

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MATHEUS QUEVEDO SIVELLI

**AVALIAÇÃO DE REDES NEURAIS DO TIPO LONG SHORT-TERM MEMORY E
MULTILAYER PERCEPTRON PARA PREDIÇÃO DO VALOR DA COTAÇÃO DAS
AÇÕES DAS CINCO EMPRESAS MAIS REPRESENTATIVAS DO IBOVESPA**

Porto Alegre

2022

MATHEUS QUEVEDO SIVELLI

**AVALIAÇÃO DE REDES NEURAIS DO TIPO LONG SHORT-TERM MEMORY E
MULTILAYER PERCEPTRON PARA PREDIÇÃO DO VALOR DA COTAÇÃO DAS
AÇÕES DAS CINCO EMPRESAS MAIS REPRESENTATIVAS DO IBOVESPA**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul como parte dos requisitos para obtenção do título de Engenheiro Elétrico.

Professor Orientador: Dr. Tiago Oliveira Weber

Porto Alegre,

2022

MATHEUS QUEVEDO SIVELLI

**AVALIAÇÃO DE REDES NEURAIS DO TIPO LONG SHORT-TERM MEMORY E
MULTILAYER PERCEPTRON PARA PREDIÇÃO DO VALOR DA COTAÇÃO DAS
AÇÕES DAS CINCO EMPRESAS MAIS REPRESENTATIVAS DO IBOVESPA**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul como parte dos requisitos para obtenção do título de Engenheiro Elétrico.

Porto Alegre, 18 de Outubro de 2022.

Banca examinadora:

Prof. Dr. Tiago Oliveira Weber

Prof. Dr. Alexandre Balbinot

Prof. Dr. Rafael Mozart da Silva

AGRADECIMENTOS

Agradeço a minha mãe, Solange, por ser a minha maior referência na vida. Por todo o apoio e por ter me ensinado tudo. Te amo.

Agradeço ao meu irmão, Bruno, por ser uma pessoa de um coração gigante e sempre estar disposto a ajudar todo mundo. Te amo.

Agradeço a minha namorada, Stephanie, por todo apoio e paciência que teve durante esse processo. Te amo.

Por fim, agradeço ao meu orientador, Tiago, por todas as diretrizes e conversas confortantes que tivemos durante este caminho.

RESUMO

O mercado de capitais é um mecanismo que possibilita o financiamento de empresas através de pessoas físicas poupadoras. Este instrumento assume um papel ímpar no desenvolvimento econômico de um país e possibilita a construção de patrimônio a longo prazo para os investidores. Entender e conseguir prever o comportamento do mercado é de grande valia para investidores. O presente trabalho utilizou técnicas de *Multilayer Perceptron* e *Long Short-Term Memory* para prever o preço futuro das cinco empresas mais representativas do Ibovespa, que é principal índice do mercado brasileiro. Para construção dos algoritmos, utilizou-se a linguagem *Python 3* e o ambiente em nuvem *Google Collaboratory*. Os modelos foram treinados e validados utilizando dados de 04 janeiro de 2010 a 05 de agosto de 2021 e testados com dados de 06 agosto até 30 de dezembro de 2021. Para encontrar o melhor conjunto de hiperparâmetros foi realizado uma otimização considerando 72 combinações distintas. Os resultados foram avaliados através do *Root Mean Squared Error* e demonstraram que as *Multilayer Perceptron* obtiveram um erro, na média, menor quando comparada as *Long Short-Term Memory*, e, portanto, foram consideradas a técnica mais segura para os investidores. Os resultados de RMSE obtidos para as empresas B3, Bradesco, Itaú, Petrobras e Vale, respectivamente, foram de 0,37470; 0,44970; 0,69881; 0,64238; e 2,00160 para MLP contra 0,36930; 0,47892; 0,74098; 0,65798; e 2,4009 para a LSTM. Por fim, os melhores resultados foram: B3 (0,36930), Bradesco (0,44970), Petrobras (0,64238), Itaú (0,69881) e Vale (2,00160).

Palavras-chaves: *Multilayer Perceptron*, *Long Short-Term Memory*, Mercado de Capitais, Ibovespa.

ABSTRACT

The capital market is a mechanism that makes it possible to finance companies through savings of individuals person. This instrument plays a unique role in the economic development of a country and makes it possible to build long-term equity for investors. Understand and be able to predict market behavior is of great value to any investors. The present work used Multilayer Perceptron and Long Short-Term Memory techniques to predict the future value of the five most representative companies on the Ibovespa, which is the main index of the Brazilian market. To build the algorithms, it was used Python 3 language and the Google Collaboratory cloud environment. The models were trained and validated using data from January 4, 2010 to August 5, 2021 and tested with data from August 6 to December 30, 2021. To find the best set of hyperparameters, an optimization was performed considering 72 different topologies. The results were evaluated using the Root Mean Squared Error and showed that the Multilayer Perceptron obtained an error, on average, lower when compared to the Long Short-Term Memory, and therefore were considered the safest technique for investors. The RMSE results obtained for companies B3, Bradesco, Itaú, Petrobras and Vale, respectively, were 0,37470; 0,44970; 0,69881; 0,64238; and 2,00160 for MLP versus 0,36930; 0,47892; 0,74098; 0,65798; and 2,4009 for the LSTM. Finally, the best results were: B3 (0,36930), Bradesco (0,44970), Petrobras (0,64238), Itaú (0,69881) and Vale (2,00160).

Keywords: *Multilayer Perceptron, Long Short-Term Memory, Capital Market, Ibovespa.*

LISTA DE FIGURAS

Figura 1 - Classificação versus Regressão.	19
Figura 2 – Tipos de aprendizado.	22
Figura 3 - Árvore de decisão.	23
Figura 4 - Modelo de um neurônio.	25
Figura 5 - Funções de ativação linear, escada e sigmoideal.	26
Figura 6 - Arquitetura das redes neurais.	27
Figura 7 - Problema linearmente separável.	28
Figura 8 - Problema não linearmente separável.	29
Figura 9 - <i>Multilayer Perceptron</i>	30
Figura 10 - Efeito de uma taxa de aprendizado mal dimensionada.	33
Figura 11 - Topologia de uma rede neural recorrente.	35
Figura 12 - Configuração da célula.	36
Figura 13 - Estrutura de uma <i>Stacked LSTM</i>	37
Figura 14 - Estrutura de uma rede bidirecional.	38
Figura 15 - Representação do cross-validation.	39
Figura 16 – Fluxograma descritivo do método utilizado.	43
Figura 17 - Ilustração do lagging.	48
Figura 18 - Divisão do conjunto de dados.	49
Figura 19 - Diagrama dos testes aplicados a base de dados.	50
Figura 20 - Etapas de como extrair o ganho de um atributo.	52
Figura 21 - Etapas da otimização de hiperparâmetros.	55
Figura 22 - Diagrama dos testes realizados.	61
Figura 23 – Valor da ação ao longo do tempo.	65
Figura 24 - Distribuição do retorno diário de cada empresa.	66
Figura 25 - Correlação entre os atributos.	69
Figura 26 - Importância dos atributos pelo <i>XGBoost</i> (em percentual).	71
Figura 27 - Comportamento do erro por <i>lagging</i> na MLP (em reais – R\$).	73
Figura 28 - Comportamento do erro por <i>lagging</i> na LSTM (em reais – R\$).	75
Figura 29 - Histograma das diferentes combinações pelo erro no conjunto de validação no modelo generalista (<i>Multilayer perceptron</i>).	77
Figura 30 - Histograma das diferentes combinações pelo erro no conjunto de validação no modelo especialista (<i>Multilayer perceptron</i>).	79

Figura 31 - Histograma das diferentes combinações pelo erro no conjunto de validação no modelo generalista (<i>Long Short-Term Memory</i>).	82
Figura 32 - Histograma das diferentes combinações pelo erro no conjunto de validação no modelo especialista (<i>Long Short-Term Memory</i>).	84
Figura 33 - RMSE por tipo de modelo e empresa, em reais (R\$).	89
Figura 34 - Valores preditos <i>versus</i> Valores reais para a MLP – em reais (R\$).	90
Figura 35 - Erro percentual da predição para <i>Multilayer Perceptron</i>	91
Figura 36 - RMSE por tipo de modelo e empresa para as LSTMs.....	95
Figura 37 - Valores preditos <i>versus</i> Valores reais para a LSTM – em reais (R\$).....	96
Figura 38 – Distribuição do erro percentual de predição para LSTM.....	97

LISTA DE QUADROS

Quadro 1 - Descritivo dos atributos de entrada.	45
Quadro 2 - Atributo de saída.	45
Quadro 3 - Bibliotecas e frameworks utilizados.	46
Quadro 4 - Otimizações e testes preliminares.	50
Quadro 5 - Modelo avaliado tanto na MLP quanto na LSTM.....	53
Quadro 6 - Valores avaliados dos hiperparâmetros da MLP.....	53
Quadro 7 - Diferença entre modelos considerados.....	53
Quadro 8 - Valores avaliados dos hiperparâmetros da MLP.....	55
Quadro 9 - Algumas combinações de hiperparametros avaliadas para MLP.....	56
Quadro 10 – Cenários de otimizações para MLP e LSTM.....	57
Quadro 11 - Arquiteturas avaliadas da LSTM.....	57
Quadro 12 - Valores avaliados dos hiperparâmetros da LSTM.	58
Quadro 13 – Algumas das combinações de hiperparâmetros avaliadas na LSTM.	59
Quadro 14 – Avaliação sob a perspectiva generalista para cada técnica.....	61
Quadro 15 - Avaliação sob a perspectiva especialista para cada técnica.	62
Quadro 16 - Margens de seguranças estipuladas pelo investidor.	63
Quadro 17 - Exemplo de atuação por tipo de investidor.	63

LISTA DE TABELAS

Tabela 1 - Empresas selecionadas para serem utilizadas neste estudo.....	43
Tabela 2 - Tamanho das bases de dados por empresa.	44
Tabela 3 - Comportamento das variáveis de interesse.	65
Tabela 4 - Ocorrência dos retornos diários propostos.	67
Tabela 5 - Importância de atributos pelo <i>XGBoost</i> (em termos percentuais).....	70
Tabela 6 - Resultados obtidos pela MLP.....	72
Tabela 7 - Resultados obtidos para cada empresa para MLP.	74
Tabela 8 - Resultados obtidos pela LSTM.	74
Tabela 9 - Resultados obtidos para cada empresa para LSTM.....	75
Tabela 10 - Cinco melhores combinações para modelo generalista para MLP.....	77
Tabela 11 - Resultados obtidos para os modelos especialistas da MLP.....	80
Tabela 12 - Cinco melhores combinações para modelo generalista para LSTM.	82
Tabela 13 - Resultados obtidos para os modelos especialistas da LSTM.	85
Tabela 14 - Combinações de hiperparâmetros selecionadas da MLP.	86
Tabela 15 - Combinações de hiperparâmetros selecionadas da LSTM.....	86
Tabela 16 - Resultados para os modelos na atuação generalista.	87
Tabela 17 - Resultados dos modelos na atuação especialista para MLP.....	88
Tabela 18 - Resultados obtidos para as MLPs.....	89
Tabela 19 - Recomendações de compra por perfil em cada cenário para MLP.	92
Tabela 20 - Retorno sobre investimento por perfil de investidor avaliado.	92
Tabela 21 - Retorno sobre investimento para um investidor <i>Buy and Hold</i>	93
Tabela 22 - Resultados para os modelos na atuação generalista para LSTM.....	94
Tabela 23 - Resultados para os modelos na atuação generalista para LSTM.....	94
Tabela 24 - Resultados obtidos para as LSTMs.	96
Tabela 25 - Recomendações de compra por perfil em cada cenário para LSTM.....	98
Tabela 26 - Retorno sobre investimento obtido por perfil de investidor.	98

LISTA DE ABREVIATURAS E SIGLAS

CNN – Convolutional Neural Network

DNN – Deep Neural Network

IPO – Initial Public Offer

LSTM – Long Short-Term Memory

MAE – Mean Absolute Error

MLP – Multilayer Perceptron

MSE – Mean Squared Error

RMSE – Root Mean Squared Error

RNN – Recurrent Neural Network

SGD – Stochastic Gradient Descent

SP & 500 – Standard & Poor 500

XGBOOST – Extreme Gradient Boosting

SUMÁRIO

1. INTRODUÇÃO	13
1.1 Objetivo geral	15
1.2 Objetivos específicos	15
2. REFERENCIAL TEÓRICO	16
2.1 Mercado Financeiro	16
2.2 Previsão de séries temporais financeiras	18
2.3 Teoria de Mercado	20
2.4 Aprendizado de Máquina	20
2.4.1 <i>XGBoost</i>	22
2.4.2 Redes Neurais Artificiais	24
2.4.3 <i>Multilayer Perceptron</i>	27
2.5 Aprendizagem da rede	30
2.5.1 <i>Backpropagation</i>	31
2.5.2 <i>Stochastic Gradient Descent</i>	31
2.5.3 <i>Adam</i>	33
2.6 <i>Deep Neural Networks</i>	34
2.6.1 <i>Recurrent Neural Network</i>	34
2.6.2 <i>Long Short-Term Memory Network</i>	35
2.6.3 <i>Stacked LSTM</i>	36
2.6.4 Bidirecional LSTM	37
2.7 Tratamento e divisão do conjunto de dados	38
2.8 Normalização	40
2.9 Métrica de erro	41
3. METODOLOGIA DE PESQUISA E PROCEDIMENTOS METODOLÓGICOS	42
3.1 Seleção e extração da base de dados	44
3.2 Ferramentas Computacionais	45
3.3 Análise da base de dados	47
3.4 Pré-processamento da base de dados	47
3.4.1 Normalização	47
3.4.2 <i>Lagging</i>	48
3.4.3 Divisão do conjunto de dados	49
3.5 Seleção de atributos e testes preliminares na base de dados	49
3.5.1 Análise de correlação	51
3.5.2 Análise de importância de atributos por <i>XGBoost</i>	51

3.5.3	Análise do <i>lagging</i> e atributos nas técnicas selecionadas	52
3.6	Otimização de hiperparâmetros	54
3.6.1	Otimização MLP	55
3.6.2	Otimização LSTM	57
3.7	Seleção dos hiperparâmetros	60
3.8	Avaliação de desempenho	60
3.8.1	Avaliação quantitativa	60
3.8.2	Avaliação simulada	62
4.	RESULTADOS	64
4.1	Análise da base de dados	64
4.2	Seleção de atributos e testes preliminares	68
4.2.1	Análise de correlação dos atributos da base de dados	68
4.2.2	<i>XGBoost</i> para importância de atributos	69
4.2.3	Seleção de atributos e <i>lagging</i>	72
4.3	Otimização de Hiperparâmetros	76
4.3.1	Otimização de hiperparâmetros para <i>Multilayer Perceptron</i>	76
4.3.2	Otimização de hiperparâmetros para <i>Long Short-Term Memory</i>	81
4.4	Seleção dos conjuntos de hiperparâmetros	85
4.5	Avaliação de desempenho	87
4.5.1	Avaliação Quantitativa - <i>Multilayer Perceptron</i>	87
4.5.2	Avaliação Simulada - <i>Multilayer Perceptron</i>	91
4.5.3	Avaliação Quantitativa - <i>Long Short-Term Memory</i>	93
4.5.4	Avaliação Simulada - <i>Long Short-Term Memory</i>	98
5.	CONCLUSÃO	100
	REFERÊNCIAS	103
	APÊNDICE A – Combinações possíveis da otimização para MLP	109
	APÊNDICE B – Combinações possíveis da otimização na LSTM	112

1. INTRODUÇÃO

O principal objetivo do mercado financeiro é conectar dois grupos de indivíduos: os poupadores e os tomadores (ASSAF NETO, 2017). Para isso, se faz necessário o uso de diversos mecanismos como, por exemplo, o mercado de capitais que é o braço responsável por unificar empresas tomadoras de crédito com pessoas físicas poupadoras de dinheiro.

Assim sendo, o mercado de capitais permite que as empresas que necessitam de subsídio financeiro tenham acesso a uma capitalização de menor custo e de longo prazo, além de distribuir o risco de tal investimento entre todos os investidores, denominados como acionistas (quem detém ações de tal empresa). Althelay, El-Alfy e Mohammed (2018) destaca que este mecanismo é uma das melhores opções para investidores gerarem patrimônio.

Nesse sentido, entender e utilizar técnicas avançadas de modelagem para prever o valor futuro das ações é de grande valia para investidores. Assim sendo, Li *et al.*, (2017) explicam que atingir tal resultado torna-se desafiador, uma vez que os dados advindos das negociações são séries temporais altamente dinâmicas, não lineares, não parametrizáveis e caóticas.

Diante desse contexto, Sezer, Gudelek e Ozbayoglu (2020) descrevem que técnicas de *machine learning* obtêm melhores resultados do que técnicas tradicionais de econometria. Além disso, técnicas de *deep learning* emergem como os métodos com maior desempenho, em potencial, para diversos campos, inclusive o financeiro.

Sistemas de *machine learning* são limitados pela falta de habilidade em processar dados brutos e, conseqüentemente, necessitam um tratamento prévio da base de dados. Em contraste a isto, *deep learning* são técnicas que não necessitam desse tratamento e conseguem aprender através de dados brutos, assim sendo, não dependem da interferência humana e, conseqüentemente, possibilitam uma alta abstração da base de dados (LECUN, BENGIO e HINTON, 2015).

Segundo Sezer, Gudelek e Ozbayoglu (2020), existem diferentes tipos de técnicas de *deep learning*, porém, dentro de cada peculiaridade, podem ser resumidas em redes neurais com múltiplas camadas de processamento. Neste cenário, as *Multilayer Perceptrons* são redes neurais do tipo *feedforward* que podem, ou não, possuir mais de uma camada oculta. Já as redes neurais recorrentes são do tipo *backward* e fazem o uso de uma retroalimentação para preservar uma possível relação de dependência entre os dados que são alimentadas para as mesmas, comumente visto em problemas de séries temporais.

As inovações propostas pelas redes neurais recorrentes trouxeram o problema do *vanishing gradient*, tal fenômeno era caracterizado pela ausência de memória da rede, ou seja, a incapacidade de manter informações de um passado distante que eram pertinentes na predição de um valor atual.

Para resolver tal problema, Schmidhuber e Hochreiter (1997) propuseram as *Long Short-Term Memory* (LSTM). As LSTMs se destacam das redes recorrentes por possuírem unidades de entrada diferentes de neurônios tradicionais. Essas entradas são denominadas células e possuem três elementos: *input gate*, *forget gate* e *output gate*. Em resumo, Lee e Yoo (2018) explicam que esses elementos atuam para preservar a característica sequencial dos dados para um longo prazo, ou seja, informações relevantes mesmo que no início da sequência são mantidas para avaliações posteriores.

Essas técnicas também foram abordadas em trabalhos como é o caso de Althelay, El-Alfy e Mohammed (2018), Li *et al.*, (2017), Lee Yoo (2018) e Kraus *et al.* (2017), onde todos os estudos avaliaram modelos de *deep learning* aplicados aos dados históricos do SP & 500, com a exceção do último estudo, que adicionou uma avaliação sob a ótica de árvores de decisões.

Por sua vez, existem estudos que aplicaram tais técnicas fora do escopo do SP & 500, como é o caso de Hiransha *et al.* (2018), que faz uso dos métodos de *Multilayer Perceptron* (MLP), *Recurrent Neural Network* (RNN), *Long Short-Term Memory* (LSTM) e *Convolutional Neural Network* (CNN) para prever o preço de empresas presentes no mercado indiano e norte-americano. Na abordagem de Samarawickrama e Fernando (2017), aplicou-se derivações das RNNs nas ações negociadas no Sri Lanka. Por fim, Chandra e Chand (2016) faz o uso apenas do preço de fechamento das ações para prever o valor futuro das empresas listadas na NASDAQ.

Neste contexto, o presente trabalho teve como objetivo avaliar as técnicas de *Deep Learning*, *Multilayer Perceptrons* e *Long Short-Term Memory* para as cinco empresas mais representativas do IBOVESPA. Além disso, buscou-se avaliar se os modelos construídos com dados de todas as empresas apresentaram um desempenho superior quando comparado a modelos que utilizaram dados de apenas uma empresa.

Para atingir tal objetivo, foram propostas abordagens distintas das que foram vistas na literatura. A primeira foi obter a melhor base de dados para cada técnica. A segunda consistiu-se em obter o melhor valor de *lagging* (técnica para atrasar valores no tempo) para cada empresa e para todas em conjuntos, caracterizando assim, os modelos que foram considerados

especialistas e generalistas. Já a terceira abordagem, buscou, para os cenários de especialista e generalista, expandir a procura pelos melhores hiperparâmetros de cada técnica - *Long Short-Term Memory* e *Multilayer Perceptron*. Por fim, foi proposto um método simulado para avaliar a aplicabilidade dos modelos. Os dados utilizados foram extraídos do *Yahoo Finance* no período de 04/01/2010 até 30/12/2021.

O trabalho foi dividida em cinco capítulos. No primeiro capítulo apresenta-se a introdução, contexto da pesquisa e os objetivos do trabalho. No segundo capítulo apresenta-se o referencial teórico que serviu de base para o desenvolvimento da pesquisa, o qual compreende os conceitos básicos de mercado financeiro, além de introduzir o conceito de inteligência artificial e explicar as principais técnicas de IA. A metodologia de pesquisa e os procedimentos metodológicos são apresentados no terceiro capítulo. No quarto capítulo é apresentado a análise e discussões dos resultados da pesquisa. Por fim, no quinto capítulo apresenta-se as conclusões da pesquisa e sugestões para futuras investigações.

1.1 Objetivo geral

O objetivo geral do trabalho foi avaliar os métodos de *Multilayer Perceptron* e *Long Short-Term Memory* para predizer os próximos cem dias do valor das ações das cinco empresas mais representativas do Ibovespa.

1.2 Objetivos específicos

Os objetivos específicos do trabalho foram:

- Identificar os melhores, dentro do cenário avaliado, atributos da base de dados para cada técnica e em cada cenário proposto.
- Identificar o melhor, dentro do cenário avaliado, conjunto de hiperparâmetros para técnica e empresa individualmente, assim como para todas as empresas coletivamente
- Identificar qual é a técnica que obteve o melhor desempenho nos cenários propostos
- Avaliar o desempenho dos modelos quantitativamente e através de simulações reais.

2. REFERENCIAL TEÓRICO

O presente capítulo teve como objetivo apresentar os principais conceitos e teorias que embasam o trabalho de pesquisa. Para tal, este capítulo está dividido em sete tópicos, são eles: mercado financeiro, previsão de séries temporais financeiras, aprendizado de máquina, aprendizado de rede, *deep neural networks*, tratamento e divisão do conjunto de dados e métricas de erro.

2.1 Mercado Financeiro

Assaf Neto (2017) explica que o sistema financeiro nacional pode ser resumido em um conjunto de instituições financeiras que através de instrumentos financeiros transferem recursos de agentes superavitários (poupadores) para agentes deficitários (tomadores). Esta definição é extrapolada para qualquer entidade, seja para pessoas, governo ou empresas.

Dentro dessas instituições, o mercado de capitais atua de forma a permitir que empresas captem recursos através de pessoas físicas (poupadores) para viabilizar e prolongar seus projetos. Fortuna (2010) explica que o mercado de capitais é essencial para o desenvolvimento econômico, pois fornece outras opções de capitalização para empresas, diminuindo o custo destes recursos e descentralizando o risco associado ao investimento.

O investimento através do mercado de capitais possibilita a formação de patrimônio a longo prazo e, simultaneamente, fornece capital e recursos a longo prazo para financiar o crescimento das empresas e do país (ASSAF NETO, 2017).

Esse processo é instrumentalizado por meio do mercado de ações, isto é, instrumentos que correspondem a frações do capital social de uma empresa e que representam o quanto o acionista, aquele que detém a ação, possui a respeito daquela sociedade. Esses instrumentos são negociados no mercado de capitais e possuem dois tipos: ordinárias e preferenciais.

As ações *ordinárias* representam a parcela de acionistas que possuem direito a voto em assembleias gerais da companhia. Enquanto as ações *preferenciais*, não detém o direito a voto, porém, em contrapartida, possui preferência no recebimento de dividendos, que são a divisão do lucro da empresa em um determinado período de exercício, além da preferência de reembolso em caso de dissolução da sociedade. Dentro deste contexto, a função do mercado de ações é possibilitar a negociação desses instrumentos sem a presença de um contrato social.

Nesse sentido, o mercado acionário pode ser dividido em dois segmentos, são eles: o mercado primário e o secundário. O mercado primário consiste na emissão de ações disponíveis ao mercado, podendo caracterizar o *inicial public offer* (IPO), quando é a primeira emissão da sociedade ou o *follow-on*, que é a emissão de ações uma vez que a empresa já possui capital aberto e precisa captar mais financiamento. Segundo Eizirik *et al.* (2008), é neste segmento que a principal função do mercado de ações é atingida, ou seja, quando a empresa capta recursos para financiamento próprio.

Já o mercado secundário possibilita a renegociação de ações adquiridas no mercado primário, seja a venda de ações por parte dos acionistas para outros acionistas, o que não gera recurso para empresa, porém, cria-se a possibilidade de que os acionistas consigam comercializar suas ações a qualquer momento a outros acionistas, regulando assim, a precificação da ação em questão.

A bolsa de valores é uma instituição que permite a negociação pública de títulos e valores mobiliários. Os preços são fixados através da lei da oferta e demanda, e devem ser regulados de tal maneira, sem qualquer manipulação e intervenção no preço. Assaf Neto (2017) destaca que a bolsa tem como função principal de proporcionar liquidez (conceito que informa o quanto determinada ação possui potencial para ser liquidada, ou seja, vendida) às ações, permitindo assim, que as negociações sejam executadas no menor tempo possível.

A principal bolsa de valores no Brasil é a Bolsa de Valores de São Paulo (BM & FBovespa), que também é conhecida como Bovespa, torna disponível o acesso a três mercados, são eles: mercado à vista, mercado a termo e o mercado de opções.

O primeiro deles, o mercado à vista, é o mercado onde os títulos adquiridos pelo investidor são entregues até o segundo dia útil após o fechamento da operação na bolsa e a liquidação do pagamento em até três dias após, já no mercado a termo, os títulos são liquidados em prazos diferentes, podendo variar de 30, 60 ou 90 dias após o negócio. Por fim, o mercado de opções negocia os direitos de compra ou venda futura do título a um preço pré-estabelecido. Sendo a principal contribuição deste mecanismo a proteção contra o risco.

O Ibovespa é o principal indicador de desempenho das ações negociadas na Bovespa. Para compor o índice, são utilizadas apenas ações e *units*, que são um conjunto de ações ordinárias e preferenciais e é baseado nos ativos de maior negociabilidade, ou seja, de maior liquidez, e representatividade no mercado acionário brasileiro.

O índice é ponderado quadrimestralmente, ou seja, de quatro em quatro meses e nenhuma empresa pode assumir a participação maior que 20%, caso isso venha acontecer, é

adequado os pesos dos ativos da empresa para este limite, distribuindo o excedente aos demais ativos que constituem a carteira.

Por fim, a B3 explica que o Ibovespa tem como critério o retorno total das ações e, portanto, reflete as variações dos ativos ao longo das operações realizadas no dia. Para isso, assume-se como o principal mecanismo para monitorar e disparar eventos de proteção ao mercado, como o *Circuit Breaker*, que é um mecanismo que interrompe as negociações temporariamente em momentos atípicos do mercado, geralmente representados por fortes quedas nos preços das ações e, conseqüentemente, no índice. Nesse sentido, o Ibovespa é o principal índice para monitorar o desempenho da bolsa de valores brasileira e, portanto, será utilizado como referência no presente trabalho.

2.2 Previsão de séries temporais financeiras

Althelay, El-Alfy e Mohammed (2018) afirmam que séries temporais são dados observados de maneira sequencial e, ao contrário de observações arbitrárias, esses dados podem ser representados em uma linha temporal, mostrando assim seu comportamento ao longo do tempo. Dada a característica do problema temporal, justifica-se o uso de técnicas mais robustas de modelagem, principalmente o uso dos diversos tipos de redes neurais.

Sezer, Gudelek e Ozbayoglu (2020) explica que a respeito dos estudos referentes a séries temporais financeiras, constata-se dois grupos distintos, são eles: aqueles que fazem o uso de dados brutos e os que não.

O primeiro grupo aglutina os estudos que fizeram uso dos dados brutos, sob a ótica de uma janela temporal, a respeito de uma ação ou ativo financeiro. Esses dados podem ser classificados diariamente, mensalmente ou até anualmente e são divididos em seis categorias, são elas: i) preço de abertura, ii) preço de fechamento, iii) preço máximo, iv) preço mínimo, v) volume transacionado, e vi) preço de fechamento ajustado. Já o segundo grupo faz uso das informações variadas, como indicadores advindos da análise técnica e fundamentalista, análise das redes sociais e de sentimentos.

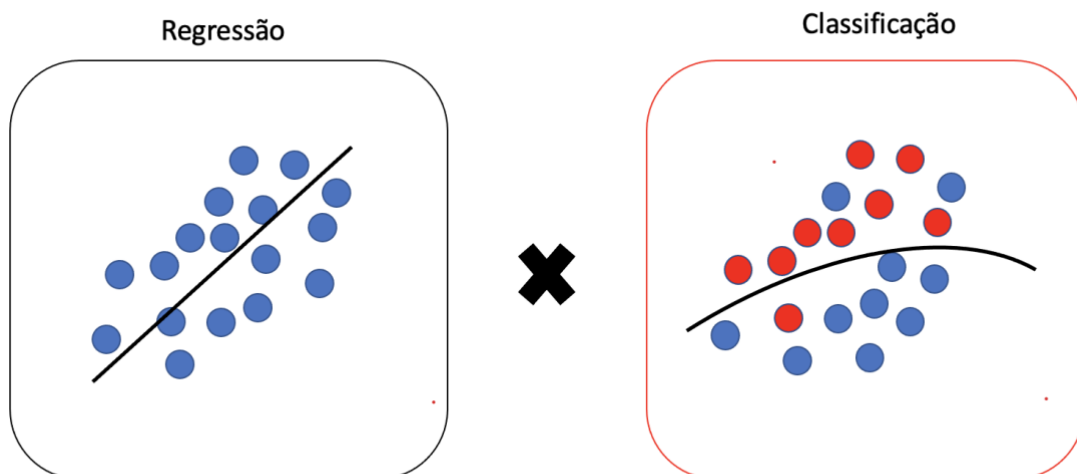
Além disso, tais pesquisas ainda podem ser classificadas em duas diferentes abordagens a respeito da predição das series temporais financeiras, são elas: a predição do preço ou do próximo movimento do ativo financeiro em questão (SEZER, GUDELEK E OZBAYOGLU, 2020).

O primeiro grupo trata-se de um problema de regressão, ou seja, o objetivo final do modelo é tentar acertar qual será o valor da ação no instante de tempo desejado. Nayak et al (2014) afirma que tipicamente os modelos tendem a prever o valor no instante $t+1$, porém, em algumas aplicações pode se obter o objetivo de prever o valor em $t+j$, sendo j o valor de dias no futuro a ser predito.

A perspectiva de prever o movimento é caracterizada como um problema de classificação, sendo assim, o objetivo final do modelo é tentar informar qual é o sentido do movimento do ativo financeiro. Este método utiliza classificações binárias em modelos com duas classes, classificando em movimentos de alta ou queda, ou em problemas de três classes, adicionando mais um estado de neutralidade do ativo em questão.

A Figura 1 ilustra a diferença de abordagem de problemas de regressão e classificação. O quadrado preto ilustra a regressão, que é quando tenta modelar o comportamento em alguma expressão que a melhor defina. Já o quadrado vermelho, ilustra a classificação de duas diferentes classes, ou seja, trata-se de agrupar os dados em finitos tipos (classes).

Figura 1 - Classificação *versus* Regressão.



Fonte: Elaborado pelo autor.

Por fim, acerca da previsão de séries temporais financeiras existem diversos subtópicos como a previsão do preço de ações, índices, volatilidade, cripto moedas, *commodities* e entre outros ativos financeiros (SEZER, GUDELEK E OZBAYOGLU, 2020).

