hill Education, 2017. ISBN 9781260026689.

CCXT. *CCXT – CryptoCurrency eXchange Trading Library*. 2022. <<https://github.com/ccxt/ccxt>>. Accessed: 22 Aug. 2022.

CELERY. *Celery - Distributed Task Queue*. 2023. <<https://docs.celeryq.dev/en/stable/>>. Accessed: 22 Aug. 2023.

CHART.JS. *Simple yet flexible JavaScript charting library for the modern web*. 2023. <<https://www.chartjs.org/>>. Accessed: 22 Sept. 2023.

CHEAH, E.-T.; FRY, J. Speculative bubbles in bitcoin markets? an empirical investigation into the fundamental value of bitcoin. *Economics letters*, Elsevier, v. 130, p. 32–36, 2015.

- CHINAGLIA, R. *8 ferramentas que traders de criptomoedas usam e não contam para ninguém*. 2022. <<https://cointelegraph.com.br/news/tools-that-successful-traders-use-and-dont-tell-anyone>>. Accessed: 13 Aug. 2022.
- CHO, K. et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- CHOLLET, F. *Deep Learning with Python*. [S.l.]: Manning Publications Company, 2017. ISBN 9781617294433.
- COINMARKETCAP. *Today's Cryptocurrency Prices by Market Cap*. 2022. <<https://coinmarketcap.com>>. Accessed: 08 May 2022.
- CROSBY, M. et al. Blockchain technology: Beyond bitcoin. *Applied Innovation*, v. 2, n. 6-10, p. 71, 2016.
- DEMIR-KAVUK, O. et al. Prediction using step-wise l1, l2 regularization and feature selection for small data sets with large number of features. *BMC bioinformatics*, Springer, v. 12, p. 1–10, 2011.
- DJANGO. *The web framework for perfectionists with deadlines*. 2022. <<https://www.djangoproject.com/>>. Accessed: 22 Aug. 2022.
- FANG, F. et al. Cryptocurrency trading: a comprehensive survey. *Financial Innovation*, SpringerOpen, v. 8, n. 1, p. 1–59, 2022.
- GÉRON, A. *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. [S.l.]: Alta Books, 2019. ISBN 9788550809021.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 9780262035613.
- GUIDES, T. S. *How to Make Money Investing in Cryptocurrency*. 2022. <<https://tradingstrategyguides.com/how-to-make-money-investing-in-cryptocurrency/>>. Accessed: 10 May 2022.
- GUO, H. et al. Bitcoin price forecasting: A perspective of underlying blockchain transactions. *Decision Support Systems*, Elsevier, v. 151, p. 113650, 2021.
- HAYKIN, S. *Redes Neurais: Princípios e Prática*. [S.l.]: Bookman Editora, 2001. ISBN 9788577800865.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.
- INDEX, C. V. *CVI - The Crypto Volatility Index*. 2023. <<https://docs.cvi.finance/>>. Accessed: 21 Sept. 2023.
- INFANTE, A.; MAZZEI, M. *Criptomoedas ganham protagonismo na guerra entre Ucrânia e Rússia*. 2022. <<https://www.cnnbrasil.com.br/business/criptomoedas-ganham-protagonismo-na-guerra-entre-ucrania-e-russia/>>. Accessed: 30 Nov. 2022.

JQUERY. *jQuery: write less and do more*. 2023. <<https://jquery.com/>>. Accessed: 03 Apr. 2023.

JUPYTER. *JupyterLab: A Next-Generation Notebook Interface*. 2023. <<https://jupyter.org/>>. Accessed: 12 Sept. 2023.

KERAS. *Adam Optimizer*. 2023. <<https://keras.io/api/optimizers/adam/>>. Accessed: 16 Sept. 2023.

KERAS. *Dropout layer*. 2023. <[https://keras.io/api/layers/regularization\\_layers/dropout/](https://keras.io/api/layers/regularization_layers/dropout/)>. Accessed: 16 Sept. 2023.

KHEDR, A. M. et al. Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. *Intelligent Systems in Accounting, Finance and Management*, Wiley Online Library, v. 28, n. 1, p. 3–34, 2021.

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

KIRCHGÄSSNER, G.; WOLTERS, J.; HASSLER, U. *Introduction to Modern Time Series Analysis*. [S.l.]: Springer Berlin Heidelberg, 2012. (Springer Texts in Business and Economics). ISBN 9783642334351.

KRAALJEVELD, O.; SMEDT, J. D. The predictive power of public twitter sentiment for forecasting cryptocurrency prices. *Journal of International Financial Markets, Institutions and Money*, Elsevier, v. 65, p. 101188, 2020.

KRISTOUFEK, L. Bitcoin meets google trends and wikipedia: Quantifying the relationship between phenomena of the internet era. *Scientific reports*, Nature Publishing Group, v. 3, n. 1, p. 1–7, 2013.

LEARN, B. *20 melhores ferramentas para negociar cripto: dos gráficos à mineração*. 2022. <<https://learn.bybit.com/pt-br/trading/best-crypto-trading-tools/>>. Accessed: 13 Aug. 2022.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LEMOES, F. de A. *Análise Técnica dos Mercados Financeiros*. [S.l.]: Saraiva Educação S.A., 2017. ISBN 9788502637832.

LEWIS, A. *The Basics of Bitcoins and Blockchains: An Introduction to Cryptocurrencies and the Technology that Powers Them*. [S.l.]: Mango Publishing, 2018. (G - Reference, Information and Interdisciplinary Subjects Series). ISBN 9781633538009.

MAHMOUD, Q. H.; LESCISIN, M.; ALTAEI, M. Research challenges and opportunities in blockchain and cryptocurrencies. *Internet Technology Letters*, Wiley Online Library, v. 2, n. 2, p. e93, 2019.

MATPLOTLIB. *Matplotlib: Visualization with Python*. 2022. <<https://numpy.org/>>. Accessed: 22 Aug. 2022.

MCNALLY, S. *Predicting the price of Bitcoin using Machine Learning*. Tese (Doutorado) — Dublin, National College of Ireland, 2016.

