

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JOÃO GUILHERME SQUINELATO DE MELO

**UM MODELO DE PREDIÇÃO PARA O MERCADO ACIONÁRIO
BASEADO EM UMA REDE NEURAL RECORRENTE E ANÁLISE
TÉCNICA DE AÇÕES**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2021

JOÃO GUILHERME SQUINELATO DE MELO

**UM MODELO DE PREDIÇÃO PARA O MERCADO ACIONÁRIO
BASEADO EM UMA REDE NEURAL RECORRENTE E ANÁLISE
TÉCNICA DE AÇÕES**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Giovani Volnei Meinerz
Universidade Tecnológica Federal do Paraná

CORNÉLIO PROCÓPIO
2021



4.0 Internacional

Esta licença permite remixe, adaptação e criação a partir do trabalho para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Nome da Diretoria
Nome da Coordenação
Nome do Curso



TERMO DE APROVAÇÃO

Um modelo de predição para o mercado acionário baseado em uma rede neural recorrente e análise técnica de ações

por

João Guilherme Squinelato de Melo

Este Trabalho de Conclusão de Curso de graduação foi julgado adequado para obtenção do Título de “Bacharel em Engenharia de Computação” e aprovado em sua forma final pelo Programa de Graduação em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, 13/12/2021

Prof. Dr. Giovani Volnei Meinerz

Prof. Dr. Danilo Sipoli Sanches

Prof. Dr. Alessandro Goedel

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

À minha avó Aparecida Carli de Melo, aos meus pais Teresa Squinelato de Melo e Antônio José Carli de Melo, à minha irmã Ana Carolina Squinelato de Melo e à minha namorada Ingrid Silva dos Santos.

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais, Teresa Squinelato de Melo e Antônio José Carli de Melo pelo fomento e incentivo aos estudos desde sempre. Além disso, agradeço à minha irmã Ana Carolina Squinelato de Melo pelos exemplos diários de resiliência e à minha namorada Ingrid Silva dos Santos pelo suporte diário.

Agradeço à psicóloga do Núcleo de Acompanhamento Psicopedagógico e Assistência Estudantil da Universidade Tecnológica Federal do Paraná, Milena de Lima Barbosa, pelas orientações e aconselhamentos e também à minha psicóloga pessoal, Beatriz Rodrigues Caldas Lourenção, por todo suporte emocional.

Agradeço ao meu orientador, Prof. Dr. Giovani Volnei Meinerz, por me instigar à pesquisa desde 2018 com a oportunidade de uma Iniciação Científica. Sou grato também aos meus amigos de pesquisa Luis Gustavo de Souza, Vitor Fabrile Guastala e Thaysla Fernanda Gomes da Cruz. Igualmente, agradeço à Profa. Dra. Marilu Martens Oliveira por primeiro me apresentar à escrita científica.

Agradeço aos amigos que fiz em minha jornada acadêmica, Aline Mara Rudsit Bini, Ana Lívia Franco, Gabriel Lechenco Vargas Pereira e Gustavo Henrique Aguilar por todo amor e suporte recebidos.

Por fim, agradeço imensamente à Universidade Tecnológica Federal do Paraná por me proporcionar uma vivência rica no ensino, pesquisa e extensão, a qual foi essencial para o meu crescimento acadêmico, profissional e também pessoal.

The original question, "Can machines think?" I believe to be too meaningless to deserve discussion. (TURING, Alan, 1950).

RESUMO

SQUINELATO, João. Um modelo de predição para o mercado acionário baseado em uma rede neural recorrente e análise técnica de ações. 2021. 50 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2021.

Prevê-se até 2025 a geração de 175 *zettabytes* de dados. A esse grande volume dá-se o nome de *Big Data*, em que 90% desses dados são do tipo não-estruturado, como, por exemplo, as transações financeiras. Em vista disso, a bolsa de valores desempenha um papel importante na geração desses dados, uma vez que possibilita a realização em alta escala de transações financeiras de compra e venda. Operada no Brasil e com quase 4 milhões de investidores, a B3 é um exemplo de bolsa de valores que apenas no ano de 2021 foi o palco para mais de 9 bilhões de transações financeiras, que juntas movimentaram mais de 23 bilhões de reais. Para entrar nesse negócio, é necessário se tornar um(a) acionista, o que pode ser feito comprando uma ação de alguma empresa listada na B3, adquirindo, assim, uma parte da empresa, que por sua vez possibilita o compartilhamento dos ganhos que obtém. Dessa forma, o(a) acionista deve buscar técnicas para analisar o mercado acionário e, então, saber a hora de comprar e vender suas ações almejando otimizar seus lucros. Para essa difícil tarefa de prever o comportamento do mercado acionário, acionistas contam com o auxílio da Análise Técnica, a qual sustenta-se na análise histórica dos preços das ações. Além disso, abordagens baseadas em Redes Neurais Recorrentes vêm ganhando relevância na tarefa da predição de séries temporais, como são as oscilações do mercado acionário, em virtude de sua capacidade de extrair, automaticamente, características relevantes dos dados. Assim, o objetivo principal deste trabalho foi desenvolver um modelo de predição de preços de fechamento das ações da Petrobrás, baseado em uma Rede Neural Recorrente e a Análise Técnica de ações. Para isso o modelo proposto não só foi treinado apenas com as ações da Petrobrás, mas também com todas as ações das empresas que formam a carteira de ativos do ETF BOVA11, para diferentes janelas temporais. Desse modo, com os experimentos, pôde-se observar que o modelo proposto obteve os melhores resultados na predição das ações da Petrobrás quando treinado com todas as empresas que formam a carteira de ativos do ETF BOVA11 para uma janela temporal de trinta dias. O modelo proposto obteve um Erro Absoluto Médio de 0,035, assim como um R^2 no valor de 0,938 e uma correlação de Pearson de 0,98111 entre suas predições e os preços de fechamento das ações da Petrobrás. Por fim, o trabalho constatou que o modelo proposto apresentou uma boa performance em diferentes janelas temporais, além de melhores resultados ao ser treinado com o auxílio do ETF BOVA11 ao invés de utilizar apenas as ações da Petrobrás.

Palavras-chave: Mercado acionário. Análise Técnica. Redes Neurais Recorrentes.

ABSTRACT

SQUINELATO, João. A stock market prediction model based on recurrent neural network and technical analysis. 2021. 50 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2021.

The generation of 175 zettabytes of data has been expected by 2025. This large volume is called Big Data, in which only unstructured data, such as financial transactions, represents 90% of this volume. Thus, the stock market plays a significant role in the context of Big Data due to the high volume of usual financial buying and selling transactions in this market. The Brazilian stock market, also known as B3, almost reached 4 millions investors in 2021 that together accomplished 9 billions of financial buying and selling transactions and moved over 23 billions reais. To enter this market and become a shareholder, it is necessary to buy a share from a company to participate in the company's profit. Therefore, the shareholder must search for techniques and methods to analyze the stock market aiming at good profits. Recognized as a difficult task, the prediction of the behavior of the stock market also relies on Technical Analysis, which employs the historical analysis of stock prices. Additionally, approaches based on Recurrent Neural Networks have been gaining importance in time series prediction, such as stock market fluctuations, due to their ability to extract relevant characteristics from the data. Thus, the main objective of this research is to develop a prediction model for closing prices of Petrobrás stocks, based on a Recurrent Neural Network and Technical Analysis. In addition, the proposed model was not only trained with Petrobrás stocks but also with all the stocks from the companies that form the asset portfolio from the ETF BOVA11 for different time windows. It was possible with all the experiments to realize the best scenario where the proposed model achieves the best predictions of the Petrobrás stocks. The best scenario occurred when training with all the stocks from the companies that form the asset portfolio from the ETF BOVA11 for a thirty-day time window. The proposed model obtained a Mean Absolute Error of 0.035, a R^2 of 0.938, and a Pearson correlation coefficient of 0.98111 between the predictions of the model and the actual closing prices of Petrobrás stocks. Finally, this research found that the proposed model performed good predictions on different time windows, besides better results when trained with all the stocks from the companies that form the asset portfolio from the ETF BOVA11 instead of using only Petrobrás stocks.

Keywords: Stock Market. Technical Analysis. Recurrent Neural Networks.

LISTA DE FIGURAS

Figura 1 – Representação Gráfica da Variação dos Preços de uma Ação ao Longo do Tempo	6
Figura 2 – Representação de uma Série Temporal de <i>Candlesticks</i>	7
Figura 3 – Representação das Cinco Ondas de Elliot	11
Figura 4 – Quatro Grandes Etapas do Processo de KDD	12
Figura 5 – Representação do Perceptron	14
Figura 6 – Topologia de um PMC	15
Figura 7 – Representação de um Neurônio Recorrente	16
Figura 8 – Neurônios Recorrentes em Camada	17
Figura 9 – <i>Workflow</i> do Desenvolvimento do Modelo de Predição de Preços de Fechamento	26
Figura 10 – Nuvem de Palavras dos Códigos de Negociação das 83 Empresas Filtradas	28
Figura 11 – Modelo Proposto de Rede Neural Recorrente	31
Figura 12 – Modelo Linear Simples	32
Figura 13 – EAM dos Modelos em Diferentes Janelas Temporais	34
Figura 14 – <i>Overfitting</i> do EAM dos Modelos em Diferentes Janelas Temporais	35
Figura 15 – Modelos em suas Melhores Janelas Temporais Segundo a Métrica de EAM	35
Figura 16 – R^2 dos Modelos em Diferentes Janelas Temporais	36
Figura 17 – <i>Overfitting</i> do R^2 dos Modelos em Diferentes Janelas Temporais	37
Figura 18 – Modelos em suas Melhores Janelas Temporais Segundo a Métrica de R^2	38
Figura 19 – Correlação de Pearson Entre o Preço de Fechamento das Ações da Petrobrás e a Predição dos Modelos	38
Figura 20 – Correlação de Pearson Entre a Predição dos Modelos e os Resíduos da Predição	39
Figura 21 – Predição dos Modelos Linear-PETR4-20 e RNR-PETR4-30	40
Figura 22 – Predição dos Modelos Linear-PETR4-20 e Linear-BOVA11-50	40
Figura 23 – Predição dos Modelos RNR-PETR4-30 e RNR-BOVA11-30	41
Figura 24 – Predição dos Modelos Linear-BOVA11-50 e RNR-BOVA11-30	41

LISTA DE QUADROS

Quadro 1 – Trabalhos Correlatos	25
---	----

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
AF	Análise Fundamentalista
AT	Análise Técnica
SVM	<i>Support Vector Machine</i>
LSSVM	<i>Least Squares Support Vector Machine</i>
SVR	<i>Support Vector Regression</i>
RBF	<i>Radial Basis Function</i>
RNs	Redes Neurais
RNA	Rede Neural Artificial
PMC	Perceptron de Múltiplas Camadas
RNR	Rede Neural Recorrente
BPTT	<i>Backpropagation Through Time</i>
LSTM	<i>Long Short-Term Memory</i>
GRU	<i>Gated Recurrent Unit</i>
AM	Aprendizado de Máquina
AP	Aprendizado Profundo
RNP	Rede Neural Profunda
ReLU	<i>Rectified Linear Units</i>
BP	<i>Backpropagation</i>
GD	<i>Gradient Descent</i>
BGD	<i>Batch Gradient Descent</i>
SGD	<i>Stochastic Gradient Descent</i>
MBGD	<i>Mini-Batch Gradient Descent</i>
B3	Brasil, Bolsa, Balcão

IBOVESPA	Índice da Bolsa de Valores de São Paulo
CSV	<i>Comma Separed Values</i>
NoSQL	<i>Not Only SQL</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
EAM	Erro Absoluto Médio
EQM	Erro Quadrático Médio
REQM	Raiz do Erro Quadrático Médio
HME	Hipótese do Mercado Eficiente
TPA	Teoria do Passeio Aleatório
IFR	Índice de Força Relativa
MME	Média Móvel Exponencial
MACD	<i>Moving Average Convergence Divergence</i>
OBV	<i>On-Balance Volume</i>
FVM	Faixa Verdadeira Média
DAIN	<i>Deep Adaptive Input Normalization</i>
RMSProp	<i>Root Mean Square Propagation</i>
API	<i>Application Programming Interface</i>
DIJA	<i>Dow Jones Industrial Average</i>
S&P 500	<i>The Standard and Poor's 500</i>
INSE	<i>Indian National Stock Exchange</i>
NYSE	<i>New York Stock Exchange</i>
NASDAQ	<i>National Association of Securities Dealers Automated Quotations System</i>
KDD	<i>Knowledge Discovery in Databases</i>
OLS	<i>Ordinary Least Squares</i>

LISTA DE SÍMBOLOS

w	Pesos do modelo
x	Entradas do modelo
y	Resposta esperada do modelo
\hat{y}	Resposta predita pelo modelo
ϕ	Função de ativação do modelo
J	Função custo do modelo
θ	Matriz de pesos do modelo

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 Contextualização	1
1.2 Motivação	2
1.3 Caracterização do Problema	2
1.4 Objetivo do Trabalho	3
1.5 Especificação de Requisitos do Trabalho	3
1.6 Visão Geral do Documento	3
2 – FUNDAMENTAÇÃO TEÓRICA	4
2.1 Bolsa de Valores	4
2.1.1 Ações	4
2.1.2 Opções	4
2.1.3 ETFs	4
2.1.4 Fundos Imobiliários	5
2.1.5 Principais Bolsas de Valores	5
2.2 Análise Técnica	5
2.2.1 Preço	5
2.2.2 Teoria de Dow	7
2.2.3 Indicadores Técnicos	8
2.2.4 Média Móvel Exponencial	9
2.2.5 <i>Moving Average Convergence Divergence</i>	9
2.3 Análise Fundamentalista	9
2.3.1 Valor Intrínseco	9
2.3.2 Hipótese do Mercado Eficiente	10
2.3.3 Ondas de Elliott	10
2.4 KDD	11
2.5 Normalização	12
2.6 Aprendizado de Máquina	13
2.6.1 Séries Temporais	14
2.6.2 Redes Neurais	14
2.6.3 Redes Neurais Recorrentes	16
2.6.4 Aprendizado Profundo	18
2.6.5 Treinamento e validação de Redes Neurais	18
2.6.6 Técnicas para Acelerar o Treinamento	19
2.6.7 Técnicas para Evitar <i>Overfitting</i>	20
2.6.8 Avaliando Modelos	21

3 – TRABALHOS CORRELATOS	23
3.1 Síntese e Lacunas dos Trabalhos Correlatos	25
4 – METODOLOGIA DO DESENVOLVIMENTO DO MODELO	26
4.1 Seleção	27
4.2 Pré-processamento	28
4.3 Mineração	30
4.4 Treinamento e Avaliação do Modelo	30
5 – ANÁLISE E DISCUSSÃO DOS RESULTADOS	33
6 – CONCLUSÃO	42
6.1 Considerações Gerais	42
6.2 Contribuições do Trabalho	42
6.2.1 Principal Contribuição	42
6.2.2 Contribuição Complementar	43
6.3 Principais Limitações e Dificuldades	43
6.4 Considerações Finais	43
6.5 Trabalhos Futuros	44
Referências	45

1 INTRODUÇÃO

Este capítulo contém uma breve introdução ao presente Trabalho de Conclusão de Curso. Dessa forma, a [Seção 1.1](#) apresenta o contexto no qual se insere este trabalho e, na sequência, [Seção 1.2](#), as motivações para o seu desenvolvimento. Ademais, a [Seção 1.3](#) disserta sobre a caracterização do problema, bem como a [Seção 1.4](#) quanto ao objetivo deste trabalho e a [Seção 1.5](#) acerca de seus requisitos. Por fim, a [Seção 1.6](#) descreve a organização dos próximos capítulos.

1.1 Contextualização

Em 2025 estima-se que serão gerados 175 *zettabytes* de dados, isto é, 175 bilhões de *terabytes*. Para elucidar o tamanho desse volume de dados, levar-se-iam 1,8 bilhão de anos para baixá-los a uma taxa de *download* de 25 *megabits* por segundo (IDC; SEAGLE, 2018).

A esse grande volume, velocidade e variedade na geração de dados dá-se o nome de *Big Data*, em que 90% desses dados são do tipo não-estruturado (KHAN et al., 2014). Um exemplo desse tipo de dado são as transações financeiras, pois representam dados desprovidos de estrutura definida.

Nesse contexto, a bolsa de valores representa um mercado onde são efetuadas transações financeiras de compra e venda de ações, ambiente conhecido como mercado acionário. Uma ação representa uma parte de uma empresa que pode ser comprada ou vendida por investidores. Logo, quem compra uma ação também se torna acionista e, por conseguinte, compartilha os ganhos que a empresa obtém (GOMES, 1997; INFOMONEY, 2021b).

Com quase 4 milhões de investidores em 2021, a Brasil, Bolsa, Balcão (B3) – a bolsa de valores brasileira – contabilizou mais de 9 bilhões de transações financeiras de compra e venda realizadas apenas entre Janeiro de 2021 e Outubro de 2021. Essas transações movimentaram mais de 23 bilhões de reais, o que torna esse mercado atrativo à investidores, os quais buscam formas de analisar e investir seu dinheiro com eficiência (B3, 2021a; B3, 2021b).

Uma das formas de identificar padrões e tendências futuras do mercado acionário é a Análise Técnica (AT), que observa os preços históricos de uma ação para determinar suas oscilações futuras (EDWARDS; MAGEE; BASSETTI, 2018). Outrossim, a AT comumente utiliza-se de indicadores técnicos, tais como: cálculos matemáticos aplicados, por exemplo, ao preço de uma ação, para predizer futuras mudanças no preço desta ação (ACHELIS, 2000).

Diversos trabalhos vêm sendo realizados com foco no desafiador problema de se prever as oscilações do mercado acionário. No que tange ao Aprendizado de Máquina, por exemplo, Bollen, Mao e Zeng (2011) empregaram um modelo de *Self-Organizing Fuzzy Neural Network* juntamente com o método de Causalidade de Granger e a Análise de Sentimentos, a fim de prever as oscilações do *Dow Jones Industrial Average* (DJIA).

Já [Hegazy, Soliman e Salam \(2013\)](#) optaram por utilizar uma abordagem clássica com um modelo de *Least Squares Support Vector Machine* (LSSVM) para prever as oscilações da *The Standard and Poor's 500* (S&P 500). Ademais, [Chen, Zhou e Dai \(2015\)](#) propuseram um modelo baseado em Rede Neural Recorrente (RNR) para prever as oscilações dos mercados acionários de Shanghai e Shenzhen. Por fim, [Pasupulety et al. \(2019\)](#) desenvolveram um modelo de predição baseado em *Ensemble Learning* e Análise de Sentimentos de *tweets* referentes ao mercado acionário.

Contudo, Redes Neurais vem se tornando uma importante abordagem para problemas de predição devido a sua capacidade de extrair automaticamente características relevantes dos dados ([CHEN; LIN, 2014](#); [SZE et al., 2017](#)). Mais especificamente, Redes Neurais Recorrentes podem ser empregadas como modelo de predição em virtude de sua habilidade em lidar com séries temporais, tais como as oscilações de preços do mercado acionário ([GÉRON, 2017](#)).

Por fim, escolher o modelo de predição mais apropriado, bem como os dados mais significativos para o seu treinamento, representa um desafio, especialmente com a complexidade que permeia o mercado acionário.

1.2 Motivação

Não obstante, Redes Neurais Recorrentes aplicadas na predição de séries temporais – como o mercado acionário – ainda não estão aperfeiçoadas como, por exemplo, as Redes Neurais Convolucionais para problemas relativos ao processamento de imagem. Isso se traduz na necessidade de um esforço maior para encontrar guias e regras de ouro, se comparado ao esforço necessário a ser empreendido na resolução de problemas que não pertençam ao contexto das séries temporais ([NIELSEN, 2019](#)).

Diante disso, o presente trabalho busca responder alguns questionamentos:

- a) É possível desenvolver um modelo de predição baseado em Rede Neural Recorrente, unindo a Análise Técnica de ações, para o contexto da B3?
- b) Se sim, será que o modelo proposto de Rede Neural Recorrente pode superar um modelo de predição linear simples?
- c) Por fim, treinar o modelo proposto de Rede Neural Recorrente com diferentes ações de diferentes empresas pode melhorar sua performance?

1.3 Caracterização do Problema

O trabalho aborda o problema caracterizado pela dificuldade em se prever o preço de fechamento futuro de ações de empresas listadas na bolsa de valores brasileira, representada atualmente pela B3.

1.4 Objetivo do Trabalho

Em vista disso, o objetivo deste trabalho é desenvolver um modelo de predição, baseado em Rede Neural Recorrente, aplicando Análise Técnica, visando prever o preço de fechamento futuro das ações da Petrobrás negociadas na B3.

1.5 Especificação de Requisitos do Trabalho

O presente trabalho buscou atender os requisitos especificados a seguir:

1. Coletar e armazenar séries históricas de preços de ações de empresas listadas na B3;
2. Pré-processar séries históricas de preços de ações por meio de indicadores técnicos;
3. Treinar e avaliar um modelo baseado em Rede Neural Recorrente para predizer preços de fechamento futuro de ações.