2.3 Teoria de Mercado

Para entender as teorias que regem o mercado de capitais é preciso compreender as duas abordagens que explicam o comportamento de uma ação, que são divididas em abordagem técnica e abordagem fundamentalista (CHITENDERU, MAREDZA E SIBANDA, 2014). A primeira abordagem, a abordagem técnica, defende que a história se repete em algum momento e nesse sentido os padrões se repetem e, dessa forma, é possível agir de maneira a modelar tal comportamento. Já a segunda abordagem compreende que todas as ações, em algum momento, possuem um valor intrínseco e que tal valor é influenciado por conceitos que fundamentam o ativo em questão. Em uma empresa, por exemplo, as variáveis como o comportamento do lucro ao longo do tempo, assim como a qualidade do gerenciamento da empresa impactam diretamente no valor intrínseco da empresa em questão (FAMA, 1995).

Chitenderu, Maredza e Sibanda (2014) explicam que as diferentes abordagens na compreensão do comportamento de ativos financeiros levaram ao surgimento de algumas teorias que tentam explicar o comportamento do mercado. A teoria do andar aleatório (do inglês, *Random Walk*) postula que o preço das ações caminham de maneira aleatória ao redor do preço intrínseco, sendo assim é impossível prever o comportamento futuro. Em contraste a isto, caso as diferenças do valor intrínseco para o valor real tenham comportamentos sistemáticos, é possível estabelecer uma possibilidade para investidores maximizarem seus ganhos através de uma modelagem que capte tal variação (FAMA, 1995).

Umoru *et al.* (2020) explica que em adição a teoria do caminhar aleatório, surgiu a teoria do mercado eficiente. Tal teoria defende que o mercado é eficiente e isso determina que o preço das ações se adaptam instantaneamente a qualquer informação e perturbação que possa existir e, portanto, qualquer valor no tempo é assumido como um valor intrínseco.

Por fim, no cenário em que alguma das teorias fossem comprovadas, qualquer tentativa de encontrar o valor futuro de uma ação seria um esforço inútil e isso invalidaria a abordagem técnica.

2.4 Aprendizado de Máquina

Arthur Samuel foi um dos pioneiros no segmento de aprendizado de máquina (em inglês, *machine learning*) e definiu tal conceito como o campo de estudo que permite computadores

aprenderem sem serem explicitamente programados para tal (SAMUEL, 1959). Em complemento a isto, a técnica de aprendizado de máquina consiste em algoritmos que são aplicados interativamente a um conjunto de dados, referente a um problema em específico, que permitem encontrar padrões complexos ou descobertas não usuais a respeito do problema em questão. Neste cenário, os avanços na área de *machine learning* permitiram inovações em diversas áreas tais como: detecção de fraudes, visão computacional, reconhecimento facial, processamento de linguagem natural e muitos outros campos do (JANIESCH, ZSCHECH E HEINRCH, 2021).

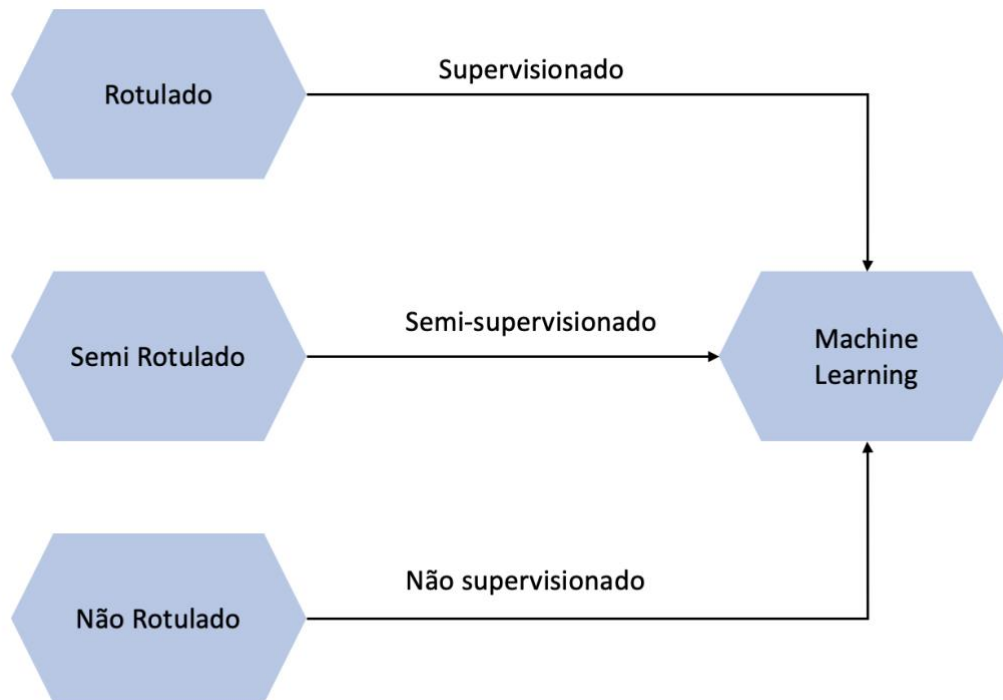
O aprendizado de máquina pode ser dividido de acordo com a característica do processo de aprendizado. Eles são classificados em três tipos: supervisionado, não supervisionado e semi-supervisionado (NAGA E MURPHY, 2015).

O primeiro tipo, **aprendizado supervisionado**, consiste no aprendizado através de exemplos rotulados, ou seja, informações presentes na base de dados, denominados de atributos, assim como o seu respectivo rótulo, a variável de resposta. Esse processo é denominado supervisionado pelo simples fato de inserir para o modelo exemplos com respostas, permitindo, assim, que o mesmo encontre um padrão a respeito do problema e extrapole tal comportamento para cenários inéditos. Essa abordagem é usualmente aplicada em problemas de regressão e/ou classificação (NAGA e MURPHY, 2015).

O segundo tipo, **aprendizado não supervisionado**, consiste em dar exemplos que não possuem os respectivos rótulos (respostas) e, portanto, possui o objetivo de encontrar algum tipo de estrutura entre os dados fornecidos (JANIESCH, ZSCHESCH E HEINRCH, 2021). Um exemplo de tal aplicação é: dado um grupo de animais, solicitar o agrupamento dos diferentes animais de acordo com a regra construída pelo modelo (sem supervisão). Este exemplo é conhecido como *clustering* e é uma das principais aplicações do aprendizado não supervisionado e esse exemplo pode ser extrapolado para qualquer contexto e não apenas animais.

O terceiro tipo é o **aprendizado semi-supervisionado** consiste na união entre os métodos anteriores, isto é, tem momentos em que os exemplos são rotulados e outras vezes não (NAGA e MURPHY, 2015). A Figura 2 ilustra os tipos de aprendizados.

Figura 2 – Tipos de aprendizado.



Fonte: Elaborado pelo autor.

Por fim, existe outro tipo de aprendizado, que não faz o uso de exemplos e rótulos, e é conhecido como **aprendizado por reforço**. Esse tipo de aprendizado faz uso de um estado inicial do sistema e a cada iteração é aplicado punições e recompensas, tendo como objetivo minimizar a punição e maximizar a recompensa (JANIESCH, ZSCHECH E HEINRICH, 2021).

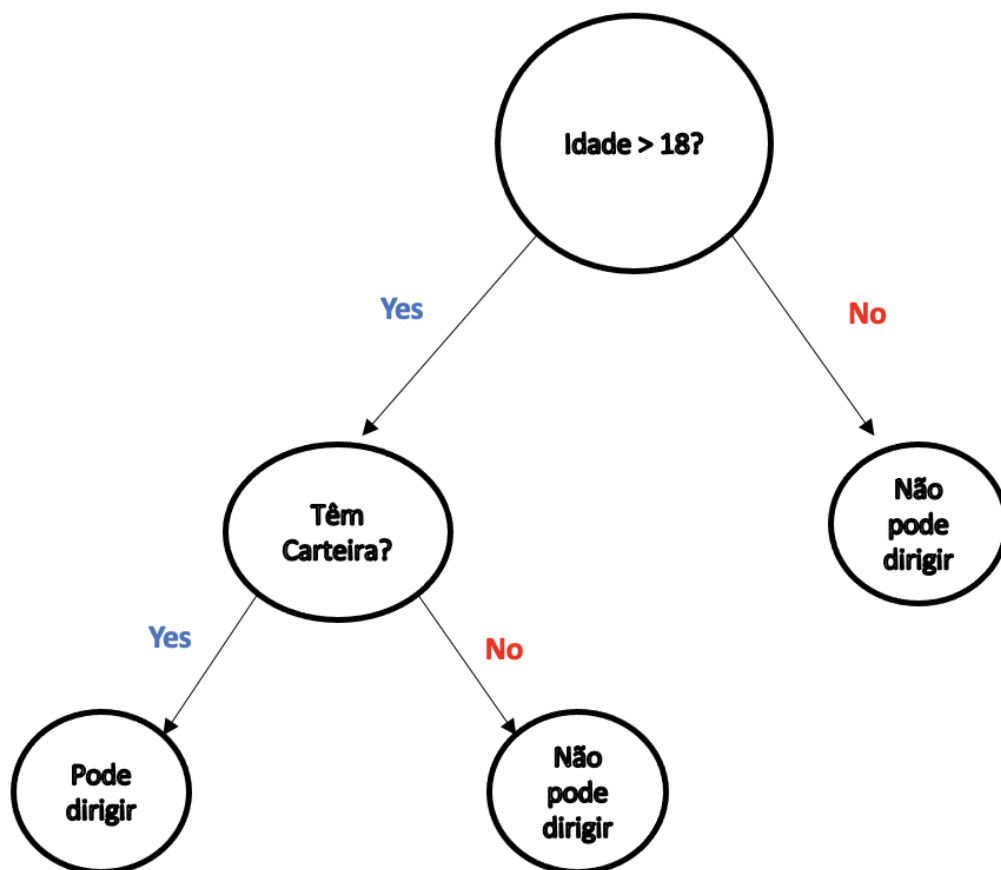
2.4.1 XGBoost

O XGBoost é um acrônimo de *Extreme Gradient Boosting* e é um algoritmo baseado em árvores de decisão aplicado à técnica de *gradient boosting*. Mitchell e Frank (2017) explicam que as árvores de decisões são métodos que aprendem através de repetidas repartições do conjunto de treinamento através de algum critério. Os preditores da árvore recebem um valor, de acordo com o treinamento, para cada atributo. No final, consolida-se em um modelo com determinadas regras que podem ser usadas em valores inéditos.

Tal característica pode submeter o modelo a um cenário em que o modelo problematize em demasia a modelagem do problema, considerando assim, o *overfitting*. Esse comportamento

acontece quando as árvores crescem sem nenhum detrimento e, portanto, podem apresentar milhares de regras que não modelam corretamente o problema e sim a amostra de dados em questão. Nesse cenário, as árvores podem apresentar uma taxa de acerto em 100% dentro do conjunto de treinamento, porém atuam com baixíssima assertividade no conjunto de testes. A Figura 3 ilustra uma árvore de decisão que prediz se o indivíduo possui ou não uma carteira de habilitação, dado um conjunto de características.

Figura 3 - Árvore de decisão.



Fonte: Elaborado pelo autor.

Nobre e Neves (2019) definem que *boosting* é uma técnica *ensemble* que busca unificar preditores fracos, cujo erro é grande, em um único preditor que tem um erro menor, definido como preditor forte. O método Ensemble trata-se de um conjunto de preditores atuando para

predizer a mesma variável alvo. No final, os preditores são unificados, por votação nos problemas de classificação e por média na regressão, para obter a resposta final.

Ainda sobre isto, o *gradient boosting* é uma variação do problema de aprendizado no *Gradient Descent*, com alterações na função de perda que mede a performance do algoritmo no conjunto de treinamento (MITCHEL E FRANK, 2017). Basicamente, em cada iteração, é treinado um preditor para minimizar a função de perda. No final de cada execução, é alterado os valores dos preditores que tiveram amostras classificadas ou regredidas de maneira incorreta, a fim de minimizar a função de perda.

O XGBoost é um algoritmo de *gradient boosting* genérico que aplica um termo de regularização na função de perda. O principal objetivo dessa etapa é combater o *overfitting*. Portanto, ao contrário do *gradient boosting* que busca minimizar a função de perda não levando em conta a complexidade do modelo, o XGBoost busca encontrar o modelo menos complexo, ou seja, com a menor quantidade de regras, e que obtém o melhor resultado.

Por fim, Chen *et al.* (2020) esclarece que o algoritmo utiliza o conceito de *gain*. Este conceito está relacionado ao ganho que cada segmentação impacta durante o treinamento do modelo. Cada segmentação é realizada através de valores para os atributos presentes na base de dados. No final, a importância de cada atributo é extraída através da média do ganho, calculada através da soma de todos os ganhos, por atributo, pelo número total de segmentações.

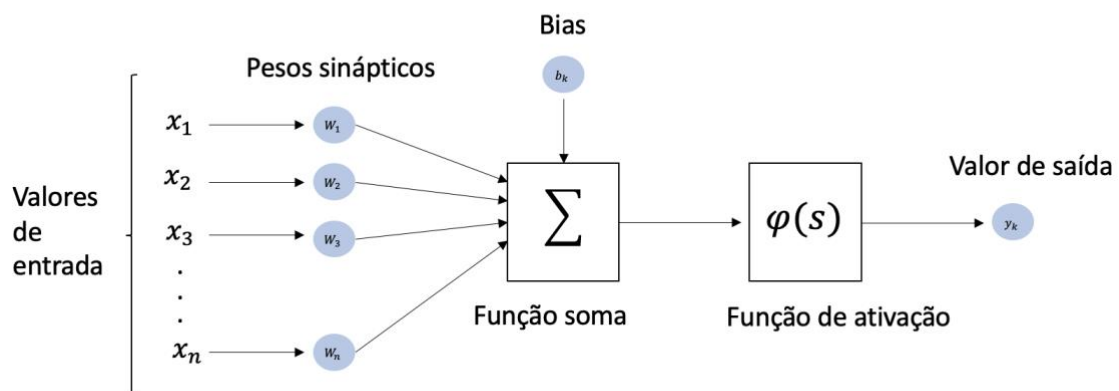
2.4.2 Redes Neurais Artificiais

Segundo Abelém, Pacheco e Vellasco (1995), redes neurais artificiais são sistemas inspirados nos neurônios biológicos e na estrutura paralela do cérebro, que possui a capacidade de adquirir, armazenar e utilizar conhecimento. Esses sistemas são compostos de diversas unidades simples, conhecidas por neurônios artificiais, ligados de maneira apropriada para obter comportamentos. O comportamento é determinado pela estrutura das ligações entre os neurônios e pelos valores das conexões, conhecido como pesos sinápticos. Conforme Haykin (2001), o conhecimento da rede é adquirido através de um processo de aprendizado e pode ser resumido em encontrar os valores dos pesos sinápticos ideais para o conhecimento desejado.

Haykin (2001) explica que o neurônio é o elemento de processo fundamental à operação da rede neural. O modelo do neurônio contém três elementos básicos: Os pesos, somador, função de ativação e o *bias*. O primeiro consiste no valor que multiplicará a entrada do neurônio, enquanto o segundo realiza a soma ponderada pelos pesos de todas as entradas mais o *bias*. O

bias, segundo Filho (2016), serve para aumentar os graus de liberdade de ajuste dos pesos. Por fim, a função de ativação é aplicada ao resultado do somador e ambos são extremamente úteis por possibilitarem a inclusão de um fator de não linearidade na arquitetura da rede (FILHO, 2016). A Figura 4 representa o modelo do neurônio artificial.

Figura 4 - Modelo de um neurônio.



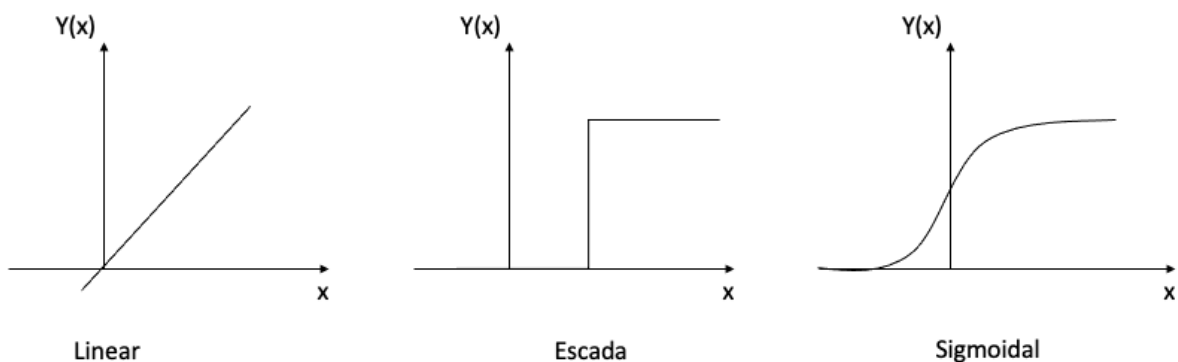
Fonte: Baseado em Haykin (2014).

O modelo matemático do neurônio é apresentado na Equação 2. A saída do neurônio será o valor da função de ativação no ponto V_k .

$$V_k = \sum w_k \cdot x_1 + b_k \quad (1)$$

Fleck *et al.*, (2016), esclarece que a função de ativação representa o efeito que a entrada interna e o estado atual de ativação exercem na definição do próximo estado de ativação da unidade. Algumas funções de ativação são ilustradas na Figura 5.

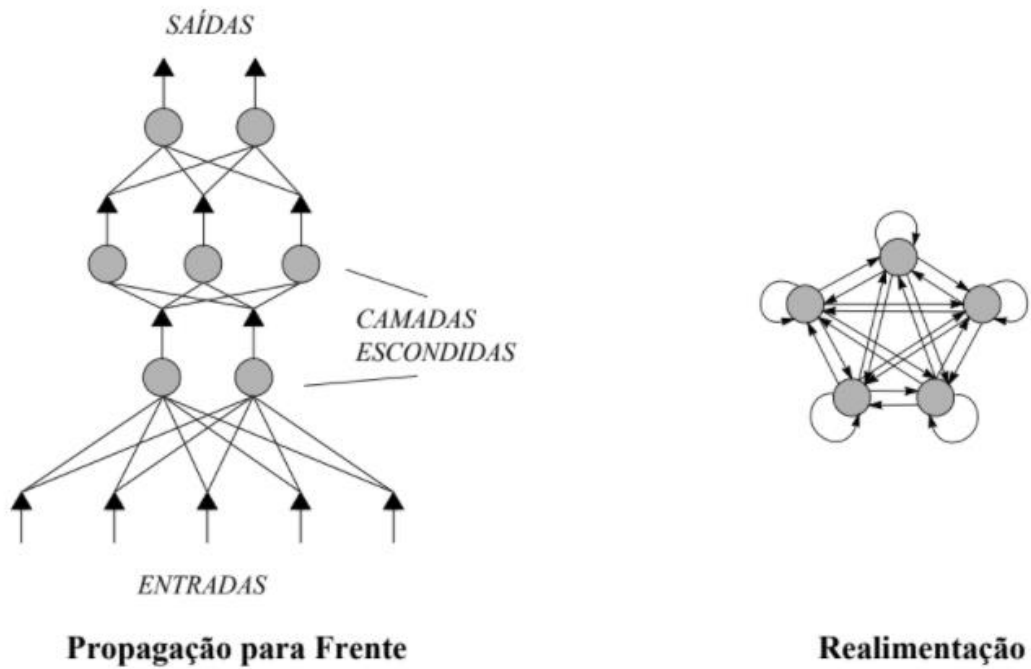
Figura 5 - Funções de ativação linear, escada e sigmoidal.



Fonte: Baseado em Haykin (2014).

Por fim, as redes neurais podem ser compostas por diversas camadas de neurônios, caracterizando assim, as camadas escondidas (ou oculta), que são as etapas intermediárias entre a entrada e saída da rede. Além disso, as redes podem estar dispostas em duas arquiteturas distintas, são elas: *feedforward* e *backward*. A primeira delas é quando as redes possuem somente um caminho entre suas camadas. Já a segunda, é quando existe alguma realimentação entre uma camada e outra, assim sendo, apresentam mais de um caminho para o tráfego de informação (DALLABRIDA, 2007). A Figura 6 ilustra as arquiteturas das redes neurais.

Figura 6 - Arquitetura das redes neurais.

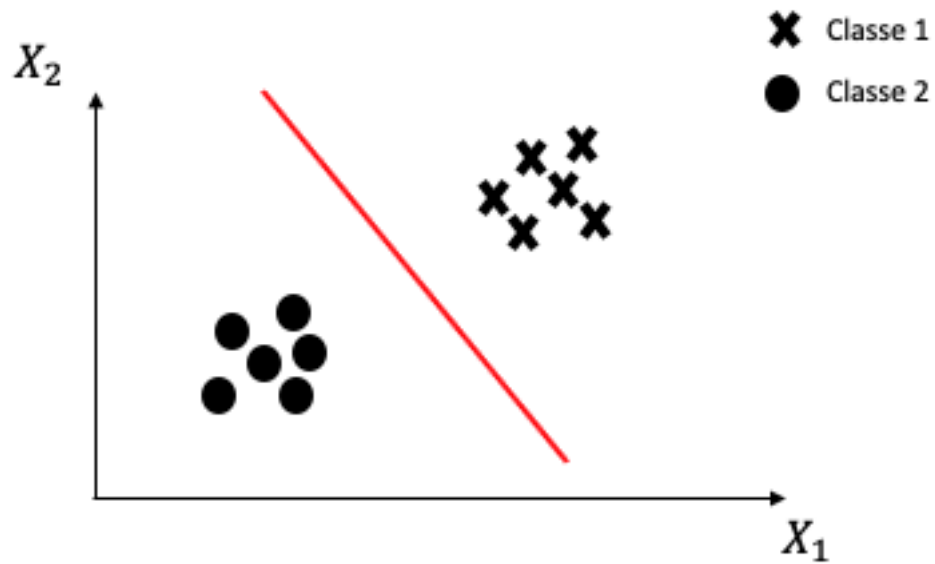


Fonte: Rauber (2005).

2.4.3 Multilayer Perceptron

Fausett (1994) explica que o modelo do *perceptron* é uma rede neural que possui a arquitetura clássica de apenas as camadas de entrada e saída e com função de ativação binárias. Sua principal aplicação é em problema de classificação, porém, somente em problemas linearmente separáveis, ou seja, não consegue atuar em problemas que não são linearmente separáveis. A Figura 7 ilustra um problema linearmente separável.

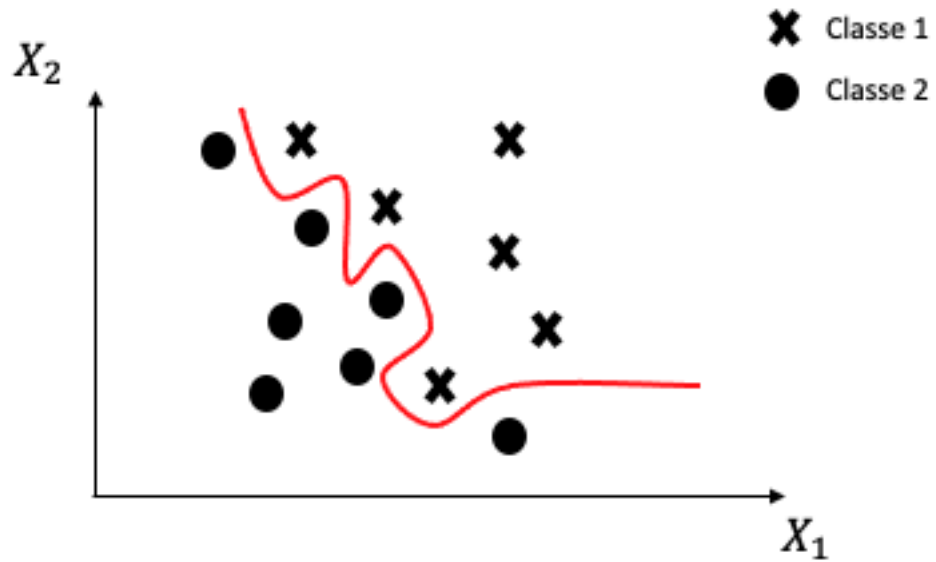
Figura 7 - Problema linearmente separável.



Fonte: Baseado em Rauber (2014).

Segundo Gonçalves *et al.*, (2008), para suprir os problemas não linearmente separáveis, surgiu o modelo das redes MLP, acrônimo de *Multilayer perceptron*. A Figura 8 ilustra a classificação de um problema não linearmente separável.

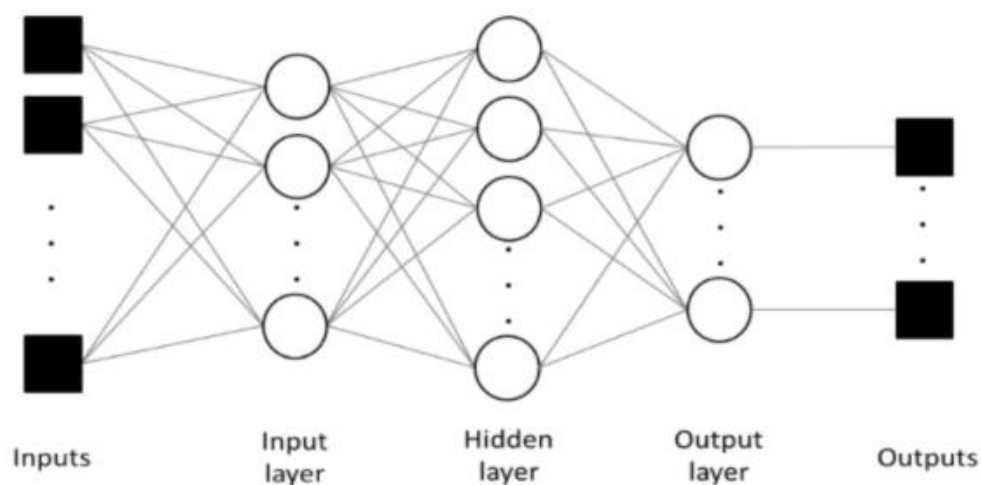
Figura 8 - Problema não linearmente separável.



Fonte: Baseado em Rauber (2014).

Dalabrida (2007) explica que uma rede neural artificial do tipo *Multilayer Perceptron* (MLP) é constituída pela camada de entrada na rede, de uma ou mais camadas ocultas e de uma camada de saída. Com exceção da camada de entrada, todas as outras camadas são constituídas por neurônios e, portanto, apresentam capacidade computacional. A Figura 9 ilustra uma MLP.

Figura 9 - *Multilayer Perceptron*.



Fonte: Haykin (2014).

A utilização de MLPs para problemas de regressão é extremamente útil, visto que o modelo da rede é um aproximador universal de funções, desde que possua a quantidade suficiente de neurônios escondidos (RAUBER, 2005).

2.5 Aprendizagem da rede

Segundo Gonçalves *et al.*, (2008), as redes neurais artificiais têm a capacidade de aprender através de exemplos. Neste tipo de aprendizado, não se procura obter regras, mas sim determinar a intensidade das sinapses entre neurônios, ou seja, o valor dos pesos. O conjunto de procedimentos para encontrar esses valores se chama: algoritmo de aprendizado. Eles basicamente se diferem pela maneira que o ajuste dos pesos é feito, cada um tem suas características que podem apresentar vantagens e desvantagens na solução de problemas.

Os métodos para treinamento de redes podem ser agrupados em dois tipos: o aprendizado supervisionado e o aprendizado não supervisionado. Na aprendizagem supervisionada cada exemplo de treino está acompanhado por um valor que é o valor desejado, ou seja, são apresentados para a rede os exemplos e as respostas, para que a rede encontre o padrão que culminou no valor final (RAUBER, 2005) No aprendizado não supervisionado, a única informação disponível são os valores desejados, portanto, a tarefa de aprendizagem é

descobrir a correlação entre os exemplos. O número de categorias não está definido. Isso significa que ela tem que desenvolver uma representação própria dos exemplos, esse processo também é conhecido como *clustering*.

Segundo Bishop *et al.*, (1995), a aprendizagem por correção do erro, consiste no ajuste dos pesos entre os neurônios da rede, baseados nos erros encontrados na saída da rede, com o objetivo de minimizar esse erro. O erro da saída consiste na diferença entre o resultado calculado pelo neurônio e o resultado desejado.

2.5.1 *Backpropagation*

Haykin (2001) explica que o algoritmo de aprendizado *backpropagation* pode ser considerado uma generalização da regra delta. O algoritmo segue o paradigma do aprendizado supervisionado e baseia-se no aprendizado por correção de erro, onde o erro é retro propagado da camada de saída para as camadas intermediárias da rede.

Neste contexto, o algoritmo aprende interativamente pelo processamento de um conjunto de exemplos no treinamento da rede. Compara-se a saída da rede para cada exemplo com o valor desejado e para cada iteração, os pesos são modificados de forma a minimizar o erro médio quadrático entre a saída da rede e o valor desejado. Estas modificações são feitas na direção reversa, ou seja, começa na camada de saída e vai através das camadas ocultas chega-se na primeira camada (HAN; KAMBER, 2000).

Por fim, o algoritmo usa um método de gradiente descendente para encontrar um conjunto de pesos que pode modelar o problema de forma a minimizar o erro da rede. Basicamente, calcula-se o gradiente local e aplica-se um decréscimo no sentido contrário do gradiente, a fim de localizar o mínimo da função em questão. A taxa de aprendizado é uma constante que multiplica o incremento ou decremento a ser realizado no peso. Ela é utilizada para que não ocorra um mínimo local no espaço de decisão, ou seja, que a rede converge para solução, mas não para a solução ótima. A taxa de aprendizado pode ser vista como um mecanismo que auxilia na busca pelo mínimo global (DALABRIDA, 2007).

2.5.2 *Stochastic Gradient Descent*

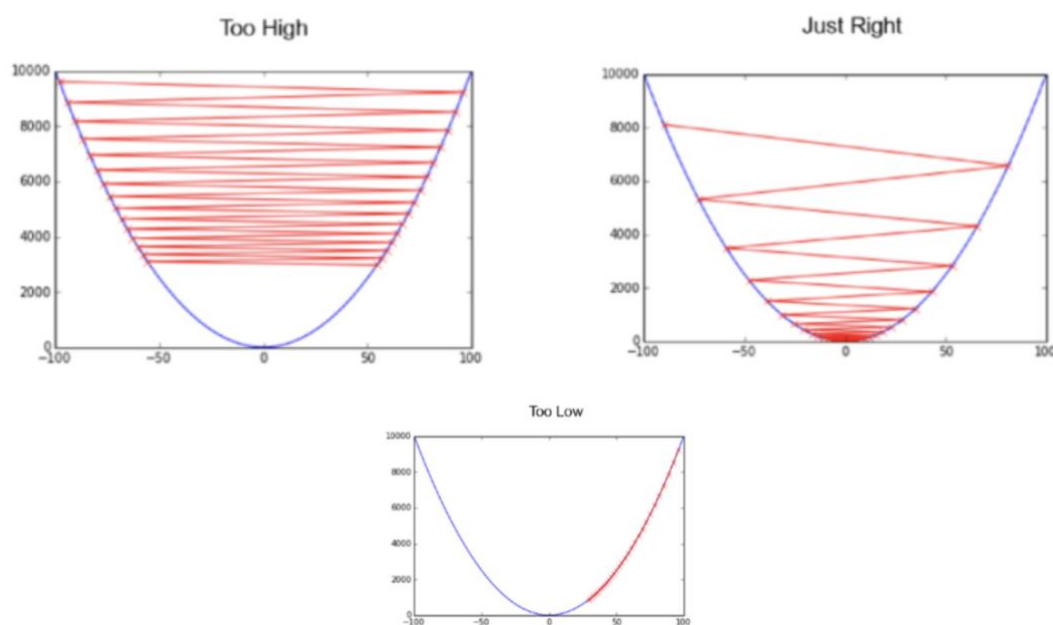
Diferente do *backpropagation*, que faz uso do *gradient descent*, responsável por atualizar os pesos de acordo com o resultado do gradiente de cada iteração, o *stochastic gradient*

descent (SGD) faz uso de uma amostra estocástica (aleatória) para calcular o gradiente e atualizar os pesos de acordo a minimizar o erro (BOTTOU, 2012). O método pode ser visto como uma simplificação do gradiente descendente.