- MELNIKOV, A.; FETTE, I. *The WebSocket Protocol*. RFC Editor, 2011. RFC 6455. (Request for Comments, 6455). Accessed: 20 Sept. 2023. Disponível em: <<https://www.rfc-editor.org/info/rfc6455>>.
- MIGUEL, T. *Séries Temporais*. 2022. <<https://aprenderdatascience.com/series-temporais/>>. Accessed: 19 July 2022.
- MOUGAYAR, W.; BUTERIN, V. *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*. [S.l.]: Wiley, 2016. ISBN 9781119300311.
- NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, p. 21260, 2008.
- NARAYANAN, A. et al. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. [S.l.]: Princeton University Press, 2016. ISBN 9780691171692.
- NICOCELI, A. *Criptomoeda luna perde 99,98% de seu valor em sete dias após falha em algoritmo*. 2022. <<https://www.cnnbrasil.com.br/business/criptomoeda-luna-perde-9998-de-seu-valor-em-sete-dias-apos-falha-em-algoritmo/>>. Accessed: 12 May 2022.
- NIELSEN, A. *Análise prática de séries temporais: Predição com Estatística e Aprendizado de Máquina*. [S.l.]: Alta Books, 2021. ISBN 9788550816937.
- NORMAN, A. *The Cryptocurrency Investing Bible*. [S.l.]: CreateSpace Independent Publishing Platform., 2017. ISBN 9781979688369.
- NUMPY. *The fundamental package for scientific computing with Python*. 2022. <<https://numpy.org/>>. Accessed: 22 Aug. 2022.
- OLIVEIRA, A. C. S. d. et al. Aplicação de redes neurais artificiais na previsão da produção de álcool. *Ciência e Agrotecnologia*, SciELO Brasil, v. 34, p. 279–284, 2010.
- PATTERSON, J.; GIBSON, A. *Deep Learning: A Practitioner's Approach*. [S.l.]: O'Reilly Media, 2017. ISBN 9781491914236.
- REACT. *Uma biblioteca JavaScript para criar interfaces de usuário*. 2022. <<https://pt-br.reactjs.org/>>. Accessed: 22 Aug. 2022.
- ROY, S.; NANJIBA, S.; CHAKRABARTY, A. Bitcoin price forecasting using time series analysis. In: IEEE. *2018 21st International Conference of Computer and Information Technology (ICCIT)*. [S.l.], 2018. p. 1–5.
- SABRY, F. et al. Cryptocurrencies and artificial intelligence: Challenges and opportunities. *IEEE Access*, IEEE, v. 8, p. 175840–175858, 2020.
- SANTOS, E. dos; CAVALCANTI, L. *Payments 4.0: As forças que estão transformando o mercado brasileiro*. [S.l.]: Linotipo Digital, 2020. ISBN 9786599026348.
- SWAMYNATHAN, M. *Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python*. [S.l.]: Apress, 2019. ISBN 9781484249468.
- SWAN, M. *Blockchain: Blueprint for a New Economy*. [S.l.]: O'Reilly, 2015. ISBN 9781491920497.



TAPSCOTT, D.; TAPSCOTT, A. *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. [S.l.]: Penguin Publishing Group, 2016. ISBN 9781101980156.

TENSORFLOW. *Introdução ao TensorFlow*. 2022. <<https://www.tensorflow.org/?hl=pt-br>>. Accessed: 22 Aug. 2022.

TENSORFLOW. *Previsão de séries temporais*. 2023. <[https://www.tensorflow.org/tutorials/structured\\_data/time\\_series?hl=pt-br](https://www.tensorflow.org/tutorials/structured_data/time_series?hl=pt-br)>. Accessed: 16 Sept. 2023.

TENSORFLOW. *Regularizer base class*. 2023. <[https://www.tensorflow.org/api\\_docs/python/tf/keras/regularizers/Regularizer](https://www.tensorflow.org/api_docs/python/tf/keras/regularizers/Regularizer)>. Accessed: 16 Sept. 2023.

TIM, C. *Black Dashboard Django*. 2023. <<https://www.creative-tim.com/product/black-dashboard-django>>. Accessed: 20 Sept. 2023.

ULRICH, F. *Bitcoin: A moeda na era digital*. [S.l.]: LVM Editora, 2017. ISBN 9788581190778.

VUE.JS. *The Progressive JavaScript Framework*. 2022. <<https://vuejs.org/>>. Accessed: 22 Aug. 2022.

YLI-HUUMO, J. et al. Where is current research on blockchain technology?—a systematic review. *PloS one*, Public Library of Science San Francisco, CA USA, v. 11, n. 10, p. e0163477, 2016.

YU, P.; YAN, X. Stock price prediction based on deep neural networks. *Neural Computing and Applications*, Springer, v. 32, n. 6, p. 1609–1628, 2020.

ZHENG, Z. et al. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, Inderscience Publishers (IEL), v. 14, n. 4, p. 352–375, 2018.

# Apêndices

# APÊNDICE A – TESTES DOS MODELOS

Os testes automatizados com os modelos ANN, RNN e LSTM foram realizados com vários parâmetros combinados e levou aproximadamente 09 horas. Os seguintes parâmetros foram utilizados na construção dos modelos por meio da biblioteca *TensorFlow*:

- *Features*: são os diferentes recursos de dados utilizados na entrada da rede neural. As *features* utilizadas nos testes foram referentes aos dados de preços (preço de abertura, máxima do dia, mínima do dia, volume de negociação e preço de fechamento) do par de negociação BTC-USD no *timeframe* diário. A qualidade dos dados de entrada dos modelos de *machine learning* são importantes para melhorar os resultados dos modelos e, portanto, um estudo para a seleção e pré-processamento dos dados deve ser levado em consideração (BADILLO et al., 2020).
- *Timesteps*: são as etapas de tempo anteriores que serão consideradas na entrada dos dados dos modelos. Nos testes realizados, é a quantidade de dias anteriores em que os dados serão selecionados e utilizados como entrada durante o treinamento dos modelos. Este parâmetro foi melhor explorado nos modelos RNN e LSTM devido a arquitetura de redes neurais recorrentes considerar sequência e tempo nos dados de entrada, diferente do modelo ANN que considera somente uma entrada sem dependência temporal entre os dados.
- *Next-days*: é a quantidade de dias à frente que serão feitas as previsões dos preços (*multi-step forecast*). Neste trabalho, foi decidido a utilização da abordagem “*One-shot*”. Nesta abordagem, a previsão de várias etapas à frente é feita diretamente e, portanto, a saída do modelo gera um *array* com vários valores correspondentes às etapas de tempo seguintes. Outra abordagem considerada, foi a previsão com um modelo autoregressivo. Nesta abordagem, o modelo faz a previsão de uma etapa, e o valor previsto é utilizado como entrada para as previsões seguintes. Na documentação do *TensorFlow*, foi realizada uma comparação com vários modelos e arquiteturas de redes neurais para previsão de temperatura a partir de dados meteorológicos. Os resultados mostraram que a previsão *multi-step* de forma direta com modelos de redes neurais ANN, Convolucionais e LSTM apresentaram um resultado um pouco melhor que um modelo LSTM autoregressivo, sendo que o modelo autoregressivo apresentou maior complexidade e um desempenho inferior no problema em específico apresentado (TENSORFLOW, 2023a).