1.6 Visão Geral do Documento

O vigente [Capítulo 1](#) apresentou o contexto e as motivações deste trabalho. Ainda, o capítulo caracterizou o problema no qual se insere o trabalho, bem como definiu o objetivo e especificou os requisitos necessários para o desenvolvimento do modelo proposto de RNR.

Já o [Capítulo 2](#) elucida conceitos-chave para a realização do trabalho como, por exemplo, a bolsa de valores, assim como a Análise Técnica e o Aprendizado de Máquina. Ademais, o [Capítulo 3](#) apresenta os trabalhos correlatos, descrevendo os métodos escolhidos, assim como os resultados obtidos para a bolsa de valores escolhida pelo estudo.

Na sequência, o [Capítulo 4](#) descreve a metodologia utilizada para se atingir os objetivos do trabalho. Assim, esse capítulo explica como foram coletados, armazenados e pré-processados os dados para, por fim, treinar e avaliar o modelo proposto de RNR. Posteriormente, o [Capítulo 5](#) apresenta os resultados obtidos pelo trabalho por meio de gráficos e discussões a cerca destes resultados.

Além disso, o [Capítulo 6](#) encerra o trabalho apresentando as considerações gerais, bem como as contribuições obtidas em detrimento das limitações e dificuldades enfrentadas, além de descrever as considerações finais e esboçar alguns possíveis trabalhos futuros. Logo após o [Capítulo 6](#), são listadas as referências utilizadas para a dissertação dos capítulos supracitados.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo diz respeito à fundamentação teórica, o qual atenta-se em definir conceitos-chave para o trabalho. Assim, a seção [Seção 2.1](#) define o que é a bolsa de valores, assim como a [Seção 2.2](#) e [Seção 2.3](#), respectivamente, elucidam como a Análise Técnica e a Análise Fundamentalista se inserem no contexto da bolsa de valores.

Ademais, a [Seção 2.4](#) explica a técnica de KDD, utilizada em tarefas como seleção, pré-processamento, mineração e avaliação de grandes volumes de dados. Já a [Seção 2.5](#) detalha a importância da aplicação da normalização nos dados. Por fim, a [Seção 2.6](#) introduz o leitor ao Aprendizado de Máquina, conceito-chave para o entendimento do modelo de predição proposto.

2.1 Bolsa de Valores

Uma bolsa de valores representa um mercado onde são efetuadas transações de compra e venda de valores mobiliários. Em vista disso, define-se como mercado o local onde são realizadas as transações, as pessoas que as realizam e o conjunto destas transações ([GOMES, 1997](#)).

Diversos são os valores mobiliários, também conhecidos como ativos, que podem ser negociados nesse ambiente. As subseções a seguir apresentam alguns desses ativos.

2.1.1 Ações

Uma ação representa a parte de uma empresa, podendo esta pertencer a empreendedores, sócios ou ser negociada no mercado acionário para ser comprada ou vendida por investidores. Assim, quem investe na compra de ações de uma empresa também se torna um(a) acionista, o(a) qual têm o objetivo de compartilhar os ganhos que a empresa obtém. Com isso, as empresas distribuem dividendos, que representam parte de seus resultados, e que são pagos proporcionalmente à quantidade de ações que o(a) acionista adquiriu ([INFOMONEY, 2021b](#)).

2.1.2 Opções

Opções são contratos que dão ao seu titular o direito de comprar ou vender um determinado ativo por um valor estipulado em uma data específica do futuro. Existem opções sobre diversos tipos de ativos, dentre eles as ações, no que fica conhecido como opções de ações ([INFOMONEY, 2021d](#)).

2.1.3 ETFs

O *Exchange Traded Funds* (ETF) é um fundo de investimento, porém com duas principais diferenças entre os fundos de investimento tradicionais. A primeira delas é que ETFs

sempre estão atrelados à índices, e.g., o ETF BOVA11 que é atrelado ao Índice da Bolsa de Valores de São Paulo (IBOVESPA). Logo, o papel do gestor, neste contexto, é utilizar os recursos dos investidores para comprar, nas mesmas proporções, ações das empresas que compõem o índice que o ETF está atrelado. Por fim, a segunda diferença é que ETFs podem ser negociados na bolsa de valores como se fossem ações ([INFOMONEY, 2021e](#)).

2.1.4 Fundos Imobiliários

Um fundo imobiliário representa um conjunto de investidores que aplicam seus recursos em conjunto no mercado imobiliário. O comum nesses fundos é que o dinheiro seja aplicado na construção ou na aquisição de imóveis que posteriormente sejam locados ou arrendados ([INFOMONEY, 2021c](#)).

2.1.5 Principais Bolsas de Valores

A seguir, [KEY et al. \(2021\)](#) listam algumas das principais bolsas de valores no mundo, com suas respectivas capitalizações em Setembro de 2021:

- a) *New York Stock Exchange* (NYSE), oriunda dos Estados Unidos, capitalizou cerca de 28,4 trilhões de dólares;
- b) *National Association of Securities Dealers Automated Quotations System* (NASDAQ), também estadunidense, capitalizou cerca de 22,33 trilhões de dólares;
- c) *Shanghai Stock Exchange*, oriunda da China, capitalizou cerca de 7,81 trilhões de dólares;
- d) *Indian National Stock Exchange* (INSE), oriunda da Índia, capitalizou cerca de 3,48 trilhões de dólares;
- e) Brasil, Bolsa, Balcão (B3), oriunda do Brasil, capitalizou cerca de 1,18 trilhão de dólares.

2.2 Análise Técnica

Ao longo dos anos, a Análise Técnica (AT) desenvolveu-se na tentativa de decifrar o mercado acionário, única e exclusivamente, por meio dele próprio. Assim, grafistas (pessoas que utilizam da AT) analisam séries históricas de ações (comumente na forma de gráficos) para, então, prever possíveis tendências, de subida ou descida, do mercado acionário ([EDWARDS; MAGEE; BASSETTI, 2018](#)).

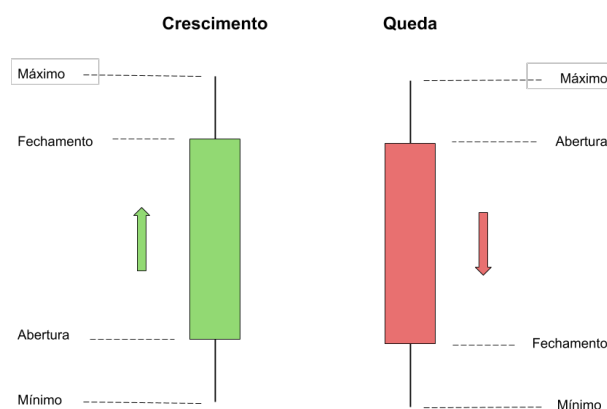
2.2.1 Preço

A AT baseia-se, quase que fundamentalmente, na análise do preço e do volume de ações. Contudo, antes de um aprofundamento em tal análise, é necessário definir algumas palavras-chave, ([ACHELIS, 2000](#)):

- Preço de abertura: é o preço da primeira negociação de uma ação em um determinado período de tempo;
- Preço máximo: é o maior preço de uma ação dado um determinado período de tempo;
- Preço mínimo: é o menor preço de uma ação dado um determinado período de tempo;
- Preço de fechamento: é o último preço de negociação de uma ação dado um determinado período de tempo. Além disso, a relação entre o preço de fechamento e o de abertura é considerada importante para a maioria dos grafistas;
- Volume: é o número de ações que foram negociadas em um determinado período de tempo. De mesma forma, o relacionamento entre o preço e o volume pode ser importante no contexto da AT.

Existem diferentes formas de representar os preços supracitados em um gráfico. Uma delas é por meio de um gráfico de *candlestick*, vide [Figura 1](#), uma versão japonesa do gráfico de barras, o qual tem a capacidade de representar os preços de abertura, fechamento, máximo e mínimo em um dado período de tempo ([MURPHY, 1999](#)). Assim, quando há um aumento no preço da ação, o *candlestick* geralmente é representado pela cor verde; queda, vermelho.

Figura 1 – Representação Gráfica da Variação dos Preços de uma Ação ao Longo do Tempo



Fonte: Autoria Própria

Além disso, *candlesticks* podem ser computados para qualquer período de tempo. *Candlesticks* podem representar o mercado diariamente, ou até em períodos de tempo menores que um dia, chamados de intra-diários. Ainda, caso deseje-se analisar uma ação ao longo dos anos, *candlesticks* semanais ou mensais podem ser utilizados. Por fim, para visualizar as variações de preço ao longo do tempo, basta utilizar uma série temporal de *candlesticks*, vide [Figura 2](#).

Ainda, na [Figura 2](#) percebe-se que *candlesticks* diários foram utilizados para representar os preços de abertura, fechamento, máximo e mínimo, em que *candlesticks* verdes representam um aumento nos preços; vermelho, queda. Desse modo, a [Figura 2](#) é capaz de representar um período temporal de quase dois anos. Outrossim, seria possível utilizar *candlesticks* para representar meses ao invés de dias, gerando assim um gráfico mais conciso.

Figura 2 – Representação de uma Série Temporal de *Candlesticks*

Fonte: ([DAMIBA et al., 2021](#))

2.2.2 Teoria de Dow

Considerado o pai da AT moderna, Charles Dow foi o primeiro a criar um índice que medisse o movimento geral dos preços de ações nos Estados Unidos. Charles despendeu cinco anos escrevendo para o *Wall Street Journal* sobre seus aprendizados no mercado acionário até que falecesse em 1902. Após sua morte, William Peter Hamilton fora o responsável por agrupar as teorias de Dow em seu livro *The Stock Market Barometer* de 1922. Entretanto, foi apenas em 1932 que Robert Rhea refinou e cunhou o que ficaria conhecido como Teoria de Dow. Embasando-se no trabalho de Hamilton, Rhea apresentou três principais hipóteses, [Kirkpatrick e Dahlquist \(2010\)](#):

1. A tendência primária é inviolável;
2. Os preços refletem tudo;
3. A Teoria de Dow não é infalível.

Segundo os princípios de Dow, existem três tendências. A primeira tendência representa as oscilações de preço que podem durar anos e resultam em um crescimento, ou queda, geral maior que 20% nos preços. Tendências primárias subdividem-se em tendências secundárias – com duração de semanas a meses – que oscilam de maneira a reagir e corrigir a tendência primária. Por fim, tendências secundárias são compostas por tendências terciárias, as quais representam oscilações com durações de dias a semanas e que são irrelevantes segundo a Teoria de Dow ([EDWARDS; MAGEE; BASSETTI, 2018](#)).

A segunda hipótese defendida por Rhea diz respeito à premissa de que os preços descritos pelo mercado já representam qualquer possível fator externo que afete, de modo geral, a oferta e a demanda do mercado. Ainda, mesmo que o mercado não consiga antecipar eventos como catástrofes ambientais, estes são rapidamente refletidos nos preços das ações ([MURPHY, 1999](#)).

Embora a Teoria de Dow seja a base para a AT moderna, os princípios de Dow não são ilibados de críticas. Uma delas está no atraso em perceber as mudanças nas tendências primárias, a ponto que estas são apenas percebidas após terem ocorrido. Ademais, a interpretação das oscilações dos preços pode ser difícil de classificar em uma tendência específica (KIRKPATRICK; DAHLQUIST, 2010).

2.2.3 Indicadores Técnicos

Como ressaltam Colby e Meyers (2003), não é necessário que um indivíduo despenda décadas de análise para, enfim, observar padrões no mercado. Nesse contexto, surgem os indicadores técnicos, os quais podem ser utilizados para detectar tendências (subida, descida ou de lateralização) no mercado, além de trazerem sinais de compra ou venda desprovidos de sentimentos humanos e, portanto, maximizando as chances de sucesso em tomadas de decisão.

Um indicador técnico, segundo Achelis (2000), é definido como um cálculo matemático aplicado ao preço ou volume de uma ação, cujo o resultado pode ser usado para prever futuras mudanças no preço da ação. A seguir, são apresentados alguns indicadores técnicos comumente utilizados por grafistas:

- Média móvel: preço médio de uma ação em um dado instante t , considerando-se um intervalo de tempo n . Investidores normalmente compram quando o preço da ação sobe acima de sua média móvel e vendem quando cai abaixo. Contudo, a desvantagem desse indicador técnico está na compra e venda tardia, i.e., se a tendência observada pela média móvel não durar o suficiente, haverá prejuízo (ACHELIS, 2000);
- Indicador de tendência: é utilizado para rastrear as tendências dos preços das ações. A estratégia deste tipo de indicador está na proposição de que eventos de cunho político e econômico geralmente refletem-se em tendências no mercado acionário, e não abruptamente em preços específicos (HU et al., 2015). Outrossim, medias móveis podem ser utilizadas como indicadores de tendência, tal qual a Média Móvel Exponencial (MME) que pondera-se pelos preços mais recentes (PEACHAVANISH, 2016);
- Indicador de momento: momento é definido no mercado acionário como a taxa de variação experienciada pelo preço de uma ação em relação a um período de tempo. Dentre os indicadores de momento tem-se o Índice de Força Relativa (IFR), o qual é capaz de indicar se o preço de uma ação está supervalorizado ou subvalorizado (AGRAWAL; JINDAL; PILLAI, 2010);
- Indicador de volatilidade: este tipo de indicador é geralmente empregado para visualizar a amplitude dos preços, além de ser utilizado para avaliar o risco, bem como o nível de suporte e resistência das ações. Exemplos de indicadores de volatilidade incluem a Banda de Bollinger e a Faixa Verdadeira Média (FVM) (HU et al., 2015);
- Indicador de volume: este tipo de indicador funda-se na hipótese de que a movimentação de preços do mercado acionário é determinada pelo entusiasmo de compradores e vendedores. Comumente, o volume de transações tende a crescer (diminuir) durante tendências de

alta (baixa) nos preços. Exemplos de indicadores de volume incluem: volume; proporção de volume; e *On-Balance Volume* (OBV) (HU et al., 2015).

2.2.4 Média Móvel Exponencial

A Média Móvel Exponencial é um tipo de média móvel a qual atribui maior peso para preços recentes, enquanto que média móvel simples atribui o mesmo peso para diferentes dias (CHEN; POTTERS, 2021). Isto posto, a Equação (1) demonstra o cálculo da MME, em que n representa os dias e s o fator de suavização.

$$MME_{atual} = Preço_{atual} \left(\frac{s}{1+n} \right) + MME_{anterior} \left(1 - \left(\frac{s}{1+n} \right) \right) \quad (1)$$

2.2.5 Moving Average Convergence Divergence

O *Moving Average Convergence Divergence* (MACD) é um indicador técnico de tendência, o qual utiliza de MMEs para explorar informações nos preços das ações do mercado acionário. Assim, por meio da subtração de uma MME de vinte e seis dias de uma MME de doze dias, é calculada a linha MACD. Logo, aplicando-se uma MME de nove dias na linha MACD, tem-se o sinal de linha. Por fim, e normalmente, a linha MACD cruzando acima do sinal de linha, representa indicativo de compra de ações. Já o contrário, indica a venda (FERNANDO; POTTERS; MUNICHIELLO, 2021).

2.3 Análise Fundamentalista

A Análise Fundamentalista (AF) – em contrapartida à AT – atenta-se em analisar ações do mercado acionário sob o ponto de vista da empresa, i.e., se esta apresenta boas perspectivas para o futuro. Essa perspectiva é gerada a partir do estudo dos fundamentos da empresa em questão, os quais encontram-se nos dados divulgados nos balanços trimestrais da empresa. Além disso, fundamentalistas costumam monitorar outros dados relativos ao mercado acionário como, por exemplo, cotações, situação econômica do país, nível de consumo, concessões e licitações ganhas pela empresa, bem como a análise do setor de atuação da empresa (DEBASTIANI; RUSSO, 2017).

2.3.1 Valor Intrínseco

Entre os objetivos da AF está o cálculo do valor intrínseco. O valor intrínseco, no que lhe concerne, representa o valor real de uma ação, isto é, o valor que ela deveria ter. Assim, diz-se que uma ação é supervalorizada (subvalorizada) quando seu preço negociado no mercado acionário é superior (inferior) ao seu valor intrínseco. Isto posto, é sugestivo que investidores vendam ações supervalorizadas (lucro) e comprem ações subvalorizadas (investimento), pois estas tendem a retornar para seu valor intrínseco (GOTTWALD, 2011).

2.3.2 Hipótese do Mercado Eficiente

A Hipótese do Mercado Eficiente (HME), diz respeito a um mercado o qual reflete totalmente nos preços todas as informações disponíveis que competem ao mercado acionário. Em outras palavras, nesse cenário, diz-se que o mercado é eficiente. Logo, para testar essa hipótese, a literatura propõem três diferentes formas, (FAMA, 1970):

- Forma forte: refere-se a crença de que nenhum indivíduo tem acesso exclusivo à informações que influenciem o mercado. Contudo, evidências de que corporativos têm acesso monopolístico à informações representam exceções à hipótese de um mercado eficiente;
- Forma semi-forte: confia que todas as informações disponíveis – que concernem ao mercado acionário – são totalmente refletidas nos preços das ações. Evidências que corroboram essa hipótese são observadas, por exemplo, quando empresas anunciam seus lucros;
- Forma fraca: presume que preços antigos não podem ser utilizados para prever preços futuros (LOC, 2006). Com isso, apesar de existirem indícios de correlações dia a dia entre preços, períodos superiores à estes endossam a existência de um jogo justo.

De acordo com Poshakwale (1996), há um jogo justo quando se recebe, em média, o lucro esperado sem que sejam obtidos ganhos anormais. Nesse cenário, tem-se, então, um mercado eficiente ao que diz respeito às informações utilizadas pelos investidores. Contudo, a HME é indissociável da Teoria do Passeio Aleatório (TPA), a qual sustenta a ideia de que preços sucessivos são independentes e, portanto, não pode-se prever o preço de amanhã embasando-se no preço de hoje.

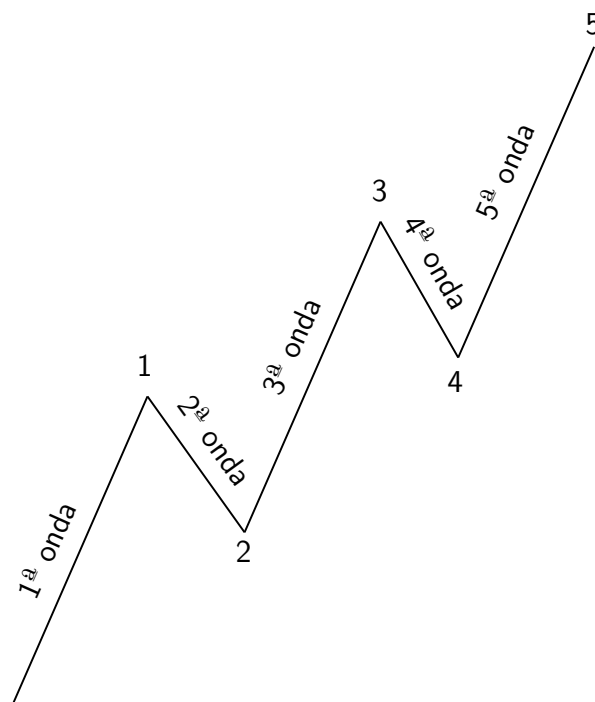
Em outras palavras, em um mercado eficiente – que reflete rapidamente nos preços todas as informações disponíveis – não há margem para tentativas de previsão de preços futuros. Dessa forma, as chances de se antecipar a direção do mercado são imprevisíveis, tal qual um passeio aleatório em que futuros passos ou direções não podem ser determinadas por ações passadas (MALKIEL, 1999).

2.3.3 Ondas de Elliott

Em 1930, ao analisar os preços do mercado acionário, Ralph Nelson Elliott foi capaz de descobrir padrões (ondas), os quais se repetiam em forma, porém, não necessariamente em tempo ou amplitude. Assim, após serem isoladas e nomeadas em cinco ondas, também denominadas de forma base, vide Figura 3, Elliott estudou como estas interagiam para formar – progressiva e estruturadamente – versões maiores delas mesmas. Esse fenômeno foi então intitulado por Elliott de O Princípio da Onda (FROST; PRECHTER, 1995).

Nessa forma, três das cinco ondas seguem a direção do movimento (subida ou descida), sendo estas: 1^a, 3^a e 5^a ondas. Em contrapartida, a 2^a e 4^a ondas seguem a tendência contrária, contrapondo-se à direção de tendência ditada pela 1^a, 3^a e 5^a ondas. Ainda segundo Frost e Prechter (1995), não é muito difícil analisar o mercado acionário sobre um escopo geral e

Figura 3 – Representação das Cinco Ondas de Elliot



Fonte: Adaptado de (FROST; PRECHTER, 1995)

observar tendências seguindo a forma base, em que as demais formas (padrões) são absorvidas pela forma base.

