

**Universidade Estadual de Campinas**  
**Faculdade de Tecnologia**

**Utilização de Séries Temporais para Predição do  
Preço de Fechamento de Criptomoedas**

**Palavras-chave:** criptomoedas, aprendizado de máquina, redes neurais  
artificiais, séries temporais

**Aluno:**

Renan de Oliveira Ferretti (FT - Unicamp)  
RA: 221034

**Orientadores:**

Prof. Dr. Guilherme Palermo Coelho (FT - Unicamp)  
Prof. Dr. Arthur Emanuel de Oliveira Carosia (IFSP - *campus* São João da Boa Vista)

---

## **1 Introdução**

O termo criptomoeda se refere a uma moeda virtual que utiliza criptografia para realizar as suas transações. Tais moedas são organizadas por uma rede ponto-a-ponto chamada *blockchain*, o que garante segurança quando são usadas. Criptomoedas vêm sendo amplamente reconhecidas como um novo método alternativo de troca de capital, o que tem implicações consideráveis para economias emergentes e, em geral, para a economia global [1].

As criptomoedas conquistaram popularidade devido à sua natureza descentralizada, livre da ação de órgãos regulamentadores e baixos custos de transação. Tal popularidade levou a um aumento no volume de negociações e volatilidade do preço deste tipo de ativo [2]. O mercado de criptomoedas possui algumas diferenças em relação aos mercados tradicionais, como maior volatilidade e menor volume [2]. Ressalta-se que um alto nível de volatilidade pode ser usado a favor do investidor para planejar estratégias de negociação a fim de otimizar seus ganhos [3].

Tanto como um mercado emergente quanto como uma linha de pesquisa, as criptomoedas e o comércio de criptomoedas têm tido um aumento notável de interesse [4]. Atualmente, alguns pesquisadores se dedicam a analisar a eficiência do mercado e a volatilidade dos preços de criptomoedas [5].

Devido a tamanha variação de preço, investidores buscam gerar receita por meio da compra e venda de criptomoedas. Para isso, eles adotam técnicas de mercado para oferecer suporte para as suas decisões. As principais técnicas utilizadas se baseiam na Análise Técnica [10] ou na Análise Fundamentalista [11].

Segundo Vidotto [30], a análise técnica se define na interpretação de gráficos de preços, volumes e outros indicadores. Este tipo de análise é uma metodologia que visa estudar o movimento de preços das ações, relacionados aos volumes negociados, para determinar

tendências de alta, de estabilidade ou de baixa, em busca da oportunidade de comprar e vender ações a preços compensadores.

Segundo Cavalcante et al. [13], a análise fundamentalista é uma metodologia de análise para determinar o preço justo de um ativo, se baseando em indicadores técnicos para chegar a um veredito. Para as criptomoedas, esse tipo de análise pode usar o valor de transação, o *whitepaper* da criptomoeda, o valor de mercado, volume, os endereços ativos, além de outros indicadores [12]. A partir desta análise, o tomador de decisão define se deve comprar ou vender determinado ativo financeiro.

A diversificação de investimentos é recomendada na gestão de portfólio, o que tem levado investidores a recorrer tanto a ativos mais tradicionais, como *commodities* e imóveis, quanto a alternativas mais recentes, como *Non-fungible tokens* (NFTs) e criptomoedas. Normalmente, os investimentos alternativos têm uma correlação histórica mais baixa com os ativos convencionais, como ações e títulos, o que proporciona uma boa diversificação da carteira. Assim, a criptomoeda pode ser uma boa forma de investimento alternativo [6], chamando atenção da literatura para entender a sua dinâmica, o que é de interesse de investidores [7].

O aprendizado de máquina é uma abordagem eficiente para desenvolver estratégias de compra e venda de criptomoeda [8], porque ele permite a inferência de relacionamentos de dados que muitas vezes não são diretamente observáveis por humanos. Para criar um modelo de aprendizado de máquina, precisamos passar pelos seus estágios de construção, sendo eles a coleta de dados, o pré-processamento dos dados, a análise exploratória dos dados, o ajuste de hiper-parâmetros, a predição e a avaliação. Em cada etapa podemos utilizar técnicas de aprendizado de máquina, que são caracterizadas por investigar como as máquinas podem adquirir conhecimento através da extração de padrões a partir de um conjunto de dados, buscando o desenvolvimento de algoritmos que permitam que computadores possam se tornar capazes de tomar decisões com certa autonomia [9].

O presente trabalho de iniciação científica buscou realizar a predição da tendência de preço de criptomoedas no próximo dia a partir de dados históricos de suas cotações, i.e. dizer ao usuário se o valor irá subir ou descer no dia seguinte ao momento atual, o que caracteriza um problema de *classificação de dados*. Foram considerados preditores baseados em aprendizado de máquina e na estatística clássica para tal efeito.

Inicialmente, foram estudadas algumas das técnicas mais utilizadas em cada estágio da construção de modelos de predição baseados em aprendizado de máquina: no pré-processamento, a normalização dos dados e o janelamento dos dados; na classificação, as máquinas de vetores-suporte (SVMs, do inglês *support-vector machines*), as redes neurais do tipo perceptron multicamadas (MLP, do inglês *multi-layer perceptron*) e memória de longo-curto prazo (LSTM, do inglês *long short-term memory*), e o modelo autoregressivo integrado de médias móveis (ARIMA, do inglês *autoregressive integrated moving average*); no ajuste de hiperparâmetros, a janela deslizante (do inglês *increasing window cross validation*) e a busca em grade (do inglês *grid search*).

Dessa forma, foi possível fazer comparações entre os quatro algoritmos de classificação implementados e chegar a observações importantes sobre as possíveis vantagens e desvantagens de cada um no contexto de previsão da tendência de preço de criptomoedas. O presente relatório está organizado da seguinte maneira: a Seção 2 possui a fundamentação teórica necessária para entender o relatório. A Seção 3 contém a revisão bibliográfica. Já, na Seção 4, é apresentada a metodologia empregada para a realização da pesquisa, enquanto que os resultados obtidos e suas discussões foram descritos na Seção 5. A Seção 6 traz as conclusões extraídas ao longo de todo o desenvolvimento do trabalho. Por fim, a Seção 7 contém o que pode ser feito futuramente neste projeto.