- *Normalization*: técnica de pré-processamento dos dados que transforma as diferentes escalas de valores das *features* em um intervalo específico definido como entre 0 e 1 ou entre -1 e 1 (ALI et al., 2014). Nos testes, foi realizada a normalização com a classe *MinMaxScaler* da biblioteca *scikit-learn* e definido os valores dos dados no intervalo entre -1 e 1, visto que a função de ativação utilizada nas camadas das redes neurais foi a tangente hiperbólica, que gera saídas em um intervalo entre -1 e 1. Desta forma, as redes neurais trabalham na mesma escala de valores com os dados e os seus parâmetros.
- *Regularization*: técnica para melhorar a capacidade de generalização dos modelos e reduzir o *overfitting* ao aplicar penalidades aos parâmetros das camadas das redes neurais e que são somadas à função de perda. Nos testes foram utilizados os regularizadores L1 e L2 para os parâmetros *kernel regularizer* (regulariza os pesos entre os neurônios da camada), *bias regularizer* (regulariza os valores de viés da camada) e *activity regularizer* (regulariza as saídas dos neurônios da camada) (TENSORFLOW, 2023b). Na regularização L1, a penalização é calculada pela multiplicação de uma variável com a soma dos valores absolutos dos parâmetros a serem regularizados. Já a regularização L2, a penalização é calculada pela multiplicação de uma variável com a soma dos quadrados dos parâmetros a serem regularizados. Na regularização L1, pode ocorrer de alguns pesos ficarem zerados, fazendo com que características irrelevantes sejam desativadas do modelo. A regularização L2 tem a finalidade de reduzir pesos muito altos, mas não chega a zerá-los, fazendo com que ajude a evitar o *overfitting* (DEMIR-KAVUK et al., 2011). Outra forma de regularização é a técnica de *dropout*, que foi utilizada nos testes com modelo ANN. Esta técnica consiste em adicionar uma camada *Dropout* entre as camadas ocultas e camada de saída da rede neural com a finalidade de desligar aleatoriamente uma quantidade de neurônios das camadas para evitar o *overfitting*. A quantidade de neurônios a serem desligados é definido por uma taxa com valor entre 0 e 1, que representa o percentual de neurônios a serem desativados durante o treinamento da rede neural (KERAS, 2023b).
- *Hidden layers*: são as camadas ocultas da rede neural. Nos testes foram utilizadas as camadas da API *Keras* do *TensorFlow* por meio da classe *Sequential*. As camadas utilizadas foram *Dense* (camada oculta tradicional), *SimpleRNN* (camada recorrente tradicional) e *LSTM* (camada recorrente com arquitetura LSTM) com 64 neurônios cada. Foram realizados experimentos com a combinação dessas camadas. No modelo ANN foi testado somente a camada *Dense*. No modelo RNN foi testada a camada *SimpleRNN* combinada com a camada *Dense*. O modelo LSTM foi testado com a camada *LSTM* combinada com a camada *Dense*. A principal evidência percebida entre essas camadas, foi a quantidade de parâmetros do modelo. Os modelos para predição de preços para o próximo dia com uma única camada oculta totalizaram os

seguintes parâmetros: 449 parâmetros para o modelo ANN, 4545 parâmetros para o modelo RNN e 18241 parâmetros para o modelo LSTM. Devido a quantidade de muitos parâmetros, o modelo LSTM é o que mais demanda tempo de treinamento.

- *Activation function*: parâmetro utilizado que define qual a função de ativação utilizada nas camadas ocultas da rede neural. Nos testes foi utilizada somente a função de ativação tangente hiperbólica, que é a função de ativação padrão das camadas *SimpleRNN* e *LSTM*.
- *Optimizer*: define qual o algoritmo otimizador dos pesos durante o treinamento da rede neural. Nos testes foi definido o algoritmo Adam. O algoritmo Adam é um método de descida do gradiente estocástico computacionalmente eficiente e adequado para problemas que envolvem grandes quantidades de dados e parâmetros. Para mais detalhes, consultar a obra de Kingma e Ba (2014). Na construção do modelo com o *TensorFlow*, é possível especificar o otimizador com a classe *Adam* e definir vários parâmetros ao instanciar a classe, sendo que o parâmetro principal é o *learning rate* que define qual a taxa de aprendizado aplicada durante o treinamento e atualização dos pesos. Por padrão a taxa de aprendizado possui o valor 0.001, sendo possível modificá-lo e testá-lo com os demais parâmetros do otimizador para buscar um valor que melhore o aprendizado da rede neural (KERAS, 2023a).
- *Loss function*: define qual a função de perda que computa o erro da rede neural. Foram realizados testes com a função MAE (*Mean Absolute Error*) e a função MSE (*Mean Squared Error*), ambas mencionadas na **Subseção 2.3.4**. Foi identificado nos testes que a função de perda MAE, foi a que mais apareceu nos melhores resultados dos modelos por conseguir computar melhor o erro durante os treinamentos e então conseguir capturar melhor os movimentos dos preços previstos.
- *Epochs*: define o número de vezes que o conjunto de dados de treinamento será passado para o modelo durante o treinamento. Nos testes, os modelos foram treinados por 100 épocas. O número de épocas para o treinamento dos modelos influencia no aprendizado do modelo, pois um número muito pequeno faz com que o modelo aprenda pouco sobre os dados e não consiga extrair padrões dos dados corretamente. Já um número muito alto de épocas faz com que o modelo se ajuste muito aos dados de treinamento e não consiga generalizar com os dados de validação.
- *Batch size*: define o tamanho do lote de dados que será utilizado durante cada iteração do treinamento. O modelo atualiza os pesos após cada iteração de treinamento com o lote de dados. Nos testes foi utilizado o tamanho de lote igual a 32. O tamanho do lote é um parâmetro importante que deve ser testado para cada conjunto de dados e problema em específico.