Segundo Ketkar (2017), a abordagem do SGD permite duas características ímpares. A primeira delas é o custo computacional de processar o gradiente de uma simples amostra contra o gradiente de todos os exemplos. Já o segundo é a incerteza na direção da otimização, causada pelo fato de que o algoritmo de aprendizagem está limitado a uma fração do *dataset* (do inglês, conjunto de dados). Essa característica fornece resultados melhores, especialmente em *datasets* com grandes quantidades de exemplos (KETKAR, 2017). Além disso, o algoritmo possui mais mecanismos para evitar mínimos locais de diferentes profundidades, característica essa, que não é vista no gradiente descendente, haja visto que o mesmo vai ficar preso no primeiro mínimo local que encontrar.

Por fim, a escolha da taxa de aprendizado, utilizada em ambos os métodos, pode ser um desafio para obter o melhor desempenho do método e deve ser devidamente escolhida (KETKAR, 2017). Uma taxa muito alta pode causar um efeito de pulo, ou seja, pulando de um lado para outro da solução esperada, enquanto uma taxa muito baixa, pode significar uma convergência muito lenta para o mínimo esperado. Alguns outros algoritmos de otimização tentam resolver problemas como este, como o Adam, que é um método baseado na estimação do momento para atualizar os pesos do algoritmo. Apresenta-se na Figura 10, o efeito de uma taxa de aprendizado mal e bem dimensionada.

Figura 10 - Efeito de uma taxa de aprendizado mal dimensionada.



Fonte: Adaptado de Ketkar (2017).

2.5.3 Adam

Segundo Kingma e Lei Ba (2017), Adam é um método estocástico de otimização de parâmetros, que requer apenas gradientes de primeira ordem e pouco consumo de memória. Além disso, o método faz uso de taxas de aprendizados independentes para cada parâmetro e é baseado no momento de primeira e segunda ordem.

O algoritmo faz uso das melhores características de métodos como o *Adadelta* e *RMSprop*, que fazem uso do decaimento exponencial da média quadrática dos gradientes passados, assim como o *Momentum*, que mantém um decaimento exponencial da média dos gradientes (RUDER, S. 2016).

Por fim, o método é simples e eficiente para otimização estocástica de funções e têm como objetivo de simplificar o aprendizado de algoritmos de *machine learning* com grandes conjuntos de dados ou de alta dimensionalidade. Além disso, os autores expõem que testes empíricos demonstram que o método funciona muito bem na prática quando comparado a outros métodos de otimização estocástica (KINGMA; LEI BA, 2017).

2.6 Deep Neural Networks

Sun *et al.* (2018) afirma que as *Deep Neural Network* (DNN) são capazes de realizarem tarefas iguais ou melhores do que humanos, como as aplicações dos carros autônomos. Cichy e Kaiser (2019) reforçam o impacto da revolução que o surgimento das DNNs permitiu em campos como visão computacional, processamento de linguagem natural e tarefas de navegação, como encontrar o caminho mais curto dado um problema.

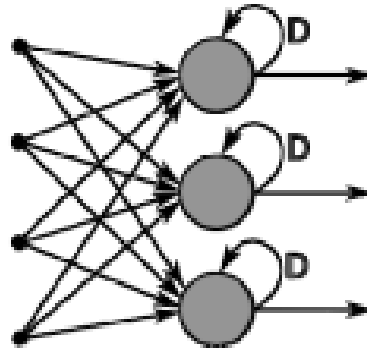
As DNNs consistem de uma rede com uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Os neurônios na camada de entrada são equivalentes à quantidade de atributos analisados pelo modelo. Na camada de saída, encontra-se um problema de classificação ou regressão que combinam com a saída esperada do modelo (KRAUSS; DO; HUCK, 2017).

Além disso, as DNNs possuem diversos tipos de arquiteturas e podem possuir milhões de parâmetros e, portanto, boa parte do sucesso há ser obtido passa pela melhor escolha de topologia para a natureza do problema. Por fim, existem dois tipos de DNNs amplamente divulgadas e validadas, são elas: As *convolutional neural networks* e as *recurrent neural networks*. A primeira delas é amplamente utilizada em visão computacional. Já a segunda é utilizada em processos em que os dados são sequenciais, como o processamento de linguagem natural e o uso de séries temporais (MIKKULAINEN *et al.*, 2019).

2.6.1 Recurrent Neural Network

De acordo com Samarawickrama e Fernando (2017), o conceito de redes neurais recorrentes foi primeiramente proposto por Hopfield, em 1983. Esta arquitetura de rede, como exposto nas seções anteriores, utiliza a arquitetura de recorrência que, em resumo, faz uso de laços de realimentação entre a saída das camadas. A Figura 11 ilustra a topologia tradicional de uma rede recorrente.

Figura 11 - Topologia de uma rede neural recorrente.



Fonte: Samarawickrama, A. J. P., & Fernando, T. G. I. (2017).

As redes neurais recorrentes são utilizadas para usarem dados atuais e antecedentes, definindo assim, dados que atuam de forma sequencial. Essa característica permite que a rede preserve informações durante o treinamento da mesma (ALTHELAYA, EL-ALFY E MOHAMMED, 2018).

Entretanto Kelly e Knottenbelt (2015) afirmam que é comum que as redes neurais recorrentes sofram do fenômeno *vanishing gradient*, que é o desaparecimento do gradiente quando o mesmo é propagado muito atrás da rede, fazendo com que a capacidade de manter e considerar registros muito históricos da rede afetada. Uma solução para este problema é o surgimento das *Long Short-Term Memory Network*, também conhecida como LSTM.

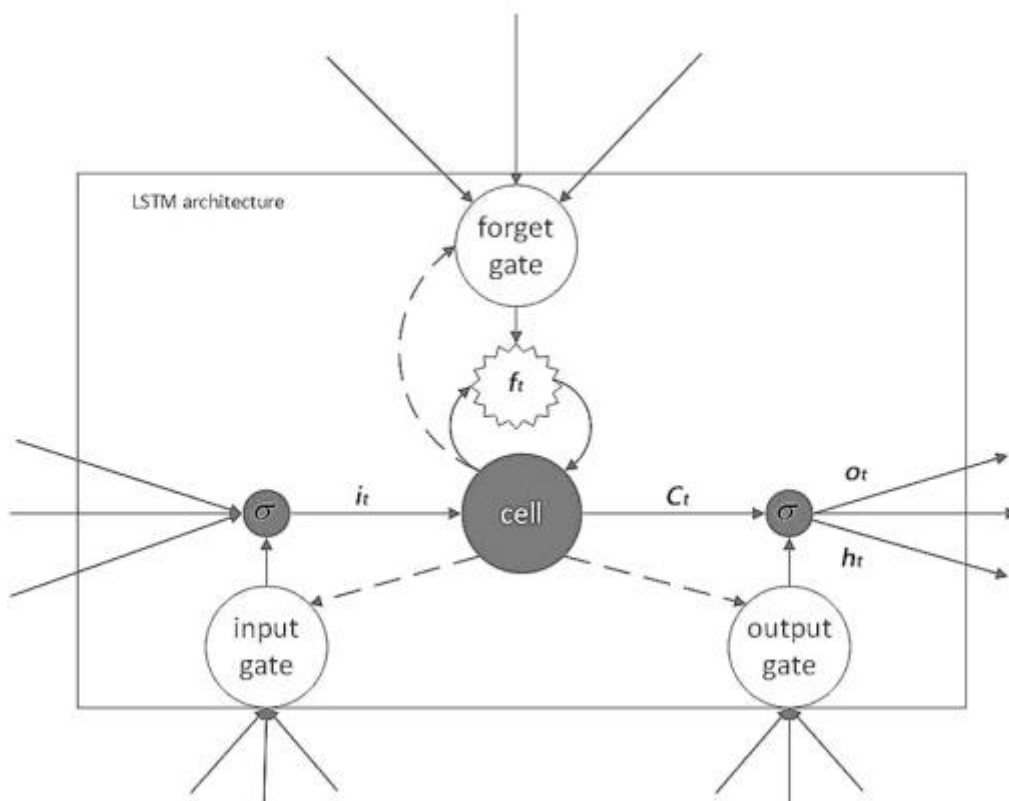
2.6.2 Long Short-Term Memory Network

Segundo Lee e Yoo (2018), o algoritmo da LSTM é uma solução composta de uma abordagem de *gating*. Em resumo, as unidades de entrada das LSTM são diferentes dos neurônios tradicionais. Tais entradas são caracterizadas como células e possuem três elementos básicos que a caracterizam, são eles: *Input gate*, *forget gate* e o *output gate* (SAMARAWICKRAMA E FERNANDO, 2017).

Lee e Yoo (2018) explica que a célula da LSTM recebe um dado de entrada, que é avaliado pelo *input gate*. Este mecanismo é responsável por dimensionar a informação para determinar qual parte deve ou não ser adicionada ao próximo estado da célula. Enquanto o *forget gate* é responsável por determinar qual parcela da informação ainda é válida para ser mantida na memória da rede (POTHUGANTI, 2021).

O *input gate* e o *output gate* são responsáveis por alimentar as camadas ocultas da rede. No entanto, o *output gate* é responsável por determinar quais são as informações úteis a serem extraídas da célula utilizando as informações da execução passada (SUN *et al.*, 2020). A Figura 12 ilustra a configuração de uma célula de LSTM.

Figura 12 - Configuração da célula.



Fonte: Sun *et al.* (2020).

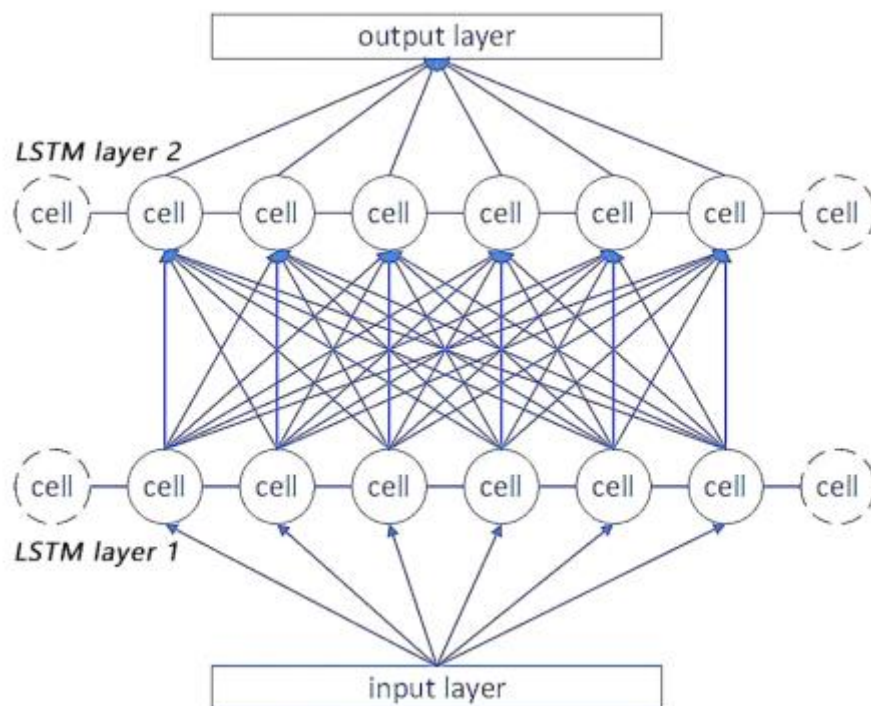
A utilização desta configuração em uma camada oculta caracteriza o modelo mais utilizado de LSTM, que são conhecidas como *Vanilla*. Por fim, a arquitetura robusta da célula da LSTM permite que a rede possa aprender efetivamente através de diferentes porções de dados, que podem estar na sequência ou não, controlando o que deve ser adicionado ou excluído da memória da rede (GREEF *et al.*, 2017).

2.6.3 Stacked LSTM

As arquiteturas *stackeds* são redes que possuem duas camadas de LSTM conectadas entre si. Na prática, são redes que utilizam o modelo de célula e possuem mais de uma camada

oculta, ou seja, a saída da primeira camada oculta é a alimentação da segunda camada (SUN *et al.*, 2020). A Figura 13 exemplifica a estrutura da *Stacked LSTM*.

Figura 13 - Estrutura de uma *Stacked LSTM*.



Fonte: Sun *et al.* (2020).

Por fim, segundo os autores, o modelo de múltiplas camadas empilhadas permite uma melhora na extração de atributos relevantes para melhor resultado do modelo.

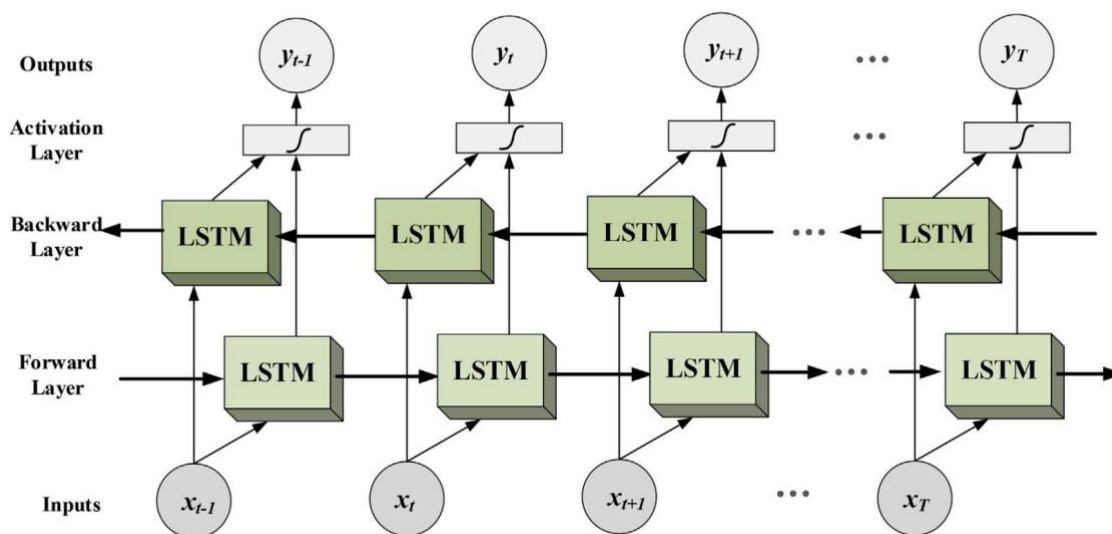
2.6.4 Bidirecional LSTM

Althelaya, El-Alfy e Mohammed (2018) esclarecem que o uso de redes bidirecionais é uma outra maneira de conectar duas camadas sequenciais, fazendo uso da informação inicial do conjunto de dados assim como a saída da célula da camada respectiva. Na prática, existem duas camadas, sendo que cada uma delas opera as informações em fluxos contrários. A primeira camada faz uso da informação na mesma sequência dos dados, operando como uma rede unidirecional. Já a segunda camada atua no sentido *backward*, ou seja, de maneira a retroalimentar a rede com a saída da mesma.

Por fim, as redes LSTM bidirecionais permitem que a saída da rede seja avaliada através de informações do passado, alimentadas através da camada *feedforward*, e de informações

futuras, que são fornecidas pela camada *backward*. Com isso, ambas informações do passado e do futuro são levadas em consideração simultaneamente pelo modelo, descartando qualquer atraso que possa existir na alimentação dessa informação (YILDIRIM, 2018). A Figura 14 ilustra a estrutura de uma LSTM bidirecional.

Figura 14 - Estrutura de uma rede bidirecional.



Fonte: Yildirim (2018).

2.7 Tratamento e divisão do conjunto de dados

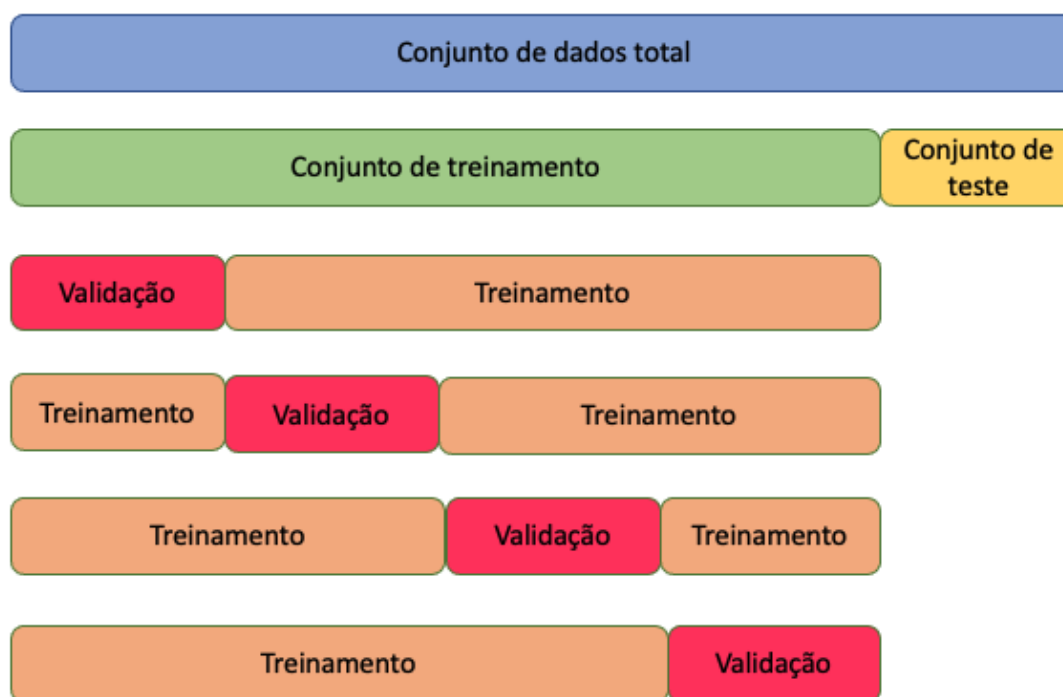
Segundo Xu e Goodacre (2018), a validação de um modelo de inteligência artificial é a parte mais importante de sua criação e, para isso, é necessária uma estratégia de separação de dados apropriada. A maioria dos modelos de inteligência artificial têm inúmeros parâmetros, que influenciam diretamente na complexidade e na capacidade de cada modelo e, portanto, também aumentam a possibilidade de caracterizar o fenômeno de *overfitting*. Este fenômeno é caracterizado quando o modelo treinado desempenha extremamente bem para o conjunto de dados utilizado no treinamento, porém, não consegue entregar o mesmo resultado para amostras desconhecidas para o mesmo.

Berrar (2018) destaca que a melhor maneira de evitar o *overfitting* e garantir a capacidade de generalização do modelo é a avaliação do mesmo através de novos dados, gerados através da mesma população que foi utilizada para o treinamento do mesmo. Na prática, para resolver este problema é necessário segmentar o conjunto de dados em dois ou mais

conjuntos, são eles: treinamento e teste. O primeiro deles serve de consumo inicial para o modelo aprender sobre os dados. Já o segundo, que não foi utilizado durante o treinamento e logo, são dados inéditos aos olhos do mesmo, serve para avaliar a capacidade de generalização do mesmo.

Um dos métodos mais utilizados para evitar o *overfitting* é o uso do *cross-validation*. Em resumo, este método consiste em dividir o conjunto de dados total em três amostras, são elas: treinamento, validação e teste. O primeiro deles é o conjunto que deve conter a maior quantidade de registros e é responsável pelo treinamento do modelo, enquanto o conjunto de validação é a parcela de dados responsável por avaliar o modelo em cada iteração do treinamento. Por fim, o conjunto de teste é responsável por avaliar a capacidade de generalização do modelo resultante, concluindo assim, se o resultado é satisfatório ou não dada a natureza do problema (TSANAS; XIFARA, 2012). A Figura 15 representa este método.

Figura 15 - Representação do *cross-validation*.



Fonte: Elaborado pelo autor.

2.8 Normalização

Moeeni e Bonakdari (2017) explicam que a normalização dos dados sempre assumiu um papel importante no sucesso de modelos estatísticos. A normalização é uma etapa do pré-processamento dos dados e têm como principal objetivo de acomodar valores de múltiplos atributos em uma mesma escala, permitindo assim, que para um conjunto de dados de inúmeros atributos, todos possuem média zero e variância equivalente a 1 (CHAKRABORTY, 2000).

A padronização dos dados de múltiplos atributos em uma mesma escala também reduz a possível influência maior de um atributo que possui uma escala maior, ou seja, reduz a ponderação indevida de atributo para atributo (NAYAK *et al.*, 2014). Diferentes técnicas podem ser usadas para atingir tal objetivo, como a técnica de *min-max*, *z-score normalization* e entre outras.

No método *min-max*, os valores de saída são pré-definidos para conterem no conjunto [0,1]. Faz-se uso dos valores máximo e mínimo de cada atributo, convertendo todos os valores presentes em uma fração referente ao valor máximo, por exemplo, o valor máximo assume o valor 1, enquanto o valor mínimo assume o valor de zero. Por fim, o maior problema na utilização desta técnica para uso de séries temporais é a imprevisibilidade do valor máximo. Para resolver este problema, é indicado transformar os valores que extrapolam essa condição no valor máximo ou mínimo, dependendo da característica do mesmo. Na Equação 1 demonstra-se o cálculo realizado para cada amostra do atributo.

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

Por fim, Nayak *et al.* (2014) explica que a técnica de *z-score normalization* também apresenta falhas em normalizar séries temporais, haja visto que a técnica utiliza a média e o desvio padrão de cada atributo para normalização e que, pela característica intrínseca das séries temporais, que é variar no tempo, impacta na consolidação da média e do desvio padrão.

2.9 Métrica de erro

Segundo Terziyska *et al.* (2018), cada método de calcular o erro tem sua peculiaridade, por exemplo, o erro quadrático médio (*MSE - Mean Squared Error*) utiliza a distância euclidiana do ponto avaliado até o respectivo resultado e, portanto, faz uso de uma punição aos *outliers* do processo, enquanto isso, o erro absoluto (*MAE - Mean absolute error*) trata de apenas encontrar a diferença entre o valor predito e o correspondente valor.

A raiz do erro médio quadrático é representada pela sigla RMSE (*root mean squared error*) e é amplamente utilizada como métrica de erro para avaliações numéricas (regressão). A diferença entre o valor predito e o valor alvo é atenuada no cálculo do RMSE e, assim como o *mean squared error (MSE)*, *outliers* são amplificados e, conseqüentemente, erros maiores são punidos na aferição da métrica (ROONDIWALA, 2017).

As Equações 3, 4, 5 apresentam o *mean squared error*, *root mean squared error* e o *mean absolute error* respectivamente. Onde y_1 assume o valor predito e y o valor real e n é o número de amostras avaliadas no intervalo.

$$MSE = \frac{1}{n} \sum_{1}^{n} (y - y_1)^2 \quad (3)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{1}^{n} (y - y_1)^2} \quad (4)$$

$$MAE = \frac{1}{n} \sum_{1}^{n} (y - y_1) \quad (5)$$

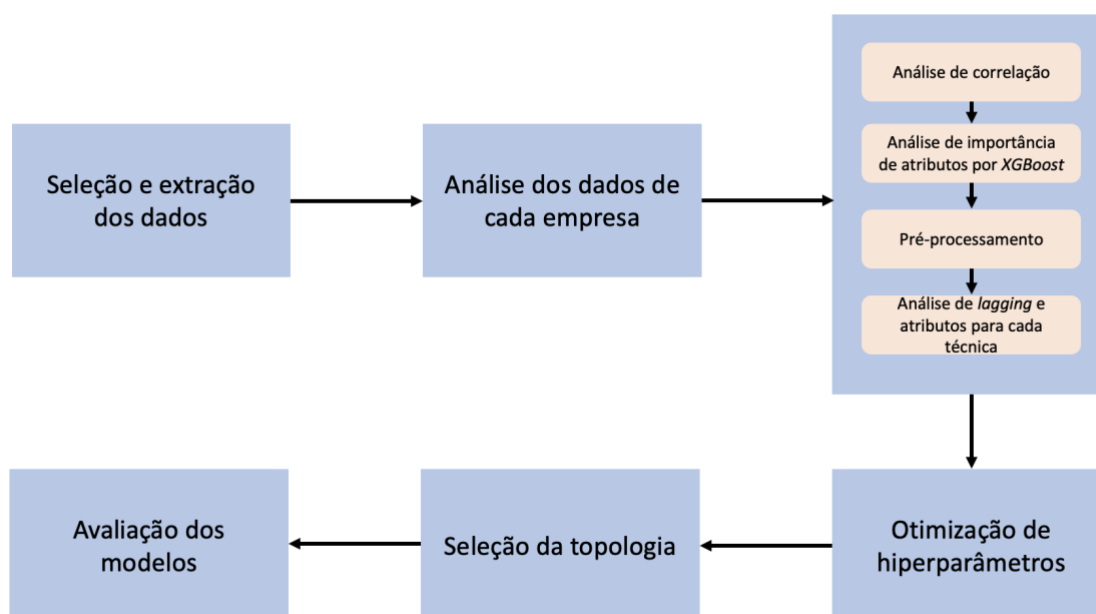
Por fim, Sezer, Gudelek, e Ozbayoglu (2020) fazem uma revisão sistemática da literatura a respeito do uso de redes neurais para a predição de séries temporais financeiras. Dentro deste estudo, as métricas mais utilizadas pelos estudos avaliados são as métricas relativas: *mean squared error* e *root mean squared error*, além do *mean absolute error*.

3. METODOLOGIA DE PESQUISA E PROCEDIMENTOS METODOLÓGICOS

O principal objetivo desta pesquisa foi avaliar o desempenho das técnicas de inteligência artificial (MLP e LSTM) na tarefa de prever o preço futuro das ações das cinco empresas mais representativas do IBOVESPA. Neste contexto, buscou-se avaliar qual é a técnica, o conjunto de hiperparâmetros e a técnica de extração de atributos que proporcionam a regressão, com menor erro, aplicadas a cada ação específica e a todas ações de forma conjunta. Dessa forma, buscou-se obter modelos ótimos para prever cada cenário das ações e um modelo para prever os cenários de todas empresas avaliadas.

Os procedimentos metodológicos foram organizados em sete etapas que estão apresentadas na Figura 13. Na etapa 1, foi realizada a seleção e extração dos dados utilizados. A etapa 2 consistiu em analisar os dados de cada empresa. Na etapa 3, foram realizados testes à respeito dos atributos presentes na base de dados. Assim sendo, o terceiro teste atuou de maneira a validar os resultados obtidos anteriormente e, para isto, foi necessário a realização do pré-processamento da base de dados, caracterizando assim, a etapa 4. A otimização dos hiperparâmetros ocorreu na etapa 5. Na etapa 6, foi realizada a seleção dos hiperparâmetros ótimos, dentro das opções testadas, para cada cenário. Por fim, na etapa 7, ocorreu o treinamento dos modelos selecionados e a avaliação de desempenho dos mesmos.

Figura 16 – Fluxograma descritivo do método utilizado.



Fonte: Elaborada pelo autor.

Este estudo se limitou a utilizar as cinco ações mais representativas da bolsa de valores, com exceção da PETR3¹, que ocuparia a posição de número cinco, porém, na prática, é a mesma empresa representada pela PETR4² e, portanto, foi desconsiderada da amostra. Para completar a amostra, foi considerada a sexta empresa mais representativa, a B3. Na Tabela 1, apresenta-se cada ação e seu respectivo peso dentro do cálculo do índice IBOVESPA, extraídos no dia 15/03/2022 do site da B3.

Tabela 1 - Empresas selecionadas para serem utilizadas neste estudo.

Ação	Participação (%)
Vale (VALE3)	15,583
Petrobras (PETR4)	6,864
Itaú Unibanco (ITUB4)	5,662
Bradesco (BBDC4)	4,606
B3 (B3SA3)	3,953

Fonte: Elaborado pelo autor (2022).

¹ PETR3 são ações ordinárias da empresa Petrobras.

² PETR4 são ações preferências da empresa Petrobras.

3.1 Seleção e extração da base de dados

De acordo com a seção 2.2 e tendo em vista o objetivo geral desta pesquisa, utilizou-se o grupo de dados brutos através de um problema de regressão para a execução desta pesquisa. Os dados históricos referentes as empresas foram extraídos do *Yahoo Finance*, que é uma plataforma centralizadora de informações referente ao mercado de capitais. Foram extraídos dados durante o período de 01/01/2010 até 30/12/2021. A escolha do período analisado deu-se em razão da disponibilidade da quantidade máxima de informações disponíveis na fonte selecionada. A Tabela 2 demonstra as empresas e as quantidades de registros extraídos, sendo que cada registro representa um dia útil de negociações.

Tabela 2 - Tamanho das bases de dados por empresa.

Ação	Número de registros
Vale (VALE3)	2976
Petrobras (PETR4)	2976
Itaú Unibanco (ITUB4)	2976
Bradesco (BBDC4)	2976
B3 (B3SA3)	2976

Fonte: Elaborado pelo autor.

O conjunto de dados foi separado em atributos de entrada e variável de interesse. Os valores de entrada são idênticos aos apresentados no primeiro grupo. Os valores de data de negociação não foram considerados neste estudo, tal conclusão explica-se pelo fato de que o valor é apenas uma representação temporal das negociações como um todo e, portanto, não caracterizam informações da empresa em questão. No Quadro 1, apresenta-se os atributos de entrada, com descrição e sua representação matemática, enquanto a variável de saída, apresentada no Quadro 2, é o preço de fechamento em $D+1$, representada por Y_1 .

Quadro 1 - Descritivo dos atributos de entrada.

Representação Matemática	Variável (R\$)	Descrição
X1	Preço de abertura	Variável que determina o preço da ação, no momento de abertura do mercado no dia.
X2	Preço máximo	É o preço máximo da ação no dia.
X3	Preço mínimo	É o preço mínimo da ação no dia.
X4	Preço de fechamento	É o preço de fechamento da ação no dia
X5	Preço de fechamento ajustado	Preço de fechamento após ajustes corporativos
X6	Volume transacionado	Volume de reais negociados por essa ação

Fonte: Elaborado pelo autor.

Quadro 2 - Atributo de saída.

Representação Matemática	Variável (R\$)	Descrição
Y1	Preço de fechamento D+1	Preço de fechamento da ação no dia seguinte (D+1)

Fonte: Elaborado pelo autor.

3.2 Ferramentas Computacionais

Neste trabalho os modelos foram construídos utilizando a linguagem de programação *Python 3* através do *Google Collaboratory*, também conhecido como *Google Colab*, que é um serviço de computação em nuvem fornecido pela Google (Alphabet Inc, Califórnia, EUA). Esse ambiente possui, nativamente, a linguagem de programação *Python 3* instalada e auxilia no desenvolvimento de aplicações de inteligência artificial, pois ele permite o uso de plataformas computacionais multicore, como as GPU's e TPU's, que são ambientes de processamento com um custo mais elevado e menos acessíveis a pessoas físicas.

Para tratamento e visualização dos dados foi utilizado as bibliotecas de código aberto: *Pandas*, *Numpy*, *SKLearn*, *Matplotlib*, *Seaborn* e *Talos*. O Quadro 3 apresenta os recursos utilizados e suas respectivas versões.

Quadro 3 - Bibliotecas e frameworks utilizados.

Ferramentas	Versão
Python	3.7.13
Pandas	1.3.5
Numpy	1.21.6
SKlearn	1.0.2
Tensorflow/Keras	2.8.2
Matplotlib	3.5.0
Talos	1.3.0
Seaborn	0.11.2

Fonte: Elaborado pelo autor.

