

UNIVERSIDADE ESTADUAL DE PONTA GROSSA
SETOR DE CIÊNCIAS SOCIAIS APLICADAS
DEPARTAMENTO DE ECONOMIA

THIAGO JOSÉ PINHEIRO

MODELOS DE DEEP LEARNING E PREVISÃO DE PREÇO DE AÇÕES: ESTUDO DE
CASOS DA BOLSA BRASILEIRA

PONTA GROSSA

2022

THIAGO JOSÉ PINHEIRO

MODELOS DE DEEP LEARNING E PREVISÃO DE PREÇO DE AÇÕES: ESTUDO DE
CASOS DA BOLSA BRASILEIRA

Dissertação apresentada como requisito parcial para
obtenção do título de mestre em Economia na
Universidade Estadual de Ponta Grossa junto ao
Programa de Pós-Graduação Stricto Sensu de
Economia Aplicada.

Orientador: Dr. Hermes Yukio Higachi.

PONTA GROSSA

2022

P654 Pinheiro, Thiago José
Modelos de Deep Learning e previsão de preço de ações: estudo de casos da bolsa brasileira / Thiago José Pinheiro. Ponta Grossa, 2022.
93 f.

Dissertação (Mestrado em Economia - Área de Concentração: Economia),
Universidade Estadual de Ponta Grossa.

Orientador: Prof. Dr. Hermes Yukio Higachi.

1. Redes neurais. 2. Random walk. 3. Validação cruzada. 4. Aprendizado profundo. 5. Preço de ações. I. Higachi, Hermes Yukio. II. Universidade Estadual de Ponta Grossa. Economia. III.T.

CDD: 330



UNIVERSIDADE ESTADUAL DE PONTA GROSSA
Av. General Carlos Cavalcanti, 4748 - Bairro Uvaranas - CEP 84030-900 - Ponta Grossa - PR - <https://uepg.br>

TERMO

TERMO DE APROVAÇÃO

THIAGO JOSÉ PINHEIRO

MODELOS DE DEEP LEARNING E PREVISÃO DE PREÇO DE AÇÕES: ESTUDO DE CASOS DA BOLSA BRASILEIRA

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Economia, da Universidade Estadual de Ponta Grossa, pela seguinte banca examinadora:

Ponta Grossa, 30 de março de 2022

Professor Dr. Hermes Yukio Higaschi – UEPG

Professor. Dr. Karlo Marques Junior - UEPG

Professor. Dr. João Basílio Pereima Neto – UFPR



Documento assinado eletronicamente por **Hermes Yukio Higachi, Professor(a)**, em 30/03/2022, às 20:31, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.

Documento assinado eletronicamente por **Karlo Marques Junior, Professor(a)**, em 31/03/2022, às 15:51, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **João Basilio Pereima Neto, Usuário Externo**, em 14/04/2022, às 07:11, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida no site <https://sei.uepg.br/autenticidade> informando o código verificador **0919804** e o código CRC **8F7D047B**.

Dedico aos meus pais, Maria Josmilda Hass e Jefferson Pinheiro, por todo amor, dedicação, carinho e compreensão que me forneceram durante a minha trajetória. Às minhas irmãs Kethlyn e Mariana, por todo carinho e incentivo. Aos meus avós Eva, Pedro e Pedro (in memoriam) e Roseli, que me apoiaram e me dedicaram muito amor e carinho.

AGRADECIMENTOS

Agradeço primeiro à Deus, pela sabedoria, energia e discernimento para realização desse trabalho.

Aos meus pais, pela dedicação e contribuição em todos os momentos me auxiliando e acreditando em mim, me apoiando e estando ao meu lado nas minhas escolhas. As minhas irmãs, Kethlyn e Mariana, pelo apoio e pela compreensão, vocês sempre acreditaram e confiaram em mim.

Ao meu orientador Dr. Hermes Yukio Higachi, por mais esta orientação, sua paciência e ajuda. Muito obrigado pela confiança e por compartilhar todo o seu conhecimento de forma clara, didática e motivadora.

Ao Luis Guilherme Ribeiro, o qual sempre esteve disponível para me auxiliar e fornecer ótimas sugestões, com certeza meu trabalho não teria sido o mesmo sem sua colaboração.

Aos professores do Programa de mestrado, por toda a contribuição durante a minha formação.

Aos colegas do Mestrado, a todos aqueles que direta ou indiretamente, contribuíram para que chegasse à conclusão de mais uma etapa de minha vida.

Eu agradeço a todas as pessoas que me auxiliaram e me ajudaram de alguma forma durante o desenvolvimento desta dissertação, e aqueles que eu esqueci de mencionar aqui. Eu espero que vocês saibam que eu sou extremamente grato por tudo o que vocês fizeram e fazem por mim.

RESUMO

Pinheiro, T. J. **Modelos de Deep Learning e Previsão de Preço de Ações: Estudo de Casos da Bolsa Brasileira**. 2022. Dissertação de Mestrado. Universidade Estadual de Ponta Grossa, Ponta Grossa, Paraná, 2022.