- *Shuffle*: define se durante o treinamento do modelo irá ser feito o embaralhamento dos lotes de dados. Habilitar o *shuffle* é uma forma de melhorar a generalização do modelo, pois o modelo não é enviesado pela ordem dos dados que são fornecidos durante o treinamento, e tende a melhorar os resultados com os dados de validação. Nos testes foi identificado que na maioria dos resultados a utilização do *shuffle* melhorou os resultados do modelo.

A **Figura 40** exibe o trecho de código que constrói o modelo LSTM com a biblioteca *TensorFlow*. Podemos ver que o modelo é criado com a classe *Sequential*, e em seguida são adicionadas as camadas sequencialmente de cima para baixo. Na adição da primeira camada é especificado o número de neurônios, a função de ativação, o formato da matriz dos dados de entrada e os regularizadores. Em seguida é adicionada a camada de saída com um único neurônio, visto que o modelo irá prever apenas uma única saída (preço para o próximo dia). Depois é definido o otimizador Adam que é referenciado no método *compile* junto com a função de perda “mae”. O modelo é então treinado a partir do método *fit*, e especificamos os dados de treinamento rotulados, o número de épocas, o tamanho do lote, o parâmetro *shuffle* e os dados de validação rotulados para que o modelo compare o erro obtido com os dados de treinamento e os dados de validação para cada época de treinamento.

```
# Cria um modelo de rede neural recorrente com a arquitetura LSTM
model = keras.Sequential(name='LSTM_ONE-STEP-FORECAST')
model.add(layers.LSTM(units=64, activation='tanh', input_shape=(timesteps, x_train.shape[2]),
    kernel_regularizer=regularizers.l1(0.0001), activity_regularizer=regularizers.l1(0.0001), bias_regularizer='l1'))
#model.add(layers.Dropout(0.2))
model.add(layers.Dense(1)) # Camada de saída

# Define o algoritmo otimizador dos pesos da rede neural e a taxa de aprendizado
opt = keras.optimizers.Adam() # learning_rate=0.001 por padrão

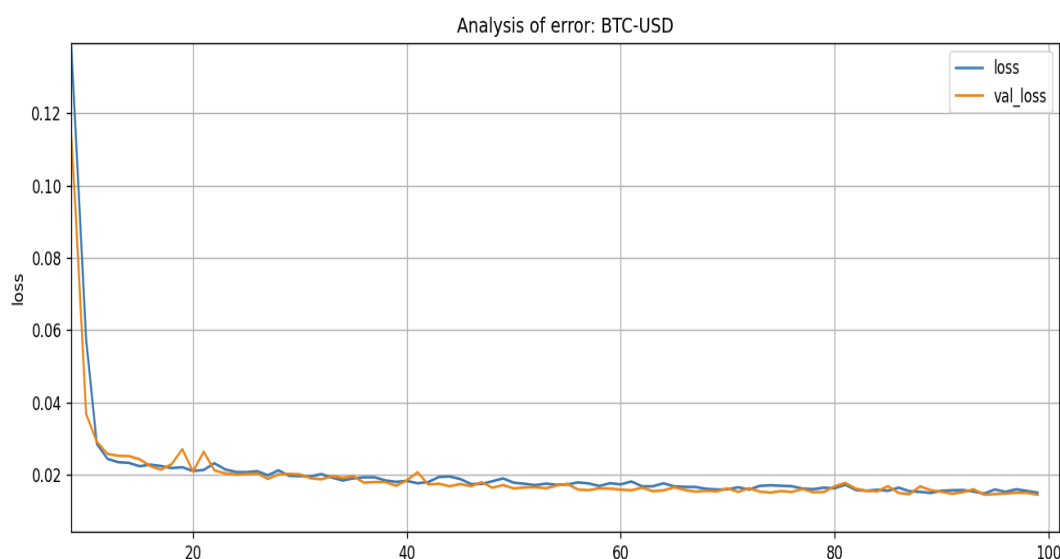
# Compila o modelo definindo a função de perda para cálculo do erro e o algoritmo otimizador.
model.compile(loss='mae', optimizer=opt)

%%time

# Treina o modelo
history = model.fit(x_train, y_train, epochs=100, batch_size=32, verbose=1, shuffle=True, validation_data=(x_test, y_test))
```

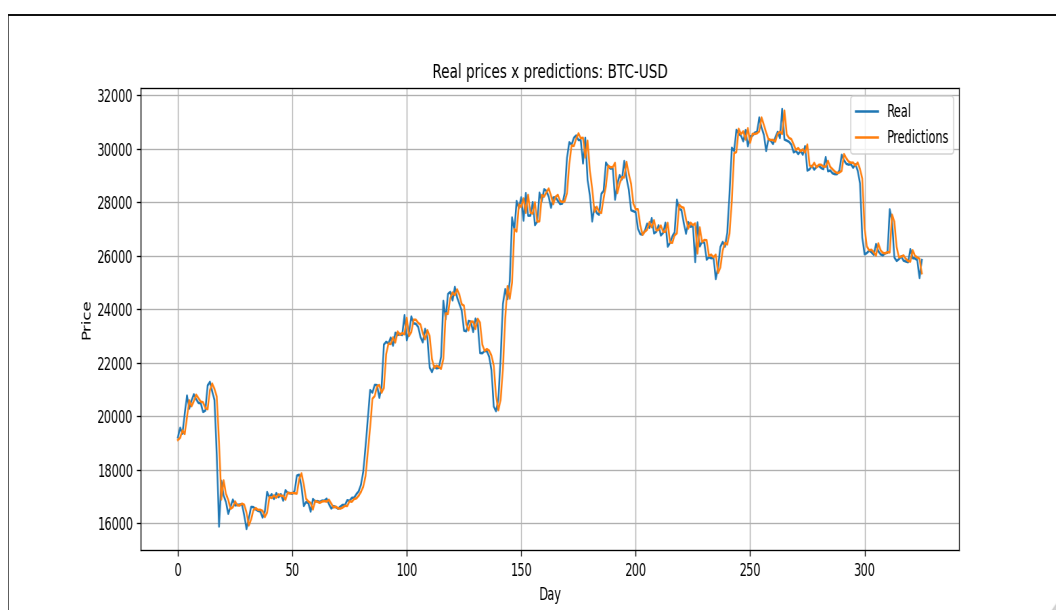
**Figura 40** – Trecho de código que demonstra a construção do modelo LSTM com a biblioteca *TensorFlow*. Fonte: Elaboração própria.