Pandas é uma biblioteca de manipulação de dados e uma das mais usadas no ambiente de inteligência artificial. Ela permite a leitura de diversos formatos de arquivos que são carregados para a memória do computador através de um *dataframe*, que é uma representação bidimensional de dados, ou seja, uma planilha. Já *Numpy* utiliza métodos algébricos para manipulação de matrizes e vetores, sendo extremamente necessário para fazer a conversão do *dataframe* para um *array* (Estrutura de dados vetorizada), que é o formato usado para alimentar o algoritmo. O *SKLearn* é pacote de *machine learning* que catalisa algumas tarefas necessárias para o uso adequado dos dados, como a normalização e a divisão em conjunto teste e treinamento, enquanto o *Matplotlib* e *seaborn*, são bibliotecas de representação gráfica dos dados, exibindo assim os resultados e análises obtidas durante a pesquisa. Por último, *Talos* é uma biblioteca de código aberto que permite a busca por hiperparâmetros de maneira distribuída.

Para desenvolver os modelos de inteligência artificial propostos, foi utilizado a plataforma de código aberto: *Tensorflow*. Essa é uma solução de ponta-a-ponta no que diz respeito ao aprendizado de máquina. Dentro dela, o pacote mais utilizado foi o *Keras* que disponibiliza métodos para a criação de MLPs e LSTMs.

Por fim, foi fixado a semente aleatória e a plataforma computacional utilizada foi a família de *Cloud TPU V2* do *Google Colab* (fornecida gratuitamente). Segundo o *Google*, empresa fornecedora do serviço, a disponibilidade do hardware varia conforme a demanda, sendo assim, a única maneira de continuar fornecendo o serviço sem custos. Existem 4 versões

disponíveis de TPU para uso no google colab, são elas: *Cloud TPU V2/V3*, e *Cloud TPU v2/v3 POD*.

3.3 Análise da base de dados

A fim de compreender e caracterizar como foi o comportamento das ações ao longo do tempo e a relação entre as empresas avaliadas, foi necessário realizar uma análise sobre as bases de dados selecionadas. Tal análise foi realizada em duas etapas.

A primeira etapa consistiu em analisar os valores de máximo, mínimo e médio do preço de abertura e fechamento para cada empresa, assim como o volume médio para cada cenário. Além disso, buscou-se analisar graficamente o comportamento do preço de fechamento de cada ação no período analisado.

Por fim, a segunda etapa consistiu em analisar o comportamento do retorno diário de cada empresa. Para tal, realizou-se a diferença do preço de fechamento no dia d com o preço no dia $d-1$. Posterior a isto, representou-se os valores obtidos em um histograma, a fim de analisar o comportamento dos retornos diários.

3.4 Pré-processamento da base de dados

A adequação da base de dados ao problema estudado é fundamental para garantir um resultado satisfatório nos modelos de *machine learning*. Esse procedimento é conhecido como pré-processamento, pois antecede a criação do modelo, e realiza processamentos que modificam a base para o problema em questão. Nesta pesquisa o pré-processamento foi dividido em três etapas, são elas: normalização, aplicação do *lagging* e divisão do conjunto de dados.

3.4.1 Normalização

Conforme abordado no referencial teórico, as redes neurais podem ser consideradas como algoritmos matemáticos que necessitam que os dados alimentados estejam em uma mesma escala para evitar que medidas com escalas maiores sejam mais significativas e também reduzir sua redundância, aumentando a integridade e desempenho das análises realizadas. Essa etapa é conhecida como normalização. Existem diversas técnicas de normalização, como a utilização da média e do desvio padrão e com o uso do valor máximo e mínimo de cada atributo.

A técnica utilizada neste trabalho é a do valor máximo e mínimo, fornecida pelo método *MinMaxScaler* da biblioteca *Sklearn* em *Python 3*.

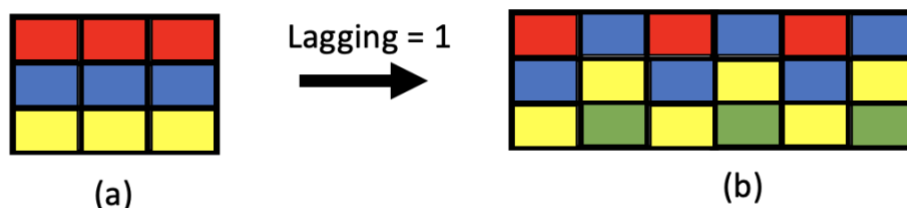
Os atributos (entrada e saída) foram normalizados em um intervalo de 0 a 1, seguindo a Equação 1. O resultado obtido pela rede também se encontra na mesma escala disponibilizada no treinamento e, para avaliar o desempenho do modelo em questão, foi necessário a desnormalização dos dados após a construção do mesmo. Esta etapa também foi realizada através do método *MinMaxScaler* do *SKLearn*.

3.4.2 Lagging

Por se tratar de séries temporais, o valor do dia (d) e o valor do dia seguinte ($d+1$) tem uma relação natural de causalidade e, portanto, é preciso adaptar a base de dados para que ela demonstre essa relação de dependência. Essa prática é conhecida como *lagging*, que se refere a quantos intervalos de tempo são passados para o modelo. Para um *lagging* igual a 3, o insumo para o modelo prever o valor do dia (d), passa a ser as informações referentes a $d-2$, $d-1$ e $d-0$, e aplica-se a todas as informações da base de dados, portanto, a predição do dia $d+2$, passa a ter a informação do dia $d+1$, d , $d-1$.

Por fim, o *lagging* foi um parâmetro a ser otimizado e encontrado para cada cenário proposto na seção subsequente. A Figura 17 ilustra uma base de dados aplicada a um *lagging* equivalente a um. Neste cenário, cada linha de cada cor representa um dia de informações, assim sendo, quando aplicado o *lagging*, cada linha de informação conterá informações da linha vermelha e azul, como exemplificado na Figura 17. Esse comportamento é expandido para todas as linhas do conjunto de dados.

Figura 17 - Ilustração do *lagging*.



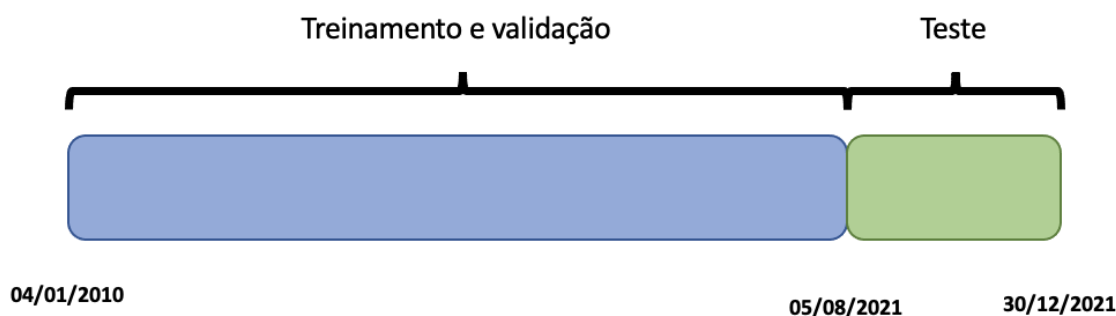
Fonte: Elaborado pelo autor.

3.4.3 Divisão do conjunto de dados

Para garantir a generalização do modelo construído, foi necessário a separação do conjunto de dados total em dois conjuntos: treinamento e validação (1) e conjunto de teste (2). O primeiro deles é o conjunto de dados que é exibido ao modelo durante o treinamento, ou seja, é com base nesse universo de dados que o modelo aprende. O segundo é o universo de dados utilizados para avaliar o desempenho do mesmo. Vale lembrar que o conjunto de teste somente é utilizado quando definido o modelo que melhor desempenhou durante o treinamento. Por fim, foi utilizada a técnica de *cross-validation* durante os treinamentos, que corresponde a uma porção de 10% do conjunto de treinamento total em cada iteração.

Como conjunto de treinamento, foram utilizados dados históricos que compreendem o período de 04/01/2010 até 05/08/2021. Para testar o resultado, utilizou-se os últimos 100 dias úteis presentes na base de dados, contendo as informações de 06/08/2021 até 30/12/2021. A Figura 18 ilustra como foi feito a divisão dos dados.

Figura 18 - Divisão do conjunto de dados.



Fonte: Elaborado pelo autor.

3.5 Seleção de atributos e testes preliminares na base de dados

A fim de obter o melhor conjunto de atributos, ou seja, os atributos que possuem a maior quantidade de informações pertinentes aos modelos e com a menor sobreposição possível para cada técnica, foi-se necessário a realização de testes sobre os mesmos. Todos os testes foram

realizados em todas as ações analisadas. Os testes foram divididos em três etapas, apresentadas no Quadro 4.

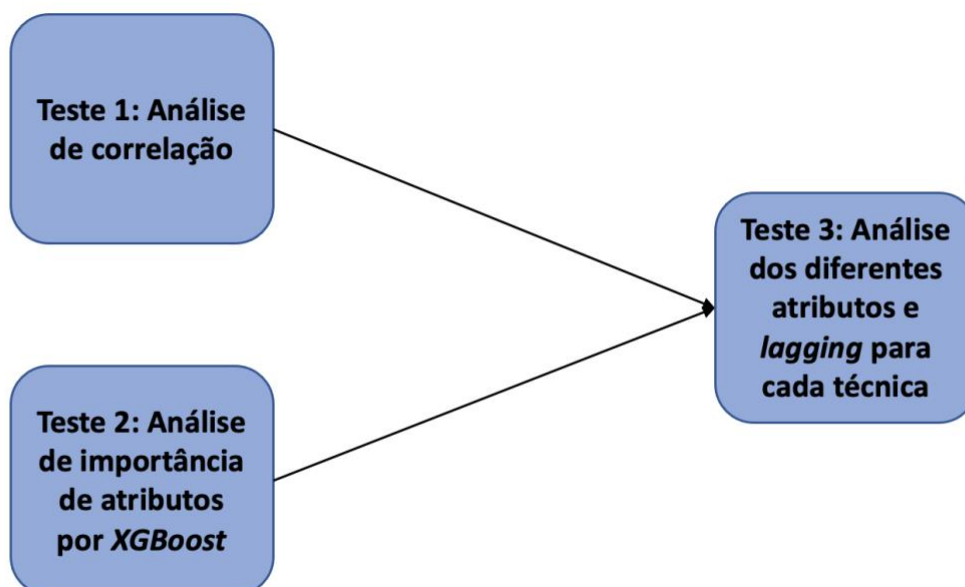
Quadro 4 - Otimizações e testes preliminares.

Teste	Descrição	Objetivo	Base de dados
1	Análise de correlação	Analisar a correlação entres os atributos considerados, permitindo ou não, o descarte de atributos semelhantes	Sem tratamentos (Atributos de entrada e variável de interesse D+1)
2	Análise de importância de atributos por XGBoost	Analisar a importância de cada atributo para um modelo genérico do <i>XGBoost</i> , permitindo ou não, o descarte de atributos não relevantes.	Sem tratamentos (Atributos de entrada e variável de interesse D+1)
3	Análise dos diferentes atributos e <i>lagging</i> para cada técnica	Analisar se o descarte de atributos produzem um resultado melhor da regressão, assim como obter o melhor valor de <i>lagging</i> para cada cenário proposto.	Base normalizada, com <i>lagging</i> variável

Fonte: Elaborado pelo autor.

A Figura 19 ilustra como os testes se complementam. O primeiro e o segundo teste são responsáveis pelas primeiras evidências. Os resultados destes testes foram utilizados para gerar os cenários avaliados no terceiro teste, responsável por validar os resultados obtidos nas etapas anteriores.

Figura 19 - Diagrama dos testes aplicados a base de dados.



Fonte: Elaborado pelo autor.

3.5.1 Análise de correlação

Este teste consistiu em uma análise de correlação entre os atributos de entrada e a variável de interesse (D+1). Tal experimento teve como objetivo de avaliar se devido a uma alta similaridade entre os atributos, houvesse cenários onde a base de dados pudesse ser reduzida. O teste foi realizado aplicando o método de *Pearson*. A Equação 6 demonstra o cálculo. O valor de n é a quantidade de amostras, enquanto os valores indexados são os valores das amostras e os barrados como a média de todas as amostras.

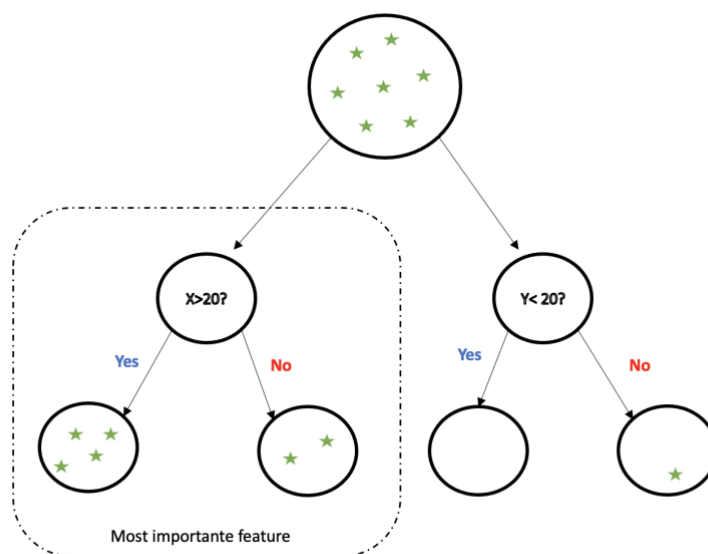
$$\rho = \frac{\sum_{i=1}^n (X_i - \bar{X}) * (Y_i - \bar{Y})}{\sqrt{[\sum_{i=1}^n (X_i - \bar{X})^2][\sum_{i=1}^n (Y_i - \bar{Y})^2]}} \quad (6)$$

3.5.2 Análise de importância de atributos por *XGBoost*

Esta etapa compreendeu-se na utilização do *XGBoost*, algoritmo de árvores de decisões baseados no gradiente descendente, para extração da importância dos atributos avaliados. Como abordado na seção 2.3.2 do referencial teórico, o *XGBoost* utiliza o conceito de *gain*, que é o ganho que cada segmentação gera no modelo durante o treinamento. Em suma, a importância dos atributos são calculadas de acordo com o ganho que suas segmentações geram para o resultado do modelo. A Figura 20 ilustra o exemplo de um algoritmo com sete exemplos, onde a segmentação de um atributo levou a classificação ou regressão de seis exemplos, assim sendo, esse atributo obtém uma importância maior quando comparado aos outros atributos.

Neste procedimento, foi utilizado os atributos de entrada e a variável de interesse (D+1). Como foi abordado na seção 4.2, foi-se necessário a realização de um teste final com o objetivo de validar as inferências realizadas dos testes anteriores.

Figura 20 - Etapas de como extrair o ganho de um atributo.



Fonte: Elaborado pelo autor.

3.5.3 Análise do *lagging* e atributos nas técnicas selecionadas

O terceiro teste se caracterizou por arbitrar um modelo genérico de cada técnica proposta, com hiperparâmetros constantes, variando, assim, as diferentes possibilidades dos atributos considerados, assim como o valor de *lagging* em cada cenário. Para a realização de tal, foi necessário submeter o pré-processamento da base de dados, que fora explicada na seção 3.4. Ressalta-se que, devido a restrições de tempo e de processamento, foi necessário restringir o escopo destes testes a um conjunto genérico de hiperparâmetros e não a uma procura entre todos os universos possíveis dentro dos valores estabelecidos, como explorado na seção 3.6.

Em suma, para cada ação, treinou-se o modelo com o conjunto de treinamento e avaliou-se o desempenho com o conjunto de teste. As métricas de avaliação consideradas foram o *root mean squared error* e o *mean absolute error*.

Para cada execução, alternou-se a base de dados, contendo todos os atributos de entrada e outra contendo apenas o preço de fechamento (X4), além da quantidade de *lagging* aplicada à mesma. Analisou-se um *lagging* variando de 3 dias a 60 dias atrás. Cada cenário foi executado três vezes, obtendo a média de cada execução como métrica final. O Quadro 5 apresenta os modelos utilizados tanto para MLP quanto LSTM, enquanto o Quadro 6 ilustra o teste.

Quadro 5 - Modelo avaliado tanto na MLP quanto na LSTM.

Hiperparâmetro	Valor
Camadas Ocultas	1
Neurônios	128
Função de ativação	Relu
Épocas	10
Tamanho de Lote	32

Fonte: Elaborado pelo autor.

Quadro 6 - Valores avaliados dos hiperparâmetros da MLP.

Empresa	Base de Dados	Lagging
Todas (B3, Itaú, Bradesco, Vale e Petrobras)	Todos atributos	[3 - 60]
	Somente preço de fechamento	[3 - 60]

Fonte: Elaborado pelo autor.

Por fim, os resultados destes testes tiveram o propósito de avaliar qual é o conjunto de atributos que possibilitam o melhor desempenho para cada técnica. Além disso, permitiu concluir o *lagging* que obteve o melhor desempenho, em média, para todas as ações avaliadas, assim como qual é o melhor valor de *lagging* para cada empresa específica. A partir deste momento, considerou-se o modelo generalista que foi construído através dos parâmetros extraídos pela média de todas execuções, não diferindo sobre qual é a empresa correspondente, enquanto o modelo especialista³ é o construído através dos melhores parâmetros obtidos para cada empresa. O Quadro 7 ilustra os modelos que foram considerados.

Quadro 7 - Diferença entre modelos considerados.

Técnica	Tipo de Modelo	Base de Dados	Lagging
MLP/LSTM	Modelo Generalista	Conjunto de atributos mais aderente a técnica correspondente	Melhor <i>lagging</i> , em média, de todas execuções
	Modelo Especialista	Conjunto de atributos mais aderente a técnica correspondente	Melhor valor de <i>lagging</i> por empresa

Fonte: Elaborado pelo autor.

³ Especialista, neste contexto, é o modelo utilizado para representar o estudo referente a uma empresa específica.

3.6 Otimização de hiperparâmetros

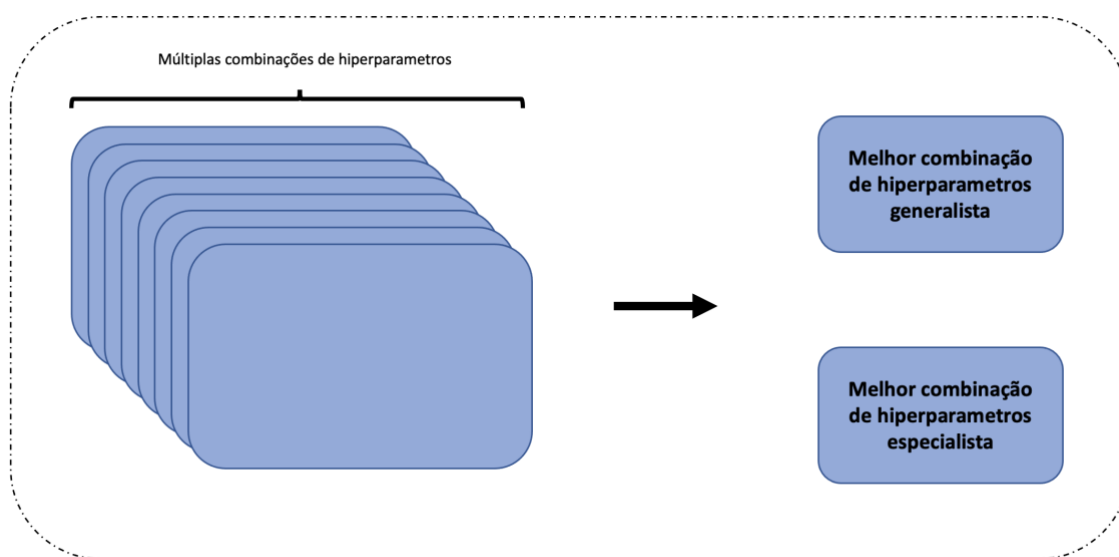
Após a realização dos testes anteriores, que permitiram a conclusão sobre a base de dados e o *lagging*, precisou-se extrapolar o universo de modelos possíveis dentro das técnicas selecionadas. Este processo é conhecido como otimização de hiperparâmetros e tem como objetivo encontrar os melhores parâmetros dado a base de dados em questão. Existem diversas técnicas para realização deste experimento. Nesta pesquisa, foi realizada uma otimização em formato de grade, cruzando todos os valores possíveis de hiperparâmetros entre si, aumentando assim a quantidade de possibilidades e, conseqüentemente, o tempo e o custo deste processo. Para atingir tal objetivo, foi utilizado o pacote *Scan* da biblioteca *Talos*, que possibilita uma busca de maneira distribuída entre hardwares *multi-threads*.

A fim de excluir qualquer dependência e fator de impacto que um hiperparâmetro possa agir em outro, influenciando ou não no resultado, foi realizada a procura dos valores ótimos através de todas as combinações possíveis dos valores estipulados. Portanto, foi necessário reduzir a quantidade de valores avaliados, uma vez que qualquer valor adicionado duplica as possibilidades que devem ser consideradas. Além disso, de acordo com Luo *et al.*, (2019), o ADAM é uma versão aprimorada do *Stochastic gradient descent* e do *Root Mean Square Propagation* (RMSProp) e, para restringir o limite de possibilidades, foi considerado como o otimizador escolhido em ambas as técnicas. Ressalta-se que este mesmo hiperparâmetro poderia ser um caso de estudo à parte, porém, não é o foco do presente estudo.

Além disso, devido a limitação de tempo e recurso deste trabalho, estipulou-se valores arbitrários para tais hiperparâmetros e, portanto, reforça a necessidade de uma busca maior entre o universo de valores para uma otimização completa (idealmente testando para infinitos valores).

A otimização dos hiperparâmetros está dividida em duas etapas. A primeira delas é a procura pelo modelo generalista, ou seja, aquele que, na média, consegue prever melhor os valores de todas as ações. Já a segunda etapa, consiste em procurar o modelo especialista a cada empresa, aplicando assim, os valores obtidos nos testes anteriores. A Figura 21 ilustra o processo de otimização aplicado às técnicas avaliadas neste estudo, as *Multilayer Perceptron* e as *Long Short-Term memory*.

Figura 21 - Etapas da otimização de hiperparâmetros.



Fonte: Elaborado pelo autor.

3.6.1 Otimização MLP

Os valores e parâmetros considerados para este processo são apresentados no Quadro 8 enquanto o Quadro 9 apresenta algumas das possibilidades analisadas que resultam em um total de 72 modelos diferentes para avaliação. Todas as combinações podem ser vistas no apêndice A.

Quadro 8 - Valores avaliados dos hiperparâmetros da MLP.

Hiperparâmetro	Valor
Camadas Ocultas	1/2
Neurônios	32/64/128
Função de ativação	Tanh/Relu/Sigmoid
Épocas	50/100
Tamanho de lote	16/32

Fonte: Elaborado pelo autor.

Quadro 9 - Algumas combinações de hiperparâmetros avaliadas para MLP.

Combinação de hiperparâmetros	Número de Camada Ocultas	Quantidade de Neurônios nas camadas ocultas	Função de pertinência da camada oculta	Número de Épocas	Tamanho do Lote
1	1	8	Tanh	50	16
2	1	8	Tanh	100	16
3	1	8	Relu	50	16
4	1	8	Relu	100	16
5	1	8	Sigmoid	50	16
6	1	8	Sigmoid	100	16
7	1	32	Tanh	50	16
8	1	32	Tanh	100	16
9	1	32	Relu	50	16
10	1	32	Relu	100	16

⋮

62	2	32	Tanh	50	32
63	2	32	Relu	100	32
64	2	32	Relu	50	32
65	2	32	Sigmoid	100	32
66	2	32	Sigmoid	50	32
67	2	128	Tanh	100	32
68	2	128	Tanh	50	32
69	2	128	Relu	100	32
70	2	128	Relu	50	32
71	2	128	Sigmoid	100	32
72	2	128	Sigmoid	50	32

Fonte: Elaborado pelo autor.

O processo de otimização utilizado foi aplicado para ambos os modelos, tanto genérico quanto o especialista (de todas as empresas). Variou-se apenas o *lagging* em cada cenário, assumindo o valor devido em cada cenário, especialista ou generalista. Devido a natureza estocástica do processo, foi executado para cada combinação e ação avaliada, três vezes o

mesmo cenário. Para isso, utilizou-se o conjunto de treinamento e validação, mencionados anteriormente. Em seguida, foi obtido o erro quadrático médio (MSE) do conjunto de treinamento e validação para cada combinação testada. O Quadro 10 ilustra os cenários.

Quadro 10 – Cenários de otimizações para MLP e LSTM.

Execuções	Métrica	Modelo	Lagging	Base de dados
3	Erro Quadrático Médio	Generalista	Lagging Médio	B3/Petrobras/Vale/Itaú/Bradesco
		Especialista	B3/Petrobras/Vale/Itaú/Bradesco	B3/Petrobras/Vale/Itaú/Bradesco

Fonte: Elaborado pelo autor.

3.6.2 Otimização LSTM

Conforme foi apresentado na abordagem do referencial teórico, as arquiteturas da LSTM são denominadas de acordo com a quantidade de camadas ocultas e o tipo do neurônio presente na mesma e, portanto, devem ser otimizadas com suas particularidades. Neste trabalho, foi considerada as arquiteturas tradicionais da LSTM, como a *vanila*, *stacked* e bidirecional. O Quadro 11 ilustra as peculiaridades de cada arquitetura adotada na presente pesquisa.

Quadro 11 - Arquiteturas avaliadas da LSTM.

Arquitetura	Camada Ocultas	Neurônio
Vanila	1	Normal
Stacked	2	Normal
Bidirecional	1	Bidirecional

Fonte: Elaborado pelo autor.

Devido à maior complexidade das *Long Short-Term Memory* e a quantidade maior de arquiteturas, foi preciso reduzir a quantidade de neurônios avaliados para apenas dois valores, ante três valores utilizados nas *Multilayer Perceptrons*. Para cada arquitetura apresentada no

Quadro 11, testou-se os hiperparâmetros e seus valores, expostos no Quadro 12, resultando assim, em 72 combinações distintas para avaliação, algumas mostradas no Quadro 13 (todas combinações podem ser vistas no apêndice B).

Quadro 12 - Valores avaliados dos hiperparâmetros da LSTM.

Hiperparâmetro	Valores
Neurônios	32/128
Função de ativação	Tanh/Relu/Sigmoid
Épocas	100/200
Tamanho de Lote	16/32

Fonte: Elaborado pelo autor.

Quadro 13 – Algumas das combinações de hiperparâmetros avaliadas na LSTM.

Combinação de hiperparâmetros	Arquitetura	Quantidade de Neurônios	Função de Pertinência da camada oculta	Número de Épocas	Tamanho do Lote
1	Vanila	32	Tanh	100	16
2	Vanila	32	Tanh	200	16
3	Vanila	32	Relu	100	16
4	Vanila	32	Relu	200	16
5	Vanila	32	Sigmoid	100	16
6	Vanila	32	Sigmoid	200	16
7	Vanila	128	Tanh	100	16
8	Vanila	128	Tanh	200	16
9	Vanila	128	Relu	100	16
10	Vanila	128	Relu	200	16

·
·
·
·

62	Bidirecional	32	Tanh	200	32
63	Bidirecional	32	Relu	100	32
64	Bidirecional	32	Relu	200	32
65	Bidirecional	32	Sigmoid	100	32
66	Bidirecional	32	Sigmoid	200	32
67	Bidirecional	128	Tanh	100	32
68	Bidirecional	128	Tanh	200	32
69	Bidirecional	128	Relu	100	32
70	Bidirecional	128	Relu	200	32
71	Bidirecional	128	Sigmoid	100	32
72	Bidirecional	128	Sigmoid	200	32

Fonte: Elaborado pelo autor.

Por fim, o processo de otimização foi idêntico ao caracterizado na seção anterior e, portanto, a representação do Quadro 10 se aplica a esta seção também. Mudou-se apenas a técnica aplicada, junto com a base de dados e *lagging* encontrados nos testes preliminares.

3.7 Seleção dos hiperparâmetros

Para cada combinação possível apresentada na seção 3.6, algumas ilustradas nos Quadros 09 e 13, foi executado o mesmo ensaio três vezes, obtendo assim, para cada execução, o resultado do erro a respeito do conjunto de validação e treinamento. Neste cenário, foi utilizada a técnica de *cross-validation*, que consiste em utilizar uma parcela (10%) do conjunto de treinamento, como ilustrado na seção 3.4.3, para validar o modelo após o término do treinamento. Vale lembrar que esse conjunto não é utilizado durante o treinamento e, portanto, possui um papel semelhante ao conjunto de teste.

Por fim, foi realizado uma média do erro apresentado no conjunto de validação para cada modelo proposto, selecionando assim, a combinação que apresentou o menor erro médio. A métrica utilizada foi o *mean squared error*, cujo cálculo é apresentado na Equação 3.

3.8 Avaliação de desempenho

Para avaliar o real desempenho de modelos de inteligência artificial, é preciso submetê-los a um conjunto de dados inédito. Este conjunto de dados é denominado de conjunto de teste. Como exposto na seção 3.4, o conjunto de testes contém 100 dias para predição da rede. Vale lembrar que a base de teste sofre o mesmo *lagging* aplicado ao treinamento e, portanto, as informações são retroalimentadas para a rede. Assim sendo, a variável de interesse é sempre $D+1$, pois o modelo, necessariamente, precisa da informação do dia anterior para predição do próximo dia de negociações. Para avaliar o resultado dos modelos, foi preciso desnormalizar os resultados para a escala original, permitindo assim, a comparação entre os valores preditos (saída do modelo) e os originais.

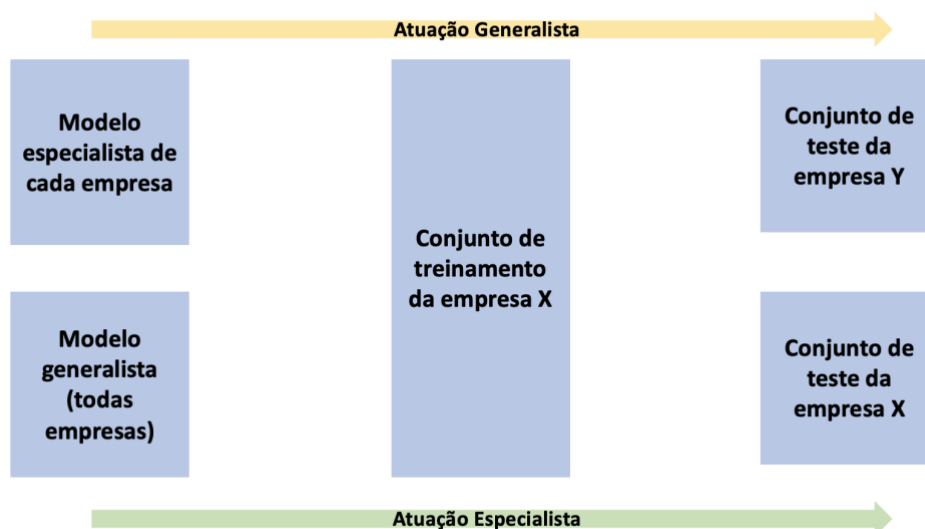
Por fim, para auferir o desempenho do modelo em completude, precisou-se dividir em duas etapas. A primeira delas é uma avaliação quantitativa e utilizou-se de métricas de erros para mensurar a performance de cada modelo em questão. Já a segunda, por se tratar de informações práticas do dia-a-dia do mercado, é uma avaliação simulada e, portanto, fará uso de regras para decidir se um negociador deve comprar ou não determinada ação.

3.8.1 Avaliação quantitativa

A fim de avaliar se os modelos construídos atuam conforme o propósito de cada um, precisou-se avaliar os modelos com o mesmo método. Para atingir tal objetivo, foi avaliado os

modelos em dois cenários. A Figura 22 representa um diagrama contendo os dois cenários avaliados.

Figura 22 - Diagrama dos testes realizados.



Fonte: Elaborado pelo autor.

O primeiro cenário diz respeito ao desempenho de todos os modelos quando os mesmos são treinados com dados de empresas diferentes das quais são avaliados, por exemplo, quando o modelo genérico é treinado com os dados da empresa X e é avaliado através dos dados da empresa Y. Após isto, agrupou-se os resultados através da média do erro, através dos métodos *root mean squared error* e do *mean absolute error*, resultando assim, no erro médio de cada modelo atuando como generalista.

O principal objetivo dessa consolidação foi submeter o mesmo método a todos os modelos e avaliar se, de fato, o modelo generalista consegue, em média, apresentar um erro menor, quando comparado aos modelos específicos. O Quadro 14 resume o teste realizado.