2.4 KDD

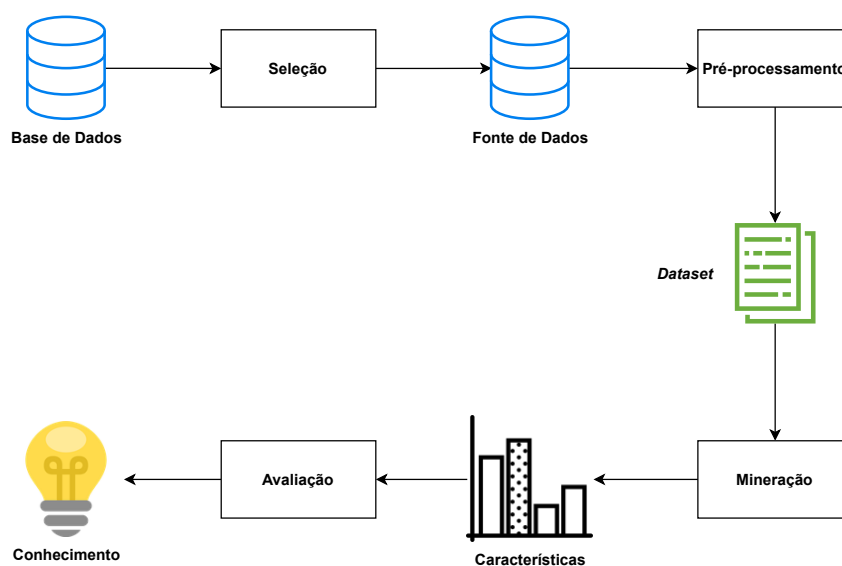
Em vista do crescente volume de dados gerados todos os dias, vê-se a necessidade da automatização de análises aplicadas a esse expressivo volume. Dessa necessidade, surge o *Knowledge Discovery in Databases* (KDD), o qual agrega múltiplos campos de pesquisa como Aprendizado de Máquina, Reconhecimento de Padrões, Banco de Dados e Visualização de Dados com o objetivo principal de extrair conhecimento de um grande volume de dados não processados (FAYYAD; PIATETSKY-SHAPIO; SMYTH, 1996).

Ademais, Fayyad, Piatetsky-Shapiro e Smyth (1996) propuseram nove etapas para o processo de KDD, as quais foram posteriormente resumidas em quatro grandes etapas por Williams e Huang (1996), com o objetivo de simplificar a identificação e execução do processo de KDD. As quatro grandes etapas do processo de KDD são ilustradas pela Figura 4.

A primeira etapa descrita por Williams e Huang (1996), é a seleção. Nessa etapa, os dados mais apropriados ao contexto do problema são selecionados de uma base de dados, formando uma fonte de dados a qual servirá de base para o resto dos processos de KDD. Em seguida, na segunda etapa, de pré-processamento, a fonte de dados será limpa e processada da forma mais adequada, gerando um *dataset*.

Já na terceira etapa, a de mineração, serão aplicados algoritmos de mineração de

Figura 4 – Quatro Grandes Etapas do Processo de KDD



Fonte: Adaptado de (WILLIAMS; HUANG, 1996)

dados no *dataset* gerado, a fim de se extrair informações (características) úteis, por exemplo, para algoritmos de Aprendizado de Máquina. Por fim, a quarta e última etapa, de avaliação, encarregar-se-á de avaliar as características obtidas da etapa de mineração com o intuito de descobrir quais destas foram úteis, ou não, para a geração de conhecimento (WILLIAMS; HUANG, 1996).

2.5 Normalização

A normalização dos dados é um processo importante para a obtenção de bons resultados em se tratando de modelos de Rede Neural. Entenda-se normalização como a aplicação de uma conversão linear na escala nos dados, em que os valores absolutos dos dados são representados em uma nova escala, porém relativamente com as mesmas variações (SOLA; SEVILLA, 1997).

Predizer séries temporais representa uma tarefa desafiadora em virtude da natureza não estacionária e multimodal dos dados. Diz-se que uma série temporal é não estacionária quando suas propriedades estatísticas, e.g., média, variância etc, mudam com o tempo. Já uma série temporal multimodal é aquela formada, por exemplo, por diferentes ações do mercado acionário que apresentem distinções significativas no preço, tendências etc. Em vista disso, Passalis et al. (2019), Mader et al. (2006) propuseram a criação da *Deep Adaptive Input Normalization* (DAIN).

Contudo, apesar dos bons resultados da DAIN para com redes neurais (PASSALIS et al., 2019), esta ainda não encontra-se oficializada em bibliotecas como TensorFlow¹, proposta por Abadi et al. (2015), para o desenvolvimento de modelos de rede neural. Isso torna a aderência

¹Biblioteca criada para a linguagem Python para o desenvolvimento de modelos de Aprendizado de Máquina.

de métodos do estado da arte mais difícil, e apesar da comunidade científica apresentar códigos extraoficiais, a citar Briola (2020), estes ainda não suprem todos os cenários.

Em vista disso, a literatura apresenta métodos de normalização clássicos, que embora não sejam ideais para series temporais, são melhores do que não normalizar os dados, como também demonstram Passalis et al. (2019). Assim, o *z-score*, ou *Standard Scaler*, é um método de normalização clássico e de fácil utilização, o qual é acessível, por exemplo, por meio de bibliotecas como SciPy², proposta por Virtanen et al. (2020), e scikit-learn³, proposta por Pedregosa et al. (2011).

Por fim, o método de normalização *z-score* é obtido por meio da Equação (2), em que x representa um conjunto de valores, \bar{x} a média deste conjunto e s_x o desvio padrão:

$$z = \frac{x - \bar{x}}{s_x} \quad (2)$$

2.6 Aprendizado de Máquina

Desde sua gênese, o Aprendizado de Máquina (AM) vem sendo cada vez mais empregado na automatização de tarefas humanas. Samuel (1959), no entanto, visionou que programar computadores a aprender por meio de experiências deveria eliminar a necessidade de programá-los explicitamente. Já Géron (2017) afirma que esse aprendizado, na verdade, provem dos dados utilizados para o treinamento de computadores a realizar tarefas. Com isso, o Aprendizado de Máquina permite a realização de tarefas que seriam difíceis de resolver caso explicitamente programados por seres humanos (GOODFELLOW et al., 2016).

O AM é capaz de realizar muitas tarefas distintas. Dentre elas, as mais comuns, conforme Goodfellow et al. (2016), compreendem: classificar dados dentre um grupo de categorias (classificação); prever valores numéricos, dada uma determinada entrada (regressão); traduzir textos de uma língua para outra; e detectar anomalias em dados.

Ademais, diferentes são os tipos de AM. Assim, caso os dados de treinamento possuam a resposta desejada, este é chamado de Aprendizado de Máquina Supervisionado. Quando não há a resposta desejada nos dados de treinamento, trata-se de um Aprendizado de Máquina Não-Supervisionado. Ainda, se apenas parte do conjunto de treinamento possui a resposta desejada, intitula-se Aprendizado de Máquina Semi-Supervisionado. Além desses, há o Aprendizado por Reforço, o qual aprende por meio de tentativa e erro, sendo a máquina recompensada ou penalizada por suas ações (GÉRON, 2017).

Atualmente, existem diversos modelos de AM disponíveis para fácil utilização. Em Python, por exemplo, bibliotecas como scikit-learn (PEDREGOSA et al., 2011), permitem o treinamento e teste desses modelos em pouquíssimas linhas de código. Alguns dos modelos contemplados por essa biblioteca incluem: *Decision Trees*; *Naïve Bayes*; *K Nearest Neighbors*; *Support Vector Machine*; e Perceptron de Múltiplas Camadas.

²Ecosistema criado para a linguagem Python utilizada para resolução de problemas matemáticos.

³Biblioteca criada para a linguagem Python para análise preditiva dos dados.

2.6.1 Séries Temporais

Como apontam Mahalakshmi, Sridevi e Rajaram (2016), diferentes são os modelos de predição utilizados no domínio de séries temporais. Quanto a modelos de predição tradicionais, exemplos incluem: regressão, regressão múltipla e *exponential smoothing*. Já no que tange a técnicas estocásticas, modelos de predição como *Support Vector Machine* (SVM), *Least Squares Support Vector Machine* (LSSVM) e *Support Vector Regression* (SVR) destacam-se. Por fim, modelos de predição baseados em Redes Neurais Artificiais e Lógica Fuzzy também são utilizados no contexto de séries temporais.

2.6.2 Redes Neurais

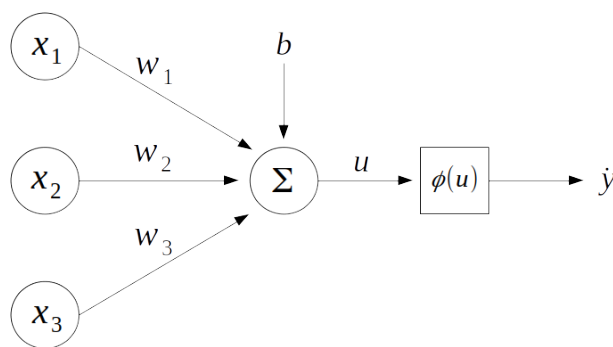
Apesar de sua atual popularidade, Redes Neurais (RNs) foram concebidas em meados da década de 40. Contudo, tais primitivas RNs não aprendiam com os dados e, portanto, estavam além da supradita definição de Aprendizado de Máquina. Fora somente em 1958 que pôde-se unir RNs com Aprendizado de Máquina Supervisionado, no que ficaria conhecido como Perceptron (SCHMIDHUBER, 2015).

Considerando cada entrada x_i associada a um peso w_i , o Perceptron é capaz de calcular a soma ponderada das entradas (GÉRON, 2017). Adicionalmente, aplica-se uma entrada b , conhecida como *bias* (HAYKIN, 2007). Essa operação pode ser observada pela Equação (3).

$$u = \sum_{i=1}^m w_i x_i + b \quad (3)$$

Na sequência, à saída u , é aplicada uma função de ativação $\phi(u)$, a citar algumas populares: função degrau, função logística; função tangente hiperbólica; e função *Rectified Linear Units* (ReLU), em português, Unidades Lineares Retificadas (DU; SWAMY, 2013). Dessa forma, o Perceptron tem o objetivo de atribuir ao conjunto de entradas x , um rótulo y_1 ou y_2 (HAYKIN, 2007). Esse funcionamento é ilustrado pela Figura 5.

Figura 5 – Representação do Perceptron



Fonte: Autoria Própria

Para que o Perceptron aprenda como associar suas entradas x às saídas y esperadas, é necessário que se ajuste os pesos w . Isso é alcançável graças à estrutura livre do Perceptron,

pois permite que esses pesos w sejam atualizados por meio de um algoritmo de aprendizado (DUDA; HART; STORK, 2012).

Mais precisamente, conforme Géron (2017), para cada predição errônea produzida pelo Perceptron, os pesos w são ajustados almejando corrigi-los para futuras predições. Já em casos de acerto, os pesos w são mantidos. Desse modo, a Equação (4) representa a referida correção iterativa dos pesos w .

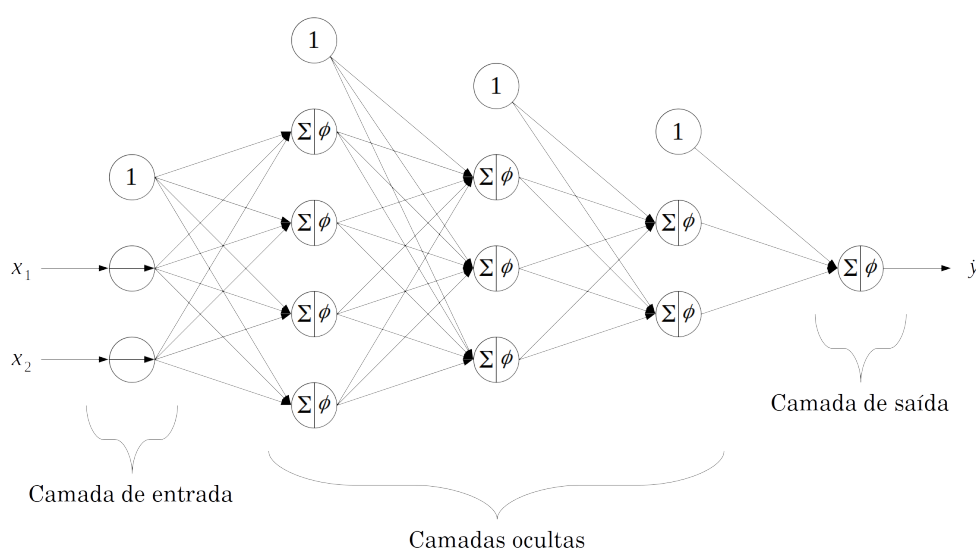
$$w'_i = w_i + \eta(y - \hat{y})x_i \quad (4)$$

Na Equação (4), w'_i representa o próximo valor que será associado ao peso w_i . Esse ajuste, porém, é regulado pela diferença entre o valor esperado y e o predito \hat{y} pelo Perceptron, multiplicado pela entrada x_i e por uma constante η , conhecida como taxa de aprendizado.

Todavia, o algoritmo de aprendizagem descrito pela Equação (4) exige que as classes do problema sejam linearmente separáveis. Se isso não for satisfeito, o algoritmo não irá convergir (DUDA; HART; STORK, 2012). Contudo, esta limitação do Perceptron pode ser superada agrupando-se Perceptrons em camadas, criando-se assim um Perceptron de Múltiplas Camadas (PMC), ou também chamada, Rede Neural Artificial (RNA) (GÉRON, 2017).

Um Perceptron de Múltiplas Camadas (PMC) tradicionalmente é formado por três tipos de camadas. A primeira delas, a camada de entrada, é populada pelo vetor x de características do problema. Em seguida, o PMC possui uma ou mais camadas ocultas, as quais contém neurônios (Perceptrons). Por fim, a camada de saída, responsabiliza-se por sintetizar os cálculos propagados pelas camadas anteriores e apresentar a predição do PMC sobre a dada instância (HAYKIN, 2007). Esta topologia é representada pela Figura 6.

Figura 6 – Topologia de um PMC



Fonte: Autoria Própria

Na Figura 6 o bias b é representado por um neurônio de valor constante 1. Esse neurônio, ao se conectar com os demais neurônios, gera um peso w encarregado de aplicar um

bias ao neurônio conectado. Ademais, para simplificar a ilustração, a função de ativação foi incluída dentro da representação, em formato circular, de um neurônio.

Não obstante, por muitos anos fora difícil encontrar um algoritmo que pudesse treinar um PMC, até que em 1986 o algoritmo de *Backpropagation* (BP), em português, retropropagação, fosse introduzido (GÉRON, 2017). O referido algoritmo, por sua vez, utiliza uma busca gradiente para minimizar o erro gerado pelas previsões do PMC. Esse erro é mensurado por meio da métrica de Erro Quadrático Médio entre o valor \hat{y} predito pelo PMC e o valor y esperado (DU; SWAMY, 2013).

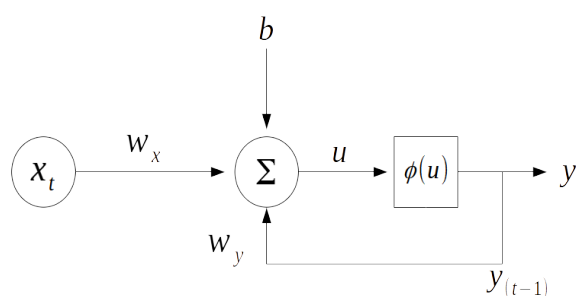
Em seguida, o erro calculado é então propagado retroativamente, partindo da camada de saída até a camada de entrada, isto é, ajustando assim cada peso w do PMC no intuito de gerar menor erro nas próximas previsões (HAYKIN, 2007).

2.6.3 Redes Neurais Recorrentes

Sem embargo, o PMC apresenta uma topologia estática, o que torna-o incapaz de lidar com dados temporais, e.i., séries temporais (DU; SWAMY, 2013). Para analisar o tempo é necessário empregar um tipo específico de RNA: a Rede Neural Recorrente (RNR), tipo de rede neural que utiliza de exemplos passados para realizar previsões futuras. Devido a essa propriedade, RNRs podem ser utilizadas em tarefas como: tradução automática de texto; análise de sentimentos; direção de carros autônomos; e também, no mercado acionário (GÉRON, 2017).

Apesar disso, uma RNR é bastante parecida com um PMC, salvo pela retroalimentação de pesos w . Imagine um neurônio o qual possui apenas uma entrada x_t e uma saída y_t em um determinado intervalo de tempo t . Agora imagine que esse neurônio também pudesse ser alimentado com a saída y_{t-1} do instante anterior (GÉRON, 2017). Dessa forma, tem-se um neurônio recorrente, vide Figura 7, que se conecta não só à entrada x_t por pesos w_x , mas também à saída defasada no tempo y_{t-1} por pesos w_y , além de um bias b .

Figura 7 – Representação de um Neurônio Recorrente



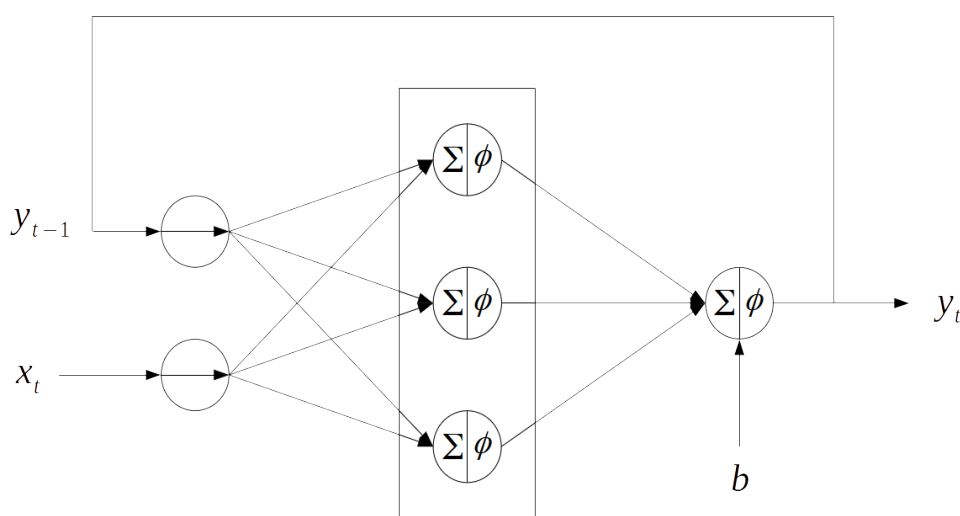
Fonte: Autoria Própria

Outra diferença entre uma RNR e um PMC está no compartilhamento de pesos. Enquanto um PMC necessita de um peso w_i para cada entrada x_i , em que cada entrada deve

ter um tamanho fixo, uma RNR suporta entradas de tamanhos variados. Isso se dá pois uma RNR não está interessada na sequência (série temporal) como um todo, e sim na transição realizada a cada instante t (GOODFELLOW et al., 2016).

Outrossim, é possível empilhar neurônios recorrentes em uma camada. Nesse cenário, a cada intervalo de tempo t , todos os neurônios recorrentes dessa camada são, então, alimentados pela entrada $x_{(t)}$ e pela saída do intervalo de tempo anterior $y_{(t-1)}$ (GÉRON, 2017). Essa explicação é ilustrada pela Figura 8 que novamente inclui a função de ativação na representação do neurônio para fins didáticos.

Figura 8 – Neurônios Recorrentes em Camada



Fonte: Autoria Própria

Para calcular o valor $y_{(t)}$ da Figura 8 são necessárias algumas generalizações. Resumindo-se todos os vetores w_x e w_y dos neurônios recorrentes de uma camada, tem-se duas matrizes de pesos W_x e W_y . Considere, ainda, b como bias e ϕ como função de ativação, chega-se à $y_{(t)}$ por meio da Equação (5) (GÉRON, 2017).

$$y_{(t)} = \phi(W_x^T x_{(t)} + W_y^T y_{(t-1)} + b) \quad (5)$$

Uma vez que a saída de um neurônio recorrente, em um instante t , representa o cômputo de todas as entradas desse neurônio no instante passado, pode-se dizer que existe uma espécie de memória. Uma célula de memória, ou apenas célula, portanto, preserva o estado do sistema ao longo do tempo, sendo formada por um ou mais neurônios recorrentes. Essa célula (Figura 7) é conhecida como célula básica perante células mais complexas e poderosas como a *Long Short-Term Memory* (LSTM) e a *Gated Recurrent Unit* (GRU) (GÉRON, 2017).

Em síntese, uma LSTM é capaz de reconhecer uma entrada importante, armazená-la e aprender a preservá-la o quanto for necessário e, por fim, descartá-la quando preciso. Dessa forma, a LSTM tem mostrado bons resultados no reconhecimento de padrões importantes ao

longo de séries temporais. Já uma GRU é uma versão simplificada de uma LSTM, que também tem apresentado bons resultados. Diante disso, essas duas células são apresentadas como uma das principais razões do sucesso de RNRs nos últimos anos (GÉRON, 2017).

Por fim, RNRs também podem ser treinadas pelo algoritmo de BP, o qual é simples de ser implementado, contudo enfrenta dificuldades para escapar de mínimos locais. Ademais, lidar com dados temporais exige algoritmos de aprendizagem mais complexos. Dessa forma, em 1990 é proposto o algoritmo de *Backpropagation Through Time* (BPTT), em português, Retro Propagação ao Longo do Tempo (MEDSKER; JAIN, 1999).

2.6.4 Aprendizado Profundo

Abordagens clássicas de AM são capazes de resolver uma grande variedade de problemas. Ainda assim, não foram bem sucedidas em resolver problemas mais complexos como, por exemplo, o reconhecimento de fala ou o reconhecimento de objetos em uma imagem. Diante dessas limitações, o desenvolvimento do Aprendizado Profundo (AP) surge em detrimento do fracasso das abordagens tradicionais de AM (GOODFELLOW et al., 2016).