## 2 Fundamentação Teórica

Os seguintes conceitos foram fundamentais para a realização deste projeto: séries temporais, aprendizado de máquina e seus algoritmos de treinamento, redes neurais artificiais e seus algoritmos de treinamento, ajuste de hiperparâmetros e o ARIMA, um modelo da estatística clássica. Portanto, foram implementados modelos de duas abordagens diferentes (aprendizado de máquina e estatística clássica) para compararmos os resultados obtidos ao final dos experimentos.

### 2.1 Séries Temporais

Uma série temporal representa uma coleção de observações feitas de maneira sequencial em um período. A previsão de séries temporais é uma prática amplamente adotada, na qual é possível estimar o valor futuro de uma variável tendo como base o seu histórico [14]. Essas estimativas são de extrema utilidade pois apoiam o processo de tomada de decisão em diferentes aplicações.

A estrutura de uma série temporal é definida por suas características, as quais afetam em sua análise e modelagem. Dentre elas, a estacionariedade de uma série é medida pela média e variância dos valores: caso elas sejam iguais ou próximas durante todo tempo observado, a série é dita estacionária. Já a sazonalidade, identifica uma repetição de comportamento durante um período específico. Um exemplo seria o consumo de energia no inverno em países frios, que sofre um acréscimo na quantidade gasta dado que as pessoas tendem a utilizar mais energia durante esse período específico do ano. A análise de uma série temporal tem o objetivo de modelar o fenômeno em questão, obter conclusões em termos estatísticos e avaliar a adequação do modelo de previsão [15].

### 2.2 Aprendizado de Máquina

O aprendizado de máquina é uma abordagem interessante para predição da tendência de preço, porque ele permite a inferência de relacionamentos de dados que muitas vezes não são diretamente observáveis por humanos [31]. As técnicas de aprendizado de máquina são caracterizadas por investigar como as máquinas podem adquirir conhecimento através da extração de padrões a partir de um conjunto de dados, buscando o desenvolvimento de algoritmos que permitam que computadores possam se tornar capazes de tomar decisões com certa autonomia [32]. Dentre as técnicas de aprendizado de máquina, encontra-se o SVM (*support vector machine* do inglês) e as redes neurais artificiais, que serão discutidos mais à frente. Ele é um dos métodos mais populares na literatura para lidar com problemas de classificação e, portanto, também foi utilizado neste projeto.

#### 2.2.1 *Support Vector Machine*

*Support vector machines* são uma família de métodos de aprendizado de máquina que inicialmente foi introduzida para lidar com o problema de classificação, mas posteriormente se generalizou para resolver outros tipos de problemas. Sua proposta combina ideias da

estatística em busca de um limite adequado no espaço de dados capaz de separar as amostras em classes distintas [41].

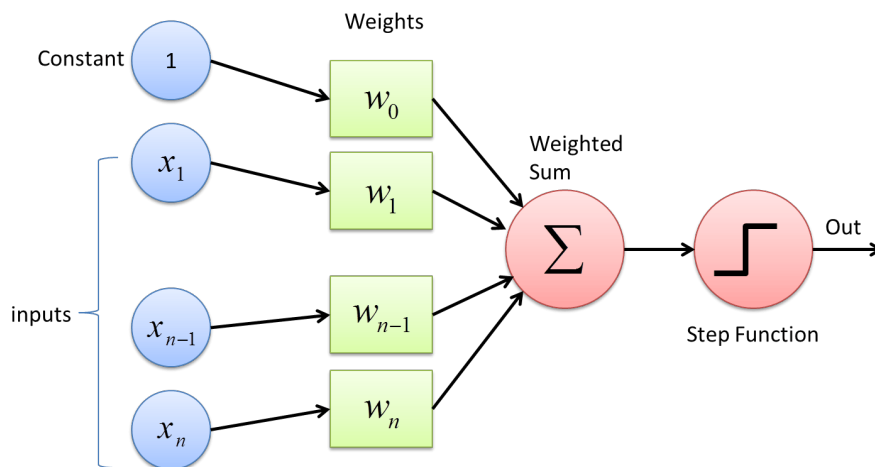
Atualmente, o SVM é empregado em diversos domínios de aplicação, como reconhecimento facial, categorização de texto, bioinformática, reconhecimento de escrita, entre muitos outros.

### 2.2.2 Redes Neurais Artificiais

As redes neurais artificiais (RNA) são modelos capazes de processar informação de forma paralela e armazenar conhecimento e experiência a fim de torná-lo disponível para uso posterior [16]. As RNA são formadas por neurônios artificiais cujo objetivo, a exemplo dos neurônios do cérebro humano, é aprender e tomar decisões baseadas em seu próprio aprendizado [17].

O Perceptron é um modelo de neurônio criado por Rosenblatt [18] para reconhecimento de padrões. Podemos observar o funcionamento do Perceptron na Figura 1. Primeiramente, multiplicamos pesos ( $W$ ) aos valores de entrada ( $X$ ) e adicionamos uma constante. Em seguida, realiza-se um somatório ( $\Sigma$ ) dos resultados obtidos. Após isso, passamos pela função de ativação a fim de introduzir um componente não-linear no Perceptron, fazendo com que eles possam aprender mais do que relações lineares entre as variáveis [19]. Por fim, o resultado obtido após a aplicação da função de ativação é o valor de saída do Perceptron.

**Figura 1:** Funcionamento do Perceptron. Disponível em [20].



Dentre todas as funções de ativação, a função ReLU é uma das mais utilizadas nos trabalhos envolvendo RNAs. Seu acrônimo significa Unidade Linear Retificada (*Rectified Linear Unit* do inglês). Seu funcionamento ocorre da seguinte maneira: o valor de saída é o valor máximo entre 0 e  $x$ , assumindo zero quando  $x$  é menor que zero e  $x$  quando esse valor é maior do que zero [33]. Foi demonstrado que essa função permite um melhor treinamento de redes neurais, gerando menor erro durante o treinamento quando comparada com as outras funções de ativações [34].

Uma coleção de Perceptrons forma uma rede neural, que deve ser configurada de modo que um conjunto de entradas produza o conjunto desejado de saídas. Uma maneira de definir os valores dos pesos é treinar a rede neural e deixá-la mudar seus próprios pesos de acordo com alguma regra de aprendizado.