O objetivo deste trabalho tem o intuito de verificar e comparar os modelos de Redes Neurais Recorrentes de aprendizado profundo RNN univariado e multivariado denominados nesse trabalho como RNN-U e RNN-M e modelos LSTM univariado e multivariado denominados LSTM-U e LSTM-M, considerando-se como estudo de caso as ações da PETR3, PETR4, ITUB4 e ABEV3. Duas foram as contribuições deste estudo: primeiro, propor e avaliar pelo teste da razão da variância se as ações selecionadas seguem um processo estocástico Random Walk ou a validade da HME na forma sua forma fraca; segundo, propor e aplicar a técnica de validação cruzada para séries temporais para avaliar o poder preditivo de modelos de RNN e LSTM univariados e multivariados na previsão dos preços de ações. Os resultados empíricos do teste da razão da variância mostraram que os retornos de ações da PETR3 e PETR4 apresentam autocorrelação serial negativa de primeira ordem, enquanto os retornos de ações da ABEV3 e ITUB4 apresentam autocorrelação serial negativa também de ordens superiores. Com efeito, como os modelos RNN LSTM-U e RNN LSTM-M possuem memória curta (1 período) e longa (múltiplos períodos), reportaram maior poder preditivo em termos das várias métricas usadas, do que os modelos RNN-U e RNN-M que possuem apenas memória curta ou de 1 período, no caso de ações da ABEV3 e ITUB4.

Palavras-chave: Redes Neurais, Random Walk, Validação Cruzada, Aprendizado Profundo, Preço de Ações.

ABSTRACT

Pinheiro, T. J. **Deep Learning Models and Stock Price Forecast: Case Study of Brazilian Stock Exchange**. 2022. Masters' Dissertation. State University of Ponta Grossa, Ponta Grossa, Paraná, 2022.

The objective of this work is to verify and compare the univariate and multivariate RNN deep learning Recurrent Neural Network models called in this work as RNN-U and RNN-M and univariate and multivariate LSTM models called LSTM-U and LSTM-M, considering as a case study the actions of PETR3, PETR4, ITUB4 and ABEV3. Two were the contributions of this study: first, to propose and evaluate by the variance ratio test whether the selected stocks follow a Random Walk stochastic process or the validity of the HME in its weak form; second, to propose and apply the cross-validation technique for time series to assess the predictive power of univariate and multivariate RNN and LSTM models in predicting stock prices. The empirical results of the variance ratio test showed that the returns of shares of PETR3 and PETR4 present negative serial autocorrelation of the first order, while the returns of shares of ABEV3 and ITUB4 also present negative serial autocorrelation of higher orders. Indeed, as the RNN LSTM-U and RNN LSTM-M models have short (1 period) and long (multiple periods) memory, they reported greater predictive power in terms of the various metrics used than the RNN-U and RNN-M that have only short or 1-period memory, in the case of ABEV3 and ITUB4 stocks.

Keywords: Neural Networks, Random Walk, Cross-Validation, Deep Learning, Stock Price.

LISTA DE FIGURAS

FIGURA 1- MODELO SIMPLES DE APRENDIZAGEM DE MÁQUINA.....	22
FIGURA 2 - MODELO DE NEURÔNIO BIOLÓGICO.....	25
FIGURA 3 - MODELO NÃO LINEAR DE UM NEURÔNIO, ROTULADO K	25
FIGURA 4- TRANSFORMAÇÃO AFIM PRODUZIDA PELA PRESENÇA E UM VIÉS..	26
FIGURA 5- OUTRO MODELO NÃO LINEAR DE UM NEURÔNIO; LEVA EM CONTA O VIÉS bk	27
FIGURA 6 – (A) FUNÇÃO LIMITE, (B) FUNÇÃO SIGMOIDE PARA PARÂMETRO DE INCLINAÇÃO VARIÁVEL a	27
FIGURA 7 - REDE FEEDFORWARD COM UMA ÚNICA CAMADA DE NEURÔNIOS	29
FIGURA 8 - REDE FEEDFORWARD TOTALMENTE CONECTADA COM UMA CAMADA OCULTA E UMA CAMADA DE SAÍDA	29
FIGURA 9 - REDE RECORRENTE SEM LOOPS DE REALIMENTAÇÃO E SEM NEURÔNIOS OCULTOS.....	31
FIGURA 10 - REDE RECORRENTE COM NEURÔNIOS OCULTOS.....	32
FIGURA 11 - ESTRUTURA DE UMA RNN COM UMA ÚNICA CAMADA OCULTA ...	33
FIGURA 12 - ESTRUTURA DE UMA RNN COM UMA DUAS CAMADAS OCULTAS	33
FIGURA 13 - COMPUTAÇÃO DE UMA RNN.....	34
FIGURA 14 - DIAGRAMA EM BLOCOS DE UMA RNA	35
FIGURA 15 - ESTRUTURA DE LSTM	40
FIGURA 16 - AÇÃO PETR3 DE 01/01/2012 A 31/12/2021	50
FIGURA 17 - CONJUNTO DE DADOS DA AÇÃO PETR3.....	50
FIGURA 18 - EXEMPLO DE ESTRUTURA DE UMA RNA.....	52
FIGURA 19 - COMPOSIÇÃO DOS 5 SPLITS PARA VALIDAÇÃO CRUZADA	55
FIGURA 20 - GRADIENTE DESCENTE PARA AÇÃO PETR3: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M.....	60
FIGURA 21 - MÉTRICAS ESTATÍSTICAS AÇÃO PETR3: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M	61
FIGURA 22 - GRADIENTE DESCENTE PARA AÇÃO PETR4: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M.....	62
FIGURA 23 - MÉTRICAS ESTATÍSTICAS AÇÃO PETR4: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M	63