Quadro 14 – Avaliação sob a perspectiva generalista para cada técnica.

Modelo	Treinamento	Conjunto Teste	Resultado
Especialista	Empresa X	Empresa Y	Média dos erros (RMSE/MAE)
Generalista	Empresa X	Empresa Y	Média dos erros (RMSE/MAE)

Fonte: Elaborado pelo autor.

O segundo cenário foi avaliar todos os modelos atuando conforme especialista. Para isto, utilizou-se cada modelo obtido da otimização e treinou-se para cada empresa específica, avaliando-o através do conjunto de testes. O Quadro 15 apresenta a avaliação realizada.

Quadro 15 - Avaliação sob a perspectiva especialista para cada técnica.

Tipo de modelo	Treinamento	Teste
Especialista/Generalista	Petrobras	Petrobras
	Vale	Vale
	Itaú	Itaú
	B3	B3
	Bradesco	Bradesco

Fonte: Elaborado pelo autor.

Por fim, o segundo teste permitiu encontrar o melhor resultado, dentro do contexto avaliado, para cada empresa e para cada técnica avaliada.

3.8.2 Avaliação simulada

A fim de avaliar a aplicabilidade dos resultados, foi feita uma simulação a partir dos resultados extraídos dos modelos. Para tal, foi realizada uma comparação entre duas metodologias de negociação. A primeira delas consiste em um investidor hipotético adepto a filosofia do *Buy and Hold*. Já a segunda, consistiu em investidores denominados *Traders*.

O investidor *buy and hold*, foi o investidor que terá comprado a ação no primeiro dia do conjunto de teste e vendido a mesma no último dia avaliado, assim sendo, define-se o retorno sobre o investimento (ROI) obtido para o *buy and holder*.

Já o investidor *Trader*, será aquele que seguirá o resultado do modelo dada uma determinada condição. Para efeitos de cálculo, será considerado que o *Trader* comprou com o preço de fechamento do dia (d) e que vendeu com o preço de (d+1) e, portanto, a diferença entre o que a rede previu e o valor real analisado no dia (d+1) é caracterizado o retorno sobre o investimento (lucro ou prejuízo) do indivíduo.

Para proteger o investidor de qualquer ruído e aleatoriedade na previsão dos modelos, precisou-se estipular condições que deveriam ser atendidas para que haja uma compra. Este conceito é conhecido como margem de segurança. Assim sendo, no cenário em que o valor predito suprisse a regra da margem de segurança estipulada o negociador compraria a ação, caso contrário, não. As regras foram pensadas em três investidores, são eles: arrojado, moderado e conservador. Os valores da margem foram baseados no comportamento histórico de retorno diário pela ação em questão, apresentados na seção 3.3. O Quadro 16 apresenta as regras utilizadas e o Quadro 17 ilustra um cenário hipotético.

Nesta etapa, foi considerado os melhores modelos de cada técnica, ou seja, os modelos que apresentaram o menor erro na seção 3.9.1. A simulação consiste em determinar se o negociante deve comprar ou não, baseado na previsão do valor de D+1 do modelo em questão.

Quadro 16 - Margens de segurança estipuladas pelo investidor.

Investidor	Valor predito
Conservador	$> 1.025 * \text{Valor (d)}$
Moderado	$> 1.012 * \text{Valor (d)}$
Arrojado	$> 1.005 * \text{Valor (d)}$

Fonte: Elaborado pelo autor.

Quadro 17 - Exemplo de atuação por tipo de investidor.

Exemplo	Retorno sobre o investimento (ROI)	Tipo do Investidor	Ação
Valor (d) = R\$ 40,00 Valor predito (d+1) = R\$ 42,32	5.8%	Conservador	Compra
		Moderado	Compra
		Arrojado	Compra
Valor (d) = R\$ 40,00 Valor predito (d+1) = R\$ 40,51	1.25%	Conservador	Nenhuma
		Moderado	Compra
		Arrojado	Compra
Valor (d) = R\$ 40,00 Valor predito (d+1) = R\$ 42,32	0,65%	Conservador	Nenhuma
		Moderado	Nenhuma
		Arrojado	Compra

Fonte: Elaborado pelo autor.

A metodologia foi desenvolvida para avaliar o comportamento das técnicas que tentaram prever o valor futuro das cinco empresas mais representativas do IBOVESPA. Além disso, procurou-se explorar as particularidades de cada técnica e os dados de cada empresa individual e coletivamente. Para tal, utilizou-se a técnica de *lagging* como extração de atributos e submeteu-se a testes preliminares, a fim de buscar o melhor *lagging* e a melhor configuração de atributos para a base de dados em cada técnica avaliada. Em seguida, realizou-se uma busca pelos hiperparâmetros ótimos aplicados a cada cenário proposto e analisou-se o resultado dos melhores candidatos sob uma ótica quantitativa e simulada. O capítulo 4 apresenta os resultados encontrados com tal metodologia.

4. RESULTADOS

Os resultados dos testes nesta pesquisa foram divididas em cinco partes, que são: Análise da base de dados; Seleção de atributos e testes preliminares; Otimização de hiperparâmetros; Seleção de topologia e Avaliação de desempenho.

4.1 Análise da base de dados

A fim de contextualizar a base de dados de cada empresa, extraiu-se os valores máximo, mínimo e médio do preço de fechamento, abertura e volume. Além disso, mapeou-se a quantidade de registros em cada uma delas. Foram retirados valores nulos e não numéricos que presentes na base de dados. Apenas a B3 apresentou tal cenário, sendo assim, não havia dados nos dias 23 de fevereiro e 03 de março de 2011.

As informações utilizadas são do primeiro dia útil de 2010 até o último dia útil de 2021. Os resultados obtidos são expostos na Tabela 3.

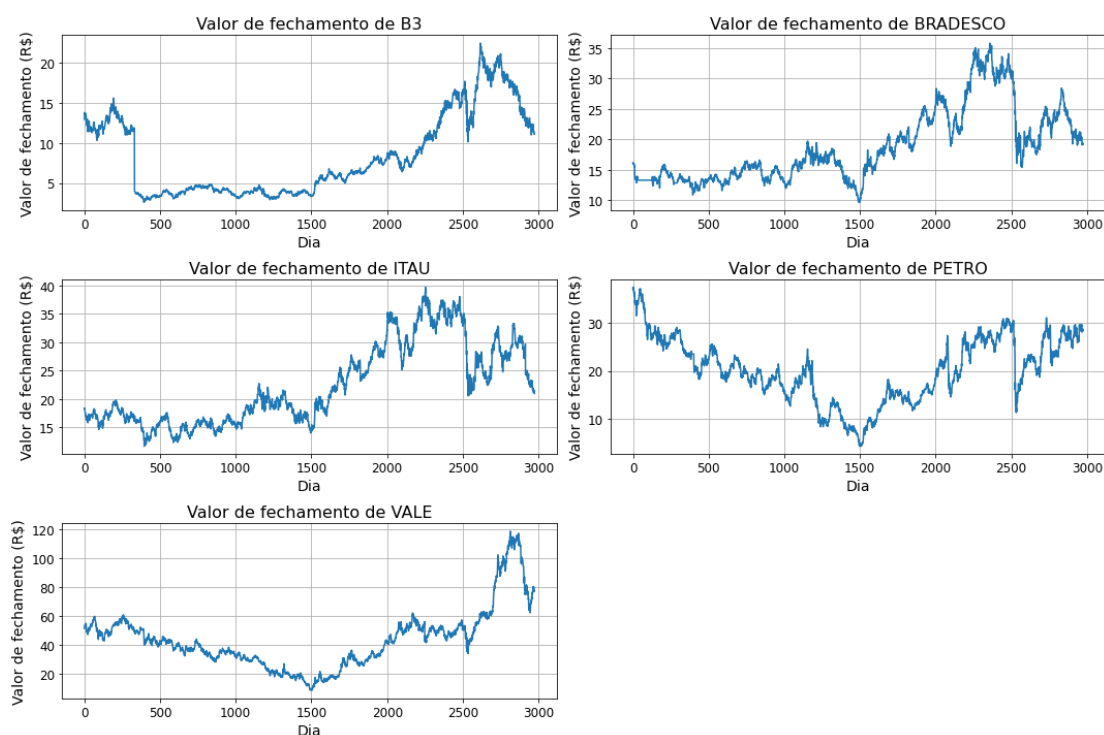
Tabela 3 - Comportamento das variáveis de interesse.

		B3	Petrobras	Bradesco	Vale	Itaú
Quantidade de registros (dias)		2973	2975	2975	2975	2975
Preço de fechamento	Máximo	R\$ 22,46	R\$ 37,50	R\$ 35,82	R\$ 118,72	R\$ 39,68
	Mínimo	R\$ 2,60	R\$ 4,20	R\$ 9,68	R\$ 8,60	R\$ 11,63
	Médio	R\$ 8,40	R\$ 20,18	R\$ 18,80	R\$ 43,54	R\$ 22,30
Preço de abertura	Máximo	R\$ 22,71	R\$ 37,43	R\$ 36,00	R\$ 119,80	R\$ 38,66
	Mínimo	R\$ 2,61	R\$ 4,20	R\$ 9,68	R\$ 8,75	R\$ 11,88
	Médio	R\$ 8,40	R\$ 20,21	R\$ 18,81	R\$ 43,58	R\$ 22,30
Volume	Médio	R\$ 33.831.050,00	R\$ 47.150.950,00	R\$ 20.631.370,00	R\$ 11.885.850,00	R\$ 22.646.290,00

Fonte: Elaborado pelo autor.

O principal objetivo deste trabalho é encontrar modelos que consigam entender o comportamento e, conseqüentemente, prever o valor da variável de interesse, que é o preço de fechamento no dia seguinte. A Figura 23 apresenta a variação do preço de fechamento, em reais, por dia de negociação para cada empresa considerada.

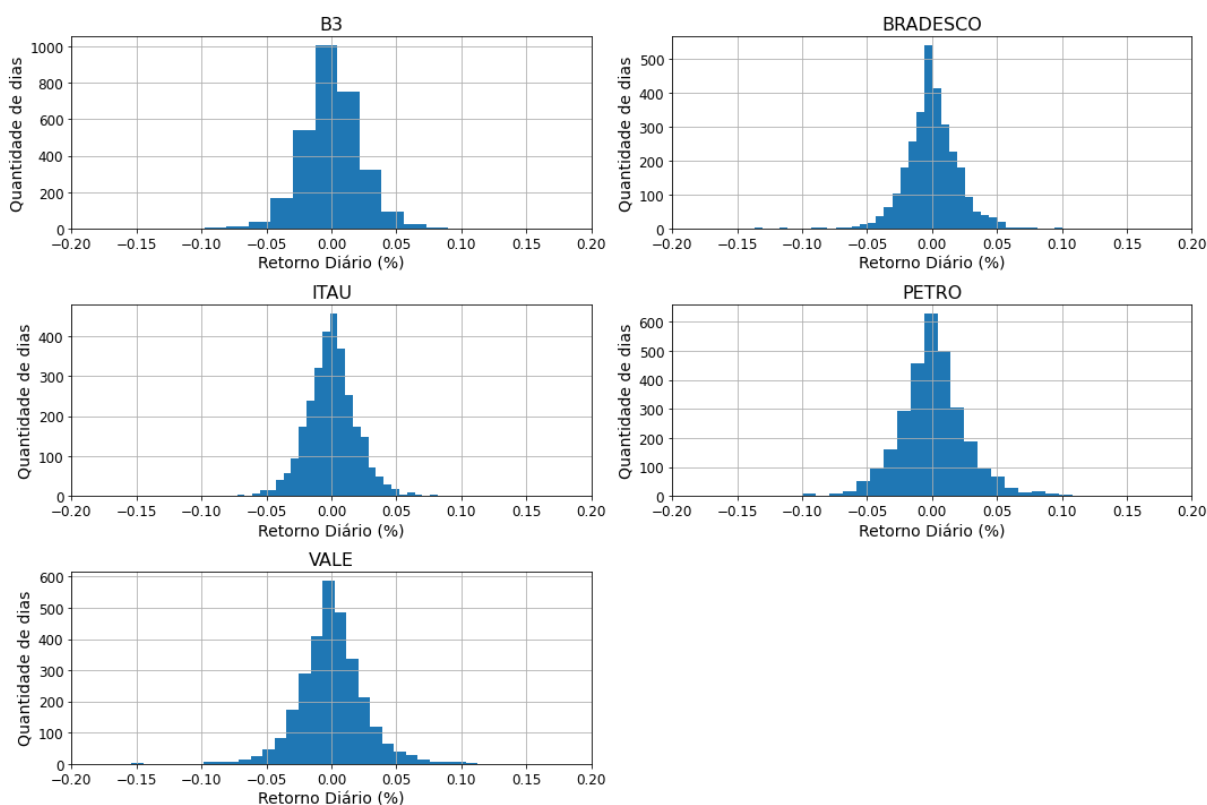
Figura 23 – Valor da ação ao longo do tempo.



Fonte: Elaborado pelo autor.

Além disso, considerando que os modelos tentam prever apenas um dia na frente, ou seja, são retroalimentados para prever o próximo dia, avaliou-se como é o retorno diário, de cada empresa, ao longo do tempo. O retorno diário, em termos percentuais, foi calculado fazendo a diferença entre o preço de fechamento no dia “ d ” pelo dia “ $d-1$ ”. Para representar o resultado, utilizou-se uma visão em histograma, a fim de entender como é a distribuição do retorno diário. O eixo X corresponde ao retorno diário, sendo positivo o cenário onde o preço de fechamento de d era maior que o de $d-1$.

Figura 24 - Distribuição do retorno diário de cada empresa.



Fonte: Elaborado pelo autor.

Nota-se que todos os gráficos têm forma similar ao de uma distribuição normal. No entanto, o mercado acionário, por se tratar de um agente caótico e muito suscetível a eventos não regulares, possuem uma difícil modelagem, como fora abordado na seção 2.1 da fundamentação teórica, e não podem ser ingenuamente classificados sob a ótica de uma distribuição normal. Cabe destacar que não é escopo deste trabalho discutir a distribuição dos retornos diários no mercado acionário que, por si só, poderia ser um caso de estudo à parte.

Nesse sentido, o interesse nesta análise é, a partir de evidências estatísticas, estabelecer uma margem de segurança para tipos de investidores considerados. O racional por trás dessa margem é estipular uma linha de corte para se o investidor deve agir ou não, dado o resultado do modelo em questão. Considerando o erro atrelado ao modelo e a não linearidade do mercado, a margem de segurança é um mecanismo que protege o investidor a ruídos e eventos não parametrizáveis e, portanto, o mesmo seguiria a previsão do modelo caso essa condição fosse cumprida.

Ao analisar os gráficos, nota-se que retornos acima de 2,5% são raros em todas as empresas consideradas, assim como a maior presença de retornos em torno do 0%. Para melhor validar tal comportamento, foi calculado, através da divisão da soma total de ocorrências pelo tamanho total do *dataset*, qual é a parcela de dias que se enquadram no cenário de um retorno diário superior a 0,5, 1,2 e 2,5%. Esse resultado é apresentado na Tabela 4.

Tabela 4 - Ocorrência dos retornos diários propostos.

Empresa	Retorno diário	Quantidade de dias	Percentual de dias
B3	0,50%	1181	39,74%
	1,25%	797	26,82%
	2,50%	366	12,31%
Bradesco	0,50%	1092	36,72%
	1,25%	681	22,90%
	2,50%	267	8,98%
Itaú	0,50%	1108	37,26%
	1,25%	677	22,76%
	2,50%	280	9,41%
Petrobras	0,50%	1192	40,08%
	1,25%	818	27,51%
	2,50%	428	14,39%
Vale	0,50%	1191	40,06%
	1,25%	786	26,44%
	2,50%	387	13,02%

Fonte: Elaborado pelo autor.

Os dados demonstram que, em média, os retornos acima de 2,5% ocorrem apenas em 10% do período analisado. Para retornos acima de 0,5% a ocorrência aumenta para 40% do

período analisado. Esses foram os valores utilizados para determinar a margem de segurança associada a cada perfil de investidor.

Nesse sentido, podemos fazer a leitura desse cenário da seguinte forma: caso haja previsão da rede de aumento no valor da ação acima de 2,5% para o dia seguinte em relação ao dia atual, todos os investidores agiriam. Se a previsão da rede estivesse acima de 1,2% do valor atual, somente os investidores moderados e arrojados agiriam e, se tivesse apenas 0,5% do valor atual, somente os investidores arrojados agiriam.

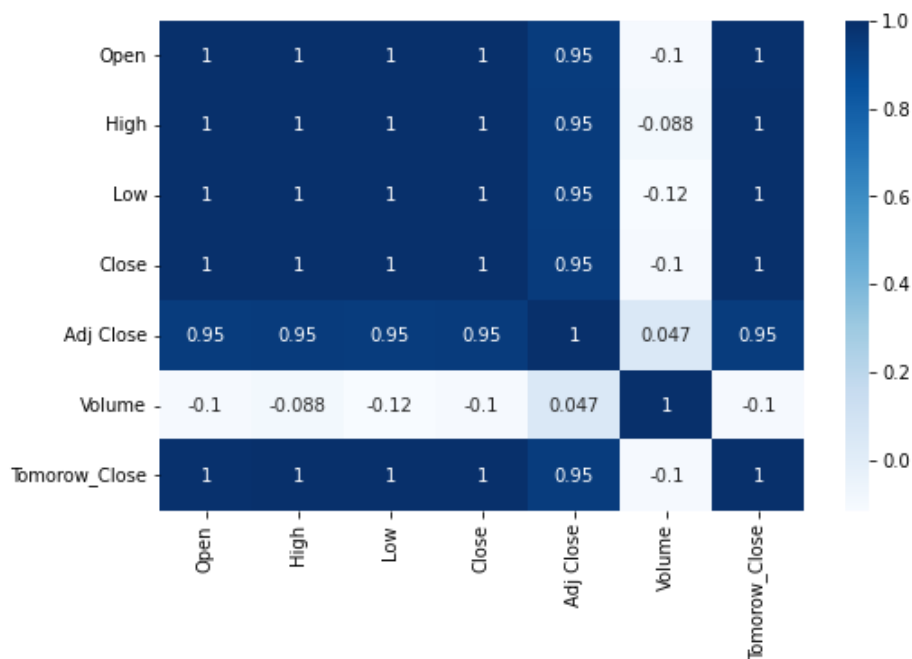
4.2 Seleção de atributos e testes preliminares

Após a análise dos dados e comportamento de cada empresa, foram realizados testes nas bases de dados (de cada empresa) para determinar o melhor conjunto de atributos para cada uma das técnicas (MLP/LSTM), assim como o melhor valor de *lagging*. A apresentação dos resultados destes testes foram divididos em três partes. A primeira foi a realização de um simples teste de correlação. O segundo teste, mais robusto, constou de treinar uma árvore de decisão com cada conjunto de dados e extrair o peso atribuído a cada um deles pelo modelo. Por fim, treinou-se um modelo genérico de cada técnica considerada neste estudo, com diferentes variações de atributos e *lagging*.

4.2.1 Análise de correlação dos atributos da base de dados

Para determinar a correlação entre os atributos, considerou-se todos os atributos e a variável de interesse, o preço de fechamento em D+. O teste foi realizado para cada empresa considerada neste estudo, porém, devido a similaridade dos resultados, decidiu-se por apresentar apenas o gráfico extraído da Petrobras. Os testes nas outras empresas ou foram idênticos, ou tiveram alterações na segunda casa decimal dos atributos de volume e preço de fechamento ajustado e, portanto, todas conclusões precedem-se as outras companhias. A Figura 25 apresenta o resultado.

Figura 25 - Correlação entre os atributos.



Fonte: Elaborado pelo autor.

Nota-se que todos os atributos apresentam correlação máxima, exceto o volume de transações, que apresenta uma correlação muito baixa. Esse comportamento é explicado pelo fato dos atributos que apresentaram a correlação máxima são valores em diferentes momentos do preço da ação em cada dia. Vale lembrar que devemos prezar pela menor sobreposição de atributos possível. O teste sugere que poderíamos utilizar apenas um dos valores de abertura, fechamento, máximo ou mínimo, além do volume, porém, o teste apresenta uma relação linear, o que pode não ser verdade para um modelo de inteligência artificial.

4.2.2 *XGBoost* para importância de atributos

Como mencionado anteriormente, um simples teste de correlação, às vezes não conseguem determinar a importância de cada atributo em situações que modelos mais complexos conseguem e, portanto, utilizou-se o *XGBoost*, que é um algoritmo de inteligência artificial, baseado em árvores de decisão, para mensurar a importância de cada atributo presente na base de dados.

Para calcular a importância de atributos, o *XGBoost* utiliza o conceito de *gain*, como fora explicado na seção 2.4.1 da fundamentação teórica. Além disso, o valor apresentado pelo modelo é relativo aos próprios atributos da base de dados, sendo assim, cria-se uma pontuação para determinar qual é o atributo mais e menos importante.

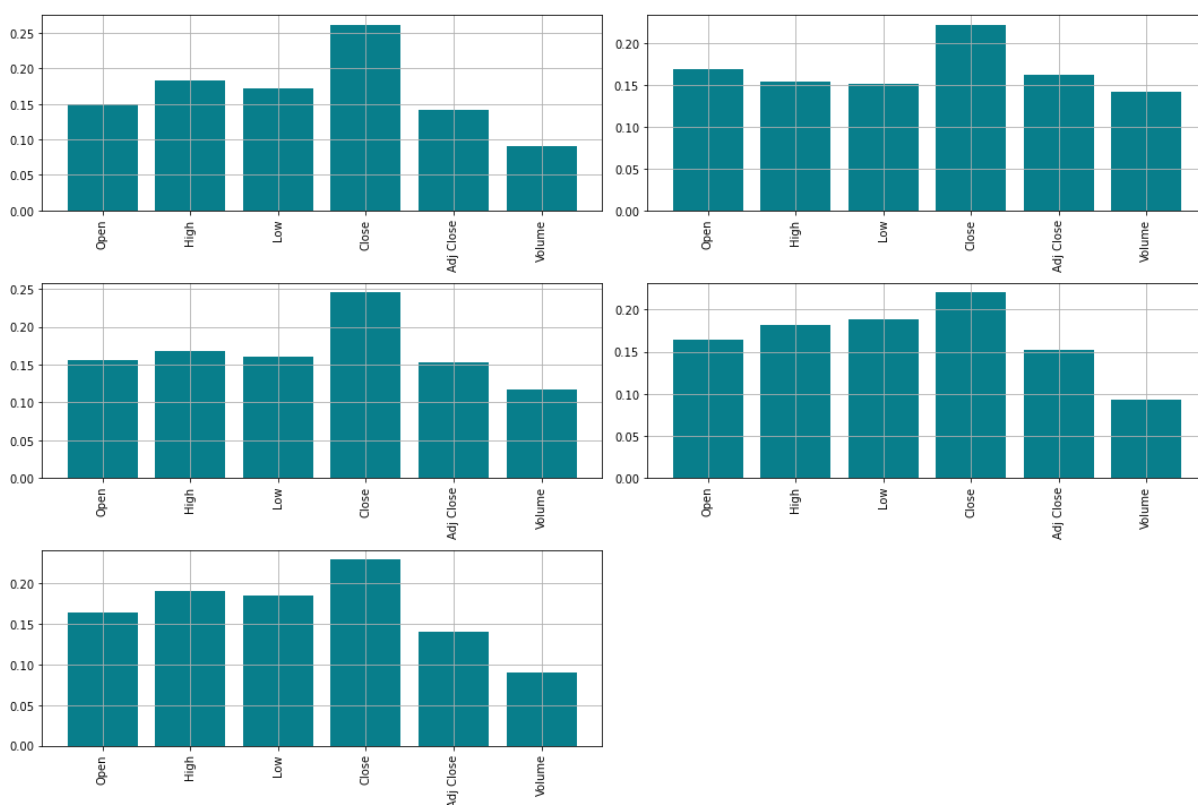
O teste consistiu em treinar um modelo genérico, com os dados de cada empresa e extrair a importância de cada um dos atributos utilizados. A Figura 26 consolida os resultados obtidos para cada empresa. Os valores numéricos, em termos percentuais, são apresentados na Tabela 5.

Tabela 5 - Importância de atributos pelo *XGBoost* (em termos percentuais).

Atributos	B3	Bradesco	Itaú	Petrobras	Vale
Close (Preço de fechamento em R\$)	0,2620	0,2221	0,2459	0,2208	0,2293
High (Preço máximo em R\$)	0,1833	0,1541	0,1676	0,1813	0,1906
Low (Preço mínimo em R\$)	0,1722	0,1516	0,1607	0,1882	0,1850
Open (Preço de abertura em R\$)	0,1496	0,1689	0,1565	0,1640	0,1649
Adj Close (Preço de fechamento ajustado em R\$)	0,1415	0,1619	0,1529	0,1520	0,1407
Volume	0,0914	0,1415	0,1165	0,0937	0,0895

Fonte: Elaborado pelo autor.

Figura 26 - Importância dos atributos pelo *XGBoost* (em percentual).



Fonte: Elaborado pelo autor.

Os resultados apresentam que, para todas empresas, o atributo de maior importância é o preço de fechamento e o de menor importância é o volume de transações. Este resultado, avaliado com o teste de correlação, indica que apesar do volume não ter correlação entre as outras variáveis, aparenta ter a menor importância de todos os atributos. Além disso, apesar de todos outros atributos possuírem uma correlação máxima, também assumem papel importante na tarefa de predição do valor final.

Por fim, devido a proximidade dos valores apresentados, o teste não permite concluir quais atributos podem gerar sobreposição de informação e, conseqüentemente, serem removidos sem perda para os modelos. Assim sendo, conforme mencionado na seção 3.5 da metodologia, foi necessário a realização de um terceiro teste.

4.2.3 Seleção de atributos e *lagging*

Ao considerar o resultado dos testes anteriores, avalia-se que, em conjunto, os testes sugerem que o preço de fechamento é o atributo mais relevante para os modelos. No entanto, devido a proximidade dos valores apresentados pelo *XGBoost*, não podemos verificar tal conclusão. Para concluir tal afirmativa foi necessário treinar um modelo genérico (conforme mencionado na seção 3.5 da metodologia, um modelo com hiperparâmetros constante para todas empresas), para cada técnica, através de duas bases de dados distintas e extrair o resultado do modelo submetido ao conjunto de testes. Além disso, variou-se o *lagging* de 3 a 60 dias aplicado a cada execução. Os dados de todas as empresas foram avaliados e cada valor de *lagging* foi executado três vezes.

As análises do teste foram divididas em duas etapas. A primeira etapa consistiu em buscar a melhor base de dados por técnica, assim como o melhor valor de *lagging*, em média, para todas as empresas juntas, definindo assim, o modelo generalista. Já a segunda, foi avaliar qual foi o melhor *lagging*, em média, para cada empresa e gerar o modelo especialista. Ambas as etapas foram executadas para as duas técnicas.

Multilayer Perceptron (MLP)

Para avaliar a melhor base de dados foi necessário agrupar todas execuções por *lagging* através da média dos valores de erro encontrado. Este resultado permitiu encontrar qual é a base de dados que apresenta a melhor capacidade de regressão dada a técnica avaliada. Os cinco melhores valores para a MLP, ou seja, o menor RMSE, estão representados na Tabela 6.

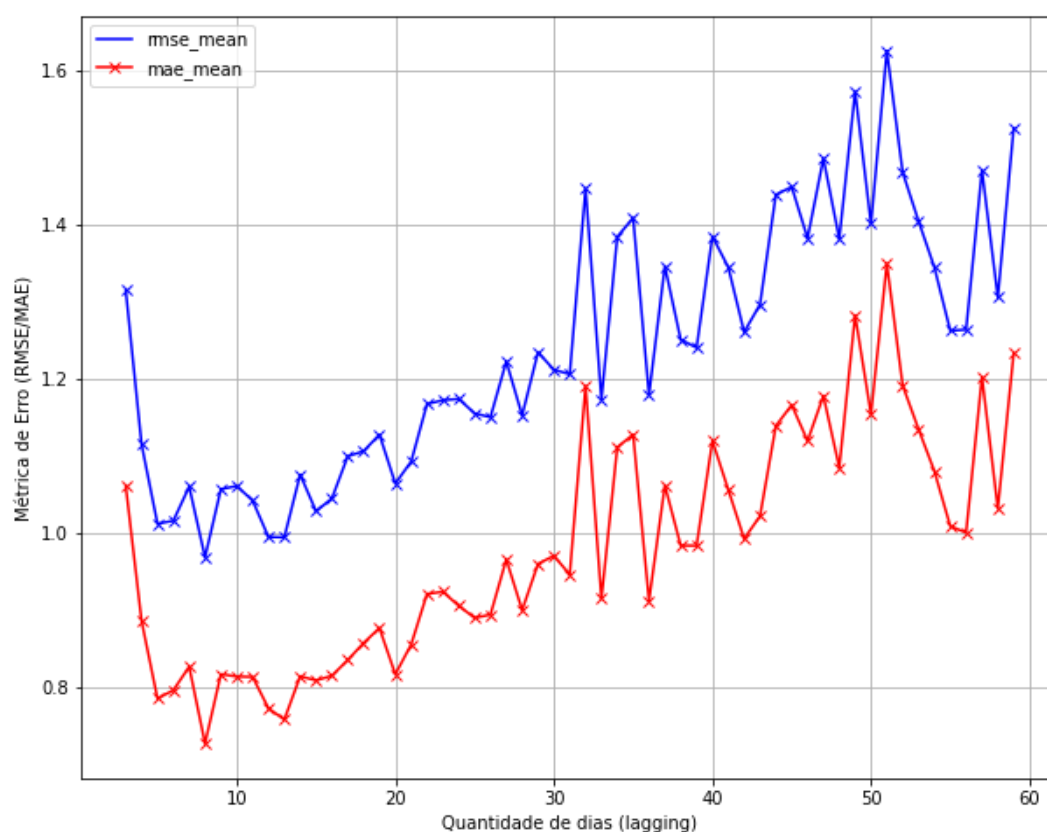
Tabela 6 - Resultados obtidos pela MLP.

Completa			Somente preço de fechamento		
<i>Lagging</i>	RMSE (R\$)	MAE (R\$)	<i>Lagging</i>	RMSE (R\$)	MAE (R\$)
5	9,912E-01	7,706 E-01	8	9,677 E-01	7,261 E-01
8	1,013E+00	7,944 E-01	13	9,935 E-01	7,570 E-01
13	1,026E+00	8,063 E-01	12	9,940 E-01	7,717 E-01
11	1,041E+00	8,130 E-01	5	1,011 E+00	7,844 E-01
8	1,056E+00	8,160 E-01	6	1,015 E+00	7,951 E-01

Fonte: Elaborado pelo autor.

Nota-se que para a MLP, a base de dados que continha a menor quantidade de atributos, em média, apresentou um erro menor, quando comparado ao mesmo cenário utilizando a base de dados com todos os atributos. Além disso, ao avaliar a Figura 27, que apresenta o comportamento do erro para cada *lagging* com apenas o preço de fechamento, percebe-se que, com o aumento do mesmo, aumenta-se o erro médio. Por essa razão, para encontrar os melhores modelos para MLP, foi utilizado a base de dados contendo apenas o preço de fechamento. Por fim, conclui-se que o *lagging* que melhor atua, em média e em todas as empresas, assume o valor de oito dias passados.

Figura 27 - Comportamento do erro por *lagging* na MLP (em reais – R\$).



Fonte: Elaborado pelo autor.

A segunda etapa consistiu em agrupar os resultados por *lagging* e por empresa. Este resultado permitiu encontrar qual é o valor de *lagging* que melhor atua, na média, para determinada empresa e, portanto, caracterizou-se como o modelo especialista. O resultado por empresa é apresentado na Tabela 7.

Tabela 7 - Resultados obtidos para cada empresa para MLP.

Empresa	Lagging	RMSE (R\$)	MAE (R\$)
B3	5	3,767 E-01	2,935 E-01
Petrobras	4	7,012 E-01	5,394 E-01
Vale	8	2,419 E+00	1,836 E+00
Bradesco	7	4,581 E-01	3,530 E-01
Itaú	3	7,059 E-01	4,398 E-01

Fonte: Elaborado pelo autor.