A **Figura 41** exibe o comportamento dos erros de treinamento e validação para cada época de treinamento. Podemos ver que nas primeiras 10 épocas o erro cai drasticamente e depois começa a estabilizar e diminuir em um ritmo mais lento no decorrer das demais épocas.



**Figura 41** – Comparação entre o erro MAE (*Mean Absolute Error*) de treinamento e validação do modelo LSTM para os dados do Bitcoin durante cada época de treinamento. Fonte: Elaboração própria.

Na **Figura 42** podemos ver a comparação entre os preços reais e previstos para o Bitcoin no próximo dia. As previsões para o próximo dia apresentou um resultado bem próximo dos preços reais, com a métrica MAPE apontando um valor de 1.6404 de erro. O resultado foi obtido a partir da validação entre os preços reais e previstos do conjunto de dados de validação que foram separados do conjunto de dados utilizados no treinamento. Os dados de validação representaram os últimos 326 dias.



**Figura 42** – Comparação entre os preços reais e previstos para o Bitcoin no próximo dia com o modelo LSTM. Resultado obtido com um MAPE de 1.6404. Fonte: Elaboração própria.

Seguem abaixo os resultados completos dos testes realizados para cada modelo.



MODELO: ANN

OBS: Valor MAPE obtido pela média de 5 treinamentos x predição do modelo.

Tempo de um único treinamento: ~19s

Total de amostras: 3278 dias (17-09-2014 a 07-09-2023 ). 2950 dias para treino, 328 dias para teste.

Configuração do PC: Windows 11 - 64 bits, processador AMD Ryzen 5 3600X - 6 núcleos (3.8 GHz), 32 GB RAM DDR4.

Features Description			Regularizer's params		
5	Open, High, Low, Volume, Close		kernel_regularizer=0.0001, activity_regularizer=0.0001, bias_regularizer=0.01		
6	Open, High, Low, Volume, Vix, Close				
2	Vix, Close		Percentage rate. Ex: 0.2 = 20%		

Date	Asset	Features	Time-steps	next-days	Normalization	Regularization	Hidden layers	Act. function	Optimizer	Learning rate	Loss function	Epochs	Batch size	Shuffle	Mape
09/09/2023	BTC-USD	5	1	1	Min-Max	L2	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,7598
09/09/2023	BTC-USD	5	1	1	Min-Max	L1	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,7859
09/09/2023	BTC-USD	5	1	1	Min-Max	-	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,7975
09/09/2023	BTC-USD	5	1	1	Min-Max	-	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1,819
09/09/2023	BTC-USD	5	1	1	Min-Max	L2, Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	1,8419
09/09/2023	BTC-USD	5	1	1	Min-Max	Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,8478
09/09/2023	BTC-USD	5	1	1	Min-Max	L1	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1,85
09/09/2023	BTC-USD	5	1	1	Min-Max	L2	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1,9208
09/09/2023	BTC-USD	5	1	1	Min-Max	L1	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,004
09/09/2023	BTC-USD	5	1	1	Min-Max	L1	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,0145
09/09/2023	BTC-USD	5	1	1	Min-Max	-	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,0379
09/09/2023	BTC-USD	5	1	1	Min-Max	L2, Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	2,0548
09/09/2023	BTC-USD	5	1	1	Min-Max	L2	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,0793
09/09/2023	BTC-USD	5	1	1	Min-Max	-	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,1295
09/09/2023	BTC-USD	5	1	1	Min-Max	L2	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,1455
09/09/2023	BTC-USD	5	1	1	Min-Max	L2, Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2,1871
09/09/2023	BTC-USD	5	1	1	Min-Max	L1, Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,2353
09/09/2023	BTC-USD	5	1	1	Min-Max	Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2,5091
09/09/2023	BTC-USD	5	1	1	Min-Max	Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,6341
09/09/2023	BTC-USD	5	1	1	Min-Max	L2, Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,7347
09/09/2023	BTC-USD	5	1	1	Min-Max	L1, Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2,8705
09/09/2023	BTC-USD	5	1	1	Min-Max	L1, Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,0696
09/09/2023	BTC-USD	5	1	1	Min-Max	L1, Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,1101
09/09/2023	BTC-USD	5	1	1	Min-Max	Dropout (0.2)	1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,1326

MODELO: RNN

OBS: Valor MAPE obtido pela média de 5 treinamentos x predição do modelo.  
Tempo de um único treinamento: ~23s  
Total de amostras: 3278 dias (17-09-2014 a 07-09-2023 ). 2950 dias para treino, 328 dias para teste.  
Configuração do PC: Windows 11 - 64 bits, processador AMD Ryzen 5 3600X - 6 núcleos (3.8 GHz), 32 GB RAM DDR4.

Features Description				Regularizers params			
N	Open, High, Low, Volume, Close	next-days	Normalization	Regularization	Hidden layers	Act. function	Optimizer
5	Open, High, Low, Volume, Close	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam
6	Open, High, Low, Volume, Vix, Close	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam
2	Vix, Close	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam
Dropout							
Percentage rate. Ex: 0.2 = 20%							

Date	Asset	Features	Time-steps	next-days	Normalization	Regularization	Hidden layers	Act. function	Optimizer	Learning rate	Loss function	Epochs	Batch size	Shuffle	Maape
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	1,7019
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1,7041
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,7093
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	1,7199
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	1,7298
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	1,7318
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	1,737
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	1,7486
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	1,766
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1,8038
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,807
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	1,8087
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	1,8204
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	1,8206
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	1,8273
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,8416
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	1,8462
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,8491
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	1,8637
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,8705
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	1,8838
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1,9111
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	1,9393
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1,9439
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	1,9472
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	1,9568
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	1,9737
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	1,9804
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	2,0112
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	2,0123
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	2,0139
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,0269
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2,0839
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2,1029
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,1064
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	2,1115
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,1335
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,2033
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	2,2115
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,2274
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2,2298
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	2,2763