### 2.2.2.1 *Multi-Layer Perceptron*

A maioria das pesquisas voltadas à predição de dados recorre à rede neural do tipo *Multi-Layer Perceptron* (MLP), consideradas aproximadoras universais de funções, para realizar este tipo de tarefa. A MLP consiste em um sistema de nós (perceptrons) interconectados. Os nós são conectados por pesos e sinais de saída que são uma função da soma das entradas para o nó modificado por uma transferência não linear simples, ou função de ativação [35]. A arquitetura de um perceptron multicamada é variável, mas em geral consistirá em várias camadas de neurônios. A camada de entrada não desempenha nenhum papel computacional, mas apenas serve para passar o vetor de entrada para a rede [36].

Aprender para o MLP é o processo de adaptar os pesos das conexões a fim de obter uma diferença mínima entre a saída da rede e a saída desejada, por isso são usados algoritmos capazes de melhorar a eficiência do aprendizado. Deles, o mais usado é o de retropropagação, o qual é baseado em técnicas de gradiente descendente [37].

### 2.2.2.2 *Long Short-Term Memory*

Outra técnica muito popular na literatura e também utilizada nesse trabalho é a *Long Short-Term Memory* (LSTM). A arquitetura LSTM consiste em um conjunto de sub-redes conectadas recorrentemente, conhecidos como blocos de memória. Esses blocos podem ser pensados como a memória de um computador [38]. Cada bloco contém uma ou mais células de memória autoconectadas e três unidades multiplicativas – as portas de entrada, saída e esquecimento – que fornecem mecanismos análogos de gravação, leitura e redefinição das operações para as células [39].

Uma rede LSTM é igual a uma rede neural artificial do tipo MLP, exceto que as unidades de soma na camada oculta são substituídas por blocos de memória. O modelo LSTM é treinado usando o gradiente, conforme apresentado por [40], para ajustar os pesos envolvidos na rede.

## 2.3 ARIMA

O modelo ARIMA (modelo de média móvel integrado auto regressivo) [22] tem uma abordagem matemática prática com impacto fundamental na análise de séries temporais e aplicações de previsão. A ideia é que a série temporal em estudo é gerada por um processo estocástico, cuja natureza pode ser representada a partir de um modelo matemático, que ainda pode ser generalizado para presença de sazonalidade [23].

O modelo ARIMA possui os parâmetros  $p, q$  e  $d$ , sendo que:

- $p$  é o número de termos autoregressivos;
- $q$  é o número de diferenciações;
- $d$  é o número de médias móveis.

A metodologia de aplicação do modelo ARIMA, envolve três etapas iterativas. A estacionariedade da série é um requisito auxiliar para utilização do modelo, portanto na parte de identificação é necessário realizar tal verificação e adaptação, caso necessário. Em sequência a definição de parâmetros é essencial para adaptação do modelo à série. A etapa final é verificar a adequação do modelo, averiguando se os erros utilizados são satisfatórios.

### 3 Revisão Bibliográfica

Algoritmos baseados em aprendizado de máquina vêm sendo utilizados para a predição da tendência no mercado de criptomoedas. Portanto, este estudo realizou uma revisão bibliográfica a fim de identificar as principais técnicas de aprendizado de máquina utilizadas para este tipo de tarefa e o tamanho da base de dados de cada criptomoeda.

A primeira etapa consistiu em realizar um levantamento bibliográfico sobre criptomoedas e *machine learning* a fim de identificar bases de dados disponíveis, quais algoritmos obtêm mais sucesso com predição de séries temporais e as métricas mais adequadas para se utilizar na medição de sucesso do modelo.

O primeiro artigo que revisamos foi o de Nakamoto [24], o qual apresentou o conceito de criptomoeda. A partir dele compreendemos melhor as características principais de nosso objeto de estudo. Após isso, começamos a nos aprofundar nos termos de aprendizado de máquina por meio de diversos artigos e livros publicados, dentre eles o livro de Haykin [25]. Com isso, conseguimos implementar métodos de pré-processamento de dados e ajuste de hiperparâmetros que melhoraram o desempenho dos algoritmos posteriormente.

O último tópico levantado nessa etapa foi o de predição de criptomoedas utilizando aprendizado de máquina, como feito em [2]. Esse tópico foi importante para nos situarmos em relação a como está a pesquisa em predição de criptomoedas atualmente. Dessa maneira, podemos observar os principais modelos de *machine learning* e métricas de avaliação de resultados utilizados nesse ramo de pesquisa.

Conforme visto em [26] [27] [28], muitos dos projetos desenvolvidos utilizam o *Multi-Layer Perceptron* (MLP) juntamente com outros tipos de redes neurais para realizar as predições. Assim, observamos que elas são amplamente utilizadas para a predição de criptomoedas. Portanto, optamos por implementá-las no projeto e iremos aprofundar mais a discussão sobre essa escolha nas próximas seções.

Em [29] podemos ver os resultados de predição obtidos por cada tipo de rede neural em um intervalo de um dia, formato que acabou sendo adotado para esse projeto. Além disso, ao invés de fazer a predição do preço exato, é feita a predição da tendência, i.e. se o preço tende a aumentar ou se tende a diminuir. Tal metodologia também foi adotada pela nossa pesquisa.

Hitam e Ismail [16] realizaram um estudo comparativo da performance dos algoritmos de aprendizado de máquina em relação a seis criptomoedas, sendo elas a Bitcoin, Ethereum, Litecoin, Nem, Ripple e Stellar. Eles chegaram à conclusão de que o algoritmo SVM é um modelo confiável para predição de criptomoeda, uma vez que ele obteve uma acurácia de 95,5%. Além disso, também foi notado que a qualidade dos dados de treinamento tem um papel importante para que se possa fazer uma predição de sucesso.

Já Livieris, Kiriakidou, Stravoyiannis e Pintelas [17], propuseram um modelo de previsão que utiliza como entrada diferentes dados de criptomoedas e os trata de forma independente para explorar informações úteis de cada criptomoeda separadamente. Foi feito um estudo empírico usando três anos de dados das três criptomoedas com a maior capitalização de mercado na época, ou seja, Bitcoin (BTC), Ethereum (ETH) e Ripple (XRP). A análise experimental detalhada revelou que o modelo proposto tem a capacidade de explorar eficientemente dados mistos de criptomoedas, reduzir o *overfitting* e diminuir o custo computacional em comparação com as redes neurais tradicionais.