FIGURA 24 - GRADIENTE DESCENTE PARA AÇÃO ITUB4: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M.....	64
FIGURA 25 - FIGURA 25 – MÉTRICAS ESTATÍSTICAS AÇÃO ITUB4: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M	65
FIGURA 26 - GRADIENTE DESCENTE PARA AÇÃO ABEV3: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M.....	66
FIGURA 27 - MÉTRICAS ESTATÍSTICAS AÇÃO ABEV3: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M	67
FIGURA 28 - VALIDAÇÃO CRUZADA AÇÃO PETR3 MODELO RNN-U	69
FIGURA 29 - VALIDAÇÃO CRUZADA AÇÃO PETR3 MODELO RNN-M	69
FIGURA 30 – VALIDAÇÃO CRUZADA AÇÃO PETR3 MODELO LSTM-U	70
FIGURA 31 - VALIDAÇÃO CRUZADA AÇÃO PETR3 MODELO LSTM-M	70
FIGURA 32 - VALIDAÇÃO CRUZADA AÇÃO PETR4 MODELO RNN-U	71
FIGURA 33 - VALIDAÇÃO CRUZADA AÇÃO PETR4 MODELO RNN-M	72
FIGURA 34 - VALIDAÇÃO CRUZADA AÇÃO PETR4 MODELO LSTM-U	72
FIGURA 35 - VALIDAÇÃO CRUZADA AÇÃO PETR4 MODELO LSTM-M	73
FIGURA 36 - VALIDAÇÃO CRUZADA AÇÃO ITUB4 MODELO RNN-U	74
FIGURA 37 - VALIDAÇÃO CRUZADA AÇÃO ITUB4 MODELO RNN-M.....	74
FIGURA 38 - VALIDAÇÃO CRUZADA AÇÃO ITUB4 MODELO LSTM-U	75
FIGURA 39 - VALIDAÇÃO CRUZADA AÇÃO ITUB4 MODELO LSTM-M	75
FIGURA 40 - VALIDAÇÃO CRUZADA AÇÃO ABEV3 MODELO RNN-U.....	76
FIGURA 41 - VALIDAÇÃO CRUZADA AÇÃO ABEV3 MODELO RNN-M	77
FIGURA 42 - VALIDAÇÃO CRUZADA AÇÃO ABEV3 MODELO LSTM-U.....	77
FIGURA 43 - VALIDAÇÃO CRUZADA AÇÃO ABEV3 MODELO LSTM-M.....	78
FIGURA 44 - COMPARATIVO DE R2 ENTRE VALIDAÇÃO CRUZADA E TREINO-TESTE-VALIDAÇÃO: PETR3 E PETR4.....	78
FIGURA 45 - COMPARATIVO DE R2 ENTRE VALIDAÇÃO CRUZADA E TREINO-TESTE-VALIDAÇÃO: ITUB4 E ABEV3	79

LISTA DE TABELAS

TABELA 1 - TESTE DA RAZÃO DA VARIÂNCIA PARA N=2, PERÍODO 01/01/2012 A 31/12/2021	58
TABELA 2 - TESTE DA RAZÃO DA VARIÂNCIA PARA N=5, PERÍODO 01/01/2012 A 31/12/2021	59
TABELA 3 - TESTE DA RAZÃO DA VARIÂNCIA PARA N=15, PERÍODO 01/01/2012 A 31/12/2021	59
TABELA 4 – HIPER PARÂMETROS PARA TREINAMENTO DA AÇÃO PETR3	61
TABELA 5 – HIPER PARÂMETROS PARA TREINAMENTO DA AÇÃO PETR4	62
TABELA 6 – HIPER PARÂMETROS PARA TREINAMENTO DA AÇÃO ITUB4.....	64
TABELA 7 – HIPER PARÂMETROS PARA TREINAMENTO DA AÇÃO ABEV3	66
TABELA 8 - RESULTADOS AÇÃO PETR3 – VALIDAÇÃO CRUZADA.....	68
TABELA 9 - RESULTADOS AÇÃO PETR4 – VALIDAÇÃO CRUZADA.....	71
TABELA 10 - RESULTADOS AÇÃO ITUB4 – VALIDAÇÃO CRUZADA	73
TABELA 11 - RESULTADOS AÇÃO ABEV3 – VALIDAÇÃO CRUZADA.....	76

SUMÁRIO

1	INTRODUÇÃO.....	13
2	REVISÃO DE LITERATURA.....	16
2.1	A HIPÓTESE DOS MERCADOS EFICIENTES	16
2.2	FINANÇAS COMPORTAMENTAIS	18
2.2.1	Heurísticas e Vieses Comportamentais	18
2.2.2	Os Limites de Arbitragem	20
2.3	INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA.....	21
2.4	REDES NEURAIS ARTIFICIAIS – RNA.....	24
2.5	RECURRENT NEURAL NETWORKS – RNN.....	31
2.6	PROCESSO DE APRENDIZAGEM	35
2.6.1	Aprendizagem por Correção de Erro	35
2.6.2	Aprendizagem Baseada em Memória	36
2.7	APRENDIZADO PROFUNDO – DEEP LEARNING	37
2.8	LONG SHORT-TERM MEMORY (LSTM)	38
2.9	REDES NEURAIS NA PREVISÃO DO MERCADO FINANCEIRO	42
2.10	MERCADO DE AÇÕES	44
3	METODOLOGIA.....	46
3.1	TESTE DA RAZÃO DA VARIÂNCIA.....	46
3.1.1	Coleta e Tratamento de Dados	49
3.2	AVALIAÇÃO DO PODER PREDITIVO DE MODELOS DE REDES NEURAIS RNN E LSTM	51
3.2.1	Modelo de Redes Neurais	51
3.2.2	Validação Cruzada em Série Temporal	54
3.2.3	Gradiente Descendente Estocástico	56
4	RESULTADOS E DISCUSSÕES.....	58
4.1	TESTE DA RAZÃO DA VARIÂNCIA PARA RANDOM WALK	58