Em virtude disso, Krizhevsky, Sutskever e Hinton (2017) propuseram um modelo de Rede Neural Profunda (RNP) composta por oito camadas, a qual obteve bons resultados com a classificação de imagens. Os autores também ressaltam que se apenas uma dessas camadas fosse removida, os resultados seriam 2% piores. Dessa forma, como apontam LeCun, Bengio e Hinton (2015), o AP enquadra-se como um ramo do AM no qual os modelos apresentam múltiplas camadas ocultas. Exemplos de modelos de AP incluem Redes Neurais Convolucionais Profundas e Redes Neurais Recorrentes Profundas.

Entretanto, toda essa robustez e complexidade, oriundas do maior número de camadas, trazem algumas ressalvas: treinamento significativamente mais lento; dificuldade em se treinar camadas ocultas profundas; e risco de *overfitting*, isto é, quando um modelo é mais robusto do que o necessário (GÉRON, 2017; HAWKINS, 2004).

2.6.5 Treinamento e validação de Redes Neurais

Para treinar modelos de RNA simples à complexos, como as RNPs, é necessário dividir o conjunto de dados relativo ao problema em três subconjuntos: conjunto de treinamento, o qual representa os dados utilizados para se treinar o modelo; conjunto de validação, utilizado para parametrizar (calibrar) o modelo, almejando melhores resultados; e conjunto de teste, o qual avaliará o quão bem generalizado está o modelo diante de dados nunca antes vistos pelo modelo (GÉRON, 2017).

Além disso, é necessário mensurar o erro do modelo durante o treinamento, o qual é representado pela função custo $J(\theta)$, em que θ são todos os pesos w do modelo de RNA (GÉRON, 2017).

Como aponta Ruder (2016), treinar um modelo de RNA significa, portanto, reduzir o erro da função custo $J(\theta)$, o que pode ser realizado por meio do algoritmo de *Gradient Descent*

(GD), em português, Descida Gradiente. Esse algoritmo, por sua vez, aponta para a direção oposta ao vetor gradiente $\nabla_{\theta} J(\theta)$ da função $J(\theta)$, indicando a direção de menor erro. Além disso, pode-se utilizar o próprio vetor gradiente como valor para atualizar θ , juntamente com uma taxa de aprendizagem η que regula a quantidade de passos tomados até que se atinja um mínimo local da função $J(\theta)$. O raciocínio descrito é representado pela [Equação \(6\)](#).

$$\theta' = \theta - \eta \cdot \nabla_{\theta} J(\theta) \quad (6)$$

Ainda de acordo com [Ruder \(2016\)](#), existem três variantes do algoritmo de GD, as quais diferem apenas na quantidade de instâncias (dados) do conjunto de treinamento utilizadas para calcular a função custo $J(\theta)$. Outrossim, a quantidade de instâncias representa um dilema entre acurácia e tempo de treinamento, i.e., uma é inversamente proporcional a outra. Dessa forma, as três variantes do algoritmo de GD são:

- A técnica clássica de *Batch Gradient Descent* (BGD), em português, Descida Gradiente em Lote, consiste em utilizar o conjunto de treinamento inteiro – também chamada de época – para realizar apenas uma atualização dos pesos θ . Apesar de precisa, essa técnica torna-se inaplicável em conjuntos de treinamento que superem a memória disponível, além de possuir um tempo de treinamento longo;
- Em contrapartida, o *Stochastic Gradient Descent* (SGD), em português, Descida Gradiente Estocástica, utiliza apenas uma instância – escolhida aleatoriamente – para atualizar os pesos θ . Com isso, além de rápida, essa técnica pode encontrar melhores mínimos locais devido a sua natureza instável. Contudo, essa mesma instabilidade dificulta a convergência da técnica, uma vez alcançado um mínimo local;
- Por fim, a variante *Mini-Batch Gradient Descent* (MBGD), em português, Descida Gradiente em Mini Lotes, representa uma técnica híbrida entre as duas já citadas. Logo, ao invés de utilizar todo o conjunto, ou apenas uma instância, a técnica em questão utiliza um subconjunto de instâncias, e.g., 32 instâncias, para atualizar os pesos θ .

2.6.6 Técnicas para Acelerar o Treinamento

Uma limitação presente em RNAs é a dissipação e a explosão de gradiente. A dissipação de gradiente faz com que seja difícil saber em qual direção os pesos devem ir para diminuir seu erro; já a explosão de gradiente torna o aprendizado instável ([GOODFELLOW et al., 2016](#)). Em outras palavras, esses fenômenos representam saturações – respectivamente, em intervalos pequenos e grandes – na atualização dos pesos à medida que o algoritmo de GD progride ([GÉRON, 2017](#)). Em vista disso, diversas técnicas foram propostas para mitigar a dissipação e explosão de gradiente nas RNAs, podendo estas serem aplicadas em conjunto ou separadamente.

Primeiramente, para mitigar esse problema, [Glorot e Bengio \(2010\)](#) propuseram que os pesos deveriam ser inicializados a partir de uma distribuição randômica uniforme, vide [Equação \(7\)](#), no que ficaria conhecido, posteriormente, como Inicialização de Xavier. Na [Equação \(7\)](#), W representa a matriz de pesos a ser inicializada, n_i representa o número de

conexões de entrada (*fan-in*) do neurônio e, enfim, n_{i+1} representa o número de conexões de saída (*fan-out*) do neurônio. Assim, ao aplicar a inicialização de Xavier nas funções de ativação sigmóide e tangente hiperbólica, os autores obtiveram bons resultados, se comparado com os métodos clássicos de inicialização.

$$W \sim U = \left[-\frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}, \frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}} \right] \quad (7)$$

Já He et al. (2015) propuseram um novo método de inicialização, porém que fosse aplicável nas funções ReLU e suas variantes. Essa, inicialização, posteriormente, ficou conhecida como Inicialização de He.

Contudo, apesar da inicialização de Xavier ou de He mitigarem a dissipação e explosão de gradiente no início do treinamento, nada impede a RNA de sofrer com a dissipação e explosão de gradiente durante o treinamento. Diante disso, a fim de mitigar esse problema, a literatura apresenta algumas técnicas como o *Batch Normalization* e o *Gradient Clipping* (GÉRON, 2017).

A técnica de *Batch Normalization*, proposta por Ioffe e Szegedy (2015), age aplicando dois parâmetros extras – γ e β – para cada função de ativação da RNA escolhida para ser normalizada. Dessa forma, ao longo do treinamento, esses dois novos parâmetros ajustam-se para melhor normalizarem os neurônios, fazendo a RNA convergir em um tempo menor. Já o *Gradient Clipping*, proposto por Pascanu, Mikolov e Bengio (2013), introduz apenas um parâmetro extra durante o treinamento, o qual é responsável por determinar um limite para o vetor gradiente e, neste caso, alterar sua escala.

Ademais, há a técnica de *Transfer Learning*, ou em português, Transferência de Aprendizado. Como o nome sugere, essa técnica consiste em aproveitar camadas de neurônios já treinadas de um modelo de RNA, de preferência que realize uma tarefa semelhante à que se deseja resolver, ao invés de treinar uma RNA desde o princípio (GÉRON, 2017).

Por fim, como frisa Géron (2017), demais técnicas para acelerar o treinamento consistem em substituir a abordagem clássica do algoritmo de GD por outros algoritmos de otimização. Esses são chamados de *Faster Optimizers* – em tradução literal para o português, otimizadores mais rápidos – dentre os quais destacam-se: *Momentum Optimization*; *Nesterov Accelerated Gradient*; *AdaGrad*; *Root Mean Square Propagation* (RMSProp); e *Adam Optimization*.

2.6.7 Técnicas para Evitar *Overfitting*

Uma RNA pode conter dezenas de milhares a milhões de parâmetros, e com tamanha capacidade de aprendizado, pode ajustar-se a uma grande variedade de dados complexos. No entanto, essa é uma faca de dois gumes, ao passo que essa grande flexibilidade torna uma RNA propícia a sofrer com o *overfitting* dos dados. Um *overfitting* dos dados de treinamento de uma RNA pode fazer com que esta não consiga generalizar-se bem, acarretando em maus resultados perante conjuntos de dados nunca antes vistos (GÉRON, 2017).

Durante o treinamento de uma RNA, é comum avaliar-se o erro desta perante o conjunto de treinamento; contudo, também faz-se uso do conjunto de validação. Esse conjunto não é utilizado no treinamento da RNA, nem para guiar seu treinamento, e sim para monitorar o erro da RNA ao longo do treinamento. Outrossim, ao longo do treinamento, comumente o erro mensurado pelo conjunto de treinamento diminui, ao passo que o erro no conjunto de validação aumenta (GOODFELLOW et al., 2016).

Uma das contra-medidas mais simples, porém eficaz, para não permitir que o modelo continue aumentando seu erro no conjunto de validação, é a técnica de *Early Stopping*, em português, Parada Antecipada. Essa técnica consiste em armazenar o melhor estado dos pesos em relação ao conjunto de validação e apenas substituí-lo quando houver um novo estado que performe um erro menor (GÉRON, 2017).

Além disso, é contado o número de passos tomados – número de atualizações dos pesos – desde o último melhor estado e interrompido o treinamento quando estes passos ultrapassam o limite estipulado. Dessa forma, a RNA pode ser configurada com o último melhor estado atingido antes que erro no conjunto de validação aumente (GÉRON, 2017).

Outra técnica consagrada para lidar com o *overfitting* é o *Dropout*, detalhada em Srivastava et al. (2014). Essa, no que lhe concerne, consiste em aleatoriamente desligar alguns neurônios durante cada passo do treinamento de uma RNA, o que previne os pesos de tornarem-se demasiadamente dependentes entre si. Além do mais, essa dependência entre os neurônios faz com as chances de *overfitting* aumentem, consequentemente resultando em uma RNA com uma baixa performance no conjunto de teste.

Em contrapartida, os autores destacam que a aplicação do *Dropout* apresenta efeitos colaterais. Vê-se que a técnica aumenta de duas a três vezes o tempo de treinamento, em virtude do treinamento da RNA representar um rearranjo diferente de neurônios a cada passo.

2.6.8 Avaliando Modelos

Existem diferentes métricas para avaliar um modelo de RNA, esteja este atendendo à um problema de classificação ou de regressão. Um vez que o presente trabalho propõe-se a prever o preço de fechamento de ações da bolsa de valores brasileira, esta subseção descreve as principais métricas utilizadas em problemas de regressão.

Antes, é importante salientar que essas métricas são aplicadas em conjuntos de n instâncias, dispondo do valor esperado y para cada instância. Logo, ao aplicar-se o modelo de RNA nesse conjunto de instâncias, será predito um valor \hat{y} o qual será comparado com o valor esperado y . Exemplos de métricas incluem:

- O Erro Absoluto Médio (EAM), vide Equação (8), que calcula a soma dos valores absolutos obtidos por meio da diferença entre o valor predito \hat{y} e o esperado y . Essa soma é então dividida pelo número de instâncias do conjunto, obtendo-se, assim, a média

da soma absoluta de todos os erros;

$$EAM(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (8)$$

- Já o Erro Quadrático Médio (EQM) apenas se difere do EAM em não calcular os valores absolutos da diferença entre \hat{y} e y , e sim, elevar essa diferença ao quadrado. Essa diferença pode ser vista na [Equação \(9\)](#), a qual ilustra o EQM;

$$EQM(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (9)$$

- No que diz respeito à Raiz do Erro Quadrático Médio (REQM), como o próprio nome sugere, apenas calcula-se a raiz quadrada do EQM. Essa métrica é, portanto, facilmente calculada e representada pela [Equação \(10\)](#);

$$REQM(y, \hat{y}) = \sqrt{EQM(y, \hat{y})} \quad (10)$$

- Ademais, a [Equação \(11\)](#) representa a métrica R^2 – também conhecida como Coeficiente de Determinação – a qual denota o quão bem generalizado é o modelo avaliado. Dessa forma, o melhor valor possível nessa métrica é 1.0, em que um R^2 de 0 representa um modelo que sempre prediz um valor \bar{y} fixo, independente de suas entradas. Caso o modelo seja pior do que isto, a métrica R^2 assumirá valores negativos;

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad \text{onde } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (11)$$

- Por fim, a Correlação de Pearson (ρ) pode ser utilizada para mensurar a correlação linear entre duas variáveis. Essa métrica pode assumir qualquer valor entre -1 e 1 , em que um valor de ρ próximo de 0 representa a falta de correlação entre as variáveis. Assim, quanto mais próximo de 1 ρ estiver, mais direta é a relação entre as variáveis; -1 , indireta. Isto é, para um ρ próximo de 1, a medida que y assumir valores maiores, \hat{y} também assumirá valores maiores na mesma proporção. Logo, para calcular a Correlação de Pearson, basta empregar a [Equação \(12\)](#).

$$\rho(y, \hat{y}) = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}} \quad (12)$$

3 TRABALHOS CORRELATOS

Este capítulo apresenta os trabalhos correlatos – isto é, trabalhos que de algum modo se assemelham com o presente trabalho – seus métodos e resultados obtidos. Por fim, a [Seção 3.1](#) sumariza os trabalhos correlatos, assim como destaca suas lacunas.

A dez anos atrás, [Bollen, Mao e Zeng \(2011\)](#) publicaram um artigo no *Journal of Computational Science* que ganhou a atenção da comunidade científica da época, intitulado: “*Twitter mood predicts the stock market*”, que em tradução literal para o português significa “O humor do Twitter prevê o mercado acionário”.

Nele, os autores utilizaram duas ferramentas capazes de medir as variações de humor expressadas por usuários do Twitter nos *tweets* publicados entre 28 de Fevereiro de 2008 e 19 de Dezembro de 2008. O intervalo temporal em questão fora escolhido por contemplar o dia de ação de graças e as eleições presidenciais dos Estados Unidos da América.

A primeira ferramenta utilizada, [OpinionFinder](#), é capaz de analisar o conteúdo de um *tweet* quanto à sua positividade e negatividade. Já o *Google-Profile of Mood States* (GPOMS), a segunda ferramenta utilizada pelos pesquisadores, é capaz de analisar um *tweet* em até seis formas: calma, alerta, certeza, vitalidade, gentileza e felicidade.

Em seguida, uma análise de causalidade de Granger, juntamente com um modelo de *Self-Organizing Fuzzy Neural Network*, foram utilizados para investigar a hipótese de que o humor expresso pelos usuários do Twitter – mensurados pelo OpinionFinder e GPOMS – é capaz de prever as oscilações do DJIA.

Como resultado, os pesquisadores evidenciaram que a predição das oscilações do DJIA pode ser significativamente melhorada com a análise do humor expresso pelos usuários do Twitter. Assim, os autores obtiveram uma acurácia de 86.7% na predição das oscilações com tendência de subida e descida do DJIA, além de uma redução na métrica de erro absoluto médio percentual da ordem de 6%.

Já [Hegazy, Soliman e Salam \(2013\)](#), propuseram um modelo de predição híbrido baseado em *Particle Swarm Optimization* (PSO) e LSSVM. O PSO é um tipo de algoritmo evolutivo, utilizado pelos autores para otimizar os parâmetros do LSSVM, que por sua vez representa um SVM regularizado pelo método de *Least Squares*.

O estudo utilizou ações da S&P 500 no período de Janeiro de 2009 até Janeiro de 2012, englobando diferentes setores do mercado, sendo estes: Tecnologia da Informação, finanças, saúde, energia, comunicações, materiais e industrial. Na sequência, para realizar a análise, cinco indicadores técnicos foram utilizados no estudo: *Relative Strength Index* (RSI); *Money Flow Index* (MFI); MME; *Stochastic Oscillator of %K* (SOK) e; MACD.

O modelo proposto de LSSVM com PSO, juntamente com um modelo LSSVM sem o PSO e uma RNA foram treinados com as ações da S&P 500, utilizando os cinco indicadores técnicos supraditos para os diferentes setores abordados no estudo. Logo, para o treinamento

dos modelos foi dedicado 70% do *dataset*, já os 30% restantes dedicados ao teste dos três modelos.

Enfim, o estudo evidenciou que o modelo proposto pelos pesquisadores – o qual uniu o método de PSO com LSSVM – apresentou os melhores resultados em todos os setores escolhidos pelo estudo, segundo a métrica de EQM. Por fim, o segundo melhor modelo fora o LSSVM sem a utilização do método de PSO, enquanto que o RNA apresentou os piores resultados.

Em outro trabalho correlato, o de [Chen, Zhou e Dai \(2015\)](#), os autores apostaram em um modelo de predição baseado em RNR e LSTM para prever as oscilações do mercado acionário Chinês. As características escolhidas para treinar o modelo de predição foram o volume, preço de abertura, preço de fechamento, preço máximo, preço mínimo e o índice de outros mercados acionários relacionados ao mercado acionário Chinês.

Em seguida, para o estudo, as características supracitadas foram obtidas por meio do Yahoo! Finance, coletando assim as séries históricas do mercado acionário Chinês, mais especificamente das bolsas de valores de Shanghai e Shenzhen. O período temporal considerado pelo estudo foi de 1 de Junho de 2013 a 31 de Maio de 2015, período este posteriormente subdividido em instâncias de 30 dias consecutivos.

Para o treinamento, os autores categorizaram o *dataset* em sete classes baseadas na taxa de ganhos. Ademais, para o treinamento do modelo, utilizaram o método de SGD, juntamente com o RMSProp, o que apresentou uma taxa de aprendizado de 0.001. Além disso, os pesquisadores utilizaram um *mini batch* de 64 instâncias, aplicando a normalização z-score.

Com o experimento, os pesquisadores descobriram que as seguintes abordagens guiaram o modelo de predição para uma melhor acurácia: emprego da normalização; utilização de índices de outros mercados acionários com impacto no mercado acionário em análise; e o treinamento do modelo de predição com uma ação por vez ao invés de utilizar todas as ações disponíveis em um mercado acionário.

Como resultado, o estudo obteve uma acurácia de 27.2% superior se comparado a modelos de predição aleatórios. No entanto, embora os resultados tenham demonstrado a capacidade das células LSTM, os autores concluíram que o mercado acionário Chinês permanece complexo e desafiador.

Por fim, [Pasupulety et al. \(2019\)](#) propuseram um modelo de *Ensemble Learning*, juntamente com uso da Análise de Sentimentos, para analisar as oscilações do INSE entre 1 de Agosto de 2018 e 31 de Janeiro de 2019. Mais especificamente, foram utilizadas características obtidas pela *Application Programming Interface* (API) Quandl, tais como, preço máximo, preço mínimo, preço de abertura, preço de fechamento, volume e *turnover* relativos ao INSE.

Ademais, também foram utilizados sete indicadores técnicos para a análise das oscilações do INSE: MACD; *Channel Commodity Index* (CCI); SOK; *Stochastic Oscillator of %D* (SOD); *21-day Rate of Change* (21ROC); RSI e; *Price Volume Trend* (PVT).

O modelo *ensemble* foi gerado a partir de um modelo de SVM, baseado em um modelo

Extremy Randomized Trees. Para realizar a combinação entre ambos, os autores utilizaram a técnica *Stacked Regressor*, mais especificamente por meio de um *meta-regressor*.

Quanto à Análise de Sentimentos, os autores extraíram *tweets*, do Twitter, que mencionassem empresas listadas na INSE. Para a tarefa de coleta, o trabalho utilizou a biblioteca *Tweepy*, disponível para a linguagem de programação Python. Em seguida, os *tweets* extraídos foram pré-processados com técnicas de Processamento de Linguagem Natural (PLN), tais como: remoção de pontuação, *stop words*, URLs, *hashtags*, menções e *emoticons*. Na sequência, aplicaram, nos *tweets* pré-processados, a técnica de *Word Embedding*, conhecida como Word2Vec, a qual mapeia as palavras para uma representação vetorial de 300 dimensões.

Após esse pré-processamento dos *tweets*, um modelo de *Random Forest* foi treinado com o *dataset Large Movie Review*, da Universidade de *Stanford*. Depois de treinado, passou então a ser utilizado para analisar o sentimento de *tweets* coletados em tempo real. Por fim, o número total de *tweets*, positivos e negativos, foi contabilizado cumulativamente para gerar a representação final do *dataset*.

Contudo, os autores destacam que a utilização da Análise de Sentimentos, juntamente com o modelo *ensemble*, representou uma melhoria de apenas 0,003%, não sendo, portanto, significativa na predição das oscilações do INSE.

3.1 Síntese e Lacunas dos Trabalhos Correlatos

Em linhas gerais, todos os trabalhos supracitados optaram por utilizar o Aprendizado de Máquina em suas predições. Outrossim, todos os trabalhos agregaram um ou mais métodos no desenvolvimento do modelo de predição que propuseram, os quais foram sumarizados pelo [Quadro 1](#).