Yiying e Yeze [12] focaram na dinâmica não estacionária de preços de três criptomoedas: Bitcoin, Ethereum e Ripple. Sua abordagem teve como objetivo identificar e entender os fatores que influenciam o valor dessas moedas digitais. Os dados coletados continham 1.030 dias de negociação em relação aos preços de abertura, alta, baixa e fechamento. Eles realizaram uma análise experimental que revelou que os modelos *Long Short Term Memory* (LSTM) levam a resultados superiores aos das RNAs clássicas, indicando que os modelos

LSTM são mais capazes de explorar informações ocultas em dados históricos. Além disso, os autores afirmaram que provavelmente a razão para a eficiência das redes LSTM é que elas tendem a depender mais da dinâmica de curto prazo, enquanto as RNAs clássicas tendem a depender mais do histórico de longo prazo. No entanto, caso sejam fornecidas informações históricas suficientes, as RNAs clássicas podem atingir uma precisão semelhante às redes LSTM.

Outro trabalho, desenvolvido por Garcia e de Oliveira [18], utilizou técnicas de inteligência computacional para a predição do mercado de criptomoedas. Sendo mais específico, a pesquisa utilizou o modelo ARIMA, a técnica de Redes Neurais do tipo LSTM e o método de regressão baseada em vetores de suporte (do inglês, *Support Vector Regression - SVR*). Cabe ressaltar que em [18] foram usados apenas conjuntos de dados com mais de 6 anos de registros diários. Por fim, foi concluído que a LSTM obteve resultados ligeiramente superiores aos demais métodos.

## 4 Materiais e Métodos

O fluxograma apresentado na Figura 2 contém a metodologia empregada no desenvolvimento deste projeto. Os blocos em amarelo correspondem às etapas de pré-processamento de dados e os vermelhos à escolha de algoritmos e ajuste de hiperparâmetros. Todo esse processo é descrito mais detalhadamente nesta seção.

### 4.1 Ambiente Computacional

Nós desenvolvemos todas as implementações e experimentos utilizando a linguagem de programação Python versão 3.8.10 e as seguintes bibliotecas: *Pandas* versão 1.5.3, *Numpy* versão 1.23.5, *Tensorflow* versão 2.12.0, *Matplotlib* versão 3.6.2, *Scikit-learn* versão 1.2.2 e *Statsmodels* versão 0.14.0. O ambiente de desenvolvimento usado foi a plataforma *Google Colab*. O código-fonte desenvolvido assim como a base de dados original e pré-processada estão disponíveis no Github<sup>1</sup>.

### 4.2 Base de Dados de Criptomoedas

Para a realização das atividades propostas foi utilizado o banco de dados público do *Yahoo Finance*<sup>2</sup>, que contém dados diários de negociação de todas as criptomoedas do mercado, desde o seu lançamento até os dias de hoje. A base de dados em questão possui dados do preço de abertura, preço de fechamento, maior preço alcançado no dia, menor preço alcançado no dia e volume de transações da criptomoeda no dia.

No presente trabalho, foram selecionadas cinco criptomoedas para realizar os experimentos, sendo que o critério de decisão utilizado para realizar esta escolha foi o valor de mercado. Valor de mercado é o termo usado para se referir ao preço total de todas as moedas juntas de determinada criptomoeda. Ele é calculado multiplicando-se o número de moedas em circulação pelo preço atual de cada uma. O próprio site do *Yahoo Finance* fez essa ordenação e as criptomoedas escolhidas foram Bitcoin (BTC), Ethereum (ETH), Tether (USDT), BNB e USD Coin (USDC), conforme a Figura 3, que apresenta a coluna valor de

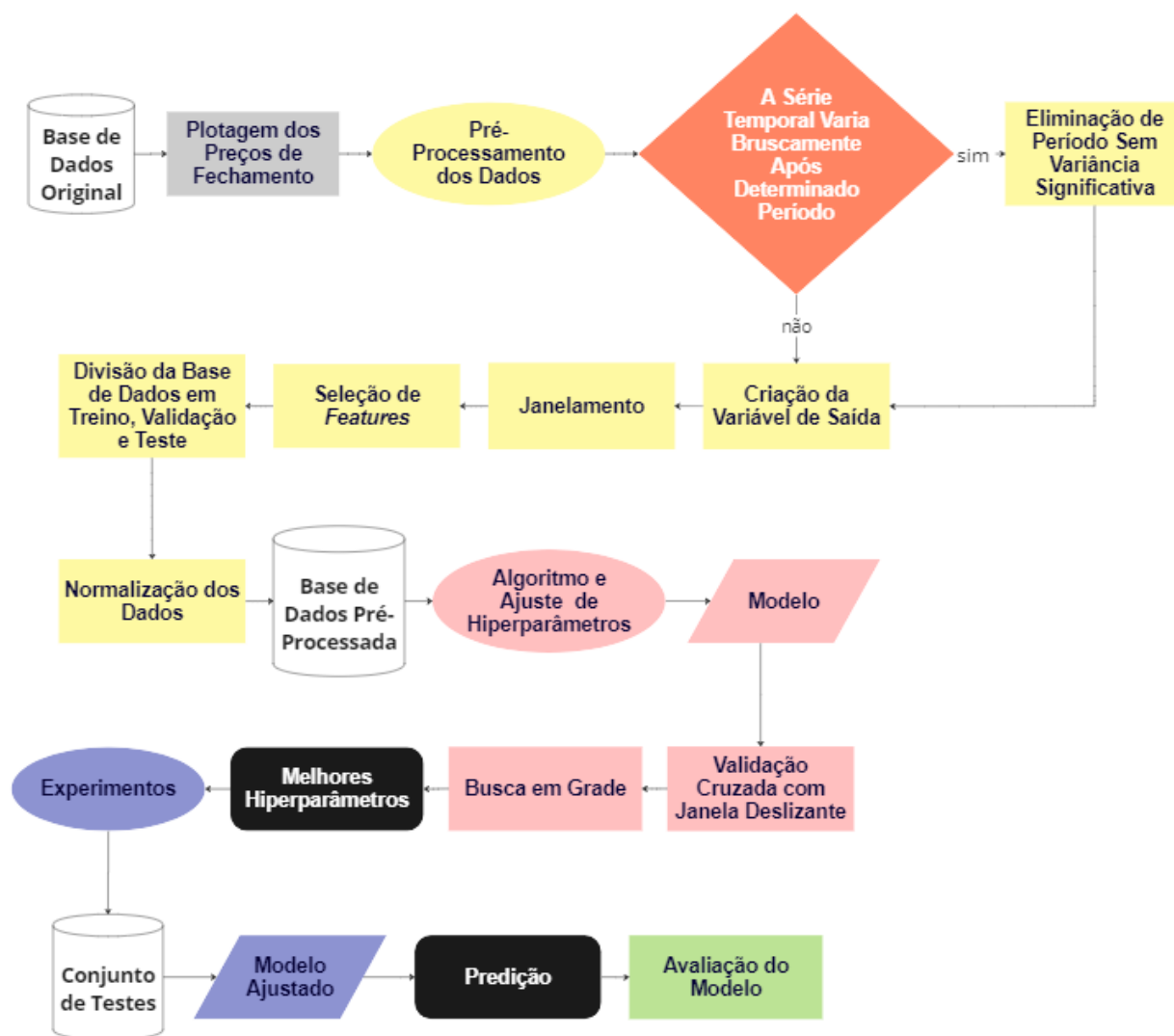
---

<sup>1</sup> <https://github.com/renan-ferretti/Crypto-Forecasting>






<sup>2</sup> <https://finance.yahoo.com/crypto/>

mercado (*market cap*, do inglês) para cada moeda considerada aqui.