4.2	AVALIAÇÃO DOS HIPER PARÂMETROS DOS MODELOS.....	60
4.2.1	Resultados da Ação PETR3.....	60
4.2.2	Resultados da Ação PETR4.....	62
4.2.3	Resultados da Ação ITUB4	63
4.2.4	Resultados da Ação ABEV3.....	66
4.3	AVALIAÇÃO DOS MODELOS POR VALIDAÇÃO CRUZADA	67
4.3.1	Validação Cruzada PETR3	68
4.3.2	Validação Cruzada PETR4	71
4.3.3	Validação Cruzada ITUB4	73
4.3.4	Validação Cruzada ABEV3	76
4.4	COMPARATIVO DOS RESULTADOS	78
5	CONSIDERAÇÕES FINAIS.....	80
	REFERÊNCIAS	81
	ANEXO A – ATIVOS QUE COMPÕEM A CARTEIRA IBOV DE SET A DEZ 2021 .	87
	ANEXO B – SCRIPT R TESTE RAZÃO DA VARIÂNCIA	89

1 INTRODUÇÃO

Nos últimos anos se observa muitos indivíduos buscando alternativas para investimentos, principalmente voltado para o mercado de ações.

Uma das hipóteses que possui grande relevância foi proposta pelo economista Eugene Fama (1970), a qual é intitulada Hipótese de Mercados Eficientes – HME, de acordo com essa hipótese o mercado é considerado eficiente uma vez que conseguiria refletir de forma rápida e competente qualquer informação disponível sobre preços dos ativos, tornando impossível obter ganhos anormais, ou seja, a detenção de tais informações sobre o mercado não ocasionaria mudança em relação ao retorno esperado.

As ações podem sofrer desvios em relação ao seu preço fundamental causados por traders irracionais, em contrapartida as operações de arbitragem realizadas por traders racionais deverão eliminar estas ineficiências de mercado, ou seja, há um equilíbrio no mercado.

A partir de 1970 houve o desenvolvimento de estudos relacionados à Economia Comportamental, pontuando que a racionalidade é limitada. Para analisar a economia de forma mais abrangente se passou a analisar fatores econômicos, da psicologia cognitiva, sociológicos, trazendo assim um campo mais amplo na análise de economia e finanças.

Daniel Kahneman e Amos Tversky (1979) incorporam a racionalidade limitada em suas pesquisas, gerando um mapeamento da aplicação deste conceito, trazendo novos indicadores sobre o comportamento humano. Nesse aspecto cabe destacar as heurísticas de julgamento e o efeito *framing*¹ em conjunto com a Teoria dos Prospectos: surgem novos elementos que passam a ser analisados nos estudos sobre tomada de decisão pelos agentes econômicos em um ambiente de incerteza e complexidade.

Os preços podem oscilar sistematicamente como observam Bondt e Thaler (1985), ou seja, os agentes estão sujeitos a reações emocionais extremas causadas por fatores externos. Segundo Kahneman e Tversky (1979) os agentes podem ter excesso de confiança em sua capacidade de prever o preço futuro das ações.

No mercado de ações há vários fatos estilizados ou propriedades estatísticas de retornos em logs como o excesso de volatilidade em relação aos fundamentos, a aglomeração de volatilidade que são considerados anomalias pela Hipótese de Mercados Eficientes (HME) ou ainda o excesso de retorno ajustado pelo risco.

¹ O efeito framing se refere à existência de falhas na percepção humana, é o viés que descreve de que modo a tomada de decisão pode ser afetada pela maneira como o problema é formulado ou pela forma como as opções são apresentadas (enquadradas).

A HME ao supor a ausência de limites a arbitragem, propõe que os analistas fundamentalistas com expectativas racionais deverão inexoravelmente realizar operações de arbitragem sem risco. O preço da ação deverá convergir rapidamente para o seu valor fundamental, sendo improvável a presença de desvio persistente do preço da ação e de dependência temporal, passíveis de serem exploradas por meio de estratégias de seleção de ações ou modelos de previsão, sendo improvável a presença de desvio persistente do preço da ação (CONT, 2000; THALER, 2005; SHLEIFER; VISHNY, 2005).

Por sua vez, a Teoria de Finanças Comportamentais propõe que há limites a arbitragem causada pelo risco fundamental, risco de noise traders e custos de transações. Desvios persistentes do preço da ação em relação ao seu valor fundamental e ineficiências como a presença de autocorrelação positiva e/ou negativa podem emergir no mercado de ações, devido a reação exagerada ou reação nula de acionistas grafistas e/ou mal-informados com racionalidade limitada, que usam as heurísticas da ancoragem, representatividade e disponibilidade no processo de julgamento.

Com o intuito de melhorar a previsibilidade e a confiabilidade dos modelos de previsão, além da complexidade dos algoritmos usados no processo de tomada de decisão, alguns pesquisadores também tentaram uma combinação de Redes Neurais Artificiais - RNA e lógica fuzzy.