Quadro 1 – Trabalhos Correlatos

Autores	Abordagem	Mercado Acionário	Período
Bollen, Mao e Zeng (2011)	Causalidade de Granger <i>Self-Organizing Fuzzy Neural Network</i> Análise de Sentimentos	DJIA	Fev/2008 Dez/2008
Hegazy, Soliman e Salam (2013)	<i>Particle Swarm Optimization</i> <i>Least Squares Support Vector Machine</i>	S&P 500	Jan/2009 Jan/2012
Chen, Zhou e Dai (2015)	Redes Neurais Recorrentes Índices de outras bolsas de valores	Shanghai e Shenzhen	Jun/2013 Maio/2015
Pasupulety et al. (2019)	<i>Ensemble Learning</i> Análise de Sentimentos	INSE	Ago/2018 Jan/2019

Fonte: Autoria Própria

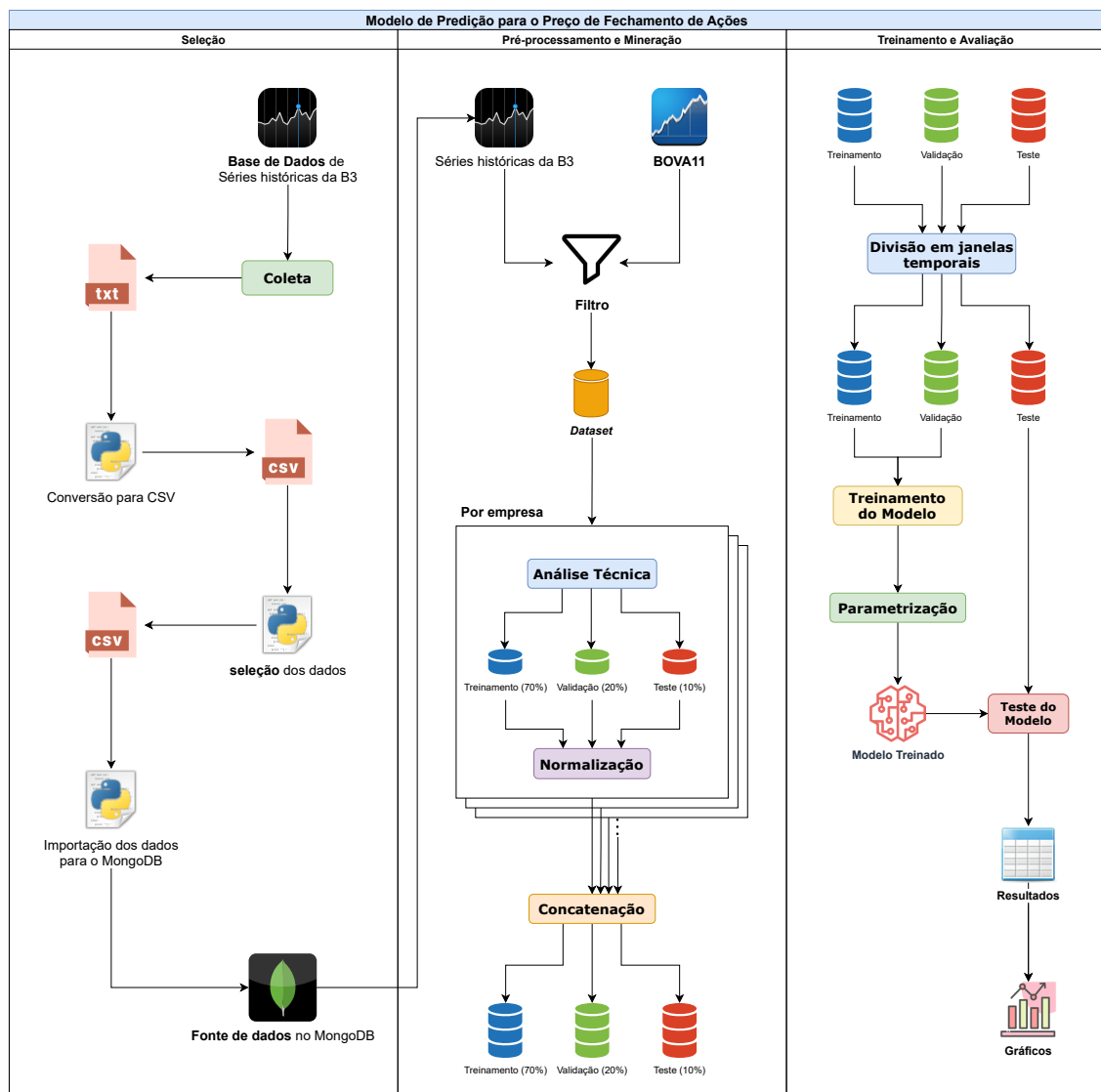
Contudo, este trabalho de conclusão de curso busca abranger um período temporal maior do que o abordado pelos trabalhos listados pelo [Quadro 1](#), coletando duas décadas de ações compreendidas entre 2001 e 2021. Outra diferença está no treinamento do modelo proposto com ações de empresas que formam a carteira de ativos do ETF BOVA11.

4 METODOLOGIA DO DESENVOLVIMENTO DO MODELO

Este capítulo detalha a metodologia do desenvolvimento do modelo de predição de preços de fechamento proposto por este trabalho, seguindo as diretrizes do KDD. A [Seção 4.1](#) descreve como foram coletadas, selecionadas e armazenadas as séries históricas de ações do mercado acionário brasileiro. Já a [Seção 4.2](#) explica como os dados armazenados foram pré-processados, culminando em um *dataset*.

Na sequência, a [Seção 4.3](#) detalha as técnicas de mineração de dados utilizadas para a extração de características do *dataset*. Por fim, a [Seção 4.4](#) elucida como o modelo de predição baseado em Rede Neural Recorrente foi treinado e avaliado a fim de prever os preços de fechamento das ações da Petrobrás. A [Figura 9](#) ilustra as etapas supracitadas.

Figura 9 – *Workflow* do Desenvolvimento do Modelo de Predição de Preços de Fechamento



Fonte: Autoria Própria

4.1 Seleção

Como explicado por [Williams e Huang \(1996\)](#), na etapa de seleção os dados coletados são apropriadamente selecionados de uma base de dados, a fim de gerar uma fonte de dados. Isto posto, todas as atividades inerentes a etapa de seleção serão descritas a seguir.

A princípio, séries históricas da B3 podem ser coletadas com facilidade no próprio [site da empresa](#), o qual disponibiliza séries históricas datadas desde 1986, subdivididos em intervalos diários. Entretanto, este trabalho limitou-se a explorar séries históricas compreendidas entre 2 de Janeiro de 2001 a 4 de Outubro de 2021, formando assim uma ampla base de dados com períodos importantes para o mercado acionário, a citar alguns: mudanças de governos; os ataques de 11 de setembro de 2001; a crise do mercado imobiliário de 2008; a greve dos caminhoneiros de 2018 e; a pandemia do novo coronavírus em 2020.

As séries históricas vêm em arquivos posicionais em formato de texto, isto é, cada linha do arquivo representa uma ação de uma empresa negociada em um determinado dia. Nessas linhas, os dados não possuem um caractere separador, pois situam-se em posições específicas nas linhas. Mais especificamente, para cada linha são 26 dados representados por colunas, as quais começam e terminam em posições específicas, como demonstra o [guia de layout do arquivo](#) de ações da B3. Assim, baixou-se 20 arquivos posicionais, cada um representando as ações diárias negociadas no período de um ano, totalizando 2.03 GB de dados.

A posteriori, foram criados três *scripts* (códigos de programação) na linguagem de programação Python, juntamente com o uso da biblioteca Pandas¹, a fim de manipular os arquivos de texto coletados:

O primeiro *script* encarregou-se de transformar os 20 arquivos posicionais em apenas um arquivo *Comma Separed Values* (CSV), facilitando a manipulação das 26 colunas, agora separadas por vírgula. Com isso, o *script* gerou um arquivo CSV com 8.830.227 registros, em que cada registro representa uma ação diária de uma determinada empresa, que juntos somam 2.22 GB de dados.

Já o segundo *script* reduziu o tamanho do arquivo CSV, selecionando apenas sete das 26 colunas de cada registro, considerando apenas ações do mercado à vista e fracionário. As colunas selecionadas foram: data do pregão, código de negociação, preço de abertura, preço máximo, preço mínimo, preço de fechamento e volume. Assim, a seleção reduziu o número de registros para 3.455.148, totalizando 160 MB de dados em um único arquivo CSV.

Por fim, o terceiro *script* incumbiu-se de importar o CSV resultante para um banco de dados MongoDB, formando-se assim a fonte de dados deste trabalho, a qual totaliza 3.455.148 registros armazenados em 476 MB de dados.

A preferência pelo banco de dados *Not Only SQL* (NoSQL) orientado à documentos MongoDB justifica-se pelo grande volume de dados coletados, e pelo não relacionamento entre os dados. Além disso, uma vez estando os dados coletados em formato CSV, estes puderam

¹Biblioteca disponível para a linguagem de programação Python, utilizada comumente para a manipulação e análise de dados.

ser facilmente importados para um banco de dados MongoDB. Isso se dá pois esse Sistema de Gerenciamento de Banco de Dados (SGBD) não exige que o(a) programador(a) esquematize o banco de dados, visto que esse processo é feito de forma automática pelo MongoDB com a inserção dos dados.

Ademais, o SGBD MongoDB estrutura os dados em coleções – que são equiparáveis às tabelas em SGBDs relacionais – as quais contêm documentos. Cada documento representa um registro, em que as sete colunas selecionadas pelo segundo *script* são armazenadas em atributos. Outrossim, o MongoDB encarrega-se de criar em todos os documentos um atributo de valor único que é utilizado para a identificação de cada documento. Logo, criou-se uma coleção de dados para as séries históricas da B3 que representa a fonte de dados deste trabalho.

4.2 Pré-processamento

Seguindo com a metodologia KDD, e com a fonte de dados devidamente criada, dá-se vez à etapa de pré-processamento. Assim, o primeiro passo para a transformação da fonte de dados em um *dataset* foi criar um *script* em um Jupyter Notebook na linguagem de programação Python, juntamente com o auxílio da biblioteca Pandas, para manipular e pré-processar com maior facilidade a fonte de dados.

Com o *script*, pôde-se filtrar as empresas de interesse deste trabalho, as quais formam a carteira de ativos do ETF BOVA11. Mais especificamente, o BOVA11 conta com 88 empresas que compõem o Índice Bovespa, as quais juntas representam mais de 80% do volume financeiro do mercado acionário brasileiro (INFOMONEY, 2021a).

Além disso, excluiu-se empresas com séries históricas inferiores a 100 dias. Por conseguinte, com essas duas filtragens, reduziu-se o número de empresas analisadas de 5671 para 83, em que o número remanescente de registros diminuiu para 253.838. Dito isso, a Figura 10 representa as 83 empresas filtradas, ilustradas por meio de seus códigos de negociação, cujos tamanhos são relativos aos seus respectivos pesos na carteira do BOVA11.

Figura 10 – Nuvem de Palavras dos Códigos de Negociação das 83 Empresas Filtradas



Fonte: Autoria Própria

Na sequência, aplicou-se a etapa de mineração de dados no *dataset*, gerando-se três novas características, as quais serão apresentadas e explicadas na [Seção 4.3](#). Em síntese, cada registro do *dataset* é composto por:

- a) Um index para identificar os registros de forma temporal. Para isso, utilizou-se a data do pregão, em que cada registro é identificado pelo ano, mês e dia que a ação foi negociado na B3;
- b) Cinco características numéricas, anteriormente representadas por colunas, sendo estas o preço de abertura, preço máximo, preço mínimo, preço de fechamento e o volume;
- c) Uma característica em formato de *string* (texto), responsável por identificar as ações diárias de uma determinada empresa. Essa característica é representada pelo código de negociação, e.g., PETR4 (Petrobrás) ou MGLU3 (Magazine Luiza);
- d) Por fim, três características resultantes da aplicação de indicadores técnicos nas ações diárias de cada empresa. As três características em questão serão melhor descritas na [Seção 4.3](#).

Prosseguindo, o *dataset* resultante, composto por 253.838 registros, 83 empresas, oito características e um index foi salvo em um arquivo de 9.05 MB no formato *parquet* para facilitar sua manipulação.

Posteriormente, o *dataset* foi agrupado por meio dos códigos de negociação, que representam as 83 empresas filtradas pelo *script*, mantendo-se a ordem temporal dos registros. Em seguida, cada grupo de registros representados por uma empresa, foi dividido entre os subconjuntos de treinamento, validação e teste, respectivamente, totalizando 70%, 20% e 10% dos registros em cada subconjunto.

Mais especificamente, para cada empresa, o conjunto de treinamento contemplou da primeira ação negociada até que se totalizasse 70% das ações coletadas, respeitado a ordem cronológica das ações. Já para o conjunto de validação, a partir da última ação que compõe o conjunto de treinamento, agrupou-se cronologicamente as ações remanescentes até que se totalizasse 20% das ações. Por fim, o conjunto de teste foi formado pelos 10% das ações restantes, também mantendo a ordem cronológica.

Ademais, aplicou-se a normalização *z-score* nas cinco características numéricas e, também, nas três características oriundas da aplicação dos indicadores técnicos. Em cada empresa, utilizou-se o conjunto de treinamento para o cálculo da normalização, a qual foi aplicada, em seguida, nos três subconjuntos.

Por fim, os subconjuntos de treinamento das 83 empresas foram concatenados, formando assim o conjunto de treinamento. Idem para os conjuntos de validação e de teste. Por último, o código de negociação de todos os conjuntos foi removido, mantendo-se apenas oito características e um index por registro.

4.3 Mineração

Na etapa de mineração, o objetivo é extrair características do *dataset* por meio da aplicação de técnicas de mineração de dados. Logo, a Análise Técnica foi utilizada para extrair novas características, em que os indicadores técnicos empregados foram escolhidos segundo os estudos de [Caetano, Meinerz e Souza \(2021\)](#), também inseridos no contexto da B3.

A primeira característica foi criada por meio de duas MMEs aplicados ao preço de fechamento. A primeira delas foi calculada com base em um período curto de cinco dias, já a segunda com base em um período longo de vinte dias. Assim, o cruzamento da MME curta por cima da MME longa indica um sinal de compra, do contrário, tem-se um sinal de venda. Desse modo, ao subtrair-se a MME longa da MME curta, cria-se uma característica, em que valores positivos indicam compra; negativos, venda.

Já a segunda característica foi gerada por meio do MACD aplicado ao preço de fechamento. Para a geração da linha MACD uma MME curta de doze dias e uma MME longa de vinte e seis dias foram empregadas. Depois, uma MME de nove dias foi escolhida para o cálculo do sinal de linha. Logo, quando a linha MACD cruza acima do sinal de linha, recomenda-se a compra, do contrário, recomenda-se a venda. Assim, ao se subtrair o sinal de linha da linha MACD, tem-se a segunda característica, em que valores positivos indicam um sinal de compra; negativos, venda.

Por fim, a última característica foi calculada por meio de uma Média Móvel Simples de dez dias aplicada ao volume. Com isso, pôde-se traduzir a força vendedora e a força compradora, em que um volume acima (abaixo) da Média Móvel Simples aplicada ao volume pode indicar a confirmação (ou não) de uma tendência.

4.4 Treinamento e Avaliação do Modelo

Com todas as características extraídas e devidamente pré-processadas, dá-se vez ao treinamento e avaliação do modelo proposto. Outrossim, além do modelo proposto de Rede Neural Recorrente, um modelo linear simples foi utilizado como patamar a ser superado pelo modelo proposto. Ambos os modelos foram treinados em dois cenários: o primeiro considerando todas as 83 empresas que formam a carteira de ativos do ETF BOVA11 – e com séries históricas superiores a 100 dias – e o segundo cenário considerando apenas as ações da Petrobrás, representada pelo código de negociação PETR4.

Além da diferença dos conjuntos de treinamento utilizados nos dois cenários, o conjunto de validação também diferiu, em que apenas o conjunto de teste manteve-se o mesmo. Isto é, independentemente de qual cenário o modelo proposto, ou o linear, foram treinados e avaliados, ambos foram testados em um conjunto nunca antes visto, representado pelas ações diárias da Petrobrás compreendidas entre 9 de Setembro de 2019 e 4 de Outubro de 2021.

Ademais, em ambos os cenários, os conjuntos de treinamento, validação e teste foram subdivididos em janelas temporais, i.e., subintervalos de dois ou mais dias. As janelas testadas

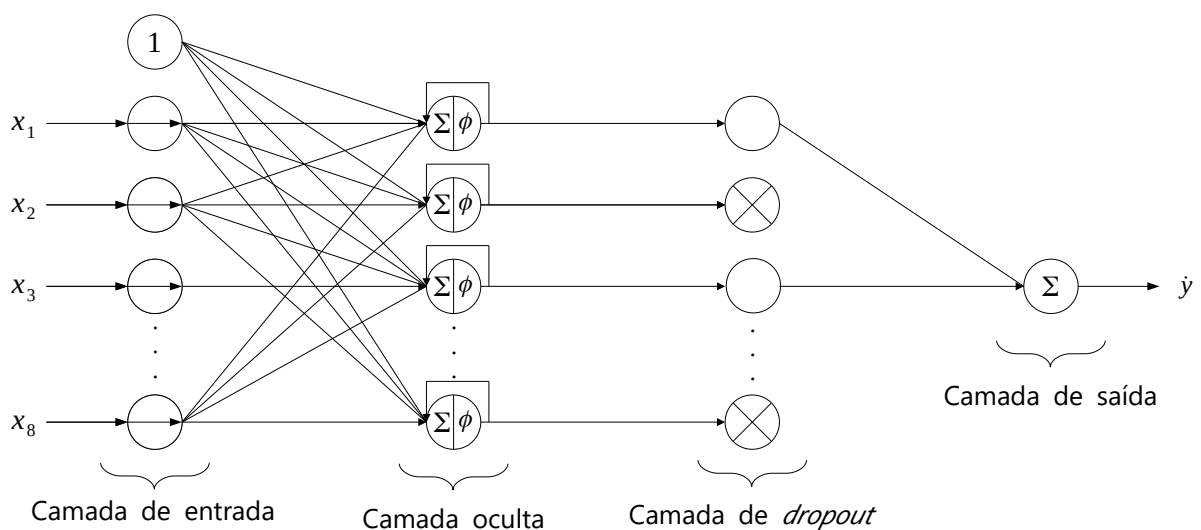
por este trabalho foram as de 10, 20, 30, 40, 50, 60 e 70 dias. De mesma forma, para todas as janelas temporais escolhidas, determinou-se um dia no futuro para ser predito o preço de fechamento pelos modelos. Com isso, dois modelos testados em dois cenários distintos com sete janelas temporais cada, resultaram em vinte e oito experimentos a fim de descobrir qual o melhor modelo de RNR para prever os preços de fechamento da Petrobrás.

Até se encontrar o melhor modelo proposto de Rede Neural Recorrente para cada experimento, diferentes configurações foram avaliadas. Variadas quantidades de células de memória bem como diferentes quantidades de camadas ocultas foram experimentadas, variando entre células de memória LSTM e GRU. Por fim, diferentes valores de *dropout* também foram testados.

Ainda no que tange ao modelo proposto, independentemente das camadas ocultas, em todos os experimentos as camadas de entrada e de saída mantiveram-se inalteradas. A camada de entrada de todos os experimentos foi composta por oito neurônios, representados pelas oito características detalhadas na Seção 4.2, mais um *bias*. Já a camada de saída manteve-se com apenas um neurônio sem função de ativação (linear), responsável por unir os neurônios das camadas ocultas a fim de prever o preço de fechamento do dia seguinte.

Dito isso, a principal diferença entre o modelo proposto e o modelo linear está nas camadas ocultas e na aplicação de *dropout*, vide Figura 11 e Figura 12. Isto é, o modelo linear apenas é composto pelas camadas de entrada e de saída, as quais são idênticas às utilizadas pelo modelo proposto. Em contrapartida, as semelhanças entre ambos os modelos estão na utilização do *Faster Optimizer* Adam, com uma taxa de aprendizado η de 0.001, além de um *batch* de 32 instâncias e da aplicação da inicialização de Xavier.

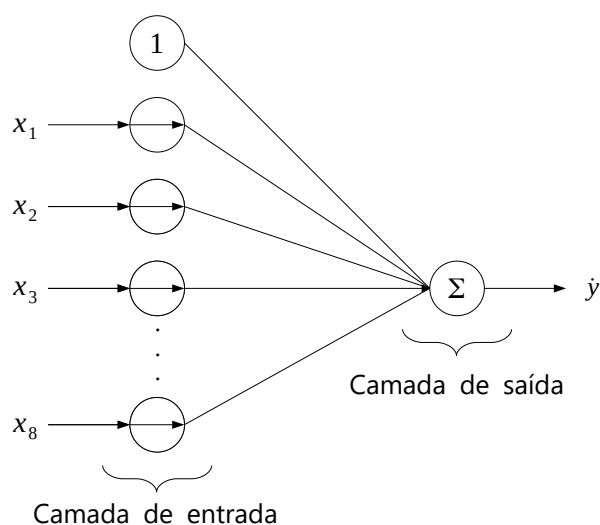
Figura 11 – Modelo Proposto de Rede Neural Recorrente



Fonte: Autoria Própria

Posteriormente, para encontrar o melhor modelo proposto de RNR para prever o preço de fechamento das ações da Petrobrás, as métricas de EAM, R^2 e a correlação de Pearson

Figura 12 – Modelo Linear Simples



Fonte: Autoria Própria

foram utilizadas. Ainda, é importante destacar que as métricas de EQM e REQM não foram consideradas por representarem a mesma informação oferecida pela EAM. Assim, para eleger a melhor configuração do modelo proposto em cada experimento, as métricas supracitadas foram calculadas para os dois modelos e, enfim, comparadas.