**Figura 2:** Diagrama da metodologia empregada.



**Figura 3:** Criptomoedas com maior valor de mercado em Agosto de 2023.

Symbol	Name	Price (Intraday)	Change	% Change	Market Cap ▾
 <b>BTC-USD</b>	Bitcoin USD	25,980.97	+126.67	+0.49%	505.996B
 <b>ETH-USD</b>	Ethereum USD	1,637.78	+1.11	+0.07%	196.888B
 <b>USDT-USD</b>	Tether USDt USD	0.999952	+0.000024	+0.00%	82.896B
 <b>BNB-USD</b>	BNB USD	214.18	-0.06	-0.03%	32.951B
 <b>USDC-USD</b>	USD Coin USD	1.0002	+0.0001	+0.01%	26.208B



### 4.3 Pré-processamento de dados

Nesta etapa aplicamos um conjunto de técnicas para converter os dados brutos em dados preparados, ou seja, dados em formatos úteis para a nossa aplicação. Inicialmente, foi constatado que a BNB, BTC e ETH não sofreram muitas variações em seu preço até determinado período. Portanto, consideramos as amostras coletadas a partir desse período de mudança, visto que, antes disso, a alteração no preço não era significativa e a sua rápida ascensão acabou comprometendo o funcionamento do modelo. A Figura 4 demonstra a eliminação das amostras entre 2015 e 2018 da base de dados da BTC utilizada no projeto, em azul temos o intervalo que foi considerado para os experimentos. Ao final dessa etapa, os dados utilizados da BNB foram de 2021 a 2023, da BTC foram de 2018 a 2023, da ETH foram de 2021 a 2023, da USDC foram de 2018 a 2023 e da USDT foram de 2017 a 2023.

**Figura 4:** Eliminação de período na base de dados da BTC.



Após a definição do período de estudo para cada criptomoeda, foi criada a variável de saída (a ser predita) com base na diferença entre o preço de fechamento da criptomoeda no dia atual e no dia anterior. Para isso, definimos a variável de saída como sendo um (1), representando uma subida no preço da criptomoeda, e zero (0), representando uma descida ou uma não-variação no preço, sendo o último extremamente raro. Na Tabela 1 podemos encontrar a distribuição entre essas duas classes nos conjuntos de treinamento e teste de cada criptomoeda.

Posteriormente, aplicamos um janelamento de quatro dias pela nossa base, ou seja, agrupamos todos os dados de três dias atrás com os dados do dia atual a fim de fornecer mais dados de entrada para o modelo. Dessa maneira, espera-se que os modelos sejam mais eficazes ao prever a tendência do preço de cada dia, visto que eles terão mais dados para tal efeito. É importante mencionar, que os dados, devidamente janelados, foram divididos cronologicamente em dois conjuntos: 80% para treinamento e validação, e 20% para teste. Esta divisão foi feita uma única vez e todos os modelos consideraram os mesmos conjuntos de treinamento/validação e teste. Por fim, utilizamos os valores máximos e mínimos do conjunto de treinamento e validação para realizar a normalização min-max em cada atributo da base de dados, com o objetivo de evitar que o algoritmo fique enviesado para as variáveis com maior ordem de grandeza. A Tabela 2 mostra todos os atributos que foram empregados como entrada dos modelos baseados em aprendizado de máquina.

**Tabela 1:** Distribuição entre subida e descida

Criptomoeda	Quantidade de amostras	Divisão no conjunto de treinamento (% subida : % descida)	Divisão no conjunto de teste (% subida : % descida)
BNB	2.058	51:49	52:48
BTC	2.971	52:48	48:52
ETH	2.058	51:49	51:49
USDC	1.725	51:49	50:50
USDT	2.058	49:51	50:50

**Tabela 2:** atributos que serviram de entrada para a SVM, MLP e LSTM

Atributos	Preço de abertura	Alta	Baixa	Preço de fechamento
-----------	-------------------	------	-------	---------------------

Dessa forma, o modelo recebe dados referentes a quatro dias desses quatro atributos, totalizando um conjunto de 16 valores sendo passados como entrada, conforme citado anteriormente. Diferentemente dos modelos baseados em aprendizado de máquina, o ARIMA utilizou apenas a série histórica do preço de fechamento.

#### 4.4 Algoritmos e ajuste de hiperparâmetros

Foram implementados para a etapa de experimentos deste projeto o algoritmo de aprendizado de máquina *support-vector machine* (SVM), a rede neural artificial do tipo *multi-layer perceptron* (MLP), a rede neural artificial do tipo *long short-term memory* (LSTM) e o modelo autoregressivo integrado de médias móveis (ARIMA). Eles foram escolhidos devido ao seu amplo uso na literatura para a predição de séries temporais [10][11][12]. Em especial, o ARIMA foi utilizado pois aborda o problema com outra perspectiva, sem utilizar aprendizado de máquina. Ele é popular na predição de séries temporais pelo fato de observar diferentes estruturas temporais nos dados, como a tendência e a sazonalidade. Para realizar os experimentos foram desenvolvidos quatro *scripts* que utilizam módulos com uma implementação pronta de cada algoritmo, utilizando a linguagem de programação Python. Portanto, os scripts desenvolvidos tiveram como objetivo criar o modelo a partir dos parâmetros que serão passados a ele.