09/09/2023	BTC-USD	5	7	1	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,3605
09/09/2023	BTC-USD	5	7	1	1	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	2,3981
09/09/2023	BTC-USD	5	3	1	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,5644
09/09/2023	BTC-USD	5	3	1	1	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,6095
09/09/2023	BTC-USD	5	3	1	1	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,8208
09/09/2023	BTC-USD	5	7	1	1	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	2,9072
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	2,9406
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,9407
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	2,9435
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2,9557
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,9645
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2,9693
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	2,9763
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	2,989
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	2,9913
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,9947
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	2,997
09/09/2023	BTC-USD	5	7	3	3	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	3,0097
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	3,0104
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,0251
09/09/2023	BTC-USD	5	3	3	3	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	3,0388
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,0461
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	3,0551
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	3,0583
09/09/2023	BTC-USD	5	3	3	3	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,0775
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,0797
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	3,0798
09/09/2023	BTC-USD	5	3	3	3	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	3,0803
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,0843
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,1008
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	3,1104
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,1341
09/09/2023	BTC-USD	5	3	3	3	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	3,1366
09/09/2023	BTC-USD	5	7	3	3	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,1469
09/09/2023	BTC-USD	5	7	3	3	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	3,1703
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	3,1797
09/09/2023	BTC-USD	5	7	3	3	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,1864
09/09/2023	BTC-USD	5	3	3	3	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,2051
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,2105
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	3,212
09/09/2023	BTC-USD	5	3	3	3	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	3,2384
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	3,2411
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	3,2501
09/09/2023	BTC-USD	5	7	3	3	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,2595
09/09/2023	BTC-USD	5	7	3	3	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	3,2635
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,267
09/09/2023	BTC-USD	5	7	3	3	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,2914
09/09/2023	BTC-USD	5	3	3	3	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,2932
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,3082
09/09/2023	BTC-USD	5	7	3	3	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	3,4664
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	3,4784
09/09/2023	BTC-USD	5	3	3	3	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,5033
09/09/2023	BTC-USD	5	7	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	3,5849
09/09/2023	BTC-USD	5	3	3	3	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,7505

09/09/2023	BTC-USD	5	3	7	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	4,7844
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	4,7951
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	4,7976
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,7981
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,8093
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,8158
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,8207
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	4,8308
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	4,8391
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	4,8483
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	4,8493
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	4,8606
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	4,8617
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	4,8626
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	4,8692
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	4,8694
09/09/2023	BTC-USD	5	7	7	7	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	4,874
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	4,889
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	4,8917
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	4,9105
09/09/2023	BTC-USD	5	3	7	7	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	4,9306
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	4,9317
09/09/2023	BTC-USD	5	3	7	7	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,9318
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	4,9526
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L1	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	4,9715
09/09/2023	BTC-USD	5	7	7	7	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,9794
09/09/2023	BTC-USD	5	3	7	7	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	5,0289
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	5,0289
09/09/2023	BTC-USD	5	7	7	7	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	5,0539
09/09/2023	BTC-USD	5	3	7	7	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	5,0581
09/09/2023	BTC-USD	5	7	7	7	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	5,0928
09/09/2023	BTC-USD	5	7	7	7	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	5,1207
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,1609
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,1857
09/09/2023	BTC-USD	5	3	7	7	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	5,1947
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	false	5,2224
09/09/2023	BTC-USD	5	7	7	7	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,226
09/09/2023	BTC-USD	5	3	7	7	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	5,2545
09/09/2023	BTC-USD	5	7	7	7	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	5,4107
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	5,4161
09/09/2023	BTC-USD	5	3	7	7	Min-Max	-	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,4502
09/09/2023	BTC-USD	5	3	7	7	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	5,4589
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	5,4711
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	5,4889
09/09/2023	BTC-USD	5	7	7	7	Min-Max	-	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	5,5916
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L2	1 SimpleRNN (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	5,718
09/09/2023	BTC-USD	5	3	7	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	5,8505
09/09/2023	BTC-USD	5	7	7	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	false	6,2001

MODELO: LSTM

OBS: Valor MAPE obtido pela média de 5 treinamentos x predição do modelo.

Tempo de um único treinamento: ~30s

Total de amostras: 3278 dias (17-09-2014 a 07-09-2023 ). 2950 dias para treino, 328 dias para teste.

Configuração do PC: Windows 11 - 64 bits, processador AMD Ryzen 5 3600X - 6 núcleos (3.8 GHz), 32 GB RAM DDR4.

N	Features Description		
	Open, High, Low, Volume, Close		
	Open, High, Low, Volume, Vix, Close	Vix, Close	
5			
6			
2			

Regularizer	Regularizets params		
	kernel_regularizer=0.0001, activity_regularizer=0.0001, bias_regularizer=0.01		
	Percentage rate. Ex: 0.2 = 20%		
L1, L2, L1_L2			
Dropout			

Date	Asset	Features	Time-steps	next-days	Normalization	Regularization	Hidden layers	Act. function	Optimizer	Learning rate	Loss function	Epochs	Batch size	Shuffle	Mape
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	1.6314
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1.6504
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	1.682
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1.7084
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	1.7205
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	1.73
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	1.7355
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	1.74
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	1.7645
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	1.7669
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	1.7791
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1.7917
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1.8158
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	1.8227
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	1.8291
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1.8634
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1.8694
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	1.891
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	1.9042
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	1.9135
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	1.9211
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	1.9251
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1.9258
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	1.9309
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	1.9387
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	1.9399
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	1.9642
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	1.9929
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	2.0041
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	2.0133
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2.0242
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2.0278
09/09/2023	BTC-USD	5	3	1	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2.0345
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2.0427
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	2.0449
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	2.0628
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2.1089
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2.1097
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	2.1441
09/09/2023	BTC-USD	5	3	1	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	2.1755

09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	2,1777
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	2,1944
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	2,2033
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,2033
09/09/2023	BTC-USD	5	7	1	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,2216
09/09/2023	BTC-USD	5	7	1	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,2274
09/09/2023	BTC-USD	5	7	1	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,6122
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	2,8901
09/09/2023	BTC-USD	5	3	1	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	2,8976
09/09/2023	BTC-USD	5	7	3	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	2,9166
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	2,9636
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	2,9763
09/09/2023	BTC-USD	5	7	3	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	2,9982
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,0115
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,0128
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,0276
09/09/2023	BTC-USD	5	7	3	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	3,0283
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,0559
09/09/2023	BTC-USD	5	7	3	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,0579
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,0596
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	3,0715
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	3,076
09/09/2023	BTC-USD	5	7	3	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,0804
09/09/2023	BTC-USD	5	3	3	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	3,085
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,0968
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,0979
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	3,1484
09/09/2023	BTC-USD	5	3	3	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,1532
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	3,1636
09/09/2023	BTC-USD	5	7	3	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	3,1736
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	3,1765
09/09/2023	BTC-USD	5	3	3	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,2112
09/09/2023	BTC-USD	5	7	3	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	3,2145
09/09/2023	BTC-USD	5	7	3	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,2145
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	3,2192
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	3,2225
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,2488
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,2513
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	3,2862
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,2891
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	3,2918
09/09/2023	BTC-USD	5	7	3	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,2984
09/09/2023	BTC-USD	5	7	3	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	3,3037
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,3185
09/09/2023	BTC-USD	5	3	3	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	3,3222
09/09/2023	BTC-USD	5	7	3	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	3,3234
09/09/2023	BTC-USD	5	3	3	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	3,3296
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,3717
09/09/2023	BTC-USD	5	3	3	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,3779
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	3,4039
09/09/2023	BTC-USD	5	3	3	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	3,4248
09/09/2023	BTC-USD	5	7	3	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	3,427
09/09/2023	BTC-USD	5	7	3	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	3,4483
09/09/2023	BTC-USD	5	3	3	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	3,4759