Dessa forma, diferentes combinações de células de memória, camadas ocultas e valores de *dropout* foram testadas até que se superasse o modelo linear, o qual sempre se manteve com apenas uma configuração fixa em todos os experimentos. Em vista disso, descobriu-se que um menor número de camadas ocultas indicavam melhores performances, e que as células de memória LSTM não conseguiram superar o modelo linear, diferentemente das células GRU. Por fim, valores muito altos de *dropout* dificultavam a convergência dos modelos, em que todos os experimentos tiveram seus treinamentos interrompidos pela técnica de *Early Stopping*.

Para o desenvolvimento, treinamento e avaliação dos modelos linear e proposto, foi escolhida a biblioteca Tensorflow por possuir suporte para a utilização de placas de vídeo no treinamento e avaliação de Redes Neurais. Já para a obtenção dos resultados, a biblioteca Pandas foi empregada e, enfim, para a criação dos gráficos a partir dos resultados obtidos, a biblioteca Plotly² foi selecionada.

Para a execução do trabalho, foi utilizado um *notebook* Acer Nitro 5, modelo AN515-54-574Q, equipado com as seguintes especificações:

- Processador Intel Core™ I5-9300H de 9ª Geração;
- 24GB de memória RAM DDR4 com frequência de 2666MHz;
- Placa de vídeo NVIDIA GeForce GTX™ 1650 com 4GB GDDR5 de VRAM;
- SSD PCIe NVMe M.2 de 512GB e HDD de 2TB.

²Biblioteca disponível para a linguagem de programação Python que permite a criação de gráficos interativos.

5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Este capítulo apresenta a análise e discussão dos principais resultados obtidos por este trabalho. Assim, com o intuito de facilitar a compreensão e dissertação dos próximos parágrafos, os modelos treinados e avaliados receberão acrônimos tanto para o cenário no qual foram treinados apenas com as ações da Petrobrás, bem como para o cenário no qual foram treinados com o auxílio do ETF BOVA11.

Logo, o modelo linear treinado apenas com as ações da Petrobrás, cujo código de negociação é o PETR4, receberá o acrônimo Linear-PETR4, enquanto o modelo linear treinado com as 83 empresas que formam a carteira de ativos do BOVA11 – e com séries históricas superiores a 100 dias – receberá o acrônimo Linear-BOVA11. De mesmo modo, RNR-PETR4 se refere ao modelo proposto de RNR treinado apenas com as ações da Petrobrás, assim como RNR-BOVA11 se refere ao modelo proposto de RNR treinado com o auxílio do ETF BOVA11.

Prosseguindo, com o treinamento, pôde-se perceber que os modelos treinados apenas com as ações da Petrobrás terminaram o treinamento em um tempo menor que os treinados com o auxílio do ETF BOVA11. Exemplificando, o modelo Linear-PETR4 encerrou seu treinamento em média com 8 ± 7 segundos em 11 ± 10 épocas, enquanto que modelo RNR-PETR4 terminou em média com 19 ± 11 segundos em 12 ± 6 épocas. Já o modelo Linear-BOVA11 concluiu seu treinamento em média com 102 ± 81 segundos em 6 ± 4 épocas, enquanto o modelo RNR-BOVA11 encerrou o treinamento em média com 382 ± 127 segundos em 9 ± 3 épocas.

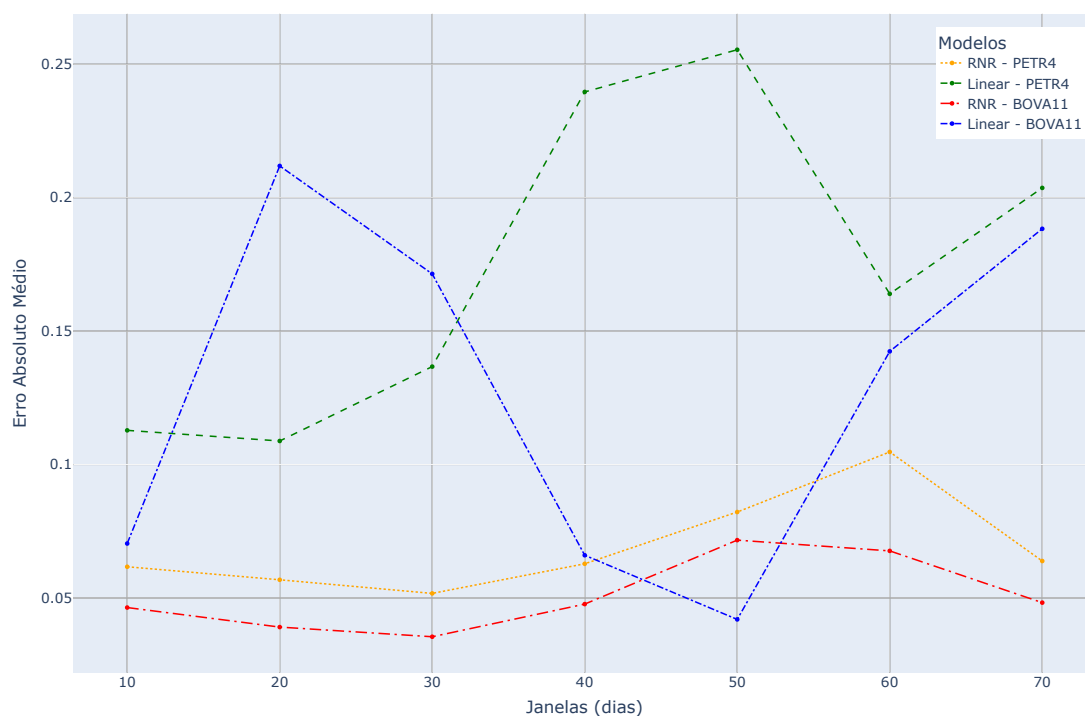
Ademais, com todos os vinte e oito experimentos realizados, selecionou-se as melhores parametrizações para o modelo proposto de RNR por meio da avaliação dos modelos mediante as métricas aplicadas ao conjunto de validação. Com isso, a avaliação ocorreu para todas as janelas temporais tanto para o conjunto de validação composto apenas pelas ações da Petrobrás como para o composto com o auxílio do ETF BOVA11.

Isto posto, observou-se que a melhor parametrização do modelo RNR-PETR4 foi aquela composta por apenas uma camada oculta com 16 células de memória GRU e um *dropout* de 10%. Em contrapartida, o modelo RNR-BOVA11 obteve sua melhor performance quando também composto por apenas uma camada oculta, porém com 128 células de memória GRU e um *dropout* de 2,5%.

Logo, a [Figura 13](#) ilustra o desempenho dos vinte e oito experimentos quanto a métrica de EAM nas sete janelas temporais performadas no conjunto de teste das ações da Petrobrás. Vê-se também que o modelo RNR-PETR4 apresentou em todas as janelas temporais um erro menor perante o modelo Linear-PETR4. Esse comportamento também foi observado nos modelos Linear-BOVA11 e RNR-BOVA11, salvo para a janela de cinquenta dias.

Ainda, observa-se que o modelo RNR-BOVA11 apresentou em todas as janelas temporais um erro inferior que o obtido pelo modelo RNR-PETR4. O mesmo é observado nos modelos Linear-BOVA11 e Linear-PETR4, salvo nas janelas temporais de vinte e trinta dias.

Figura 13 – EAM dos Modelos em Diferentes Janelas Temporais



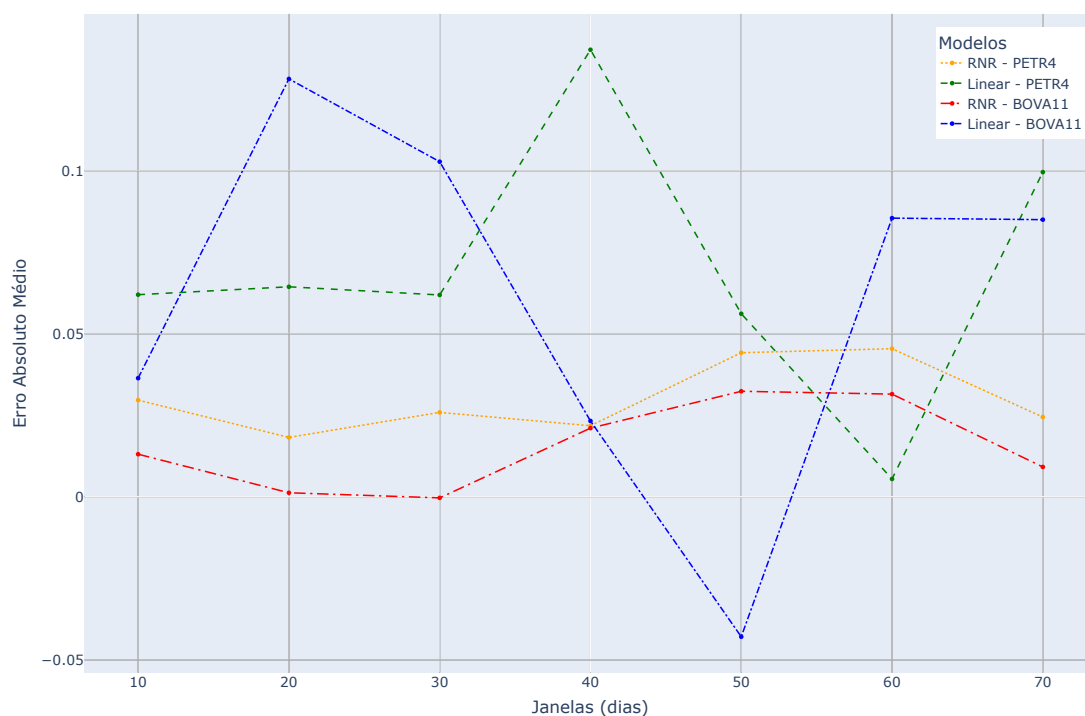
Fonte: Autoria Própria

Por fim, com a [Figura 13](#), percebe-se que os modelos RNR-PETR4 e RNR-BOVA11 demonstram estabilidade em todas as janelas temporais, performando quase sempre erros inferiores aos obtidos pelos modelos Linear-PETR4 e Linear-BOVA11. Contudo, apesar do modelo Linear-BOVA11 performar o menor erro para janela temporal de cinquenta dias, também foi capaz de apresentar o maior erro para a janela temporal de vinte dias.

Aprofundando as análises relativas à métrica de EAM, a [Figura 14](#) exibe o *overfitting* dos modelos nas sete janelas temporais. Mais especificamente, o *overfitting* foi mensurado pela diferença entre o EAM obtido no conjunto de teste das ações da Petrobrás com o EAM obtido pelo conjunto de validação das ações da Petrobrás.

Com isso, valores positivos de *overfitting* na [Figura 14](#) indicam que o modelo performou um erro menor no conjunto de validação que o obtido pelo conjunto de teste. Naturalmente, um *overfitting* com valores próximos de zero indica pouca diferença entre os erros obtidos nos dois conjuntos. Já um *overfitting* com valores negativos indica um cenário de exceção em que o modelo performou melhor no conjunto de teste do que no de validação.

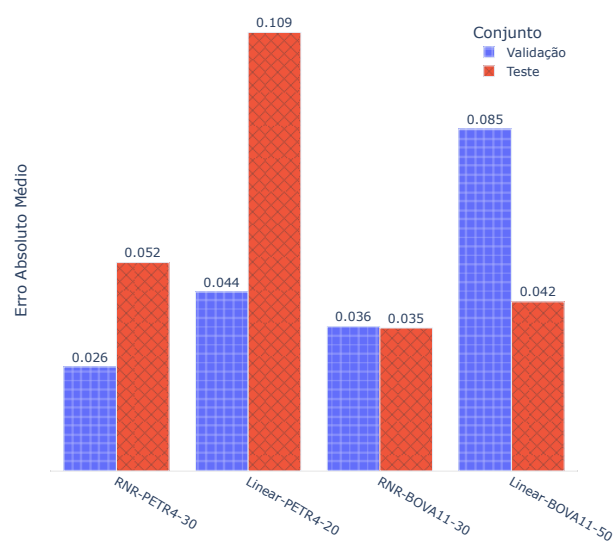
Dito isto, percebe-se que o único cenário em que o modelo Linear-BOVA11 superou o modelo RNR-BOVA11, ocorreu na janela temporal de cinquenta dias, a qual representa uma exceção. Outrossim, nota-se que tanto o modelo RNR-PETR4 quanto o RNR-BOVA11 apresentaram um baixo *overfitting* ao longo das janelas temporais, em especial na janela de trinta dias, o que não se repete nos modelos Linear-PETR4 e Linear-BOVA11.

Figura 14 – *Overfitting* do EAM dos Modelos em Diferentes Janelas Temporais

Fonte: Autoria Própria

Ademais, a Figura 15 elucida a melhor performance dos modelos Linear-PETR4, RNR-PETR4, Linear-BOVA11 e RNR-BOVA11 quanto a métrica de EAM. Logo, RNR-PETR4-30 representa o modelo RNR-PETR4 com uma janela temporal de trinta dias; Linear-PETR4-20, vinte dias; RNR-BOVA11-30, trinta dias; Linear-BOVA11-50, cinquenta dias.

Figura 15 – Modelos em suas Melhores Janelas Temporais Segundo a Métrica de EAM

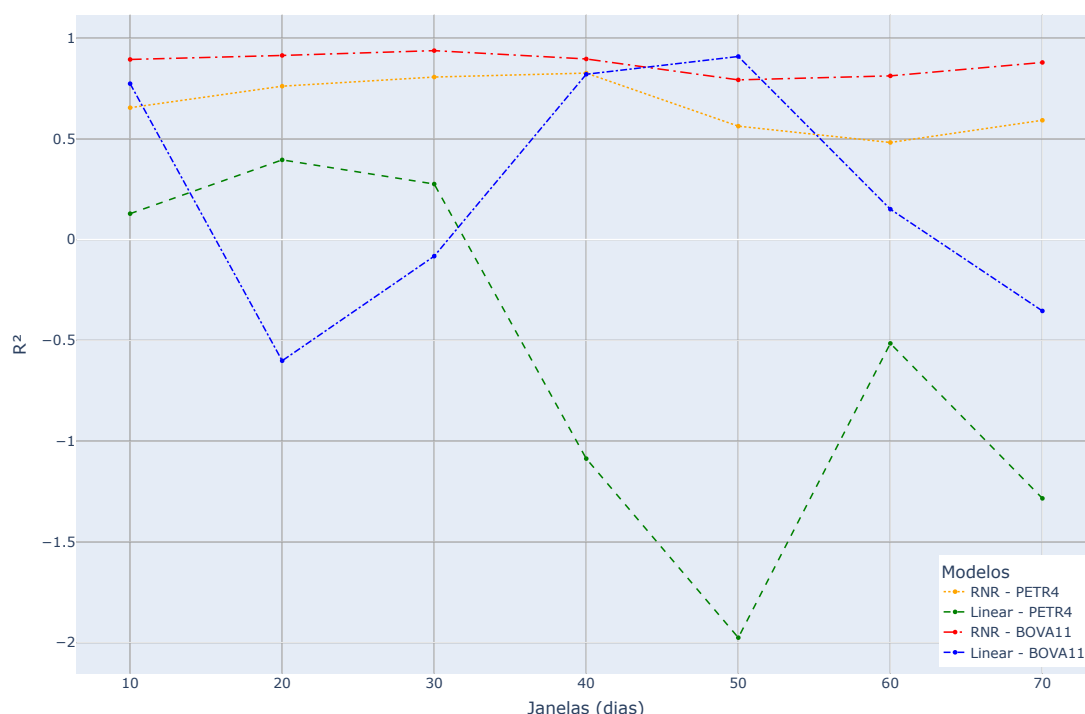


Fonte: Autoria Própria

Na Figura 15, observa-se que o modelo RNR-PETR4-30, apesar de possuir o menor erro no conjunto de validação, sofreu *overfitting* e não obteve o menor erro no conjunto de teste. Em contrapartida, o modelo RNR-BOVA11-30 não apresentou *overfitting*, enquanto que o modelo Linear-PETR4-20 apresentou o maior *overfitting*. Por fim, o modelo Linear-BOVA11-50 caracteriza uma exceção ao obter um erro menor no conjunto de teste que no de validação.

Em seguida, a métrica R^2 no conjunto de teste das ações da Petrobrás foi utilizada para mensurar o quão bem generalizados estão os modelos, o que é ilustrado pela Figura 16. Novamente, nota-se que os modelos treinados com o auxílio do ETF BOVA11 obtiveram uma melhor performance se comparados aos modelos treinados apenas com as ações da Petrobrás.

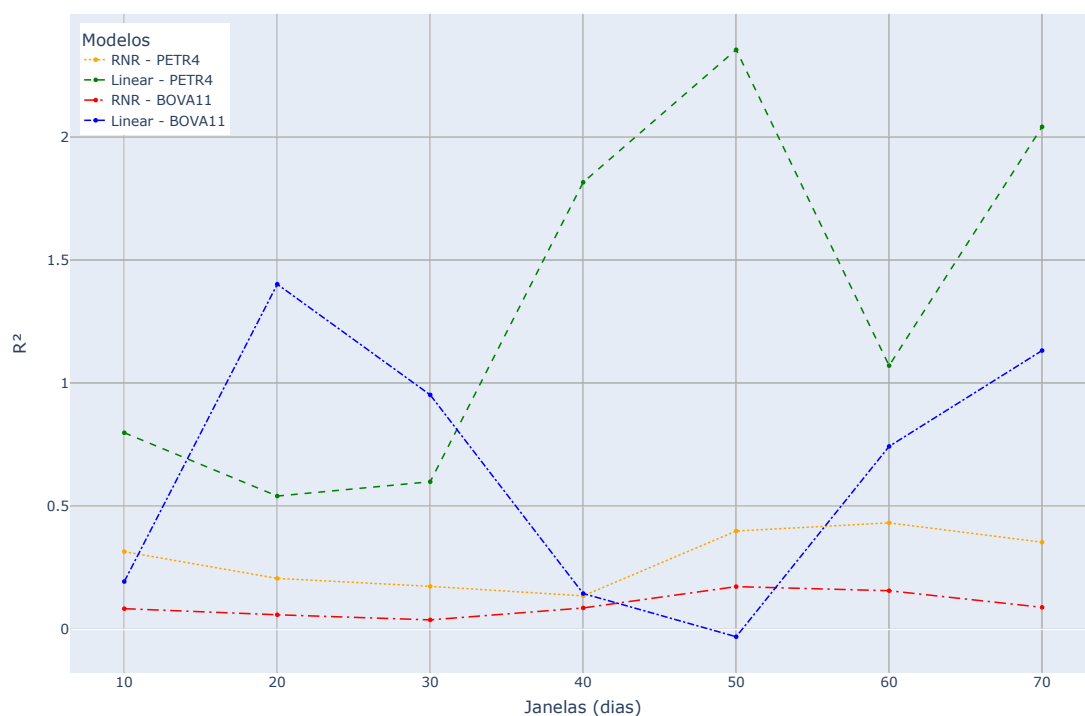
Figura 16 – R^2 dos Modelos em Diferentes Janelas Temporais



Fonte: Autoria Própria

Além disso, na Figura 16, vê-se que o modelo RNR-BOVA11 apresentou melhores resultados em todas as janelas temporais quando comparado ao modelo RNR-PETR4. Já os modelos Linear-PETR4 e Linear-BOVA11 apresentaram instabilidade ao longo das janelas temporais, além de valores negativos de R^2 , enquanto que os modelos RNR-PETR4 e RNR-BOVA11 mantiveram-se estáveis e sempre com valores positivos de R^2 .

Prosseguindo, a Figura 17 representa o *overfitting* da métrica R^2 dos dois modelos ao longo das janelas temporais. Novamente, vê-se que os modelos RNR-PETR4 e RNR-BOVA11 mantêm-se estáveis, além de performarem um baixo *overfitting*. Ainda, verifica-se que o modelo RNR-BOVA11 apresentou um *overfitting* menor que o modelo RNR-PETR4 para todas as janelas temporais.

Figura 17 – *Overfitting* do R^2 dos Modelos em Diferentes Janelas Temporais

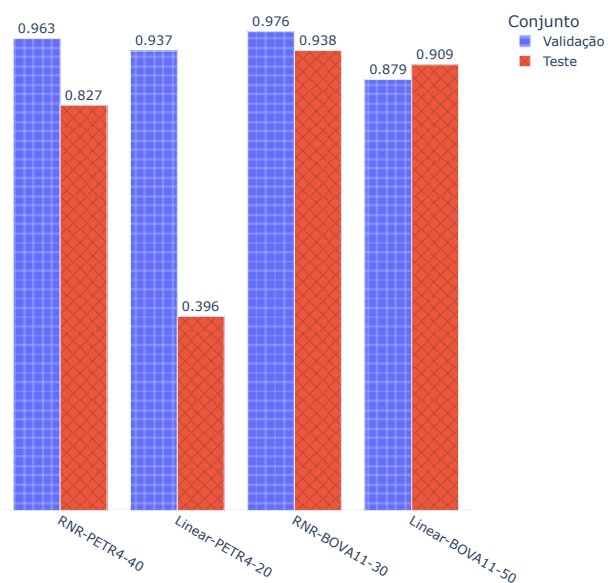
Fonte: Autoria Própria

Em contrapartida, os modelos Linear-PETR4 e Linear-BOVA11 demonstraram instabilidade ao longo das janelas temporais. Em especial, na janela temporal de vinte dias, o modelo Linear-BOVA11 apresentou o pior *overfitting*, bem como na janela temporal de cinquenta dias o modelo linear-BOVA11 representa, mais uma vez, uma exceção ao performar um valor de R^2 maior no conjunto de teste que no conjunto de validação.