Para a obtenção de uma boa combinação de hiperparâmetros dos diferentes modelos de classificação baseados em aprendizado de máquina, realizamos uma busca em grade (*grid search*). Ela se trata de uma pesquisa exaustiva apoiada em todas as combinações possíveis dentro de um subconjunto definido do espaço de hiperparâmetros, na qual o desempenho de cada combinação é avaliado usando métricas pré-definidas. Além disso, implementamos uma validação cruzada com janela deslizante de 5 pastas (*5-folds*) em conjunto com a busca em grade. Essa espécie de validação cruzada consiste em deixar um tamanho dinâmico para o conjunto de treinamento e um fixo para a validação, desse modo a janela de treinamento cresce a cada iteração, conforme o exemplo fornecido na Figura 5.

A busca em grade levou em conta, para as redes neurais, a quantidade de camadas ocultas, a quantidade de neurônios em cada camada e a taxa de aprendizado do algoritmo de treinamento, conforme a Tabela 3. Já para o SVM foi considerado o tipo de kernel a ser usado e seu coeficiente, e um limite superior na fração de erros de margem e um limite inferior na fração de vetores de suporte ( $\nu$ ), ambos representados por um mesmo valor, conforme a Tabela 4. Ao final, a combinação de valores dos hiperparâmetros que levou cada modelo ao melhor desempenho em termos da média da métrica *F1-score* no conjunto de validação foi a escolhida.

Para o modelo ARIMA, não foi utilizada a busca em grade no seu ajuste de hiperparâmetros. Ao invés disso seguimos as seguintes etapas: primeiramente, realizamos o teste de *Dickey-Fuller* a fim de verificar a estacionariedade da série temporal dos preços de fechamento de cada criptomoeda. As séries da BNB, BTC e ETH foram identificadas como não estacionárias, portanto foi aplicada uma diferenciação nos dados delas para adaptação ao modelo, o que não foi o caso da USDC e USDT. Após isso, em sua construção, definimos os parâmetros como ARIMA (3,0,1) ou ARIMA (3,1,1), a depender da criptomoeda. Sendo o primeiro parâmetro indicador de quantos dias anteriores ao atual ele deve considerar, o segundo a quantidade de diferenciações realizadas na série temporal e o terceiro a ordem da média móvel.

**Figura 5:** Validação cruzada com janela deslizante de 3 pastas [21]



**Tabela 3:** valores de hiperparâmetros da MLP e LSTM testados na busca em grade.

hiperparâmetros	valores			
quantidade de camadas ocultas	1	2	3	4
neurônios em cada camada	10	30	50	70
taxa de aprendizado	0,001	0,005	0,01	

## 5 Resultados

Nessa seção serão apresentados os resultados obtidos através da implementação da metodologia apresentada anteriormente. Um repositório com o código desenvolvido, bem

como os dados pré-processados utilizados, está disponível no Github<sup>3</sup>.

**Tabela 4:** valores de hiperparâmetros da SVM testados na busca em grade

hiperparâmetros		valores		
kernel	linear	poly	rbf	sigmoid
coeficiente do kernel	scale		auto	
nu	0,2	0,4	0,6	0,8

Na Tabela 5 podemos encontrar os resultados da etapa de ajuste de hiperparâmetros para as redes neurais MLP e LSTM, com o número de camadas ocultas, número de neurônios por camada e taxa de aprendizado selecionados. Já na Tabela 6, temos a seleção de hiper-parâmetros da SVM, com o tipo de kernel e seu coeficiente, e o valor do limite superior na fração de erros de margem e do limite inferior na fração de vetores de suporte (nu).

A Tabela 7 apresenta os desempenhos obtidos por cada modelo de aprendizado de máquina para o conjunto de testes em termos de *F1-score* e acurácia, para cada uma das criptomoedas selecionadas para este projeto.

Os melhores hiperparâmetros encontrados para as redes neurais se alteram para cada criptomoeda, exceto a quantidade de camadas ocultas da MLP e LSTM. Uma RNA com duas camadas ocultas é a arquitetura mais vezes encontrada dentre os hiperparâmetros com melhor desempenho no conjunto de validação, sendo vista em 70% dos casos. Já o valor do limite superior na fração de erros de margem e do limite inferior na fração de vetores de suporte (nu), após se ajustar os hiperparâmetros da SVM, está presente em 100% dos casos.

Notamos que a USDC e a USDT levaram aos mesmos valores de hiperparâmetros. Uma hipótese para isso ter ocorrido é que as séries temporais dessas criptomoedas são muito similares, portanto os hiperparâmetros também são parecidos. Outro ponto a se ressaltar é que os valores definidos para o kernel não são muito diversos entre si, dado que foram fornecidos quatro tipos diferentes e apenas dois conseguiram as melhores métricas no conjunto de validação.

Na Tabela 7 foram destacadas as melhores medidas obtidas para cada criptomoeda. De maneira geral, nos experimentos realizados tivemos comportamentos diferentes por parte de cada algoritmo. A SVM e a LSTM apresentaram resultados satisfatórios para todas as cinco criptomoedas, com destaque para a BNB, BTC e ETH. A arquitetura usada pela SVM para essas três criptomoedas foi kernel linear, coeficiente do kernel automático e 0,2 de nu (BNB); kernel polinomial, coeficiente do kernel escalar e 0,2 de nu (BTC); e kernel linear, coeficiente do kernel automático e 0,2 de nu (ETH). Já a arquitetura empregada pela LSTM nelas foi de 2 camadas ocultas, 30 neurônios por camada e 0,001 como taxa de aprendizado (BNB); 1 camada oculta, 50 neurônios por camada e 0,005 como taxa de aprendizado (BTC); e 3 camadas ocultas, 70 neurônios por camada e 0,001 como taxa de aprendizado (ETH). Em relação à rede neural do tipo MLP, ela teve um ótimo desempenho na predição do preço de fechamento da BNB e ETH, porém não conseguiu ter o mesmo rendimento para as demais criptomoedas. Já o ARIMA atingiu resultados inferiores aos modelos baseados em aprendizado de máquina para todas criptomoedas. A SVM obteve o melhor desempenho entre

<sup>3</sup> <https://github.com/renan-ferretti/Crypto-Forecasting>

todas as predições ao realizar a previsão para a BNB, com F1-score de 98,94% e acurácia de 98,89%.