09/09/2023	BTC-USD	5	7	3	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	3,5103
09/09/2023	BTC-USD	5	7	3	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	3,5549
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	4,8382
09/09/2023	BTC-USD	5	3	7	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,8388
09/09/2023	BTC-USD	5	7	7	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,88
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,8882
09/09/2023	BTC-USD	5	7	7	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	4,8911
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	4,8995
09/09/2023	BTC-USD	5	3	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	4,9295
09/09/2023	BTC-USD	5	7	7	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	4,9407
09/09/2023	BTC-USD	5	7	7	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	4,9459
09/09/2023	BTC-USD	5	3	7	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	4,9494
09/09/2023	BTC-USD	5	7	7	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,9524
09/09/2023	BTC-USD	5	7	7	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	4,9649
09/09/2023	BTC-USD	5	3	7	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	4,9736
09/09/2023	BTC-USD	5	7	7	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	4,976
09/09/2023	BTC-USD	5	3	7	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	4,9856
09/09/2023	BTC-USD	5	7	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	4,9899
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	4,9918
09/09/2023	BTC-USD	5	3	7	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	5,0176
09/09/2023	BTC-USD	5	3	7	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	5,0218
09/09/2023	BTC-USD	5	3	7	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,0259
09/09/2023	BTC-USD	5	7	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	5,0698
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	5,092
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	5,1099
09/09/2023	BTC-USD	5	7	7	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	5,1162
09/09/2023	BTC-USD	5	3	7	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	5,1189
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	false	5,1418
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	5,1568
09/09/2023	BTC-USD	5	7	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	5,1593
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	5,1764
09/09/2023	BTC-USD	5	3	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	5,1895
09/09/2023	BTC-USD	5	3	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	5,1906
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,2005
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	false	5,2689
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,2981
09/09/2023	BTC-USD	5	3	7	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,3575
09/09/2023	BTC-USD	5	7	7	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	5,3619
09/09/2023	BTC-USD	5	7	7	Min-Max	L1	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	true	5,4101
09/09/2023	BTC-USD	5	3	7	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	5,5223
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	5,5532
09/09/2023	BTC-USD	5	7	7	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	true	5,6029
09/09/2023	BTC-USD	5	3	7	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	5,6887
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MSE	100	32	false	5,7528
09/09/2023	BTC-USD	5	7	7	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	5,8437
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	5,9048
09/09/2023	BTC-USD	5	3	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	5,9853
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	6,0316
09/09/2023	BTC-USD	5	7	7	Min-Max	-	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	6,145
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MSE	100	32	false	6,4199

## COMPARATIVO DOS MODELOS

**OBOS:** Valor MAPE obtido pela média de 5 treinamentos x predição do modelo.

**Total de amostras:** 3278 dias (17-09-2014 a 07-09-2023). 2950 dias para treino, 328 dias para teste.

**Configuração do PC:** Windows 11 - 64 bits, processador AMD Ryzen 5 3600X - 6 núcleos (3.8 GHz) 32 GB RAM DDR4.

N	Features Description	Regularizer	Regularizers params
5	Open, High, Low, Volume, Close	L1, L2, L1_L2	kernel_regularizer=0.0001, activity_regularizer=0.0001, bias_regularizer=0.01
6	Open, High, Low, Volume, Vix, Close	Dropout	Percentage rate. Ex: 0.2 = 20%
2	Vix, Close		

MELHOR RESULTADO DE CADA MODELO - PREDIÇÃO PRÓXIMO DIA															
Date	Asset	Features	Time-steps	next-days	Normalization	Regularization	Hidden layers	Act. function	Optimizer	Learning rate	Loss function	Epochs	Batch size	Shuffle	Mape
09/09/2023	BTC-USDT	5	7	1	Min-Max	L1	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	1.6314
09/09/2023	BTC-USDT	5	7	1	Min-Max	L1	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	1.7019
09/09/2023	BTC-USDT	5	1	1	Min-Max	L2	1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	1.7598

Date	Asset	Features	Time-steps	next-days	Normalization	Regularization	MELHOR RESULTADO DE CADA MODELO - PREDIÇÃO PRÓXIMOS 3 DIAS								
							Hidden layers	Act. function	Optimizer	Learning rate	Loss function	Epochs	Batch size	Shuffle	MApe
09/09/2023	BTC-USD	5	7	3	Min-Max	-	1 LSTM (64), 1 Dense (64)	tanh	Adam	0.001	MAE	100	32	true	2.8901
09/09/2023	BTC-USD	5	3	3	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MAE	100	32	true	2.9406

MELHOR RESULTADO DE CADA MODELO - PREDIÇÃO PRÓXIMOS 7 DIAS															
Date	Asset	Features	Time-steps	next-days	Normalization	Regularization	Hidden layers	Act. function	Optimizer	Learning rate	Loss function	Epochs	Batch size	Shuffle	Mape
09/09/2023	BTC-USD	5	3	7	Min-Max	L2	1 SimpleRNN (64)	tanh	Adam	0.001	MSE	100	32	true	4,7844
09/09/2023	BTC-USD	5	7	7	Min-Max	L2	1 LSTM (64)	tanh	Adam	0.001	MAE	100	32	true	4,8382



# APÊNDICE B – MODELAGEM DO BANCO DE DADOS

Durante o projeto do sistema, foi realizada a modelagem do banco de dados com a ferramenta *MySQL Workbench*. Apesar da ferramenta ser do banco de dados *MySql*, o banco de dados relacional utilizado no sistema foi o *PostgreSQL*.

O documento a seguir apresenta o diagrama de entidade e relacionamento elaborado.