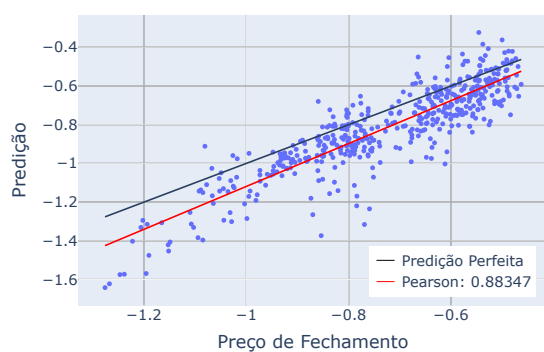
Dito isso, a Figura 18 ilustra a melhor performance dos modelos Linear-PETR4, RNR-PETR4, Linear-BOVA11 e RNR-BOVA11 quanto a métrica de R^2 . Assim como já supracitado, RNR-PETR4-40 representa o modelo RNR-PETR4 com uma janela temporal de quarenta dias; Linear-PETR4-20, vinte dias; RNR-BOVA11-30, trinta dias; Linear-BOVA11-50, cinquenta dias.

Ainda, no que tange à Figura 18, percebe-se que todos os modelos performaram um valor alto de R^2 no conjunto de validação. Sem embargo, ao performar no conjunto de teste, o modelo Linear-PETR4-20 obteve o maior *overfitting*, ao passo que o modelo Linear-BOVA11-50, novamente, representa uma exceção. Apesar disso, o modelo RNR-BOVA11-30 obteve o maior valor de R^2 nos experimentos realizados.

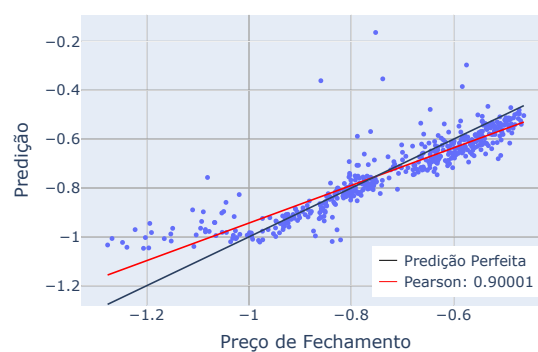
Ademais, a Figura 19 agrupa a performance dos modelos representados pela Figura 15, quanto à correlação de Pearson. Assim, utilizando-se do conjunto de teste das ações da Petrobrás, os resultados presentes na Figura 19 foram obtidos por meio do cálculo da correlação de Pearson entre o preço de fechamento das ações da Petrobrás com a predição do preço de fechamento gerado pelos modelos.

Figura 18 – Modelos em suas Melhores Janelas Temporais Segundo a Métrica de R^2 

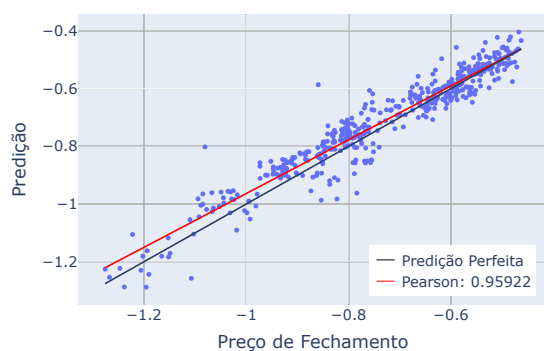
Fonte: Autoria Própria



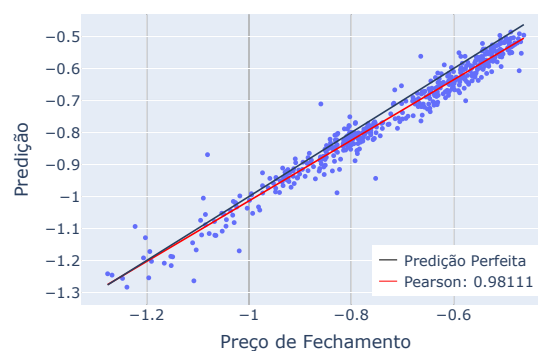
(a) Modelo Linear-PETR4-20



(b) Modelo RNR-PETR4-30



(c) Modelo Linear-BOVA11-50



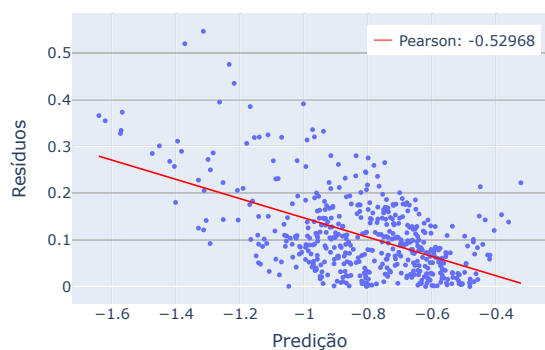
(d) Modelo RNR-BOVA11-30

Figura 19 – Correlação de Pearson Entre o Preço de Fechamento das Ações da Petrobrás e a Predição dos Modelos

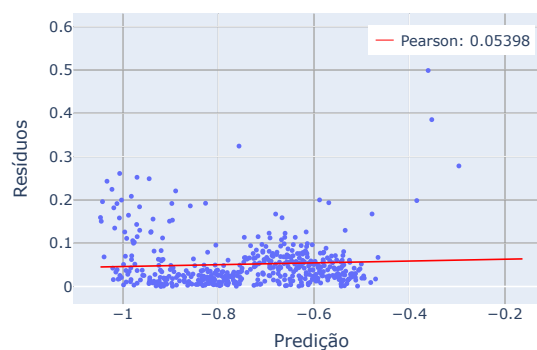
Mais especificamente, na [Figura 19](#), o eixo y representa o preço de fechamento predito pelo modelo em questão, já o eixo x representa o preço fechamento das ações da Petrobrás, em que ambos os eixos foram normalizados pela técnica de z -score. Desse modo, caso o modelo performasse uma predição perfeita, todos os pontos azuis estariam sobre a linha preta. Já a linha vermelha é obtida por meio da técnica de *Ordinary Least Squares* (OLS) e representa o quão próximo a predição do modelo está da linha preta, ao passo que quanto mais próximo de 1 a correlação de Pearson for, melhores predições o modelo fará.

Prosseguindo com as análises, vê-se que o modelo Linear-PETR4-20 ([Figura 19a](#)), cuja correlação de Pearson é de 0,88347, obteve a pior correlação entre os modelos, enquanto que o modelo Linear-BOVA11-50 ([Figura 19c](#)) ultrapassou o modelo RNR-PETR4-30 ([Figura 19b](#)), respectivamente com correlações de Pearson de 0,95922 e 0,90001. Todavia, fora o modelo RNR-BOVA11-30 ([Figura 19d](#)) que obteve a maior correlação de Pearson entre os modelos, atingindo o valor de 0,98111.

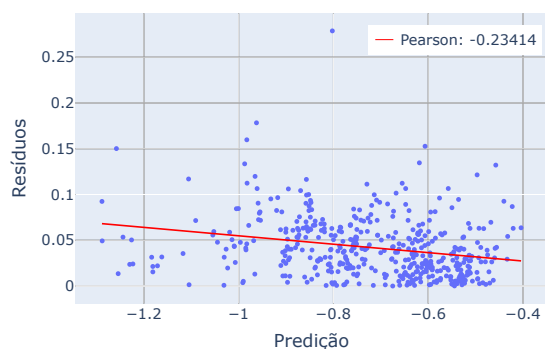
Outrossim, a [Figura 20](#) representa no eixo y os resíduos da predição, e.i., a diferença absoluta entre o preço de fechamento das ações da Petrobrás e a predição do modelo. Já o eixo x representa a predição do modelo em questão, em que ambos os eixos foram normalizados pela técnica de z -score. Dessa forma, novamente a técnica de OLS é utilizada para se traçar a tendência dos dados, bem como calculada a correlação de Pearson entre os eixos.



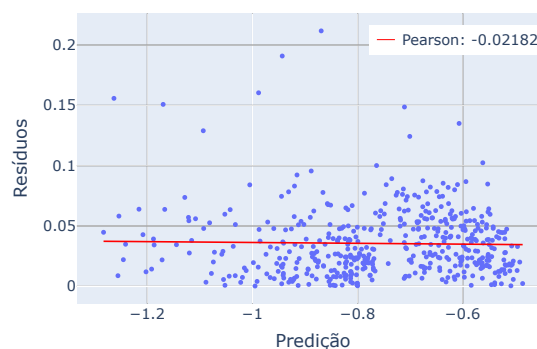
(a) Resíduos do Modelo Linear-PETR4-20



(b) Resíduos do Modelo RNR-PETR4-30



(c) Resíduos do Modelo Linear-BOVA11-50



(d) Resíduos do Modelo RNR-BOVA11-30

Figura 20 – Correlação de Pearson Entre a Predição dos Modelos e os Resíduos da Predição

Ainda na [Figura 20](#), nota-se que com uma correlação de Pearson de $-0,52968$ o modelo Linear-PETR4-20 ([Figura 20a](#)) possui uma correlação moderadamente negativa. Isso significa que quanto menor o preço de fechamento predito, maior será o erro do modelo; maior o preço de fechamento, menor o erro. Em contrapartida, o modelo RNR-PETR4-30 ([Figura 20b](#)) apresenta uma correlação de Pearson de $-0,05398$, o que representa a falta de viés nas previsões do modelo, pois há pouca correlação entre os dois eixos.

Além disso, o modelo Linear-BOVA11-50 ([Figura 20c](#)) obteve uma leve correlação negativa de $-0,23414$, resultando em um modelo que performa sutilmente melhor na previsão de altos preços de fechamento das ações da Petrobrás. Já o modelo RNR-BOVA11-30 ([Figura 20d](#)) apresentou o menor viés entre os modelos, com a correlação de Pearson mais próxima de zero.

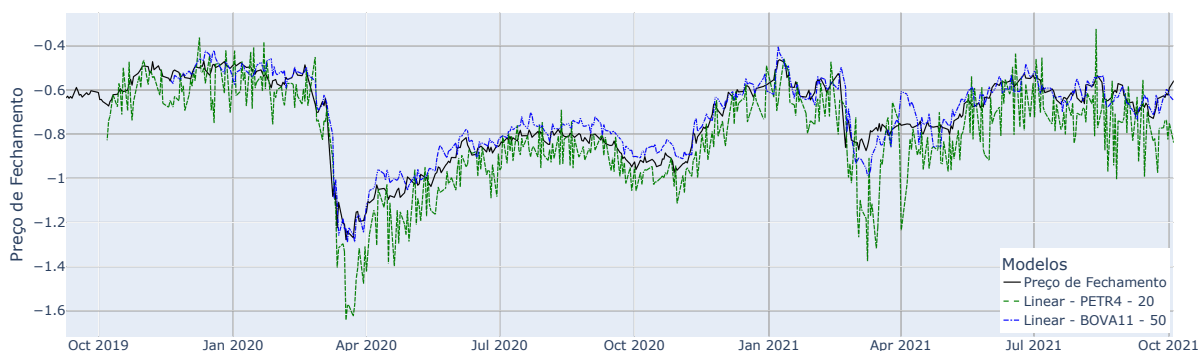
Por fim, os modelos presentes na [Figura 15](#) performaram previsões do preço de fechamento no conjunto de teste das ações da Petrobrás, em que as ações da Petrobrás são representadas pela linha preta normalizada pela técnica de *z-score* nas próximas figuras. Assim, na [Figura 21](#), vê-se que o modelo Linear-PETR4-20 apresentou uma previsão pior que a obtida pelo modelo RNR-PETR4-30, tendo em vista sua instabilidade ao longo do tempo. Contudo, quando o modelo Linear-PETR4-20 é comparado ao modelo Linear-BOVA11-50, vide [Figura 22](#), nota-se a melhoria na previsão ao se treinar o modelo linear com o auxílio do ETF BOVA11.

Figura 21 – Predição dos Modelos Linear-PETR4-20 e RNR-PETR4-30



Fonte: Autoria Própria

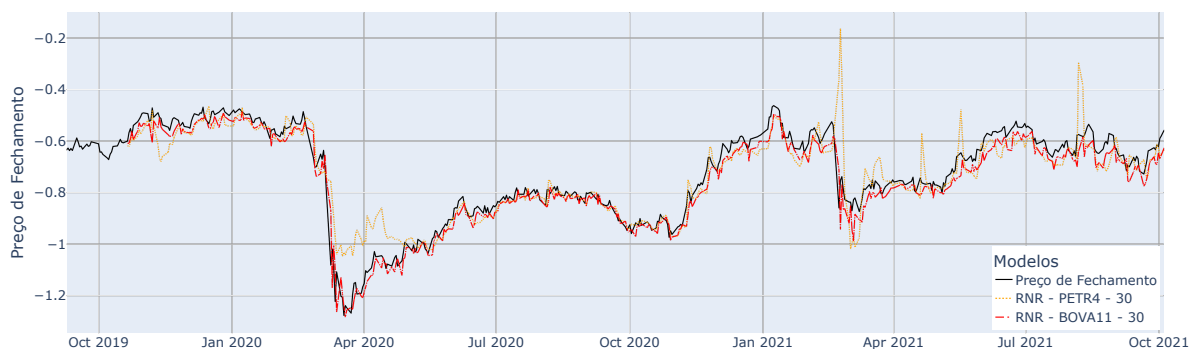
Figura 22 – Predição dos Modelos Linear-PETR4-20 e Linear-BOVA11-50



Fonte: Autoria Própria

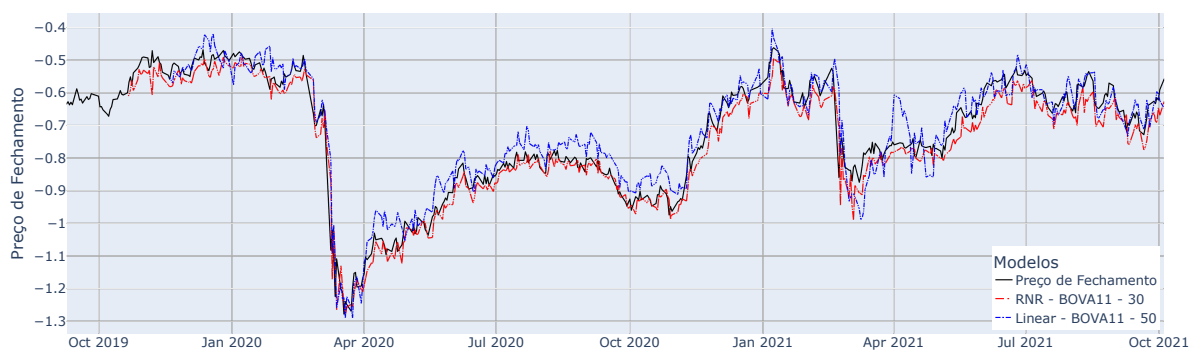
Outrossim, na [Figura 23](#), percebe-se que o modelo RNR-BOVA11-30 performou uma melhor predição se comparado ao modelo RNR-PETR4-30, especialmente em cenários extremos como o do início da pandemia do novo coronavírus em 2020. Enfim, comparando-se os modelos Linear-BOVA11-50 e RNR-BOVA11-30, vide [Figura 24](#), observa-se que o modelo proposto de RNR manteve-se mais próximo do preço de fechamento das ações da Petrobrás, o que justifica seu menor erro em comparação aos demais modelos.

Figura 23 – Predição dos Modelos RNR-PETR4-30 e RNR-BOVA11-30



Fonte: Autoria Própria

Figura 24 – Predição dos Modelos Linear-BOVA11-50 e RNR-BOVA11-30



Fonte: Autoria Própria

Por fim, no próximo capítulo, são sumarizados todos os resultados e discussões expostos neste capítulo. Além disso, o [Capítulo 6](#) apresenta as considerações gerais e as principais contribuições obtidas em detrimento das limitações e dificuldades enfrentadas. Por fim, tem-se as considerações finais, assim como os possíveis trabalhos futuros.

6 CONCLUSÃO

Este capítulo apresenta as considerações gerais, assim como as principais contribuições obtidas em detrimento das dificuldades e limitações enfrentadas. Por fim, o capítulo disserta sobre as considerações finais, como também sobre os possíveis trabalhos futuros a serem desenvolvidos a partir dos resultados obtidos por este trabalho de conclusão de curso.

6.1 Considerações Gerais

Neste trabalho de conclusão de curso, foram apresentadas diversas técnicas e métodos inerentes ao desenvolvimento de um modelo de predição. A priori, com a coleta, seleção e armazenamento dos dados, foi possível explorar tarefas relativas à engenharia de dados.

Além disso, com o pré-processamento e mineração dos dados, pôde-se lapidar os dados brutos armazenados em informações úteis, isto é, características relevantes para o treinamento do modelo de predição. Outrossim, tanto o contexto da bolsa de valores como a aplicação da Análise Técnica, estenderam o escopo deste trabalho de conclusão de curso para além do ensinado no curso de Engenharia de Computação.

Por fim, o treinamento do modelo de predição possibilitou a experimentação de diversas técnicas e configurações possíveis no que tange às Redes Neurais Recorrentes. Ainda, com os resultados obtidos pelo modelo, diferentes tipos de gráficos puderam ser elaborados.

6.2 Contribuições do Trabalho

Conforme anteriormente apresentado, todos os trabalhos correlatos utilizaram métodos complementares ao desenvolvimento de seus modelos de predição baseados em Aprendizado de Máquina. Contudo, ambos utilizaram um intervalo temporal de no máximo três anos, além de não testarem o treinamento de seus modelos de predição com empresas que formam uma carteira de ativos, tal como o ETF BOVA11.

Assim sendo, a seguir são destacadas as contribuições deste trabalho de conclusão de curso.

6.2.1 Principal Contribuição

A principal contribuição obtida pelo trabalho foi encontrar o melhor cenário de treinamento para o modelo proposto de RNR, utilizando um intervalo temporal compreendido entre 2001 e 2021. Assim, com os sete e oito experimentos realizados, pôde-se observar que os modelos obtiveram melhores predições do preço de fechamento das ações da Petrobrás quando treinados com o auxílio do ETF BOVA11, e não apenas com as ações da Petrobrás.

6.2.2 Contribuição Complementar

Como contribuição complementar, tem-se a criação dos *scripts* responsáveis por automatizar o processo de coleta, seleção e armazenamento das séries históricas da B3, o que até o momento não possui suporte oficial da B3.

6.3 Principais Limitações e Dificuldades

O trabalho encontrou quatro principais dificuldades durante sua realização. A primeira delas diz respeito à falta de suporte da B3 para a obtenção das séries históricas de ações via linguagem de programação. Isto é, o suporte ofertado pela B3 se restringe à importação das séries históricas para uma planilha do Excel, o que é feito de forma árdua, manual e repetitiva, gerando a necessidade do desenvolvimento de *scripts* para automatizar esse processo.

Já a segunda dificuldade enfrentada por este trabalho surge com o zelo por não enviar o modelo proposto de RNR durante seu treinamento. Para isso, a mineração de dados, baseada em Análise Técnica, foi aplicada separadamente para cada uma das empresas, e suas ações, que formam a carteira de ativos do ETF BOVA11, além da normalização *z-score* ser calculada sempre com base no subconjunto de treinamento de cada empresa. Por fim, todos os experimentos foram avaliados em um conjunto de teste nunca antes visto pelos modelos, proporcionando uma comparação justa.

Ademais, a terceira dificuldade dá-se com o treinamento e parametrização do modelo proposto de RNR. Redes Neurais possuem uma gama de possíveis parametrizações, que juntamente com um alto custo computacional de treinamento, torna o trabalho de ajustá-las ainda mais árduo, se comparado à parametrização de modelos clássicos.

Por fim, a quarta dificuldade diz respeito à obtenção de *tweets*, do Twitter, para fins de realização de Análise de Sentimentos, conforme previsto na proposta original deste trabalho de conclusão de curso, o que auxiliaria o modelo proposto de RNR. Embora o Twitter permita acesso à sua base de *tweets*, o acesso se limita a apenas sete dias no passado. Logo, para obter *tweets* históricos gratuitamente, porém, ainda com limitações, é necessário submeter uma solicitação que atenda a uma série de requisitos, tornando dispendioso o processo, cuja avaliação não se sabe como é realizada, nem quais critérios são adotados para permitir o acesso solicitado. Por outro lado, há versões de licenças que permitem realizar requisições ilimitadas às publicações da plataforma, entretanto, são pagas.