Long Short-Term Memory (LSTM)

Executou-se o mesmo procedimento para a LSTM. Ao analisar o resultado da primeira etapa, exposto na Tabela 8, conclui-se que, diferente da MLP, a base de dados que apresentou a melhor média (menor erro para o RMSE e MAE, nessa ordem) de resultados foi a base completa, ou seja, com todos os atributos avaliados pelo modelo. Além disso, nota-se que o melhor *lagging*, na média das três execuções, para todas empresas em conjunto foi de 36 dias passados.

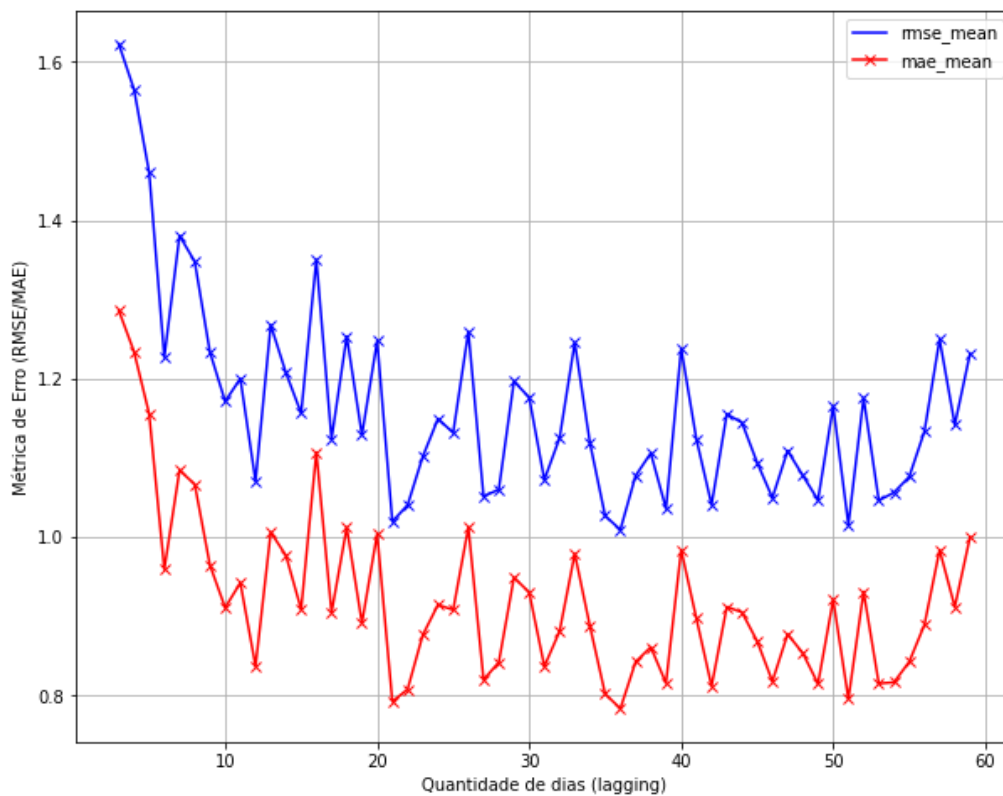
Tabela 8 - Resultados obtidos pela LSTM.

Completa			Somente preço de fechamento		
Lagging	RMSE (R\$)	MAE (R\$)	Lagging	RMSE (R\$)	MAE (R\$)
36	1,009 E+00	7,834 E-01	3	1,013 E+00	7,934 E-01
51	1,016 E+00	7,955 E-01	4	1,041 E+00	8,116 E-01
21	1,020 E+00	7,923 E-01	13	1,047 E+00	8,150 E-01
35	1,026 E+00	8,017 E-01	8	1,051 E+00	8,193 E-01
39	1,036 E+00	8,156 E-01	7	1,056 E+00	8,200 E-01

Fonte: Elaborado pelo autor.

A Figura 28 apresenta o comportamento do erro com os diversos *lagging* aplicados a LSTM com a base de dados completa. Esta representação demonstra um comportamento inverso ao da MLP, ou seja, com o aumento da complexidade (mais atributos e *lagging* maior) da base de dados, menor o erro apresentado pela rede, em média. Tal comportamento pode ser explicado pela capacidade maior que as LSTM possuem para manter a informação pertinente dentro da rede, como explicado na seção 2.8 do referencial teórico.

Figura 28 - Comportamento do erro por *lagging* na LSTM (em reais – R\$).



Fonte: Elaborado pelo autor.

Por fim, a segunda etapa permitiu encontrar qual é o melhor valor de *lagging* por empresa, na média das três execuções, caracterizando, assim, o modelo especialista relacionado às LSTM. O resultado é apresentado no Tabela 9.

Tabela 9 - Resultados obtidos para cada empresa para LSTM.

Empresa	Lagging	RMSE (R\$)	MAE (R\$)
B3	25	3,793 E-01	3,067 E-01
Petrobras	37	7,239 E-01	5,604 E-01
Vale	49	2,422 E+00	1,879 E+00
Bradesco	55	5,161 E-01	4,017 E-01
Itaú	37	7,593 E-01	5,202 E-01

Fonte: Elaborado pelo autor.

4.3 Otimização de Hiperparâmetros

Os resultados nos testes preliminares permitiram definir a características dos modelos generalistas e especialistas, ou seja, o valor de *lagging* para cada cenário e técnica. Posterior a isto, foi preciso avaliar qual combinação de hiperparametros que apresenta o menor erro médio para os modelos em cada cenário (generalista e especialista). As diferentes combinações de hiperparemetros são apresentadas na seção 3.6 da metodologia. Cada combinação de hiperparâmetro foi avaliada três vezes. Os resultados são as médias do erro no conjunto validação e treinamento de todas as execuções para cada combinação.

Este processo foi dividido em duas etapas para MLP e LSTM. São elas:

- Buscar conjunto de hiperparâmetros para o modelo generalista (considerando o *lagging* médio)
- Buscar conjunto de hiperparâmetros para o modelo generalista (considerando o *lagging* cada empresa)

4.3.1 Otimização de hiperparâmetros para *Multilayer Perceptron*

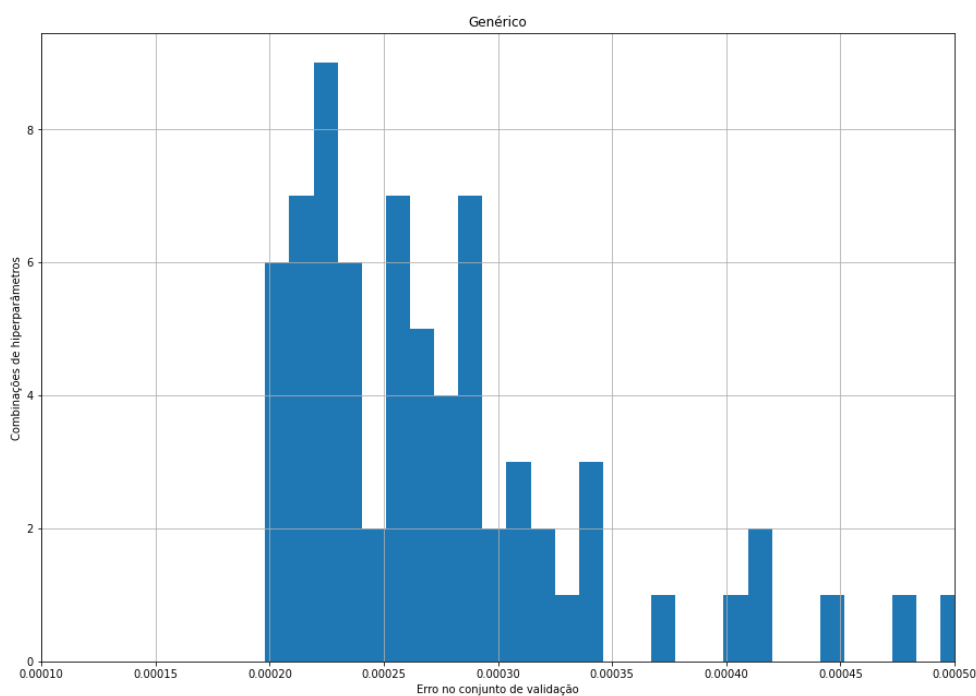
A partir dos testes anteriores, concluiu-se que a base de dados ideal para as *Multilayer Perceptrons*, dentro do contexto avaliado, foi a que apresentou a menor complexidade. Portanto, a base de dados utilizada para encontrar os diferentes tipos de modelos foi a que possuía apenas o preço de fechamento como atributo, aplicado ao *lagging* de cada cenário (generalista ou especialista).

Modelo generalista - *Multilayer Perceptron*

A título de encontrar a melhor combinação de hiperparâmetros, em média, para todas as empresas em conjunto, executou-se a procura considerando o *lagging* de 8 dias. Ao analisar a Figura 29, que apresenta a distribuição das distintas combinações dado o erro no conjunto de validação, nota-se que as combinações apresentaram variações no erro, sendo assim, a busca pela melhor combinação conseguiu diminuir o erro no conjunto de validação. Portanto, tal etapa

demonstra que houveram ganhos (redução de erro) no conjunto de validação, sendo assim, uma boa candidata a ser testada no conjunto de teste (conjunto nunca visto pelo modelo).

Figura 29 - Histograma das diferentes combinações pelo erro no conjunto de validação no modelo generalista (*Multilayer perceptron*).



Fonte: Elaborado pelo autor.

A Tabela 10 apresenta as cinco combinações com o melhor resultado, dado o contexto analisado. Vale lembrar que para o modelo generalista, os dados de todas as empresas foram considerados.

Tabela 10 - Cinco melhores combinações para modelo generalista para MLP.

Camadas Ocultas	Tamanho de Lote	Função de Ativação	Épocas	Neurônios	Erro para Conjunto Treinamento (R\$)	Erro para conjunto Validação (R\$)
1	32	relu	100	32	2,377 E-04	1,980 E-04
1	32	relu	50	128	2,444 E-04	1,993 E-04
1	32	tanh	100	32	2,384 E-04	2,045 E-04
1	32	relu	50	64	2,362 E-04	2,065 E-04
1	16	relu	50	32	2,422 E-04	2,075 E-04

Fonte: Elaborado pelo autor.

Nota-se que, em média, as configurações com apenas uma camada oculta obtiveram um resultado melhor quando comparado com os modelos mais robustos, com duas camadas ocultas. Além disso, os resultados demonstram que a função de ativação *Rectified Linear (Relu)*, assim como o tamanho do lote igual a 32 foram predominantes no objetivo de diminuir o erro apresentado no conjunto validação.

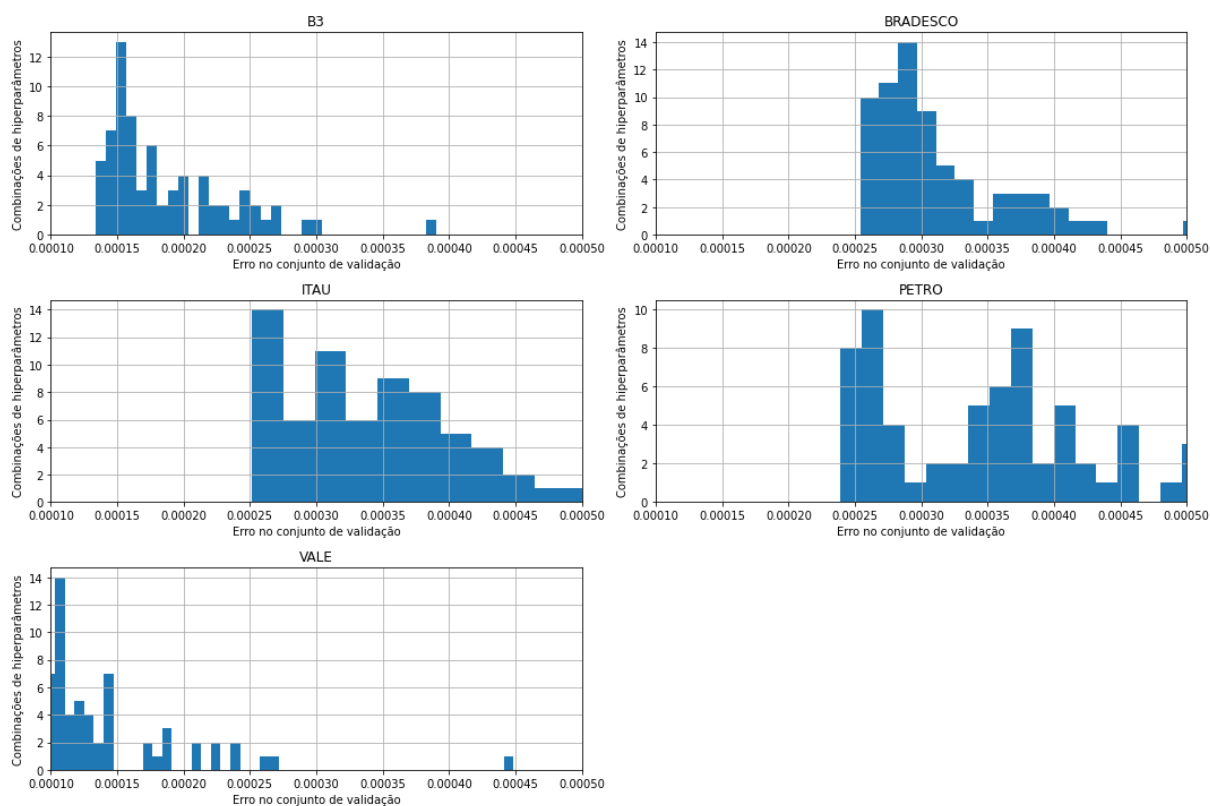
Por fim, os resultados demonstram um erro no conjunto de validação menor do que no conjunto de treinamento. Uma das razões pode ser explicado pela característica do processo de treinamento. O treinamento ocorre por épocas, sendo assim, a época é caracterizada quando todos os dados são alimentados ao modelo. Os dados são alimentados por lotes, definido como um dos hiperparâmetros. Portanto, o erro de treinamento é calculado a cada lote inserido, enquanto o erro de validação é mensurado ao término de uma época e, portanto, uma parcela com erro maior no início do treinamento tem maior influência no término do processo.

Modelo especialista - *Multilayer Perceptron*

Para encontrar o modelo especialista, foi aplicado a mesma metodologia dos modelos generalistas, apenas modificou-se o *lagging* que fora aplicado para cada empresa conforme descrito nas seções anteriores.

A Figura 30 apresenta a distribuição das diferentes combinações por empresa. Assim como no modelo generalista, o comportamento da otimização foi semelhante. Nota-se que houve flutuações no erro do conjunto de validação apresentado para cada empresa, principalmente para Petrobras, Itaú e Bradesco. Sendo assim, evidencia-se que a etapa reduziu o erro no conjunto de validação e que pode impactar na avaliação dos modelos sob a perspectiva do conjunto de teste

Figura 30 - Histograma das diferentes combinações pelo erro no conjunto de validação no modelo especialista (*Multilayer perceptron*).



Fonte: Elaborado pelo autor.

A Tabela 11 apresenta as cinco melhores configurações de hiperparâmetros de cada empresa.

Tabela 11 - Resultados obtidos para os modelos especialistas da MLP.

Empresa	Camadas Ocultas	Tamanho do Lote	Função de Ativação	Épocas	Neurônios	Erro para Conjunto Treinamento (R\$)	Erro para conjunto Validação (R\$)
B3	2	32	tanh	50	32	2,124 E-04	1,626 E-04
	2	16	relu	100	64	2,256 E-04	1,689 E-04
	2	16	relu	50	64	2,191 E-04	1,714 E-04
	2	16	relu	100	32	2,082 E-04	1,715 E-04
	2	32	tanh	100	128	2,291 E-04	1,772 E-04
Bradesco	1	32	relu	100	128	2,847 E-04	2,246 E-04
	1	16	relu	100	32	2,905 E-04	2,251 E-04
	1	32	relu	100	64	2,837 E-04	2,282 E-04
	1	16	relu	100	64	2,836 E-04	2,306 E-04
	1	32	relu	100	32	2,838 E-04	2,325 E-04
Itaú	1	32	tanh	50	128	2,996 E-04	2,514 E-04
	1	32	tanh	100	64	3,030 E-04	2,529 E-04
	1	16	relu	100	128	2,981 E-04	2,538 E-04
	1	16	tanh	100	32	3,035 E-04	2,559 E-04
	1	32	relu	50	128	2,983 E-04	2,559 E-04
Petrobras	2	16	tanh	100	64	3,184 E-04	2,005 E-04
	2	32	relu	100	128	3,022 E-04	2,045 E-04
	2	16	tanh	50	64	3,297 E-04	2,144 E-04
	2	16	relu	50	32	3,099 E-04	2,148 E-04
	2	32	relu	100	64	2,874 E-04	2,171 E-04
Vale	1	16	relu	50	128	1,021 E-04	8,090 E-05
	1	32	tanh	50	128	1,025 E-04	8,130 E-05
	1	16	relu	100	128	9,850 E-05	8,170 E-05
	1	16	tanh	100	64	1,012 E-04	8,200 E-05
	1	32	relu	100	64	1,003 E-04	8,350 E-05

Fonte: Elaborado pelo autor.

Ao analisar o resultado por empresa, nota-se que para os casos da B3 e da Petrobras, os modelos com mais de uma camada oculta foram os que obtiveram o melhor resultado, diferente do que foi avaliado nas outras empresas e no modelo generalista.

Por fim, ao comparar os resultados obtidos pelos modelos especialistas com o modelo genérico, percebe-se que os modelos especialistas (que fazem o uso de um *lagging* próprio), fez o uso de mais quantidade de épocas e neurônios para reduzir o erro no conjunto de validação.

4.3.2 Otimização de hiperparâmetros para *Long Short-Term Memory*

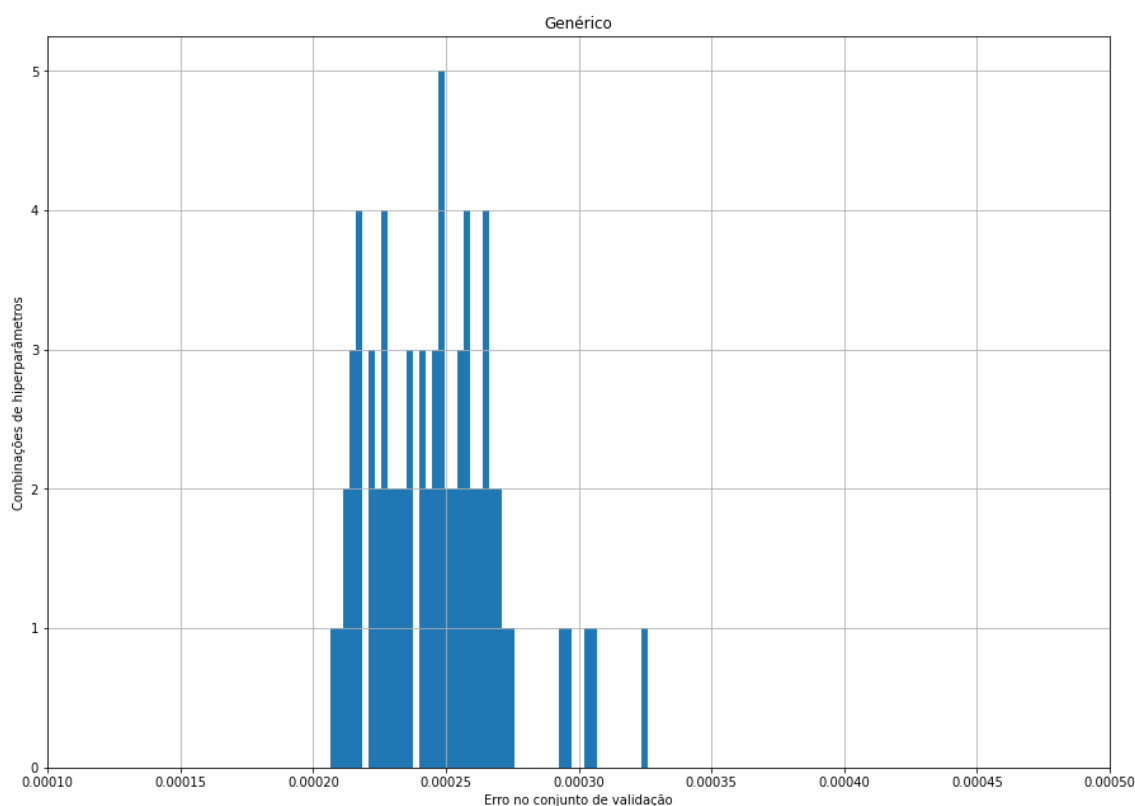
Diferente do apresentado para as *Multilayer Perceptrons*, as *Long Short-Term Memory* demonstraram uma capacidade maior em traduzir a complexidade (maior quantidade de atributos) da base de dados em uma regressão com erro menor, quando comparado a uma base de dados mais simples. Em vista disso, utilizou-se todos os atributos presentes na base de dados. Além disso, para as LSTMs, considerou-se três tipos de arquiteturas diferentes, são elas: *Vanilla*, *Stacked* e Bidirecional, representados pelos valores de 1, 2 e 3, respectivamente.

Modelo generalista - *Long Short-Term Memory*

Assim como foi apresentado nos testes preliminares, as *Long Short-Term Memory* obtiveram um sucesso maior quando submetidas a valores de *lagging* maiores. Dentro do contexto avaliado, o valor de 36 dias foi o que obteve o melhor resultado, em média, para todas as empresas. Posterior a isto, avaliou-se, dentro das combinações selecionadas, qual foi a arquitetura que obteve o menor erro médio no conjunto de validação para o *lagging* em questão. Utilizou-se os dados de todas as empresas. Cada configuração foi executada três vezes e as métricas, através do *mean squared error*, resultantes foram consolidadas na média das execuções.

A Figura 31 apresenta a distribuição das combinações testadas pelo erro apresentado no conjunto de validação. Nota-se, como no caso da MLP, que apresentaram flutuações no erro apresentado por cada combinação e, portanto, reforçando a importância da procura pela melhor combinação.

Figura 31 - Histograma das diferentes combinações pelo erro no conjunto de validação no modelo generalista (*Long Short-Term Memory*).



As cinco melhores combinações foram consolidadas e apresentadas na Tabela 12.

Tabela 12 - Cinco melhores combinações para modelo generalista para LSTM.

Arquitetura	Tamanho do Lote	Função de Ativação	Épocas	Neurônios	Erro para Conjunto Treinamento (R\$)	Erro para conjunto Validação (R\$)
3	16	relu	100	32	2,264 E-04	2,064 E-04
3	16	relu	100	128	2,207 E-04	2,111 E-04
2	32	tanh	100	32	2,408 E-04	2,122 E-04
3	16	relu	200	32	2,070 E-04	2,124 E-04
3	32	tanh	100	128	2,318 E-04	2,137 E-04

Fonte: Elaborado pelo autor.

A melhor configuração foi escolhida através do menor erro médio apresentado no conjunto de validação. Nota-se que a arquitetura Bidirecional apresentou o melhor erro médio quando avaliada para todas as empresas em conjunto, uma vez que assumiu quatro posições das

cinco melhores. Além disso, a *Vanilla* foi a arquitetura, das avaliadas, com o pior resultado apresentado. O melhor resultado proveniente de uma *Vanilla* obteve a posição 36 de 72 combinações distintas.

Diferente do que foi apresentado nas MLPs, as melhores combinações de hiperparâmetros que apresentaram o menor erro e, conseqüentemente, maior capacidade de regressão, em média, foram as mais robustas, com duas camadas ocultas ou com o estilo Bidirecional.

O resultado permite também comparar que durante a otimização de hiperparâmetros, as LSTMs obtiveram um resultado levemente melhor quando comparado com os apresentados pelas MLPs. No entanto, este resultado não garante que esse comportamento seja replicado quando avaliado sob a ótica do conjunto de testes.

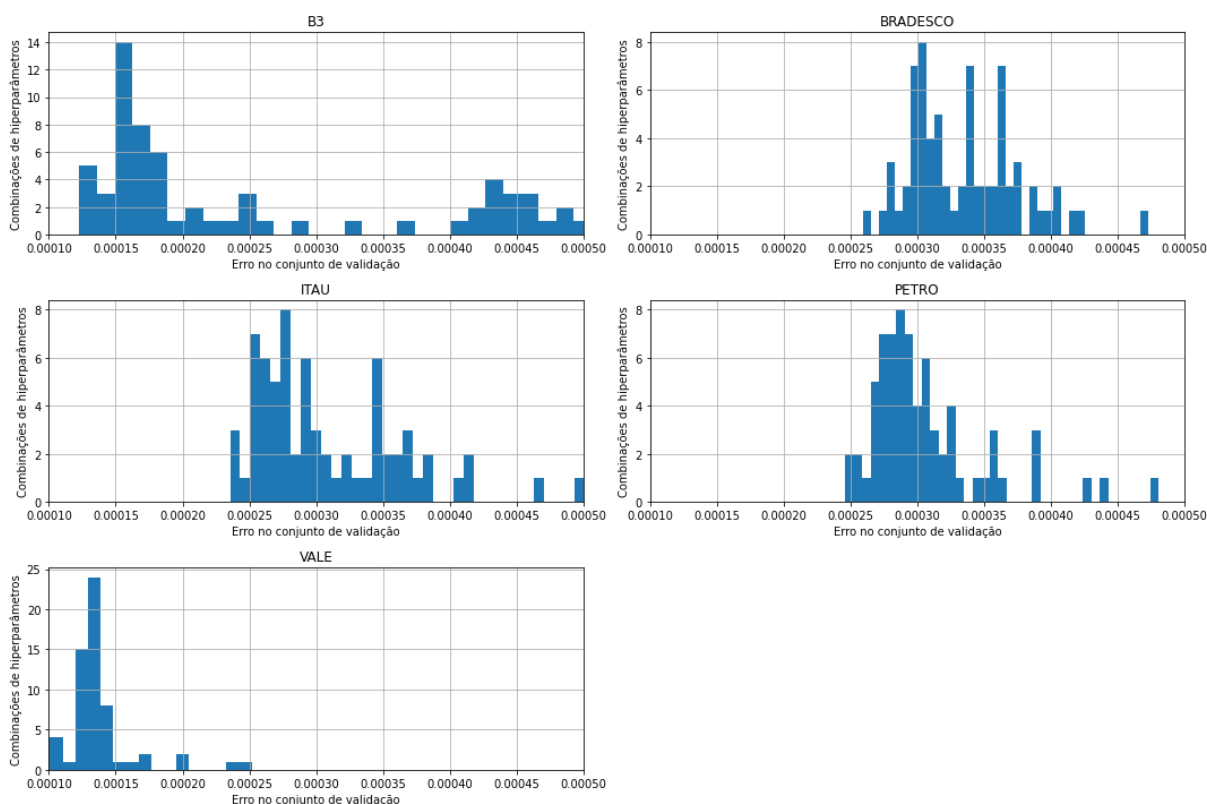
Modelo especialista - *Long Short-Term Memory*

A fim de encontrar o modelo especialista para cada empresa, executou-se a mesma busca de hiperparâmetros que fora realizada no modelo generalista. Neste caso, aplicou-se o *lagging* obtido para cada empresa. Esses valores foram obtidos através dos testes preliminares e podem ser consultados na seção anterior.

A Figura 32, que apresenta a distribuição das combinações testadas por cada empresa para as *Long Short-Term Memory*, demonstra a mesma flutuação no erro médio por combinação avaliada.

Neste caso, o resultado apresentado pelas LSTM são mais acentuados, principalmente para B3, Bradesco, Itaú e Petrobras. O maior espaçamento entre as diferentes combinações reforça a importância que a otimização dos hiperparâmetros obteve na construção das LSTM, assumindo assim, um papel mais significativo do que nas MLPs.

Figura 32 - Histograma das diferentes combinações pelo erro no conjunto de validação no modelo especialista (*Long Short-Term Memory*).



Fonte: Elaborado pelo autor.

Os cinco modelos, para cada empresa, que apresentaram o menor erro foram consolidados na Tabela 13.

Os resultados (combinações mais robustas, com duas camadas ocultas ou com o neurônio bidirecional) apresentados pelos modelos especialistas, em conjunto com o generalista, reforçam a capacidade da *Long Short-Term Memory* em traduzir a complexidade de suas variáveis, hiperparâmetros e base de dados, em um erro médio menor. Comportamento este, que não foi evidenciado durante o mesmo processo das MLPs.

Além disso, ressaltam-se dois pontos. O primeiro é a presença dominante da arquitetura *Stacked LSTM*, em relação aos resultados obtidos no generalista. Isso indica que, apesar dos melhores resultados, por empresa, terem sido advindos de uma *Stacked LSTM*, que na média, as redes bidirecionais obtiveram um resultado melhor, quando analisada sob a ótica de todas as empresas. Já o segundo é o mesmo comportamento evidenciado nas MLPs, um erro sucessivamente menor no conjunto de validação do que no de treinamento. A explicação para esta conduta é a mesma.

Tabela 13 - Resultados obtidos para os modelos especialistas da LSTM.

Empresa	Camadas Ocultas	Tamanho do Lote	Função de Ativação	Épocas	Neurônios	Erro Treinamento (R\$)	Erro Validação (R\$)
B3	3	32	relu	100	128	1,851 E-04	1,229 E-04
	3	32	relu	200	32	1,749 E-04	1,246 E-04
	3	16	tanh	200	32	1,637 E-04	1,311 E-04
	3	16	tanh	100	32	1,788 E-04	1,336 E-04
	3	16	relu	100	32	1,803 E-04	1,346 E-04
Bradesco	2	16	tanh	100	32	2,809 E-04	2,595 E-04
	2	32	tanh	100	128	2,917 E-04	2,728 E-04
	2	32	relu	100	32	2,760 E-04	2,786 E-04
	2	16	tanh	100	128	2,665 E-04	2,786 E-04
	2	16	relu	200	32	2,545 E-04	2,788 E-04
Itaú	2	16	tanh	100	128	2,809 E-04	2,356 E-04
	2	32	tanh	100	128	2,786 E-04	2,360 E-04
	2	32	sigmoid	200	32	3,057 E-04	2,387 E-04
	2	32	tanh	200	128	2,792 E-04	2,494 E-04
	2	16	tanh	100	32	2,809 E-04	2,509 E-04
Petrobras	2	32	tanh	100	32	3,103 E-04	2,461 E-04
	2	32	relu	200	32	2,812 E-04	2,491 E-04
	2	16	tanh	200	32	2,570 E-04	2,551 E-04
	2	16	sigmoid	200	32	3,204 E-04	2,570 E-04
	2	32	tanh	200	128	2,707 E-04	2,613 E-04
Vale	2	16	relu	200	32	9,410 E-05	8,220 E-05
	2	16	tanh	200	32	9,040 E-05	8,270 E-05
	2	16	tanh	200	128	9,820 E-05	8,280 E-05
	2	32	tanh	200	32	9,540 E-05	8,460 E-05
	2	16	sigmoid	200	128	1,385 E-04	8,490 E-05

Fonte: Elaborado pelo autor.

4.4 Seleção dos conjuntos de hiperparâmetros

Posterior a otimização de hiperparâmetros, precisou-se selecionar a melhor combinação de hiperparâmetros para cada cenário (generalista e especialista). As Tabelas 10, 11, 12 e 13 ilustraram a média das três execuções para cada combinação e, portanto, definiu-se o melhor resultado aquele que apresentou o menor erro, no conjunto de validação, para cada situação. A

Tabela 14 e 15, consolida as características dos hiperparâmetros selecionados, dado o tipo de modelo, empresa e técnica.

Tabela 14 - Combinações de hiperparâmetros selecionadas da MLP.

Técnica	Modelo	Empresa	Camadas Ocultas	Tamanho do Lote	Função de Ativação	Épocas	Neurônios
MLP	Generalista	Todas	1	32	relu	100	32
	Especialista	B3	2	32	tanh	50	32
		Bradesco	1	32	relu	100	128
		Itaú	1	32	tanh	50	128
		Petrobras	2	16	tanh	100	32
		Vale	1	16	relu	50	128

Fonte: Elaborado pelo autor.

Tabela 15 - Combinações de hiperparâmetros selecionadas da LSTM.