**Tabela 5:** Melhores hiper-parâmetros da MLP e LSTM.

<b>Criptomoeda</b>	<b>Modelo</b>	<b>Camadas ocultas</b>	<b>Neurônios por camada</b>	<b>Taxa de aprendizado</b>
BNB	MLP	2	70	0,001
	LSTM	2	30	0,001
BTC	MLP	4	70	0,01
	LSTM	1	50	0,005
ETH	MLP	2	70	0,005
	LSTM	3	70	0,001
USDC	MLP	2	50	0,001
	LSTM	2	10	0,001
USDT	MLP	2	70	0,005
	LSTM	2	10	0,01

**Tabela 6:** Melhores hiper-parâmetros da SVM.

<b>Criptomoeda</b>	<b>Kernel</b>	<b>Coefficiente do kernel</b>	<b>nu</b>
BNB	linear	auto	0,2
BTC	poly	scale	0,2
ETH	linear	auto	0,2
USDC	poly	scale	0,2
USDT	poly	scale	0,2

Dessa forma, considerando a complexidade do modelo e seu desempenho, verificamos que os modelos mais simples, como as SVM, são competitivos com opções inspiradas em redes neurais artificiais, podendo até superar determinadas abordagens.

**Tabela 7:** Desempenho dos modelos para cada criptomoeda.

Criptomoeda	Modelo	F1-score	Acurácia
BNB	SVM	<b>98,94%</b>	<b>98,89%</b>
	MLP	95,74%	95,58%
	LSTM	95,18%	95,03%
	ARIMA	49,85%	48,46%
Criptomoeda	Modelo	F1-score	Acurácia
BTC	SVM	<b>97,04%</b>	<b>97,16%</b>
	MLP	64,75%	47,88%
	LSTM	93,06%	92,27%
	ARIMA	45,55%	47,50%
Criptomoeda	Modelo	F1-score	Acurácia
ETH	SVM	<b>94,91%</b>	<b>95,02%</b>
	MLP	85,32%	82,32%
	LSTM	92,48%	92,82%
	ARIMA	48,18%	48,97%
Criptomoeda	Modelo	F1-score	Acurácia
USDC	SVM	82,91%	<b>84,34%</b>
	MLP	72,72%	62,61%
	LSTM	<b>84,96%</b>	83,48%
	ARIMA	56,01%	61,19%
Criptomoeda	Modelo	F1-score	Acurácia
USDT	SVM	85,47%	83,45%
	MLP	71,40%	59,85%
	LSTM	<b>87,50%</b>	<b>85,89%</b>
	ARIMA	58,37%	59,61%

## 6 Conclusões

Este trabalho de iniciação científica visou utilizar séries temporais de cotações diárias de criptomoedas a fim de realizar um estudo comparativo entre diferentes algoritmos de aprendizado de máquina e da estatística clássica na tarefa de predição da tendência do preço no dia seguinte. Para isso implementamos a SVM, as redes neurais MLP e LSTM, e o ARIMA para comparar os seus desempenhos ao final.

O projeto mostrou, através de diferentes modelos de predição, que o emprego de técnicas baseadas em aprendizado de máquina menos complexas podem levar a desempenhos tão bons, se não melhores, que modelos baseados em redes neurais artificiais. Dentre as opções testadas, a SVM foi a que atingiu o melhor desempenho médio entre as cinco criptomoedas utilizadas; a LSTM também alcançou resultados bastante satisfatórios, mas um pouco inferiores aos da SVM. Em relação às criptomoedas, a BNB foi a que obteve os melhores resultados no conjunto de testes ao se considerar todos os classificadores.

## 7 Perspectivas de Continuidade

Como trabalhos futuros, recomenda-se buscar uma melhora na performance do modelo ARIMA por meio de uma melhor configuração de seus hiperparâmetros, a união dos quatro modelos em um conjunto de preditores (*ensemble*) e que, a partir disso, seja feita uma simulação de investimento para cada uma das criptomoedas empregadas neste trabalho a fim de verificar se haveria lucro ao final e, se houver, de quanto seria. Outro ponto que pode-se evoluir é o emprego de mais criptomoedas a fim de verificar se o desempenho dos modelos se manterá parecido para elas.

Além disso, seria interessante desenvolver uma aplicação que utilize esse *ensemble* para fazer a predição em tempo real. O mesmo possuiria uma integração com a base de dados fornecida pelo *Yahoo Finance*, ficando sempre atualizado e, assim, diminuindo a probabilidade de erro.

Por fim, há a intenção de continuar esse projeto e enviar um artigo para publicação em congresso ou revista científica. Para isso algumas das sugestões citadas anteriormente precisam ser desenvolvidas, como a melhora do ARIMA e a simulação de investimento para as criptomoedas.

## Agradecimentos

Gostaria de expressar minha profunda gratidão a todos que contribuíram para a realização deste trabalho. Sem o apoio e incentivo de pessoas especiais, essa conquista não seria possível. Portanto, gostaria de dedicar esta sessão de agradecimentos para expressar minha sincera gratidão a cada um de vocês. À minha família, meus pais Soraya e Wladimir, minha cachorrinha Shakira, meus avós, tios e primos. Aos meus orientadores Guilherme e Arthur. Aos meus amigos e professores. Por fim, agradeço ao CNPq pelo incentivo através da bolsa de estudos.

## Bibliografia

- [1] NASIR, M.A.; HUYNH, T.L.D.; NGUYEN, S.P.; DUONG, D. Forecasting cryptocurrency returns and volume using search engines. **Financ. Innov.** 2019, 5, 2.
- [2] ALONSO-MONSALVE, Saúl; SUÁREZ-CETRULO, Andrés L.; CERVANTES, Alejandro; QUINTANA, David. Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. **Expert Systems With Applications**, v. 149, jul. 2020.
- [3] BOURI, Elie; LAU, Chi Keung Marco; LUCEY, Brian; ROUBAUD, David. Trading volume and the predictability of return and volatility in the cryptocurrency market. **Finance Research Letters**, v. 29, p. 340-346, jun. 2019.
- [4] FARELL, Ryan, "An Analysis of the Cryptocurrency Industry" (2015). **Wharton Research Scholars**. 130.
- [5] BALCILAR, Mehmet; BOURI, Elie; GUPTA, Rangan; ROUBAUD, David. Can volume predict Bitcoin returns and volatility? A quantiles-based approach. **Economic Modelling**, [S.L.], v. 64,, p. 74-81, ago. 2017.