Os modelos de Redes Neurais Artificiais (RNA) são utilizados para melhorar a precisão das previsões em séries temporais. Vários modelos são baseados em Redes Neurais Recorrentes (HOPFIELD, 1982) e LSTMs (HOCHREITER; SCHMIDHUBER, 1997).

Redes Neurais Recorrentes (RNNs) têm sido amplamente adotadas em áreas de pesquisa relacionadas a dados sequenciais, como texto, áudio e vídeo. A memória de longo prazo (LSTM) se adapta na solução de problemas das dependências de longo prazo. O LSTM tornou-se o foco do aprendizado profundo.

Algoritmos de aprendizado profundo possuem a capacidade de identificar padrões existentes nos dados e explorá-los por meio de um processo de aprendizagem simples (BENGIO; GOODFELLO; COURVILLE, 2015). Ao contrário de outros modelos estatísticos, as arquiteturas de aprendizado profundo são capazes de encontrar dependências de curto e longo prazo nos dados e fornecer previsões ao encontrar essas relações ocultas.

Mesquita et al (2020), realizaram um estudo de dados reais de 10 séries temporais referentes a ativos da Bovespa, com o objetivo de analisar as características das séries e utilizar uma rede neural LSTM para predição de valores futuros. Foram analisados os resultados de 5 diferentes testes de razão da variância nas séries e sua relação com o desempenho da

classificação da rede neural. O trabalho propõe a aplicação dos testes estatísticos no conjunto de treinamento da rede com intuito de destacar previamente as séries que possuirão melhor desempenho de acordo com os resultados encontrados. Os resultados do trabalho demonstraram que 4 das 10 ações rejeitaram a hipótese de passeio aleatório por meio dos testes da razão da variância, e que as mesmas ações obtiveram os melhores resultados em termos de classificação.

Apesar de Mesquita et al (2020) aplicarem o teste da razão de variância para verificar se o retorno de uma ação segue ou não uma *Random Walk* e usarem modelos de redes neurais recorrentes LSTM univariados para formular previsões de um dia, faltou avaliar o poder preditivo de modelos e arquiteturas de redes neurais recorrentes LSTM univariados e multivariados por meio da técnica da validação cruzada para series temporais, no sentido de controlar o *overfitting*².

As Redes Neurais Recorrentes (RNN) demonstram capacidade de resolver problemas complexos relacionados a séries temporais de forma satisfatória, sendo assim, o objetivo dessa pesquisa visa verificar qual modelo ou arquitetura de Redes Neurais, RNN, LSTM (univariada ou multivariada), possui maior poder preditivo do preço de fechamento das ações selecionadas do IBOVESPA fora da amostra de treinamento.

O objetivo geral foi verificar se os modelos de redes neurais recorrentes com memória curta e longa (RNN LSTM) possuem maior poder preditivo de preços de ações do que os modelos de redes neurais recorrentes de memória curta (RNN).

A hipótese deste trabalho é que se a HME na forma fraca for rejeitada, as imperfeições do mercado de ações como a presença de autocorrelação negativa ou positiva de primeira ordem ou de ordens superiores, nos retornos de ações, poderão ser modelados e explorados por redes neurais com memória curta e longa.

Por sua vez, o primeiro objetivo específico foi verificar se os retornos de ações da ABEV3, ITUB4, PETR3 e PETR4, seguem um processo estocástico Random Walk ou a validade da HME na forma fraca.

O segundo objetivo específico foi avaliar o poder preditivo através da aplicação da metodologia de validação cruzada dos modelos de redes neurais recorrentes de memória curta (RNN) e de modelos de Redes Neurais Recorrentes de memória curta e longa (RNN LSTM), univariados e multivariados.

² Overfitting ocorre quando um modelo estatístico se ajusta muito bem ao conjunto de dados de treino, mas se mostra ineficaz para prever novos resultados.

2 REVISÃO DE LITERATURA

Essa seção remete uma breve introdução sobre a Hipótese dos Mercados Eficientes e Economia Comportamental, além de trazer uma explicação e o arcabouço teórico sobre Redes Neurais Artificiais, Aprendizado Profundo, Modelo Long Short-Term Memory LSTM e uma abordagem do mercado de ações além de trabalhos e pesquisas que abordaram esses temas.

2.1 A HIPÓTESE DOS MERCADOS EFICIENTES

Os primeiros relatos do surgimento da Hipótese da Eficiência de Mercado (HEM), aparecem com o desenvolvimento de trabalhos relacionados ao comportamento dos preços de ativos, sendo um dos primeiros trabalhos a tese de doutorado em Matemática de Louis Bachelier, *Theorie de la Speculation* (1900), sob a direção do eminente matemático Henry Poincaré. Em sua tese, Bachelier investiga a teoria da especulação e descreve o que é conhecida atualmente por Random Walk Hypothesis³, ou hipótese do passeio aleatório, admitindo que as alterações nos preços dos ativos são independentes e distribuídas de forma probabilística. Porém o trabalho foi esquecido e somente por volta dos anos de 1950 Paul Samuelson descobre esse trabalho e realiza a tradução para o inglês.

No entanto, somente por meio dos estudos de Roberts (1967) e Fama (1970), a eficiência de mercado passou a ocupar um lugar de destaque na teoria de finanças, com grande argumentação de teóricos a favor e contra a sua existência. Eugene Fama (1970), com o trabalho *“Efficient Capital Markets: a review of theory and empirical work”* passa a apresentar o tema com mais robustez e com a sistematização com a qual ele passa a ser conhecido e mais trabalhado.