Técnica	Modelo	Empresa	Arquitetura	Tamanho do Lote	Função de Ativação	Épocas	Neurônios
LSTM	Generalista	Todas	3	16	relu	100	32
	Especialista	B3	3	32	relu	100	128
		Bradesco	2	16	tanh	100	32
		Itaú	2	16	tanh	100	128
		Petrobras	2	32	tanh	100	32
		Vale	2	16	relu	200	32

Fonte: Elaborado pelo autor.

Nota-se que as combinações de hiperparâmetros que obtiveram os melhores resultados para o modelo generalista (que é otimizado para todas as empresas), não foram as mesmas que obtiveram os menores erros para os modelos especialistas (otimizado para cada empresa). Este comportamento pode indicar o fato de ser a combinação de hiperparâmetros que se adaptou melhor ao cenário generalista, porém, o mesmo será avaliado na seção que se sucede.

4.5 Avaliação de desempenho

A fim de avaliar o melhor conjunto de hiperparâmetros selecionados, submeteu-se os modelos aos conjuntos de teste (dados nunca vistos pelo modelo) divididos em duas etapas. A primeira consistiu em uma avaliação quantitativa a respeito dos erros enquanto a segunda consistiu em realizar um processo simulado com os modelos que apresentaram o melhor resultado na primeira etapa.

4.5.1 Avaliação Quantitativa - *Multilayer Perceptron*

Inicialmente, cada modelo selecionado na seção anterior foi submetido ao teste ilustrado pelo Quadro 14. Em síntese, treinou-se cada modelo com os dados de cada empresa e avaliou-se com os conjuntos de teste distintos (de cada empresa, exceto a do treinamento). Extraiu-se a média dos erros. O resultado permitiu concluir se o modelo generalista conseguiu, melhor que os outros, em média, predizer o valor de uma ação de outro universo que não o seu. Os resultados são demonstrados na Tabela 16.

Tabela 16 - Resultados para os modelos na atuação generalista.

Tipo de modelo	Conjunto de Treinamento	RMSE (R\$)	MAE (R\$)
Generalista	B3	1,092 E+00	8,288 E-01
	Bradesco	9,542 E-01	6,960 E-01
	Itaú	9,294 E-01	7,046 E-01
	Petrobras	9,295 E-01	6,841 E-01
	Vale	6,622 E-01	4,759 E-01
Especialista	B3	1,093 E+00	8,495 E-01
	Bradesco	9,602 E-01	6,912 E-01
	Itaú	1,151 E+00	9,251 E-01
	Petrobras	8,840 E-01	6,448 E-01
	Vale	7,801 E-01	6,276 E-01

Fonte: Elaborado pelo autor.

Os resultados mostram que, quando submetido ao conjunto de testes de todas as empresas, o modelo generalista obteve desempenho superior aos outros modelos na maioria dos

casos analisados. O modelo da Petrobras, quando avaliado pelo conjunto de outras empresas, obteve um erro menor.

No entanto, o resultado indica que apesar do modelo generalista não ter obtido o melhor resultado quando comparado aos outros modelos, apresentou ser, na média, o modelo com maior estabilidade. Esta característica é evidenciada pela proximidade de seus resultados e, pela natureza estocástica do problema, e isso explica porque o modelo da Petrobras possa ter tido um resultado melhor neste ensaio.

Posteriormente, os mesmos modelos foram submetidos a avaliação dos conjuntos de testes pertencentes às mesmas empresas que o treinaram. O resultado é apresentado na Tabela 17.

Tabela 17 - Resultados dos modelos na atuação especialista para MLP.

Empresa	Modelo	RMSE (R\$)	MAE (R\$)
B3	Generalista	3,747 E-01	2,889 E-01
	Especialista	4,958 E-01	4,038 E-01
Bradesco	Generalista	4,497 E-01	3,422 E-01
	Especialista	5,315 E-01	4,209 E-01
Itaú	Generalista	6,988 E-01	4,238 E-01
	Especialista	7,302 E-01	4,489 E-01
Petrobras	Generalista	6,736 E-01	5,165 E-01
	Especialista	6,424 E-01	4,786 E-01
Vale	Generalista	2,002 E+00	1,510 E+00
	Especialista	4,771 E+00	3,985 E+00

Fonte: Elaborado pelo autor.

Percebe-se que o modelo generalista teve maior sucesso, exceto na Petrobras, que os outros modelos quando submetido ao conjunto de dados pertencentes a mesma empresa.

O resultado da avaliação anterior, em conjunto com a dominância do modelo generalista quando atuando de maneira especialista, reforça e permite concluir que os modelos construídos através dos parâmetros obtidos de todas as empresas, obtiveram, na média, um resultado mais estável também quando submetidos ao conjunto de dados da mesma empresa.

As evidências da avaliação quantitativa permitem concluir, para as MLPs, que, na média, os modelos generalistas tiveram um resultado superior que os modelos especialistas até

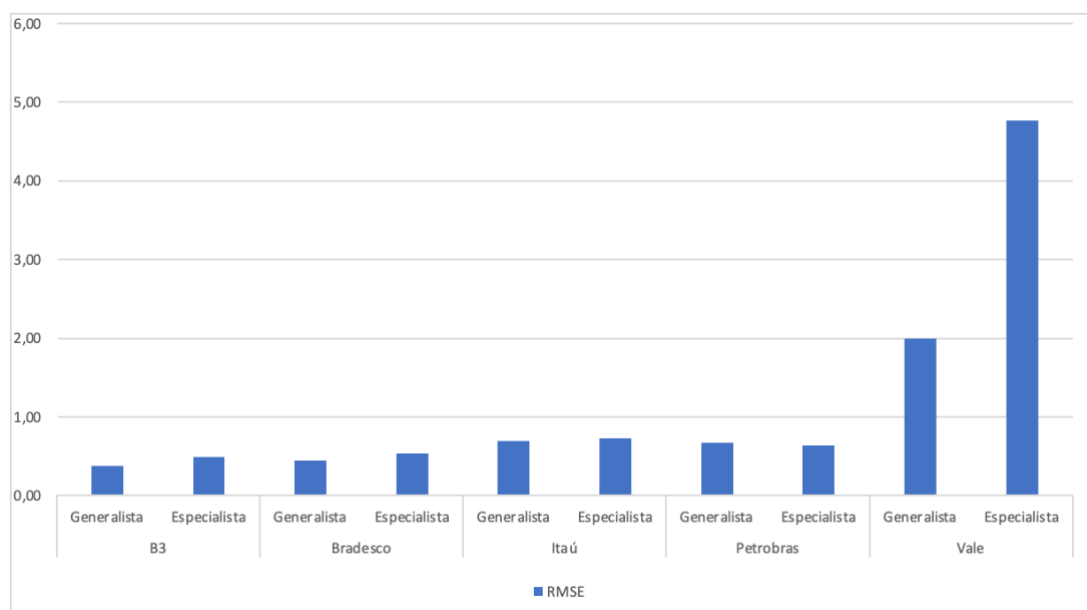
mesmo operando como tal. A Figura 33 apresenta graficamente o erro por tipo de modelo e empresa.

Tabela 18 - Resultados obtidos para as MLPs.

Empresa	Modelo	RMSE (R\$)	MAE (R\$)
B3	Generalista	3,747 E-01	2,889 E-01
Bradesco	Generalista	4,497 E-01	3,422 E-01
Itaú	Generalista	6,988 E-01	4,238 E-01
Petrobras	Especialista	6,424 E-01	4,786 E-01
Vale	Generalista	2,002 E+00	1,510 E+00

Fonte: Elaborado pelo autor.

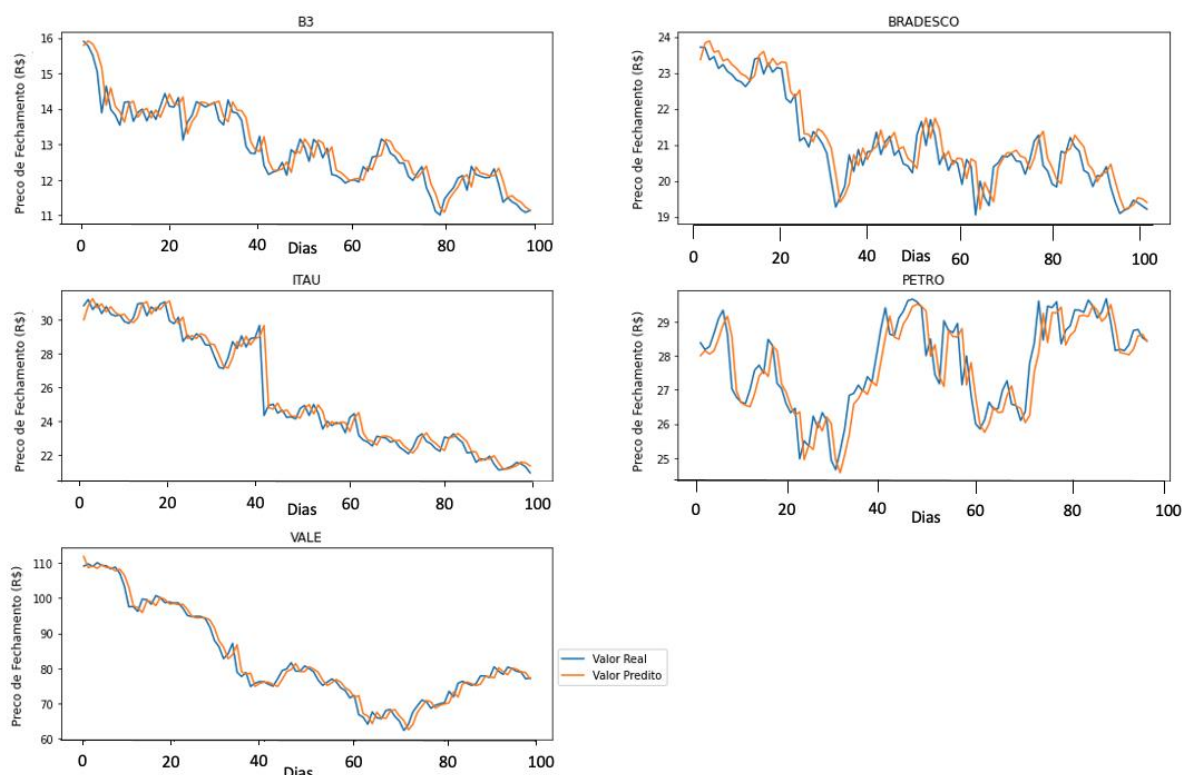
Figura 33 - RMSE por tipo de modelo e empresa, em reais (R\$).



Fonte: Elaborado pelo autor.

A Tabela 18 consolidou qual foi o melhor modelo e o menor erro para cada empresa através das *Multilayer Perceptrons*. A Figura 34 apresenta a comparação entre os valores preditos pela rede, representados em laranja, contra os valores reais, representados em azul.

Figura 34 - Valores preditos *versus* Valores reais para a MLP – em reais (R\$).

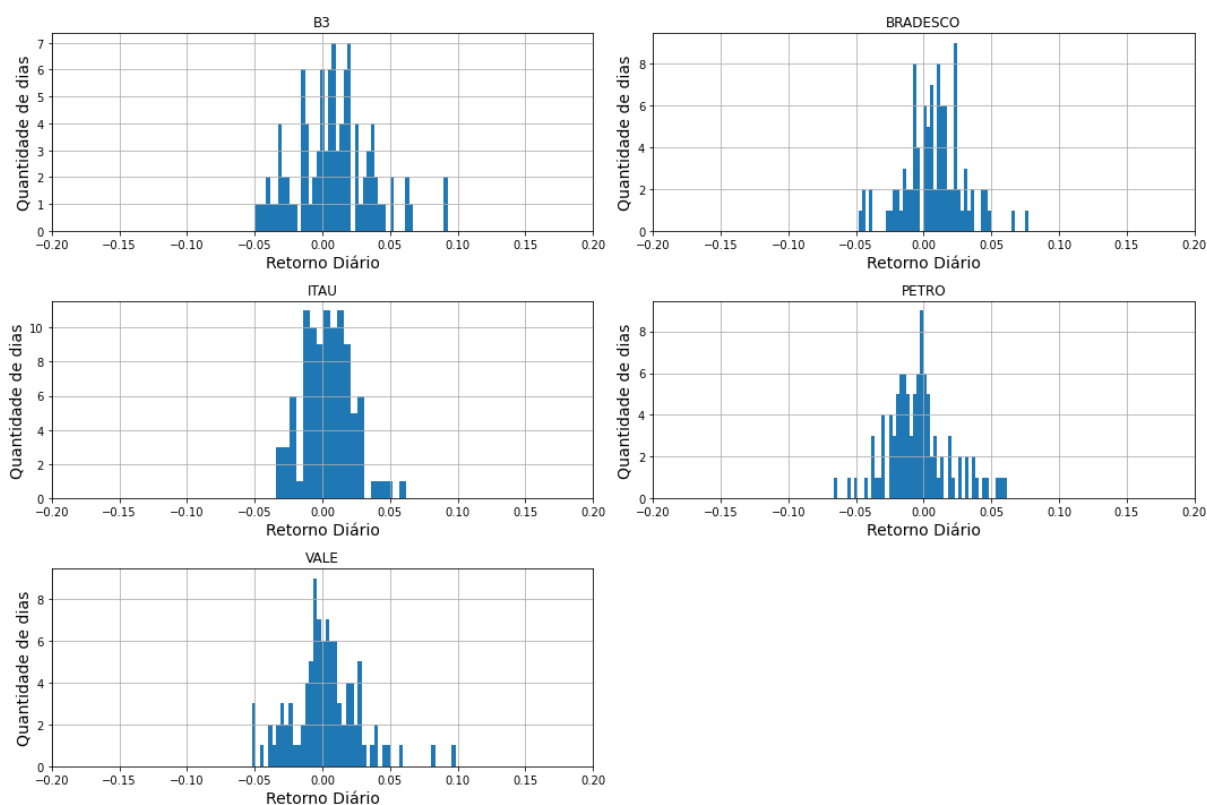


Fonte: Elaborado pelo autor.

A Figura 35 apresenta a distribuição do erro percentual para cada empresa. O erro foi calculado como a diferença do valor predito para o valor real. Ao comparar com o retorno histórico apresentado na Figura 24, nota-se que os erros apresentados pelo modelo seguem o comportamento semelhante, onde a grande maioria das ocorrências estiveram em torno do zero. Além disso, percebe-se que nos cenários de Itaú e Vale, o erro ficou mais concentrado no zero do que as outras empresas. Este comportamento pode ser explicado pelo gráfico de ambas as empresas na Figura 34, que apresentam um comportamento menos caótico (menos oscilatório) que as outras empresas.

Idealmente, os erros apresentados na Figura 35 ficariam todos concentrados em zero, ou seja, um cenário onde o modelo fosse perfeito na tarefa de predição do valor futuro. No entanto, atingir tal resultado não é possível.

Figura 35 - Erro percentual da predição para *Multilayer Perceptron*.



Por fim, os erros apresentados pelo modelo na aferição do conjunto de teste, além dos gráficos ilustrados na Figura 34 e 35, indicam que o modelo, conseguiu acompanhar o comportamento da ação ao longo do tempo, além de manter a maior parte das previsões perto do zero. No entanto, a aplicabilidade (se gerou lucro) do modelo foi avaliada na próxima seção, quando se realizou uma simulação com os valores preditos.

4.5.2 Avaliação Simulada - *Multilayer Perceptron*

Para avaliar a aplicabilidade do modelo, foi necessário submeter os resultados a cenários simulados e, para tal, foram considerados os critérios para cada tipo de investidor mencionados nas seções anteriores, comparado com o investidor *buy and holder*.

Para todas as empresas consideradas, o perfil do investidor conservador não havia atuado em nenhuma situação, uma vez que nenhum resultado predito pelo modelo final estimou que haveria um aumento no valor da ação superior a 2.5%. Já o perfil moderado teve apenas três situações em que o investidor atuaria, de acordo com a predição da rede. A Tabela 19 apresenta quantas vezes cada tipo de investidor teria atuado conforme a previsão do modelo.

Tabela 19 - Recomendações de compra por perfil em cada cenário para MLP.

	B3	Bradesco	Itaú	Petro	Vale
Arrojado	37	62	2	0	2
Moderado	2	0	1	0	0
Conservador	0	0	0	0	0

Fonte: Elaborado pelo autor.

Em todos os cenários avaliados, o perfil arrojado teria realizado prejuízos para apenas duas empresas, a Vale e o Bradesco. Ainda somando-se os ROI (retorno sobre investimento – diferença percentual entre o preço de venda e de compra), o mesmo teria obtido lucro caso estivesse operado em todas as empresas. O resultado consolidado de retorno por tipo de investidor é apresentado na Tabela 20. O perfil conservador e a empresa Petrobras foram retiradas da Tabela 20 por não apresentarem nenhuma situação de compra, no contexto analisado.

Tabela 20 - Retorno sobre investimento por perfil de investidor avaliado.

	B3	Bradesco	Itaú	Vale
Arrojado	9,14%	-6,08%	2,55%	-1,53%
Moderado	9,22%	0%	2,42%	0%

Fonte: Elaborado pelo autor.

Para dimensionar se o resultado exposto da Tabela 20 é insatisfatório, foi preciso comparar o desempenho com um investidor que não tivesse seguido as diretrizes do modelo. Para tal, considerou-se um investidor *buy and holder*, que havia comprado a ação no primeiro dia e vendido no último dia presente no conjunto de teste. O desempenho deste caso é exposto na Tabela 21.

Tabela 21 - Retorno sobre investimento para um investidor *Buy and Hold*.

Empresa	Preço inicial (R\$)	Preço Final (R\$)	ROI (%)
Vale	109,08	77,25	-29,18%
B3	15,90	11,14	-29,94%
Bradesco	23,71	19,21	-18,98%
Itaú	30,84	20,95	-32,07%
Petrobras	28,39	28,45	0,21%

Fonte: Elaborado pelo autor.

Ao analisar o retorno do investidor *Buy and Hold*, percebe-se que o mercado acionário no período analisado sofreu uma forte queda, também conhecida como *bear market*. Essa constatação, comparada aos resultados do modelo, indicam que os modelos conseguiram proteger os investidores de não realizarem as perdas caso não houvessem tomado nenhuma decisão, como o *buy and holder*.

Além disso, o índice do IBOVESPA, no período analisado, sofreu perdas de 17,2%, sendo assim, os investidores que houvessem atuado conforme a diretriz do modelo, também atenuariam tais perdas.

Por fim, os resultados apresentados pela MLP permitiram concluir que os modelos que foram construídos considerando dados de mais de uma empresa, denominados generalistas, obtiveram uma maior estabilidade no erro apresentado, quando analisado sob os conjuntos de testes. Além disso, a análise simulada demonstrou que os modelos foram capazes de proteger investidores que seguissem as recomendações de acordo com as regras estipuladas.

4.5.3 Avaliação Quantitativa - *Long Short-Term Memory*

Assim como fora realizado para as *Multilayer Perceptron*, executou-se o treinamento de cada modelo selecionado na Tabela 15, com seu respectivo conjunto de treinamento, e submeteu-se ao conjunto de teste de empresas distintas. O principal objetivo é avaliar o desempenho dos modelos selecionados atuando como generalistas. Após isso, consolidou-se os resultados na média do erro apresentado pelo mesmo. O resultado foi apresentado na Tabela 22.

Tabela 22 - Resultados para os modelos na atuação generalista para LSTM.

Tipo de modelo	Conjunto de Treinamento	RMSE (R\$)	MAE (R\$)
Generalista	B3	1,097 E+00	8,430 E-01
	Bradesco	1,096 E+00	6,324 E-01
	Itaú	9,313 E-01	7,242 E-01
	Petrobras	9,331 E-01	7,094 E-01
	Vale	6,349 E-01	4,932 E-01
Especialista	B3	1,106 E+00	8,667 E-01
	Bradesco	1,076 E+00	8,265 E-01
	Itaú	1,046 E+00	8,255 E-01
	Petrobras	9,563 E-01	7,137 E-01
	Vale	6,159 E-01	4,952 E-01

Fonte: Elaborado pelo autor.

Assim como visto nas MLPs, não necessariamente, o modelo generalista foi o que obteve o melhor resultado em todos os casos analisados, porém, foi o que apresentou o resultado mais estável (valores mais próximos) em todos os cenários analisados. Ressalta-se que para o caso da Vale, a média dos erros foi consideravelmente mais baixa, uma vez que o seu conjunto de teste foi o que obteve o maior erro e neste caso, não foi avaliado.

Após isto, avaliou-se os modelos sob a perspectiva especialista, ou seja, auferiu-se o erro para um conjunto de teste referente a mesma empresa que lhe fora treinado. Os resultados foram apresentados na Tabela 23.

Tabela 23 - Resultados para os modelos na atuação generalista para LSTM.

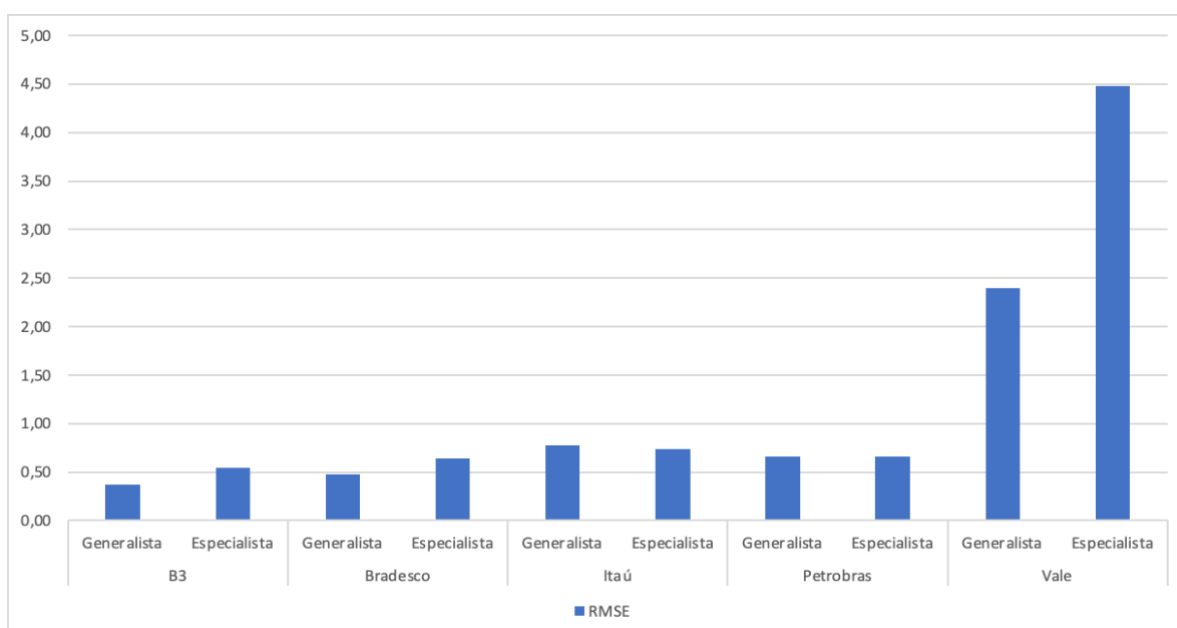
Empresa	Modelo	RMSE (R\$)	MAE (R\$)
B3	Generalista	3,693 E-01	2,933 E-01
	Especialista	5,479 E-01	4,560 E-01
Bradesco	Generalista	4,789 E-01	3,646 E-01
	Especialista	6,419 E-01	5,210 E-01
Itaú	Generalista	7,798 E-01	5,169 E-01
	Especialista	7,410 E-01	4,759 E-01
Petrobras	Generalista	6,640 E-01	5,113 E-01

	Especialista	6,580 E-01	5,099 E-01
Vale	Generalista	2,401 E+00	1,888 E+00
	Especialista	4,483 E+00	3,312 E+00

Fonte: Elaborado pelo autor.

Ao comparar o desempenho do modelo especialista com o generalista em cada empresa considerada, nota-se que, assim como nas *Multilayer Perceptron*, os modelos generalistas obtiveram um erro, médio, menor. Este comportamento pode ser visualizado na Figura 36, que apresenta o *root mean squared error* em cada cenário avaliado. Isso reforça que, em média, modelos construídos combinando dados de diversos contextos (diversas empresas) tendem a ter um resultado mais regular.

Figura 36 - RMSE por tipo de modelo e empresa para as LSTMs.



Fonte: Elaborado pelo autor.

Além disso, a Tabela 24 evidencia os melhores modelos e resultados em cenário avaliado para as *Long Short-Term Memory*.

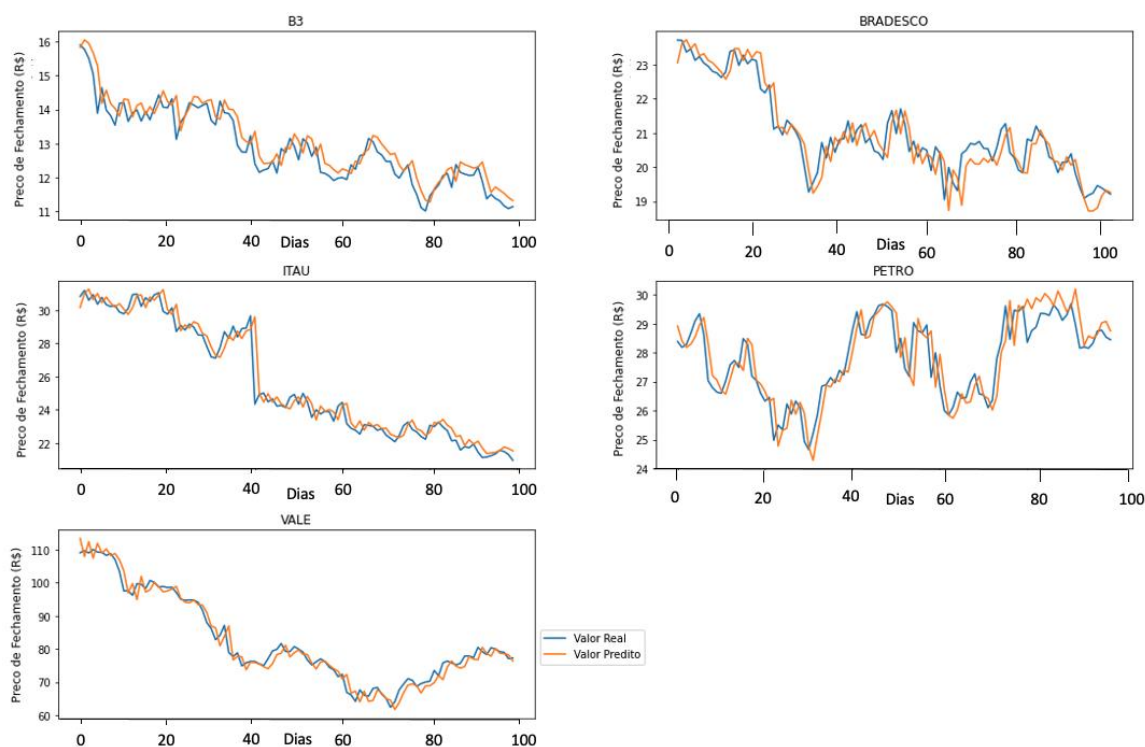
Tabela 24 - Resultados obtidos para as LSTMs.

Empresa	Modelo	RMSE (R\$)	MAE (R\$)
B3	Generalista	3,693E-01	2,933E-01
Bradesco	Generalista	4,789E-01	3,646E-01
Itaú	Especialista	7,410E-01	4,759E-01
Petrobras	Especialista	6,580E-01	5,099E-01
Vale	Generalista	2,401E+00	1,888E+00

Fonte: Elaborado pelo autor.

A Figura 37 representa graficamente o comportamento da ação e dos valores preditos de cada empresa.

Figura 37 - Valores preditos versus Valores reais para a LSTM – em reais (R\$).



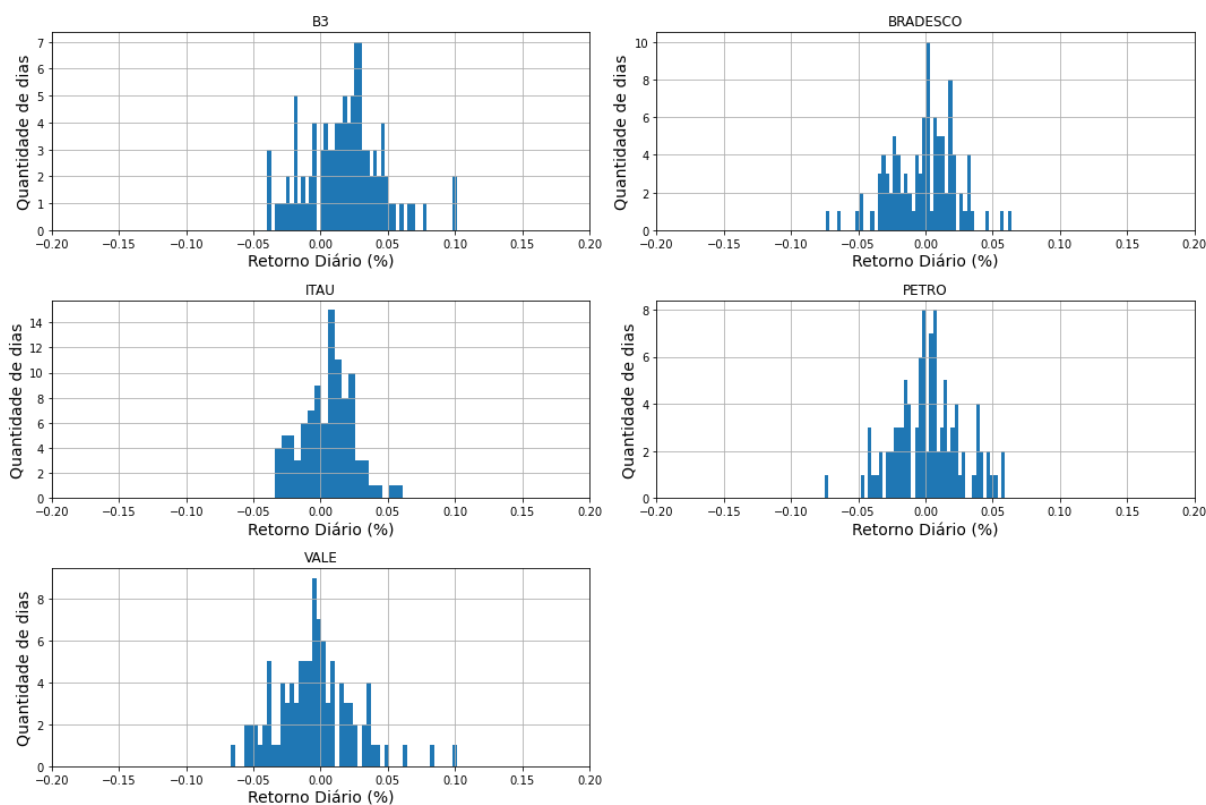
Fonte: Elaborado pelo autor.

Os gráficos obtidos pelas LSTMs, apresentado na Figura 37, comparado com os das MLP, demonstra que em determinados momentos de perturbação acentuada no mercado, as LSTMs demoraram mais para se acomodar em relação aos valores reais. Esse comportamento é visto nos gráficos da Petrobras e do Bradesco, onde tem uma queda ou subida abrupta. Além

disso, a Figura 38, que apresenta a distribuição do erro percentual para cada empresa para a LSTM, demonstra que para as LSTMs a concentração perto do zero ficou menor quando comparado às das MLPs, reforçando assim, o resultado inferior obtido pelas mesmas.

Em suma, os resultados apresentados pelas MLPs foram levemente melhores para cada cenário avaliado, exceto no caso da B3. As LSTMs apresentaram o menor erro e o maior erro para a predição de um dia, enquanto as MLPs apresentaram um range de erro menor. Portanto, esse comportamento, em conjunto com os resultados anteriores, permitem concluir que na média, as MLPs tiveram uma atuação melhor do que as *Long Short-Term Memory* em prever o valor futuro de cada ação.

Figura 38 – Distribuição do erro percentual de predição para LSTM.



Fonte: Elaborado pelo autor.