# APÊNDICE C

## – IMPLEMENTAÇÕES DO SISTEMA

O trecho de código **C.1** exibe a implementação da tarefa que executa as previsões de preços para o sistema. A tarefa recebe como parâmetros: o símbolo do criptoativo, o *id* do criptoativo no *website* CoinMarketCap, o número de *timesteps* e número de dias a serem previstos. Primeiramente é feito o carregamento do modelo que foi exportado em um arquivo após o treinamento e avaliação no JupyterLab. Em seguida é realizada a coleta dos dados de entrada do modelo na API do *website* CoinMarketCap. Depois é realizado o pré-processamento dos dados de entrada com a normalização e o redimensionamento da matriz dos dados. Em seguida é realizada a previsão de preços por meio do método “*predict*” que recebe os dados de entrada normalizados. Após obtidos os valores da previsão, é realizada a desnormalização dos valores previstos para a escala original. Por fim, os valores previstos são passados para uma tarefa responsável por salvá-los no banco de dados.

O trecho de código **C.2** apresenta a implementação referente às condições da estratégia automatizada. Antes das condições serem avaliadas para cada ordem definida com a estratégia, as variáveis utilizadas pelas condições são atualizadas com os valores obtidos do *cache*. Após obtidos os valores das variáveis necessárias, as condições apresentadas na **Subseção 3.2.10** são avaliadas e, caso todas forem satisfeitas, é acionada a tarefa que envia a ordem para a *exchange*.

O trecho de código **C.3** exibe a implementação da tarefa responsável pelo envio das ordens automatizadas para a *exchange*. A tarefa recebe como parâmetro, um dicionário com os dados da ordem local. Primeiramente são obtidas as credenciais do usuário responsável pela ordem. Em seguida o valor do montante da ordem é ajustado com a precisão de dígitos esperada pela *exchange* e então a ordem é enviada para a mesma. Por fim, se a ordem foi enviada com sucesso, a ordem local é atualizada no banco de dados a partir dos dados de execução retornados pela *exchange*.

---

```

@shared_task(ignore_result=True)
def task_LSTM_predictions(symbol, id_crypto_cmc, time_steps, n_predictions):
    try:
        if n_predictions == 1:
            model = load_model_keras(model_name='LSTM_ONE-STEP-FORECAST',
                                     symbol=symbol)
        else:
            model = load_model_keras(model_name=f'LSTM_MULTI-STEP-FORECAST-{n_pre}
            dictions}D',
                                     symbol=symbol)

        if model:
            # Obtém os dados de entrada dos últimos 3 dias
            input_data = get_period_data_crypto(id_crypto_cmc=id_crypto_cmc,
                                                n_days=time_steps)

            # Faz a normalização dos dados de entrada
            input_data_norm, scaler =
            normalize_data(input_data=input_data[['Open', 'High', 'Low',
            'Volume', 'Close']], symbol=symbol, model_name=model.name)

            # Redimensiona os dados para o formato esperado pela rede neural LSTM
            input_data_norm = input_data_norm.reshape(-1, time_steps,
            input_data_norm.shape[1])

            # Faz as previsões a partir dos dados normalizados
            predictions = model.predict(input_data_norm)

            # Transforma os valores previstos para a escala original
            predictions = transform_orignal_scale(scaler=scaler,
            array_data=predictions, symbol=symbol, model_name=model.name)

            # Redimensiona o array para uma única dimensão e arredonda os valores
            para 4 casas decimais
            predictions = predictions.flatten().round(4)

            # Aciona tarefa para salvar um json da previsão no banco de dados
            predictions_json = {'model': model.name, 'predictions':
            list(predictions)}
            task_save_predictions_db.delay(id_crypto_cmc, predictions_json)
        else:
            raise Exception('Falha no carregamento do modelo...')
    except Exception as error:
        print(f'Falha na previsão de preço: {error}')

```

**Listing C.1** – Tarefa para executar as previsões de preços dos modelos LSTM.

---

```

# Cria um dicionário com as principais informações da ordem local
order_info = {
    'id_user': order.id_user.id,
    'id_local': order.id,
    'symbol': symbol,
    'amount': amount,
    'side': order_type,
}

# Verifica todas as condições da estratégia para o tipo da ordem
if order_type == 'BUY':
    rsi_ok = rsi_ok < 30
    strategy_ok = expected_price_change >= min_percent_gains and
    sentiment_bullish and cvi_ok and sma20_ok and rsi_ok
    if price_limit:
        strategy_ok = strategy_ok and current_price >= price_limit

    # Executa a tarefa para criar uma ordem a mercado caso todas as condições
    da estratégia forem satisfeitas
    if strategy_ok:
        task_auto_market_order.delay(order_info)
elif order_type == 'SELL':
    rsi_ok = rsi_ok > 70
    strategy_ok = expected_price_change >= min_percent_gains and
    sentiment_bullish and cvi_ok and sma20_ok and rsi_ok
    if price_limit:
        strategy_ok = strategy_ok and current_price <= price_limit

    # Aciona a tarefa para criar uma ordem a mercado caso todas as condições da
    estratégia forem satisfeitas
    if strategy_ok:
        task_auto_market_order.delay(order_info)

```

**Listing C.2** – Verificação das condições da estratégia automatizada.

---

```

@shared_task(ignore_result=True)
def task_auto_market_order(local_order_dict):
    if local_order_dict:
        # Obtém as credenciais de api do usuário
        user_credentials =
        ApiKeys.objects.get(user_id=local_order_dict['id_user'])
        binance = get_ccxt_instance(user_credentials)

        # Ajusta a precisão do montante conforme exigido pela exchange
        amount_exchange =
        binance.amount_to_precision(symbol=local_order_dict['symbol'],
        amount=local_order_dict['amount'])

        order = None
        if local_order_dict['side'] == 'BUY':
            order =
            binance.create_market_buy_order(symbol=local_order_dict['symbol'],
            amount=amount_exchange)
        elif local_order_dict['side'] == 'SELL':
            order =
            binance.create_market_sell_order(symbol=local_order_dict['symbol'],
            amount=amount_exchange)

        if order:
            print(f'Ordem local {local_order_dict["id_local"]} executada pela
            estratégia!')
            print(f'Market order: {order["id"]}')

            # Atualiza a ordem no banco de dados
            update_local_order_db.delay(order,
            id_local=local_order_dict['id_local'])
        else:
            print('Ordem local não recebida...')

```

**Listing C.3** – Execução de ordens de compra e venda pela estratégia automatizada.