6.4 Considerações Finais

Com o trabalho, foi possível perceber a estabilidade e a boa capacidade de adaptação das Redes Neurais Recorrentes em cenários adversos. Isto é, em diferentes janelas temporais, o modelo proposto de RNR foi capaz de performar um baixo EAM, bem como um alto valor de R^2 , em que ambas as métricas apresentaram um baixo *overfitting*. O modelo proposto de

RNR também apresentou uma forte correlação de Pearson entre sua predição e o preço de fechamento das ações da Petrobrás.

Ademais, especialmente quando o modelo proposto de RNR foi treinado com o auxílio do ETF BOVA11, ao invés de apenas as ações da Petrobrás, pôde-se perceber a clara melhoria de performance em todas as métricas testadas para todas as janelas temporais.

Todavia, apesar do objetivo do modelo linear ser auxiliar na parametrização do modelo proposto de RNR, também foi possível notar uma melhor performance quanto às métricas supracitadas no cenário onde o modelo linear foi treinado com o auxílio do ETF BOVA11. Descartando-se o cenário de exceção da janela temporal de cinquenta dias, o modelo Linear-BOVA11, por exemplo, na janela temporal de quarenta dias quase obteve um EAM menor que o obtido pelo modelo proposto de RNR treinado apenas com as ações da Petrobrás.

6.5 Trabalhos Futuros

No que tange à possíveis trabalhos futuros, como destacado no [Capítulo 3](#), modelos apontados pela literatura como bons preditores para o mercado acionário poderiam ser comparados com o modelos proposto de RNR obtido por este trabalho de conclusão de curso.

Além disso, com a Análise de Sentimentos o modelo proposto de RNR poderia agregar em seu treinamento uma nova característica. Com isso, a Análise de Sentimentos poderia ser aplicada na etapa de mineração de dados com o objetivo de interpretar o sentimento causado por notícias que impactam no mercado acionário brasileiro.

Em vista disso, o Twitter, com seus *tweets*, serviria de base de dados para a Análise de Sentimentos. Não obstante, para isso se realizar, seria necessário um esforço da universidade na construção dessa colaboração com o Twitter, visando acesso ilimitado à sua base de *tweets* (com retroativo até 2006, quando foi fundado), de forma a permitir que seus docentes possam desenvolver suas pesquisas em tópicos tão diversos quanto as conversas no próprio Twitter.

Ademais, outras empresas presentes na B3, e que também formam a carteira de ativos do ETF BOVA11, poderiam servir de conjunto de teste para o modelo proposto de RNR. Desse modo, seria possível saber se em diversos cenários o modelo proposto de RNR treinado com o auxílio do ETF BOVA11 superaria os modelos treinados apenas com as ações específicas da empresa em análise.

Por fim, após todas as melhorias supraditas, seria interessante testar o modelo proposto de RNR em um cenário real com uma conta de teste em uma plataforma de *day trade* da B3. Desse modo, o modelo proposto de RNR, então, operaria na bolsa de valores, comprando e vendendo ações, a fim de validar seu potencial de lucro.

A seguir são listadas todas as referências utilizadas por este trabalho de conclusão de curso, essenciais para a dissertação do [Capítulo 2](#) e [Capítulo 3](#). Assim, com as referências pôde-se definir conceitos-chave, pesquisar trabalhos correlatos e, também, desenvolver o modelo proposto de RNR com o auxílio da literatura.

Referências

ABADI, M. et al. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>. Citado na página 12.

ACHELIS, S. B. **Technical Analysis from A to Z**. 2. ed. Nova Iorque, Nova Iorque, Estados Unidos: McGraw Hill Education, 2000. ISBN 9780071380119. Disponível em: <<https://books.google.com.br/books?id=XuiF-2eWHYUC>>. Citado 3 vezes nas páginas 1, 5 e 8.

AGRAWAL, S.; JINDAL, M.; PILLAI, G. Momentum analysis based stock market prediction using adaptive neuro-fuzzy inference system (anfis). In: **Proceedings of the international multiconference of engineers and computer scientists**. The Royal Garden Hotel, Kowloon, Hong Kong: Newswood Limited, 2010. v. 1, p. 526–531. Disponível em: <http://www.iaeng.org/publication/IMECS2010/IMECS2010_pp526-531.pdf>. Citado na página 8.

B3. **Investidor Pessoa Física**. B3, 2021. Acesso em 1 de Dezembro de 2021. Disponível em: <https://www.b3.com.br/pt_br/noticias/porcentagem-de-investidores-pessoa-fisica-cresce-na-b3.htm>. Citado na página 1.

B3. **Resumo das operações**. B3, 2021. Acesso em 27 de Novembro de 2021. Disponível em: <https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/consultas/mercado-de-derivativos/resumo-das-operacoes/estatisticas/>. Citado na página 1.

BOLLEN, J.; MAO, H.; ZENG, X. Twitter mood predicts the stock market. **Journal of computational science**, Elsevier, v. 2, n. 1, p. 1–8, 2011. ISSN 18777503. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S187775031100007X>>. Citado 3 vezes nas páginas 1, 23 e 25.

BRIOLA, A. **Keras/Tensorflow implementation of the Deep Adaptive Input Normalization Layer for Time Series Forecasting - First Release**. Zenodo, 2020. Disponível em: <<https://zenodo.org/record/4066525>>. Citado na página 13.

CAETANO, I. N.; MEINERZ, G. V.; SOUZA, E. F. de. Sentech: Um combinador de análise técnica e de sentimentos para o mercado de ações. In: **Tecnologias, métodos e teorias na engenharia de computação 2**. Ponta Grossa, Paraná, Brasil: Atena Editora, 2021. p. 24–33. ISBN 978-65-5706-945-5. Disponível em: <<https://www.atenaeditora.com.br/post-artigo/48442>>. Citado na página 30.

CHEN, J.; POTTERS, C. **Exponential Moving Average (EMA)**. Investopédia, 2021. Acesso em 3 de Dezembro de 2021. Disponível em: <<https://www.investopedia.com/terms/e/ema.asp>>. Citado na página 9.

CHEN, K.; ZHOU, Y.; DAI, F. A lstm-based method for stock returns prediction: A case study of china stock market. In: **2015 IEEE International Conference on Big Data (Big Data)**. Santa Clara, Califórnia, Estados Unidos: IEEE, 2015. p. 2823–2824. ISBN 9781479999262. Disponível em: <<https://doi.org/10.1109/bigdata.2015.7364089>>. Citado 3 vezes nas páginas 2, 24 e 25.

CHEN, X.-W.; LIN, X. Big data deep learning: Challenges and perspectives. **IEEE Access**, v. 2, p. 514–525, 2014. ISSN 21693536. Disponível em: <<https://ieeexplore.ieee.org/document/6817512>>. Citado na página 2.

COLBY, R. W.; MEYERS, T. A. **The encyclopedia of technical market indicators**. 2. ed. Nova Iorque, Nova Iorque, Estados Unidos: McGraw Hill Education, 2003. ISBN 9780070120570. Disponível em: <<https://books.google.com.br/books?id=SYWkGQawUgAC>>. Citado na página 8.

DAMIBA, J. et al. **Candlestick Charts in Python**. Plotly, 2021. Acesso em 23 de Fevereiro de 2021. Disponível em: <<https://plotly.com/python/candlestick-charts/>>. Citado na página 7.

DEBASTIANI, C. A.; RUSSO, F. A. **Avaliando empresas, investindo em ações: a aplicação prática da análise fundamentalista na avaliação de empresas**. Rua Luís Antônio dos Santos, 110, São Paulo, São Paulo, Brasil: Novatec Editora Ltda., 2017. ISBN 9788575225974. Disponível em: <<https://books.google.com.br/books?id=WX05DwAAQBAJ>>. Citado na página 9.

DU, K.-L.; SWAMY, M. N. S. Recurrent neural networks. In: **Neural Networks and Statistical Learning**. London: Springer London, 2013. p. 337–353. ISBN 9781447155713. Disponível em: <https://doi.org/10.1007/978-1-4471-5571-3_11>. Citado 2 vezes nas páginas 14 e 16.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification**. 2. ed. Hoboken, Nova Jérsea, Estados Unidos: John Wiley & Sons, Inc, 2012. ISBN 9781118586006. Disponível em: <<https://books.google.com.br/books?id=Br33IRC3PkQC>>. Citado na página 15.

EDWARDS, R. D.; MAGEE, J.; BASSETTI, W. C. **Technical analysis of stock trends**. 11. ed. Nova Iorque, Nova Iorque, Estados Unidos: CRC press, 2018. ISBN 9781351631433. Disponível em: <<https://books.google.com.br/books?id=cVJmDwAAQBAJ>>. Citado 3 vezes nas páginas 1, 5 e 7.

FAMA, E. F. Efficient capital markets: A review of theory and empirical work. **The Journal of Finance**, [American Finance Association, John Wiley & Sons, Inc], v. 25, n. 2, p. 383–417, 1970. ISSN 00221082, 15406261. Disponível em: <<http://www.jstor.org/stable/2325486>>. Citado na página 10.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI Magazine**, v. 17, n. 3, p. 37, Mar. 1996. Disponível em: <<https://ojs.aaai.org/index.php/aimagazine/article/view/1230>>. Citado na página 11.

FERNANDO, J.; POTTERS, C.; MUNICHIELLO, K. **Moving Average Convergence Divergence (MACD)**. Investopedia, 2021. Acesso em 4 de Novembro de 2021. Disponível em: <<https://www.investopedia.com/terms/m/macd.asp>>. Citado na página 9.

FROST, A. J.; PRECHTER, R. R. **Elliott Wave Principle: Key to Market Behavior**. 7. ed. Gainesville, Geórgia, Estados Unidos: New Classics Library, 1995. ISBN 9780932750235. Disponível em: <<https://books.google.com.br/books?id=KurtAAAAMAAJ>>. Citado 2 vezes nas páginas 10 e 11.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. Rodovia Gravenstein, 1005, Sebastopol, Califórnia, Estados Unidos: O'Reilly Media, 2017. ISBN 9781491962268. Disponível em:

<<https://books.google.com.br/books?id=khpYDgAAQBAJ>>. Citado 10 vezes nas páginas 2, 13, 14, 15, 16, 17, 18, 19, 20 e 21.

GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: TEH, Y. W.; TITTERINGTON, M. (Ed.). **Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics**. Chia Laguna Resort, Sardinia, Itália: PMLR, 2010. (Proceedings of Machine Learning Research, v. 9), p. 249–256. Disponível em: <<http://proceedings.mlr.press/v9/glorot10a.html>>. Citado na página 19.

GOMES, F. R. A bolsa de valores brasileira como fonte de informações financeiras. **Perspectivas em ciência da informação**, v. 2, n. 2, 1997. Citado 2 vezes nas páginas 1 e 4.

GOODFELLOW, I. et al. **Deep learning**. 1. ed. Londres, Inglaterra: MIT Press, 2016. v. 1. (Adaptive Computation and Machine Learning series, v. 1). ISBN 9780262035613. Disponível em: <<https://books.google.com.br/books?id=Np9SDQAAQBAJ>>. Citado 5 vezes nas páginas 13, 17, 18, 19 e 21.

GOTTWALD, R. The dependence between stock price and intrinsic value of a stock. **Ad Alta: Journal of Interdisciplinary Research**, v. 1, p. 38–40, 2011. Disponível em: <<http://www.magnanimitas.cz/ADALTA/0101/papers/gottwald.pdf>>. Citado na página 9.

HAWKINS, D. M. The problem of overfitting. **Journal of Chemical Information and Computer Sciences**, ACS Publications, v. 44, n. 1, p. 1–12, 2004. PMID: 14741005. Disponível em: <<https://doi.org/10.1021/ci0342472>>. Citado na página 18.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. Avenida Angélica, 1091, Higienópolis, São Paulo, Brasil: Artmed, 2007. ISBN 9788577800865. Disponível em: <<https://books.google.com.br/books?id=bhMwDwAAQBAJ>>. Citado 3 vezes nas páginas 14, 15 e 16.

HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: **2015 IEEE International Conference on Computer Vision (ICCV)**. Los Alamitos, Califórnia, Estados Unidos: IEEE Computer Society, 2015. p. 1026–1034. ISSN 23807504. Disponível em: <<https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.123>>. Citado na página 20.

HEGAZY, O.; SOLIMAN, O. S.; SALAM, M. A. A machine learning model for stock market prediction. **International Journal of Computer Science and Telecommunications**, abs/1402.7351, 2013. Disponível em: <<http://arxiv.org/abs/1402.7351>>. Citado 3 vezes nas páginas 2, 23 e 25.

HU, Y. et al. Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. **Applied Soft Computing**, v. 36, p. 534–551, 2015. ISSN 15684946. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S156849461500438X>>. Citado 2 vezes nas páginas 8 e 9.

IDC; SEAGLE. **Data age 2025: the digitization of the worlds, from edge to core**. 2018. Disponível em: <<https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>>. Citado na página 1.

INFOMONEY. **ETFs (BOVA11)**. 2021. Acesso em 6 de Novembro de 2021. Disponível em: <<https://www.infomoney.com.br/cotacoes/etf-bova11/>>. Citado na página 28.

INFOMONEY. **O que são ações?** InfoMoney, 2021. Acesso em 27 de Novembro de 2021. Disponível em: <<https://www.infomoney.com.br/guias/como-investir-em-acoes/>>. Citado 2 vezes nas páginas 1 e 4.

INFOMONEY. **O que são fundos imobiliários?** InfoMoney, 2021. Acesso em 27 de Novembro de 2021. Disponível em: <<https://www.infomoney.com.br/guias/fundos-imobiliarios/>>. Citado na página 5.

INFOMONEY. **O que são opções?** InfoMoney, 2021. Acesso em 27 de Novembro de 2021. Disponível em: <<https://www.infomoney.com.br/guias/mercado-de-opcoes/>>. Citado na página 4.

INFOMONEY. **O que é um ETF?** InfoMoney, 2021. Acesso em 27 de Novembro de 2021. Disponível em: <<https://www.infomoney.com.br/guias/etf-exchange-traded-fund/>>. Citado na página 5.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: BACH, F.; BLEI, D. (Ed.). **Proceedings of the 32nd International Conference on Machine Learning**. Lille, França: PMLR, 2015. (Proceedings of Machine Learning Research, v. 37), p. 448–456. Disponível em: <<http://proceedings.mlr.press/v37/ioffe15.html>>. Citado na página 20.

KEY, B. et al. **Largest stock exchange operators worldwide as of September 2021, by market capitalization of listed companies**. Statista, 2021. Acesso em 27 de Novembro de 2021. Disponível em: <<https://www.statista.com/statistics/270126/largest-stock-exchange-operators-by-market-capitalization-of-listed-companies/>>. Citado na página 5.

KHAN, N. et al. Big data: survey, technologies, opportunities, and challenges. **The Scientific World Journal**, Hindawi Publishing Corporation, v. 2014, 2014. Disponível em: <<https://doi.org/10.1155/2014/712826>>. Citado na página 1.

KIRKPATRICK, C.; DAHLQUIST, J. **Technical Analysis: The Complete Resource for Financial Market Technicians**. 2. ed. Upper Saddle River, Bergen County, Nova Jérsea, Estados Unidos: Pearson Education, 2010. ISBN 9780132599627. Disponível em: <<https://books.google.com.br/books?id=l5SgX5q5sQEC>>. Citado 2 vezes nas páginas 7 e 8.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Commun. ACM**, Association for Computing Machinery, Nova Iorque, Nova Iorque, Estados Unidos, v. 60, n. 6, p. 84–90, maio 2017. ISSN 00010782. Disponível em: <<https://doi.org/10.1145/3065386>>. Citado na página 18.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Disponível em: <<https://doi.org/10.1038/nature14539>>. Citado na página 18.

LOC, T. **Equitisation and stock-market development: the case of vietnam**. Tese (Doutorado) — University of Groningen, 2006. Disponível em: <<https://research.rug.nl/en/publications/equitisation-and-stock-market-development-the-case-of-vietnam>>. Citado na página 10.

- MADER, H. et al. **Statistics in Volcanology**. Londres, Inglaterra: Geological Society, 2006. (Special Publications of the International Association of Vol). ISBN 9781862392083. Disponível em: <https://books.google.com.br/books?id=e5Y_RRPxdyYC>. Citado na página 12.
- MAHALAKSHMI, G.; SRIDEVI, S.; RAJARAM, S. A survey on forecasting of time series data. In: **2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)**. IEEE, 2016. Disponível em: <<https://doi.org/10.1109/icctide.2016.7725358>>. Citado na página 14.
- MALKIEL, B. **A Random Walk Down Wall Street: Including a Life-cycle Guide to Personal Investing**. Quinta Avenida, 500, Nova Iorque, Nova Iorque, Estados Unidos: Norton, 1999. ISBN 9780393047813. Disponível em: <<https://books.google.com.br/books?id=v8ENTFP29tkC>>. Citado na página 10.
- MEDSKER, L.; JAIN, L. **Recurrent Neural Networks: Design and Applications**. Avenida Corporate N.W., 2000, Boca Raton, Flórida, Estados Unidos: CRC Press, 1999. (International Series on Computational Intelligence). ISBN 9781420049176. Disponível em: <<https://books.google.com.br/books?id=ME1SAkN0PyMC>>. Citado na página 18.
- MURPHY, J. **Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications**. 1. ed. Strand, 80, Londres, Inglaterra: Penguin Publishing Group, 1999. ISBN 9781101659199. Disponível em: <<https://books.google.com.br/books?id=teitAAAQBAJ>>. Citado 2 vezes nas páginas 6 e 7.
- NIELSEN, A. **Practical Time Series Analysis: Prediction with Statistics and Machine Learning**. Rodovia Gravenstein, 1005, Sebastopol, Califórnia, Estados Unidos: O'Reilly Media, 2019. ISBN 9781492041627. Disponível em: <<https://books.google.com.br/books?id=odCwDwAAQBAJ>>. Citado na página 2.
- PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. In: DASGUPTA, S.; MCALLESTER, D. (Ed.). **Proceedings of the 30th International Conference on Machine Learning**. Atlanta, Geórgia, Estados Unidos: PMLR, 2013. (Proceedings of Machine Learning Research, 3), p. 1310–1318. Disponível em: <<http://proceedings.mlr.press/v28/pascanu13.html>>. Citado na página 20.
- PASSALIS, N. et al. **Deep Adaptive Input Normalization for Time Series Forecasting**. 2019. Citado 2 vezes nas páginas 12 e 13.
- PASUPULETY, U. et al. Predicting stock prices using ensemble learning and sentiment analysis. In: **2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)**. Sardinia, Itália: IEEE, 2019. p. 215–222. ISBN 9781728114880. Disponível em: <<https://ieeexplore.ieee.org/document/8791689>>. Citado 3 vezes nas páginas 2, 24 e 25.
- PEACHAVANISH, R. Stock selection and trading based on cluster analysis of trend and momentum indicators. In: AO, S. I. et al. (Ed.). **Proceedings of the International Multi-Conference of Engineers and Computer Scientists**. The Royal Garden Hotel, Kowloon, Hong Kong: Newswood Limited, 2016. v. 1, p. 317–321. ISBN 9789881925381. ISSN 20780966. Disponível em: <http://www.iaeng.org/publication/IMECS2016/IMECS2016_pp317-321.pdf>. Citado na página 8.

- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Disponível em: <<https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>>. Citado na página 13.
- POSHAKWALE, S. Evidence on weak form efficiency and day of the week effect in the indian stock market. **Finance India**, v. 10, n. 3, p. 605–616, 1996. Disponível em: <<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.470.3815>>. Citado na página 10.
- RUDER, S. An overview of gradient descent optimization algorithms. **CoRR**, abs/1609.04747, 2016. Disponível em: <<http://arxiv.org/abs/1609.04747>>. Citado 2 vezes nas páginas 18 e 19.
- Samuel, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210–229, 1959. Disponível em: <<https://doi.org/10.1147/rd.33.0210>>. Citado na página 13.
- SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural Networks**, v. 61, p. 85–117, 2015. ISSN 08936080. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0893608014002135>>. Citado na página 14.
- SOLA, J.; SEVILLA, J. Importance of input data normalization for the application of neural networks to complex industrial problems. **IEEE Transactions on Nuclear Science**, Institute of Electrical and Electronics Engineers (IEEE), v. 44, n. 3, p. 1464–1468, jun. 1997. Disponível em: <<https://doi.org/10.1109/23.589532>>. Citado na página 12.
- SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, n. 56, p. 1929–1958, 2014. Disponível em: <<http://jmlr.org/papers/v15/srivastava14a.html>>. Citado na página 21.
- SZE, V. et al. Efficient processing of deep neural networks: A tutorial and survey. **Proceedings of the IEEE**, v. 105, n. 12, p. 2295–2329, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8114708>>. Citado na página 2.
- VIRTANEN, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in python. **Nature Methods**, Springer Science and Business Media LLC, v. 17, n. 3, p. 261–272, fev. 2020. Disponível em: <<https://doi.org/10.1038/s41592-019-0686-2>>. Citado na página 13.
- WILLIAMS, G.; HUANG, Z. Modelling the kdd process a four stage process and four element model. 07 1996. Disponível em: <https://www.researchgate.net/publication/242376693_Modelling_the_KDD_Process_A_Four_Stage_Process_and_Four_Element_Model>. Citado 3 vezes nas páginas 11, 12 e 27.