Por fim, este resultado, sob uma perspectiva momentânea do mercado, não indica que necessariamente um investidor hipotético obterá um lucro maior através das MLPs e, portanto, a fim de comparar a aplicabilidade (capacidade de gerar lucro) das técnicas, submeteu-se as LSTMs a mesma avaliação simulada que fora submetida as MLPs.

4.5.4 Avaliação Simulada - *Long Short-Term Memory*

A fim de avaliar o desempenho real das LSTMs, submeteu-se a mesma simulação realizada para as MLPs. A simulação seguiu o mesmo método que foi apresentado anteriormente.

Diferente do que foi observado na simulação para MLP, nas LSTMs todos os perfis tiveram recomendação de compra. A empresa em que o modelo a maior quantidade de compra foi a B3, somando-se a 69 recomendações. Além disso, a LSTM indicou 19 momentos em que algumas das empresas avaliadas teriam uma valorização superior a 2,5%, criando assim, a recomendação de compra para o conservador. A Tabela 25 apresenta quantas atuações teriam sido realizadas por tipo de investidor, caso seguisse as margens de segurança estipuladas.

Tabela 25 - Recomendações de compra por perfil em cada cenário para LSTM.

	B3	Bradesco	Itaú	Petro	Vale
Arrojado	69	15	34	26	6
Moderado	47	1	6	8	5
Conservador	11	0	1	4	3

Fonte: Elaborado pelo autor.

A maior quantidade de recomendações, quando comparado às advindas da MLP, reforçam que as LSTMs obtiveram um range de erro maior na predição de um dia para o outro, uma vez que em diversos cenários a rede indicou que a rede subiria ao menos 0,5% no dia seguinte. Ao utilizar a analogia dos perfis de investidores, este comportamento sugere que enquanto as MLPs atuaram conforme um conservador, as LSTMs se expuseram mais ao risco e, portanto, realizaram mais indicações e com um potencial de valorização maior. A Tabela 26 apresenta o retorno obtido por cada perfil de investidor avaliado.

Tabela 26 - Retorno sobre investimento obtido por perfil de investidor.

	B3	Bradesco	Itaú	Petro	Vale
Arrojado	-25,84%	-5,00%	-11,49%	-1,14%	-6,73%
Moderado	35,01%	1,81%	3,18%	0,14%	-5,90%
Conservador	12,84%	0,00%	1,41%	3,45%	-2,23%

Fonte: Elaborado pelo autor.

Ao comparar o resultado das LSTMs com o investidor *buy and hold*, nota-se que em todos os cenários, o modelo teria tido uma performance superior, exceto nos perfis de arrojado e moderado para a Petrobras.

Além disso, compara-se os resultados dos modelos com o desempenho do índice IBOVESPA, no mesmo período, que obteve perdas de 17%. Sendo assim, o investidor atuando pela diretriz do modelo, somente realizaria perdas maiores para o investidor arrojado na B3. Esse resultado, em conjunto com a comparação do *buy and holder*, reforça a capacidade do modelo de proteger o patrimônio do investidor.

Nota-se que quando levado em consideração os retornos apresentados pelas duas técnicas, as LSTMs tiveram o maior lucro e prejuízo. Isto reforça a analogia de que enquanto as LSTMs teriam um perfil arrojado, as MLPs teriam um perfil conservador.

Além disso, os retornos reforçam a característica de que o menor erro obtido pelas MLPs não garantem um lucro maior, quando analisado em uma janela de tempo definida, como fora realizado.

O resultado dessa análise permite concluir que não existe uma técnica certa para ser utilizada. Na média, as MLPs apresentaram um erro menor e expuseram o investidor a um risco menor. No entanto, as LSTMs obtiveram retornos e prejuízos maiores, e, conseqüentemente, expondo o investidor a um risco maior.

Por fim, os modelos que foram construídos através da avaliação de todas as empresas em conjunto, obtiveram, na média, um resultado melhor tanto pela perspectiva especialista quanto generalista.

5. CONCLUSÃO

O presente trabalho buscou analisar o desempenho das técnicas de *deep learning*: *Multilayer perceptron* e *Long Short-Term memory* aplicadas às cinco empresas mais significativas do principal índice do mercado de capitais brasileiro, o Ibovespa.

O objetivo, além de avaliar o desempenho das técnicas, foi validar se as abordagens propostas no estudo foram relevantes para construção de modelos que obtivessem um sucesso maior no objetivo geral, que é prever o valor futuro de ações.

A primeira abordagem consistiu em procurar o melhor conjunto de dados para cada técnica, assim sendo, avaliou-se quais atributos eram mais relevantes em cada cenário. A segunda abordagem, tratou-se de encontrar os valores mais apropriados de *lagging* para cada empresa individualmente, assim como para todas as empresas coletivamente.

A partir disto, a proposição foi construir modelos conforme dois tipos, generalistas e especialistas. O primeiro tipo consistiu no grupo de modelos que foram construídos através dos dados referentes a todas as empresas. O segundo grupo foi denominado especialista por fazerem uso dos dados de apenas uma empresa para obtenção de seus hiperparâmetros.

Para cada técnica avaliada, foram propostas 72 combinações distintas de hiperparâmetros, a qual o principal objetivo foi encontrar o melhor conjunto para cada grupo considerado, especialista e generalista.

Para avaliar os conjuntos selecionados, foram realizadas duas avaliações, a quantitativa e a simulada. A primeira delas teve como objetivo de avaliar os modelos conforme a atuação proposta e obter os modelos que obtiveram o menor erro no conjunto teste. Para tal, utilizou-se o RMSE e o MAE como métricas de erro. A segunda avaliação, a fim de avaliar a aplicabilidade dos modelos, consistiu em submeter os modelos selecionados na primeira avaliação à uma simulação de retorno para cada perfil de investidor considerado: arrojado, moderado e conservador.

Os resultados demonstraram que as *Long Short-Term Memory* obtiveram menor erro de predição para conjunto de dados com maior quantidade de atributos. Apresentaram uma redução no erro médio na medida que o valor de *lagging* era incrementado, além de obter um desempenho superior quando submetida a base de dados com todos os atributos disponíveis. Este comportamento não foi replicado pelas *Multilayer Perceptrons*, onde se observou um aumento no erro médio na medida que o valor de *lagging* fosse maior, além de obter um desempenho superior quando introduzida somente ao atributo do preço de fechamento da ação.

A avaliação quantitativa demonstrou que as *Multilayer Perceptron* obtiveram um erro menor do que as *Long Short-Term Memory*, exceto no caso da B3. Os menores erros, avaliados pelo método do RMSE, para Bradesco, Itaú, Petrobras e Vale foram de 0,44970, 0,69881, 0,64238, 2,00160 obtidos pelas MLPs, enquanto o melhor resultado para a B3 foi de 0,36930, obtido pela LSTMs. Além disso, os modelos generalistas foram os que obtiveram melhor resultado em quatro de cinco empresas para as MLPs, assim como obtiveram o melhor resultado em três, para as LSTMs.

Para avaliar o real desempenho dos modelos, utilizou-se a avaliação simulada. Tal resultado demonstrou que as LSTMs obtiveram mais recomendações de compra, quando comparada às das MLPs. Inclusive no cenário conservador, onde não foi obtida recomendação de compra para as MLPs. Além disso, as LSTMs obtiveram o maior lucro e prejuízo, enquanto as MLPs obtiveram ganhos e perdas menores.

Este comportamento permite concluir que as técnicas avaliadas possuem comportamento distintos em eventos que geram grande variação no valor da ação. As LSTMs possuem uma aptidão maior a grandes variações no valor futuro, enquanto as MLPs conseguiram se adaptar mais rápido após tais eventos, atenuando as perdas e o lucro.

Os resultados demonstraram que ambas as técnicas obtiveram sucesso no objetivo principal, haja visto que o investidor que tivesse seguido as recomendações de modelo (MLP/LSTM), haveria atenuado as perdas ou até mesmo lucrado em algumas ocasiões. Além disso, as MLPs obtiveram um erro médio menor, além de apresentar lucros e prejuízos menores, quando comparado as LSTMs. Portanto, conclui-se que neste experimento as MLPs demonstraram ser uma abordagem mais segura para o investidor, no entanto, tal característica deve ser mais artefato de mais estudos, uma vez que as LSTMs, no geral, são mais adaptadas as características das séries temporais.

Ademais, constatou-se que a construção de modelos através das informações de múltiplas empresas apresentou, na média, resultados melhores quando avaliados sob a perspectiva do conjunto de teste.

Por fim, como trabalhos futuros, indica-se a realização de três ações, sendo duas a respeito das técnicas e uma a respeito da base de dados.

A primeira delas é a utilização de ambas as técnicas para obter um sistema de predição híbrido, buscando assim, obter a melhor característica de cada. Neste cenário, para predições com pouca variação de um dia para o outro, utilizaria o resultado proveniente das *Multilayer*

Perceptron, enquanto para variações abruptas, haveria alguma ação seguindo a recomendação da *Long Short-Term Memory*.

A segunda ação trata-se de expandir as técnicas utilizadas neste estudo, analisando assim, se outras técnicas de *Deep Learning* podem ter um resultado superior a estas, além de avaliar se a proposição dos modelos generalistas se mantém para as mesmas.

Por último, recomenda-se um estudo a respeito das técnicas de *Deep Learning* quando submetidas a base de dados com frequências intra diárias. Além disso, sugere-se um tratamento para eliminar as primeiras transações do dia, que refletem os eventos ocorridos durante o período de fechamento do mercado. Acredita-se que neste cenário, tais eventos não impactariam tanto na variação do preço da ação e, conseqüentemente, no resultado do modelo.

REFERÊNCIAS

- ABELÉM, A. J. G.; PACHECO, M. A. C.; VELLASCO, Marley M. B. R. Modelagem de redes neurais artificiais para previsão de séries temporais. In: Simpósio Brasileiro de Redes Neurais, 2, 1995, São Carlos. Anais. São Carlos: Universidade Federal de São Carlos, 1995.
- ALTHELAYA, K. A; EL-ALFY, E.-S. M; MOHAMMED, S. (2018). Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU). 2018 21st Saudi Computer Society National Computer Conference (NCC). doi:10.1109/ngc.2018.8593076. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8593076>. Acesso em 12 mai de 2022.
- ASSAF NETO, Alexandre. Finanças corporativas e valor. 7. ed. São Paulo : Atlas, 2014.
- BERRAR, Daniel. (2018). Cross-Validation. 10.1016/B978-0-12-809633-8.20349-X. Disponível em: https://www.researchgate.net/profile/Daniel-Berrar/publication/324701535_Cross-Validation/links/5cb4209c92851c8d22ec4349/Cross-Validation.pdf. Acesso em 12 mai. 2022.
- BISCHOP, Christopher M. Neural networks for pattern recognition. Oxford university press, 1995.
- BOTTOU, L. Stochastic Gradient Descent Tricks. Neural Networks: Tricks of the Trade, 421–436. 2012. doi:10.1007/978-3-642-35289-8_25. Disponível em: https://link.springer.com/chapter/10.1007/978-3-642-35289-8_25. Acesso em 12 maio de 2022.
- CHAKRABORTY, G.; CHAKRABORTY, B. (2000). A novel normalization technique for unsupervised learning in ANN. IEEE Transactions on Neural Networks, 11(1), 253–257. doi:10.1109/72.822529. Disponível em: <https://ieeexplore.ieee.org/abstract/document/822529/>. Acesso em 12 mai. 2022.
- CHANDRA, R.; CHAND, S. 2016. Evaluation of co-evolutionary neural network architectures for time series prediction with mobile application in finance. Applied Soft Computing, 49:462–473, December 2016. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1568494616304239>. Acesso em 05 set 2022.
- CHEN, C.; ZHANG, Q.; YU, B.; YU, Z.; SKILLMAN-LAWRENCE, P. J.; MA, Q.; ZHANG, Y. (2020). Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier. Computers in Biology and Medicine, 103899. doi:10.1016/j.combiomed.2020. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0010482520302481>. Acesso em 05 set 2022.
- CHEN, K.; ZHOU, Y.; DAI, F. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market. IEEE International Conference on Big Data (Big Data), 2015, pp. 2823-2824, doi: 10.1109/BigData.2015.7364089. Disponível em: <https://ieeexplore.ieee.org/document/7364089>. Acesso em 05 set 2022.
- CHITENDERU, T. T.; MAREDZA, A.; SIBANDA, K. The Random Walk Theory And Stock Prices: Evidence From Johannesburg Stock Exchange. International Business & Economics

Research Journal (IBER), 13(6), 1241–1250. <https://doi.org/10.19030/iber.v13i6.8918>. Disponível em: <https://clutejournals.com/index.php/IBER/article/view/8918/8873>. Acesso em 07 out de 2022.

CICHY, R. M.; KAISER, D. Deep Neural Networks as Scientific Models. Trends in Cognitive Sciences. 2019. doi:10.1016/j.tics.2019.01.009. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1364661319300348>. Acesso em 12 mai de 2022.

DALABRIDA, D. Protótipo de um Sistema de Extração de Regras Simbólicas de Redes Neurais Artificiais Utilizadas na Tarefa de Classificação em Data Mining. Orientador: Mauro Marcelo Mattos. 2007. Trabalho de Conclusão de Curso; (Graduação em Bacharelado em Ciências da Computação) - Fundação Universidade Regional de Blumenau;

EL NAGA, I.; MURPHY, M. J. What is machine learning?. 2015. In machine learning in radiation oncology (pp. 3-11). Springer, Cham. Disponível em: <https://app.dimensions.ai/details/publication/pub.1011680030>. Acesso em 07 out de 2022.

EIZIRIK, Nelson et al. Mercado de capitais: regime jurídico. 2. ed. Rio de Janeiro: Renovar,

FAMA, E. F. Random Walks in Stock Market Prices. Financial Analysts Journal. 1995. 51(1), 75–80. doi:10.2469/faj.v51.n1.1861. Disponível em: <https://www.jstor.org/stable/4469865>. Acesso em 07 out de 2022.

FAUSETT, L. Fundamentals of Neural Networks, 1st ed.; Pearson: London, U.K., 1994

FILHO, R. T. A. Aplicação de redes neurais no controle de tuberculose bovina. 2016. Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2016

FLECK, L.; TAVARES, M. H.; YNG, E., HELMANN, A., ANDRADE, M. A. Redes Neurais Artificiais: Princípios Básicos. Revista Eletrônica Científica Inovação e Tecnologia, v. 1, n. 13, p. 47-57, jan./jun. 2016. Disponível em: <https://periodicos.utfpr.edu.br/recit/article/download/4330/Leandro> Acesso em 12 mai de 2022

FORTUNA, Eduardo. Mercado financeiro: produtos e serviços. Rio de Janeiro: Qualitymark, 2010.

GONÇALVES, R. M. Modelagem preditiva de linha de costa utilizando redes neurais artificiais. Boletim de Ciências Geodésicas, v. 16, n. 3, p. 420-444, 2010.

GREFF, K.; SRIVASTAVA, R. K.; KOUTNIK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. LSTM: A Search Space Odyssey. 2017. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2222–2232. doi:10.1109/tnnls.2016.2582924. Disponível em: <https://arxiv.org/abs/1503.04069>. Acesso em 12 maio 2022.

HAN, J.; KAMBER, M. Data mining. Concepts and techniques. São Francisco: Morgan Kaufmann Publishers, 2000.

HAYKIN, S. Redes neurais: princípios e prática. Porto Alegre: Bookman, 2001.

HIRANSHA, M.; GOPALAKRISHNAN, E. A.; MENON, V. K.; SOMAN, K. P. (2018). NSE Stock Market Prediction Using Deep-Learning Models. Procedia Computer Science, 132, 1351–1362. doi:10.1016/j.procs.2018.05.050 Disponível em:

<https://www.sciencedirect.com/science/article/pii/S1877050918307828>. Acesso em 12 mai. 2022

HOCHREITER, S.; SCHMIDHUBER, J. (1997). Long Short-term Memory. *Neural computation*, 9, 1735-80. doi:10.1162/neco.1997.9.8.1735. Disponível em: https://www.researchgate.net/publication/13853244_Long_Short-term_Memory. Acesso em 05 set 2022.

JANIESCH, C.; ZSCHECH, P.; HENRICH, K. Machine learning and deep learning. 2021. *Electronic Markets*, 31(3), 685-695. Disponível em: https://ideas.repec.org/a/spr/elmark/v31y2021i3d10.1007_s12525-021-00475-2.html. Acesso em 07 out de 2022.

KELLY, J.; KNOTTENBELT, W. Neural NILM. *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*. 2015. doi:10.1145/2821650.2821672. Disponível em <https://dl.acm.org/doi/abs/10.1145/2821650.2821672>. Acesso em 12 mai 2022.

KETKAR, N. (2017). Stochastic gradient descent. In *Deep learning with Python* (pp. 113-132). Apress, Berkeley, CA. Disponível em: https://link.springer.com/chapter/10.1007/978-1-4842-2766-4_8. Acesso em 12 mai de 2022.

KINGMA, D. P.; BA, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. Disponível em: <https://arxiv.org/abs/1412.6980> Acesso em 12 mai de 2022.

KRAUSS, C., DO, X. A.; HUCK, N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. 2017. *European Journal of Operational Research*, 259(2), 689–702. doi:10.1016/j.ejor.2016.10.031. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221716308657>. Acesso em 12 mai de 2022.

KRAUSS, C.; AHN DO, X.; HUCK, N. 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research*, 259 (2):689–702, June 2017. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221716308657>. Acesso em 05 set 2022.

LECUN, Y.; BENGIO, Y.; HINTON, G. 2015. Deep learning. *Nature*, 521(7553), 436–444. doi:10.1038/nature14539. Disponível em: <https://www.nature.com/articles/nature14539>. Acesso em 05 set 2022.

LEE, S. I.; YOO, S. J. Threshold-based portfolio: the role of the threshold and its applications. 2018. *The Journal of Supercomputing*. doi:10.1007/s11227-018-2577-1. Disponível em: <https://sejong.elsevierpure.com/en/publications/threshold-based-portfolio-the-role-of-the-threshold-and-its-appli>. Acesso em 12 mai de 2022.

LI, X.; YANG, L.; XUE, F; ZHOU, H. Time series prediction of stock price using deep belief networks with intrinsic plasticity. 2017 29th Chinese Control And Decision Conference (CCDC). doi:10.1109/ccdc.2017.7978707. Disponível em <https://ieeexplore.ieee.org/abstract/document/7978707/>. Acesso em 12 mai de 2022.

LUO, L.; XIONG, Y; LIU, Y; SUN, X. (2019). Adaptive gradient methods with dynamic bound of learning rate. arXiv preprint arXiv:1902.09843. Disponível em: <https://arxiv.org/abs/1902.09843>. Acesso em 12 mai de 2022.

MIKKULAINEN, R; LIANG, J; MEYERSON, E; RAWAL, A; FINK, D; FRANCON, O. HODJAT, B. Evolving Deep Neural Networks. Artificial Intelligence in the Age of Neural Networks and Brain Computing, 293–312. 2019. doi:10.1016/b978-0-12-815480-9.00015-3. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780128154809000153>. Acesso em 12 mai de 2022.

MITCHELL, R; FRANK, E. (2017). Accelerating the XGBoost algorithm using GPU computing. PeerJ Computer Science 3:e127 <https://doi.org/10.7717/peerj-cs.127>. Disponível em: <https://peerj.com/articles/cs-127/>. Acesso em 05 set 2022.

MOEENI, H.; BONAKDARI, H. (2017). Impact of Normalization and Input on ARMAX-ANN Model Performance in Suspended Sediment Load Prediction. Water Resources Management, 32(3), 845–863. doi:10.1007/s11269-017-1842-z. Disponível em: <https://link.springer.com/article/10.1007/s11269-017-1842-z>. Acesso em 12 mai 2022.

NAYAK, S.; MISRA, B.; BEHARA, H. (2014). Impact of Data Normalization on Stock Index Forecasting. International Journal of Computer Information Systems and Industrial Management Applications. 6. 357-369. Disponível em: http://www.mirlabs.org/ijcisim/regular_papers_2014/IJCISIM_24.pdf. Acesso em 12 mai. 2022.

NOBRE, J.; NEVES, R. F. (2019). Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to trade in the financial markets. Expert Systems with Applications, 125, 181–194. doi:10.1016/j.eswa.2019.01.083. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0957417419300995>. Acesso em 05 set 2022.

POTHUGANTI, Karunakar. Long Short-Term Memory (LSTM) Algorithm Based Prediction of Stock Market Exchange (January 20, 2021). International Journal of Research Publication and Reviews, Volume 2, Issue 1, ISSN 2582-7421, page 90-93. Disponível em: <https://ssrn.com/abstract=3770184>. Acesso em 12 maio de 2022.

RAUBER, T. W. Redes neurais artificiais. Universidade Federal do Espírito Santo, 2005; Disponível em: https://www.researchgate.net/profile/Thomas-Rauber-2/publication/228686464_Redес_neurais_artificiais/links/02e7e521381602f2bd000000/Redes-neurais-artificiais.pdf. Acesso em 12 mai de 2022.

ROONDIWALA, M; PATE, H.; VARMA, S. (2017). Predicting Stock Prices Using LSTM. International Journal of Science and Research (IJSR). 6. 10.21275/ART20172755. Disponível em: https://www.researchgate.net/publication/327967988_Predicting_Stock_Prices_Using_LSTM. Acesso em 05 set 2022.

RUDER, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747. Disponível em: <https://arxiv.org/abs/1609.04747> Acesso em 12 mai de 2022.

SAMARAWICKRAMA, A. J.; FERNANDO, T. G. I. A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. 2017. IEEE International Conference on Industrial and Information Systems (ICIIS). doi:10.1109/iciinfs.2017.83003. Acesso em 12 mai de 2022.

SAMUEL, A. L. Some Studies in Machine Learning Using the Game of Checkers. 1959 In: IBM Journal of Research and Development, vol. 3, no. 3, pp. 210-229, July 1959, doi: 10.1147/rd.33.0210. Disponível em: <https://ieeexplore.ieee.org/document/5392560>. Acesso em 10 out de 2022.

SEZER, O. B.; GUDELEK, M. U.; OZBAYOGLU, A. M. (2020). Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. Applied Soft Computing, 106181. doi:10.1016/j.asoc.2020.106181. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1568494620301216>. Acesso em 12 mai. 2022.

SHUANGLONG, L.; ZHANG, C.; MA, J. Cnn-lstm neural network model for quantitative strategy analysis in stock markets. In Neural Information Processing, pages 198–206. Springer International Publishing, 2017. Disponível em: https://www.researchgate.net/publication/320631410_CNN-LSTM_Neural_Network_Model_for_Quantitative_Strategy_Analysis_in_Stock_Markets. Acesso em 05 set 2022.

SINATORA, José Roberto Pereira. Mercado de capitais. Londrina : Editora e Distribuidora Educacional S.A., 2016. 224 p.

SUN, L; WANG, Y; HE, J.; LI, H.; PENG, D.; WANG, Y. (2020). A stacked LSTM for atrial fibrillation prediction based on multivariate ECGs. Health Information Science and Systems, 8(1). doi:10.1007/s13755-020-00103-x. Disponível em: <https://link.springer.com/article/10.1007/s13755-020-00103-x>. Acesso em 12 mai de 2022.

SUN, Y.; HUANG, X.; KROENING, D.; SHARP, J.; HILL, M.; ASHMORE, R. Testing deep neural networks. 2018. arXiv preprint arXiv:1803.04792. Disponível em: <https://arxiv.org/pdf/1803.04792.pdf> Acesso em 12 mai de 2022

TERZIYSKA, M.; TODOROV, Y.; DOBREVA, M. (2018). Efficient error based metrics for Fuzzy-Neural Network Performance Evaluation. Doi: [dx.doi.org/10.1007/978-3-319-65530-7_17](https://doi.org/10.1007/978-3-319-65530-7_17). Acesso em 07 out de 2022.

UMORU, B.; UDOBI-OWOLOJA, P. I.; NZEKWE, G. U.; IYIEGBUNIWE, W. C.; EZIKER, J. E. Are stock returns predictable? The myth of efficient market hypothesis and random walk theory using Nigerian market data. 2020. International Journal of Economics, Business and Management Research, 4(7), 115-130. Disponível em: https://ijebmr.com/uploads/pdf/archivepdf/2020/IJEBMR_580.pdf. Acesso em 07 out de 2022.

XU, Y.; GOODACRE, R. On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. 2018. Journal of Analysis and Testing, 2(3), 249–262. doi:10.1007/s41664-018-0068-2. Disponível em: <https://link.springer.com/article/10.1007/s41664-018-0068-2>. Acesso em 12 mai. 2022.

YILDIRIM, Ö. (2018). A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. *Computers in Biology and Medicine*, 96, 189–202. doi:10.1016/j.combiomed.2018.03. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0010482518300738>. Acesso em 12 mai de 2022.

APÊNDICE A – Combinações possíveis da otimização para MLP

O apêndice tem como objetivo de representar todas as possibilidades avaliadas na otimização de hiperparâmetros para MLP.

Combinação de hiperparâmetros	Número de Camada Ocultas	Quantidade de Neurônios nas camadas ocultas	Função de pertinência da camada oculta	Número de Épocas	Tamanho do Lote
1	1	8	Tanh	50	16
2	1	8	Tanh	100	16
3	1	8	Relu	50	16
4	1	8	Relu	100	16
5	1	8	Sigmoid	50	16
6	1	8	Sigmoid	100	16
7	1	32	Tanh	50	16
8	1	32	Tanh	100	16
9	1	32	Relu	50	16
10	1	32	Relu	100	16
11	1	32	Sigmoid	50	16
12	1	32	Sigmoid	50	16
13	1	128	Tanh	100	16
14	1	128	Tanh	50	16
15	1	128	Relu	100	16
16	1	128	Relu	50	16
17	1	128	Sigmoid	100	16
18	1	128	Sigmoid	50	16
19	2	8	Tanh	100	16
20	2	8	Tanh	50	16
21	2	8	Relu	100	16
22	2	8	Relu	50	16
23	2	8	Sigmoid	100	16
24	2	8	Sigmoid	50	16

Combinação de hiperparâmetros	Número de Camada Ocultas	Quantidade de Neurônios nas camadas ocultas	Função de pertinência da camada oculta	Número de Épocas	Tamanho do Lote
25	2	32	Tanh	100	16
26	2	32	Tanh	50	16
27	2	32	Relu	100	16
28	2	32	Relu	50	16
29	2	32	Sigmoid	100	16
30	2	32	Sigmoid	50	16
31	2	128	Tanh	100	16
32	2	128	Tanh	50	16
33	2	128	Relu	100	16
34	2	128	Relu	50	16
35	2	128	Sigmoid	100	16
36	2	128	Sigmoid	50	16
37	1	8	Tanh	100	32
38	1	8	Tanh	50	32
39	1	8	Relu	100	32
40	1	8	Relu	50	32
41	1	8	Sigmoid	100	32
42	1	8	Sigmoid	50	32
43	1	32	Tanh	100	32
44	1	32	Tanh	50	32
45	1	32	Relu	100	32
46	1	32	Relu	50	32
47	1	32	Sigmoid	100	32
48	1	32	Sigmoid	50	32
49	1	128	Tanh	100	32
50	1	128	Tanh	50	32
51	1	128	Relu	100	32
52	1	128	Relu	50	32
53	1	128	Sigmoid	100	32
54	1	128	Sigmoid	50	32
55	2	8	Tanh	100	32

Combinação de hiperparâmetros	Número de Camada Ocultas	Quantidade de Neurônios nas camadas ocultas	Função de pertinência da camada oculta	Número de Épocas	Tamanho do Lote
56	2	8	Tanh	50	32
57	2	8	Relu	100	32
58	2	8	Relu	50	32
59	2	8	Sigmoid	100	32
60	2	8	Sigmoid	50	32
61	2	32	Tanh	100	32
62	2	32	Tanh	50	32
63	2	32	Relu	100	32
64	2	32	Relu	50	32
65	2	32	Sigmoid	100	32
66	2	32	Sigmoid	50	32
67	2	128	Tanh	100	32
68	2	128	Tanh	50	32
69	2	128	Relu	100	32
70	2	128	Relu	50	32
71	2	128	Sigmoid	100	32
72	2	128	Sigmoid	50	32

APÊNDICE B – Combinações possíveis da otimização na LSTM

O apêndice tem como objetivo de representar todas as possibilidades avaliadas na otimização de hiperparâmetros para LSTM.

Combinação de hiperparâmetros	Arquitetura	Quantidade de Neurônios	Função de Pertinência da camada oculta	Número de Épocas	Tamanho do Lote
1	Vanila	32	Tanh	100	16
2	Vanila	32	Tanh	200	16
3	Vanila	32	Relu	100	16
4	Vanila	32	Relu	200	16
5	Vanila	32	Sigmoid	100	16
6	Vanila	32	Sigmoid	200	16
7	Vanila	128	Tanh	100	16
8	Vanila	128	Tanh	200	16
9	Vanila	128	Relu	100	16
10	Vanila	128	Relu	200	16
11	Vanila	128	Sigmoid	100	16
12	Vanila	128	Sigmoid	200	16
13	Stacked	32	Tanh	100	16
14	Stacked	32	Tanh	200	16
15	Stacked	32	Relu	100	16
16	Stacked	32	Relu	200	16
17	Stacked	32	Sigmoid	100	16
18	Stacked	32	Sigmoid	200	16
19	Stacked	128	Tanh	100	16
20	Stacked	128	Tanh	200	16
21	Stacked	128	Relu	100	16
22	Stacked	128	Relu	200	16
23	Stacked	128	Sigmoid	100	16
24	Stacked	128	Sigmoid	200	16
25	Bidirecional	32	Tanh	100	16
26	Bidirecional	32	Tanh	200	16

Combinação de hiperparâmetros	Arquitetura	Quantidade de Neurônios	Função de Pertinência da camada oculta	Número de Épocas	Tamanho do Lote
27	Bidirecional	32	Relu	100	16
28	Bidirecional	32	Relu	200	16
29	Bidirecional	32	Sigmoid	100	16
30	Bidirecional	32	Sigmoid	200	16
31	Bidirecional	128	Tanh	100	16
32	Bidirecional	128	Tanh	200	16
33	Bidirecional	128	Relu	100	16
34	Bidirecional	128	Relu	200	16
35	Bidirecional	128	Sigmoid	100	16
36	Bidirecional	128	Sigmoid	200	16
37	Vanila	32	Tanh	100	32
38	Vanila	32	Tanh	200	32
39	Vanila	32	Relu	100	32
40	Vanila	32	Relu	200	32
41	Vanila	32	Sigmoid	100	32
42	Vanila	32	Sigmoid	200	32
43	Vanila	128	Tanh	100	32
44	Vanila	128	Tanh	200	32
45	Vanila	128	Relu	100	32
46	Vanila	128	Relu	200	32
47	Vanila	128	Sigmoid	100	32
48	Vanila	128	Sigmoid	200	32
49	Stacked	32	Tanh	100	32
50	Stacked	32	Tanh	200	32
51	Stacked	32	Relu	100	32
52	Stacked	32	Relu	200	32
53	Stacked	32	Sigmoid	100	32
54	Stacked	32	Sigmoid	200	32
55	Stacked	128	Tanh	100	32
56	Stacked	128	Tanh	200	32
57	Stacked	128	Relu	100	32

Combinação de hiperparâmetros	Arquitetura	Quantidade de Neurônios	Função de Pertinência da camada oculta	Número de Épocas	Tamanho do Lote
58	Stacked	128	Relu	200	32
59	Stacked	128	Sigmoid	100	32
60	Stacked	128	Sigmoid	200	32
61	Bidirecional	32	Tanh	100	32
62	Bidirecional	32	Tanh	200	32
63	Bidirecional	32	Relu	100	32
64	Bidirecional	32	Relu	200	32
65	Bidirecional	32	Sigmoid	100	32
66	Bidirecional	32	Sigmoid	200	32
67	Bidirecional	128	Tanh	100	32
68	Bidirecional	128	Tanh	200	32
69	Bidirecional	128	Relu	100	32
70	Bidirecional	128	Relu	200	32
71	Bidirecional	128	Sigmoid	100	32
72	Bidirecional	128	Sigmoid	200	32