Em seu artigo clássico, Fama (1970) define um mercado financeiro eficiente na forma fraca, semiforte e forte. No primeiro caso, informações históricas já estariam refletidas nos preços; no segundo, informações públicas em geral, e no terceiro, informações privadas. A Hipótese de Mercado Eficiente (EMH) na forma fraca postula que o processo gerador de dados de retornos em logs das ações é um processo estocástico Random Walk, ao supor que :(i) os retornos das ações são independentes e (ii) identicamente distribuídos.

Logo, a HME na sua forma fraca possui várias limitações teóricas e empíricas:

³ Teoria financeira que afirma que os preços do mercado de ações evoluem de acordo com um passeio aleatório (portanto, as mudanças de preço são aleatórias) e, portanto, não podem ser previstos.

Pelo Teorema do Limite Central, a distribuição de retornos e de preços convergem, para amostras muito grandes, para a distribuição de probabilidade normal.

Apesar de explicar um fato estilizado do mercado de ações, a ausência de autocorrelação de retornos em logs, considera os demais fatos estilizados como anomalias: (a) o excesso de volatilidade de retornos, (b) a distribuição caldas pesadas e leptocúrticas de retornos, (c) a aglomeração de volatilidade caracterizada pela queda lenta e gradual da autocorrelação de retornos absolutos, (d) os retornos anormais ajustados pelo risco gerados pelos indicadores e setups da análise técnica.

Fama (1970), define “mercado eficiente” sendo aquele em que os preços dos ativos financeiros fornecem um sinal conciso para a aplicação de recursos, e os seus retornos precisam proporcionar independência sequencial, ou seja, as alterações dos preços precisam ser linearmente independentes, com correlação sequencial igual a zero. Sendo assim, é impossível que os investidores utilizem de informações e/ou estratégias para conseguirem obter vantagens anormais em relação aos demais investidores, desde que o preço dos ativos negociados sempre reflete inteiramente as informações disponíveis sobre eles.

Segundo Fama (1970), para os mercados serem eficientes eles precisam respeitar algumas condições suficientes: a) Não existem custos de transação na negociação dos ativos; b) Toda a informação está disponível para todos os participantes sem custos; c) Concordância comum nas expectativas dos investidores quanto aos efeitos das informações sobre os preços atuais dos ativos, assim como sobre suas distribuições futuras.

Destarte Fama (1970), apresenta três níveis de forma de eficiência de mercado, sendo os mais difundidos e utilizados. A HME na forma fraca, propõe que os retornos e preços de ativos seguem um processo estocástico Random Walk, sugerindo que nenhum investidor obtém retornos em excesso baseado em dados do passado, matematicamente como:

$$P_t = P_{t-1} + \text{Retorno Esperado} + \varepsilon_t \quad (1)$$

em que o preço do período de t é função do preço do período $t-1$, acrescido do retorno esperado do título e de um componente aleatório que representa os choques exógenos imprevisíveis que são rapidamente e completamente incorporados ao preço do ativo em t . O retorno esperado do título é função de seu risco e pode ser obtido pelos modelos tradicionalmente utilizados pelo mercado, como o CAPM. O componente aleatório, por sua vez, é resultante de novas informações a respeito do ativo sob precificação, podendo assumir valores positivos e negativos, sendo, porém, seu valor esperado igual a zero (RABELO JUNIOR; IKEDA, 2004)

Por sua vez, a HME na sua forma forte, além de supor a validade da sua forma fraca, propõe que os preços atuais ou correntes incorporam rapidamente e completamente também

qualquer informação pública disponível como informações de resultados de desempenho da firma, do setor e do ambiente macroeconômico.

A forma forte dos mercados eficientes é a que sofre mais crítica, a qual propõe que os indivíduos de posse de informações privilegiadas não conseguem obter qualquer ganho de retorno acima dos demais investidores.

A HME na sua forma forte, além de supor a validade da sua forma semiforte, propõe que a assimetria de informação possui efeito neutro sobre os retornos esperados: garante que os indivíduos que possuem informações privilegiadas não conseguem obter excesso de retorno ajustado ao risco. A HME na forma forte vem sendo rejeitada por diversos testes econométricos como em Lo (2004), o próprio Eugene Fama (1970) revela que não realizou um modelo HEM para a realidade nesta forma. Nessa mesma linha de acordo com Ross et al. (1995), nem os maiores defensores da HME iriam se surpreender com resultados que os mercados não são eficientes na sua forma forte, tal que, dada a obtenção de informação privilegiada e exclusiva, é plausível que se possa obter ganhos de alguma forma.

2.2 FINANÇAS COMPORTAMENTAIS

O comportamento do investidor vem sendo estudado por diversos anos, mostrando que os investidores sempre estão tentando buscar obter e realizar ganhos acima dos demais investidores, apresentando assim diversos estudos de que o mercado não é tão eficiente como proposto pela HME (MALKIEL, 2003).

No meio acadêmico a área de economia que visa avaliar e acompanhar os impactos do comportamento do investidor com influência de fatores psicológicos na tomada de decisão além de todos os demais fatores que afetam na definição dos preços dos ativos é chamado de Finanças Comportamentais ou Economia Comportamental.

Daniel Kahneman e Amos Tversky (1979), são um dos principais autores que buscam estudar o processo de julgamento e decisão de agentes com racionalidade limitada em situações de incerteza e/ou complexidade.