- [6] TRIMBORN, Simon; LI, Mingyang; HÄRDLE, Wolfgang Karl. Investing with Cryptocurrencies: a liquidity constrained investment approach. **Journal Of Financial Econometrics**, v. 18, n. 2, p. 280-306, 3 jun. 2019.
- [7] NASIR, Muhammad Ali et al. Forecasting cryptocurrency returns and volume using search engines. **Financial Innovation**, v. 5, n. 1, p. 1-13, 2019.
- [8] MCNALLY, Sean; ROCHE, Jason; CATON, Simon. Predicting the Price of Bitcoin Using Machine Learning. **2018 26Th Euromicro International Conference On Parallel, Distributed And Network-Based Processing (Pdp)**, mar. 2018.
- [9] RUSSEL, Stuart; NORVIG, Peter. **Artificial Intelligence**. 3. ed., Elsevier, 1995. 1132 p.
- [10] MATSURA, E. Comprar ou Vender? Como investir na bolsa utilizando análise gráfica. Saraiva, 7th ed., 2013. ISBN 9788502125759.
- [11] BULKOWSKI, T.N. Fundamental Analysis and Position Trading: Evolution of a Trader. Wiley, 2012. ISBN 1118464206.
- [12] AKGÜL, Akif; ŞAHİN, Eyyüp Ensari; ŞENOL, Fatma Yıldız. Blockchain-based cryptocurrency price prediction with chaos theory, on-chain analysis, sentiment analysis and fundamental-technical analysis. **Chaos Theory and Applications**, v. 4, n. 3, p. 157-168, 2022.
- [13] CAVALCANTE, Francisco; MISUMI, Jorge Yoshio; RUDGE, Luiz Fernando. Mercado de capitais: o que é, como funciona. Rio de Janeiro: Elsevier, 2005.
- [14] ADHIKARI, R.; AGRAWAL, R. K. An introductory study on time series modeling and forecasting. arXiv preprint arXiv:1302.6613, 2013.
- [15] CASTRO, M. C. F. de. **Predição não-linear de series temporais usando redes neurais RBF por decomposição em componentes principais**. 2001. Tese (Doutorado) - University of Campinas, Brazil.
- [16] KUBAT, M. Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7. **The Knowledge Engineering Review**, Cambridge University Press, v. 13, n. 4, p. 409–412, 1999.
- [17] FLECK, L. et al. Redes neurais artificiais: Principios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 1, n. 13, p. 47–57, 2016.
- [18] ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- [19] SHARMA, S.; SHARMA, S.; ATHAIYA, A. Activation functions in neural networks. **Towards Data Science**, v. 6, n. 12, p. 310–316, 2017.



- [20] RAMCHOUN, H.; GHANOU, Y.; ETTAOUIL, M.; JANATI IDRISSE, M. A. Multilayer perceptron: Architecture optimization and training. **International Journal of Interactive Multimedia e Artificial Intelligence**, 2016.
- [21] DE OLIVEIRA CAROSIA, Arthur Emanuel. **Previsão do mercado de ações brasileiro com o uso de análise de sentimentos, indicadores técnicos e valores de ações**. 2022. Tese de Doutorado. [sn].
- [22] BOX, G. E. et al. **Time series analysis: forecasting and control**. [S.l.]: John Wiley & Sons, 2015.
- [23] BAYER, F. M. **Wavelets e modelos tradicionais de previsão: um estudo comparativo**. Rev. Bras. Biom, v. 28, n. 2, p. 40–61, 2010.
- [24] NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system. **Decentralized Business Review**, 2008.
- [25] HAYKIN, Simon, **Neural Networks and Learning Machines**. 3. ed., Prentice Hall, 2008. 926 p.
- [26] DERBENTSEV, Vasily; MATVIYCHUK, Andriy; SOLOVIEV, Vladimir N. Forecasting of cryptocurrency prices using machine learning. **Advanced Studies of Financial Technologies and Cryptocurrency Markets**, p. 211-231, 2020.
- [27] AL-NEFAIE, Abdullah H.; ALDHYANI, Theyazn HH. Bitcoin Price Forecasting and Trading: Data Analytics Approaches. **Electronics**, v. 11, n. 24, p. 4088, 2022.
- [28] KUMAR, Deepak; RATH, S. K. Predicting the trends of price for ethereum using deep learning techniques. In: **Artificial Intelligence and Evolutionary Computations in Engineering Systems**. Springer Singapore, 2020. p. 103-114.
- [29] ZHONG, Xiao; ENKE, David. Forecasting daily stock market return using dimensionality reduction. **Expert Systems with Applications**, v. 67, p. 126-139, 2017.
- [30] VIDOTTO, R. S., MIGLIATO, A.L.T, ZAMBON, A.C. **O moving average convergence divergence como ferramenta para a decisão de investimentos no mercado de ações**. ANPAD. RAC, Curitiba, v. 13, n. 2, art. 7, p. 291-309, Abr./Jun. 2009.
- [31] CAMPOREALE, E. The challenge of machine learning in space weather: Nowcasting and forecasting. **Space Weather**, Wiley Online Library, v. 17, n. 8, p. 1166–1207, 2019.
- [32] XU, Y. et al. Artificial intelligence: A powerful paradigm for scientific research. **The Innovation**, Elsevier, v. 2, n. 4, p. 100179, 2021.
- [33] HAHNLOSER, R. H. et al. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. **nature**, Nature Publishing Group UK London, v. 405, n. 6789, p. 947–951, 2000.

- [34] HARA, K.; SAITO, D.; SHOUNO, H. Analysis of function of rectified linear unit used in deep learning. In: IEEE. 2015 **international joint conference on neural networks (IJCNN)**. [S.l.: s.n.], 2015. P. 1–8.
- [35] GARDNER, M. W.; DORLING, S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric environment**, Elsevier, v. 32, n. 14-15, p. 2627–2636, 1998.
- [36] MURTAGH, F. Multilayer perceptrons for classification and regression. **Neurocomputing**, Elsevier, v. 2, n. 5-6, p. 183–197, 1991.
- [37] SALOMON, D. **Data compression: the complete reference**. [S.l.]: Springer Science & Business Media, 2004.
- [38] GRAVES, A.; GRAVES, A. Long short-term memory. **Supervised sequence labelling with recurrent neural networks**, Springer, p. 37–45, 2012.
- [39] HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997.
- [40] GRAVES, A.; SCHMIDHUBER, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. **Neural networks**, Elsevier, v. 18, n. 5-6, p. 602–610, 2005.
- [41] MAMMONE, Alessia; TURCHI, Marco; CRISTIANINI, Nello. Support vector machines. **Wiley Interdisciplinary Reviews: Computational Statistics**, v. 1, n. 3, p. 283-289, 2009.