2.2.1 Heurísticas e Vieses Comportamentais

De acordo com Kahneman (2012), como os indivíduos detêm recursos e tempo limitado para a tomada de decisão em condições de incerteza e complexidade, podem recorrer a heurísticas de forma automática. Quando é feita uma tomada de decisão o cérebro procura

uma escolha assertiva, com base em estatísticas e análise centralizada de pequenas amostras, com isso, as tomadas de decisões ocorrem de forma escassa de informações.

Segundo Sternberg (2009), as informações disponíveis no mundo são filtradas pelo cérebro assim como os demais processos plausíveis no momento da decisão, sendo produzida junto de outras informações disponíveis em um curto período.

Heurísticas podem ser representadas como atalhos que nosso cérebro se utiliza para a tomada de decisões, mas com essa simplificação há uma tendência a erros de julgamento e escolhas que fogem da racionalidade ou da estatística, esses erros ocorrem de forma sistemática e previsível e em determinadas ocasiões, são denominados de vieses comportamentais.

Tversky e Kahneman (1974) apresentam três heurísticas para avaliação e realização da previsão valorativa utilizadas pelo nosso comportamento mental:

- a) Heurística da Representatividade: A avaliação acontece com base no estereótipo, semelhança com um caso típico ou comum, ou seja, o acontecimento de um fato é avaliado baseado na analogia de ocasiões anteriores que foram utilizados resultados ou estimativas de sucesso.
- b) Heurística da Disponibilidade: Nessa heurística é julgado o processo das possibilidades de acontecer um fato segundo a facilidade de conseguir lembrar das ocorrências dele, junto a limitação de manter concentração mental ao mesmo tempo. Essa heurística também é conhecida como “Lei dos Números Pequenos”, intenção de generalizar e extrair conclusões de poucas amostras, que não representam um todo. Tversky e Kahneman (1974), trazem como exemplo a queda de um avião divulgada, de acordo com eles, tal mensagem impacta temporariamente as emoções e avaliação probabilística das pessoas, que terão o medo de viajar de avião maior em relação a viajar de carro, mesmo os riscos de as viagens de carro serem maiores. Pode-se observar que muitos investidores tendem a usar essa heurística para fundamentar suas escolhas de investimentos.
- c) Heurística de Ancoragem e Ajustamento: Pode ser definida como uma propensão que dificulta o indivíduo de alterar a avaliação inicial à novas informações disponibilizadas, dado um valor primitivo, realizam-se suas estimativas e ajustamentos para chegar à decisão final.

Para Kahneman (2012), o ponto de partida da ancoragem pode ser apontado por um antecedente histórico, pelo qual o problema é proporcionado, ou, por uma informação aleatória.

A heurística de ancoragem também pode ocorrer, quando os indivíduos se baseiam em previsões de resultados que contemplam cálculos incompleto. O julgamento pode ser

tendencioso em direção a uma âncora irrelevante, por exemplo, um número arbitrário fornecido pelo pesquisador. Tversky e Kahneman (2012, p. 158) exibem um exemplo, baseado em uma experiência com corretores de imóveis que alcançaram taxa de ancoragem próxima a de estudantes universitários de administração, referente à oportunidade de aferir o valor de uma casa que estava disponível no mercado, onde os autores concluíram que tanto os corretores e estudantes foram influenciados pela âncora, porém, apenas os estudantes admitiam a influência enquanto os corretores a negavam.

Assim, as heurísticas e os vieses comportamentais, como a lei do pequenos números, o viés de auto atribuição, a aversão a perda no domínio de ganhos, propensão ao risco no domínio de perdas, o efeito disposição, efeito manada, viés de confirmação, e o excesso de confiança, podem induzir os agentes com racionalidade limitada a cometer erros sistemáticos no processo de julgamento e na tomada de decisões sobre a compra e venda de ativos: fenômenos como *overreacion* (reação exagerada) e *Underreaction* (reação lenta e gradual) são propriedades emergentes de mercados de ações.

2.2.2 Os Limites de Arbitragem

De acordo com a Hipótese de Mercados Eficientes (HME), em um mercado de ações no qual os agentes são racionais e não há fricções, o preço da ação é inexoravelmente igual a seu valor fundamental: soma descontada do fluxo de caixa futuro a uma taxa de desconto que é consistente com uma especificação de preferências normativamente aceitável. Em um mercado eficiente devido a operações de arbitragem sem risco, realizado por agentes racionais, não há possibilidade da existência de estratégias de investimento, seja de Valuation ou da análise técnica, que permitam um retorno anormal ajustado pelo risco. O argumento é que se o preço de uma ação sofre uma redução e torna-se menor do que o seu preço fundamental, cria-se uma oportunidade atrativa que os agentes racionais deverão inexoravelmente explorar, comprando esta ação e para fazer hedge, vendendo a descoberto outra ação substituta com fluxo de caixa similar: a maior pressão compradora deverá causar o retorno do preço da ação que sofreu queda para o seu valor fundamental (BARBERIS; THALER, 2005).

Por sua vez, a Finança Comportamental argumenta que algumas características de preços de ativos de risco como ações são mais plausivelmente interpretados como desvios do valor fundamental, e que estes desvios são gerados pela presença e interações de agentes irracionais ou com racionalidade limitada com agentes racionais. A teoria da arbitragem

limitada mostra que no mercado de ações se os agentes irracionais causam desvios do valor fundamental, agentes racionais frequentemente serão incapazes de fazer qualquer coisa.

O argumento é que quando há um desvio do valor fundamental, uma má precificação, não é criada uma oportunidade de investimento atrativo. Estratégias projetadas por agentes racionais para explorar e corrigir a má precificação podem ser tanto arriscadas quanto custosas, tornando-as não viáveis e arriscadas devido a presença do risco fundamental quando o próprio preço fundamental da ação pode sofrer redução e não há uma ação substituta perfeita para fazer hedge, além da existência do risco do Trader irracional ou aleatório quando a má precificação explorada pelo arbitrador piora no curto prazo causado pelo excesso de pessimismo de traders irracionais ou aleatórios. Por outro lado, os custos de implementação de explorar a má precificação de ações são os custos de transações como as taxas de corretagem, emolumentos, taxas de custódia, impacto no preço ação, a taxa de juros pagos para o doador das ações, e custo de liquidez associado a ausência de ações para realizar operações de venda a descoberto.

Em suma, para as Finanças Comportamentais, a existência de elevados custos e riscos de operações de arbitragem em mercados de ações reais, conduz o afastamento do preço da ação do seu valor fundamental e a persistir ao longo do tempo: a má precificação não é corrigida conforme postula a HME.

2.3 INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA

Atualmente a Inteligência Artificial é assunto comumente abordado em vários estudos, o termo “IA” inclui inúmeras tecnologias, sendo um termo muito amplo que descreve todos os sistemas que buscam imitar e ou simular a inteligência humana, sendo sistemas que podem aprender, raciocinar, julgar e entender a linguagem natural, em suma, assim como as habilidades cognitivas dos seres humanos.

De acordo com Sage (1990), o objetivo da Inteligência Artificial IA é o desenvolvimento de paradigmas ou algoritmos que requeiram máquinas para realizar tarefas cognitivas, para quais os humanos são atualmente melhores. Além disso, o autor menciona que o sistema de IA deve ser capaz de realizar três coisas:

- a) Armazenar conhecimento;
- b) Aplicar o conhecimento armazenado para resolver problemas;
- c) Adquirir novos conhecimentos através da experiência;

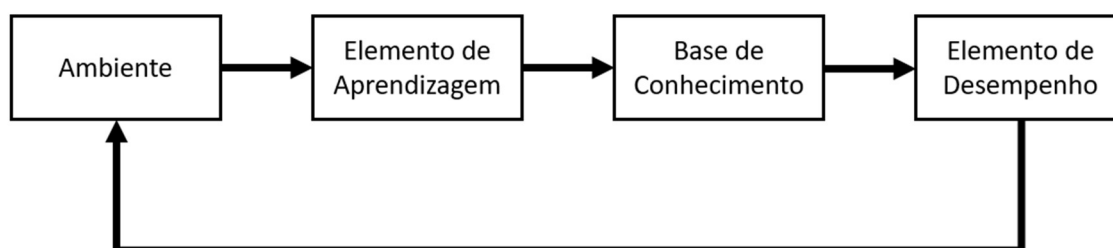
O autor elenca também que um sistema IA possui três componentes fundamentais:

- I) **Representação:** Provavelmente a característica mais distintiva da IA seja o uso difundido de uma linguagem de estruturas simbólicas para representar tanto o conhecimento genérico sobre um domínio do problema de interesse como o conhecimento específico sobre a solução do problema. “*Conhecimento*”, como é utilizado pelos pesquisadores de IA, é apenas mais um termo para dados, ele pode ser do tipo *declarativo* ou *procedimental*. Em uma representação declarativa, o conhecimento é representado como uma coleção estática de fatos com um pequeno conjunto de procedimentos gerais utilizados para manipular os fatos. Em uma representação procedimental, o significado do conhecimento está incorporado em um código executável que representa o significado do conhecimento.
- II) **Raciocínio:** Na sua forma mais básica, raciocínio é a habilidade de resolver problemas. De acordo com Fischler e Firschein (1987), para um sistema ser qualificado como um sistema de raciocínio, ele deve satisfazer certas condições:
- 1) O sistema deve ser capaz de expressar e resolver uma vasta gama de problemas e tipos de problemas;
 - 2) O Sistema deve ser capaz de tornar conhecidas para ele tanto a informação explícita como a informação implícita;
 - 3) O sistema deve ter um mecanismo de controle que determine quais operações devem ser aplicadas para um problema particular, quando uma solução para esse problema for obtida, ou quando deve ser encerrado o tratamento desse problema.

A resolução de problemas pode ser vista como um problema de busca. Uma maneira comum de lidar com a “busca” é utilizar regras, dados e controle (NILSON, 1980).

- III) **Aprendizagem:** No modelo simples de aprendizagem de máquina representado na Figura 1, o ambiente fornece algumas informações para um elemento de aprendizagem.

FIGURA 1- MODELO SIMPLES DE APRENDIZAGEM DE MÁQUINA



O elemento de aprendizagem utiliza esta informação para aperfeiçoar a base do conhecimento, e finalmente o elemento de desempenho utiliza a base de conhecimento para executar a sua tarefa. Normalmente, a informação que o ambiente fornece para a máquina é imperfeita, resultando que o elemento de desempenho não sabe previamente como preencher os detalhes ausentes ou ignorar os detalhes que não são importantes, portanto, a máquina opera inicialmente por suposição e depois recebe realimentação dos elementos de desempenho. O mecanismo de realimentação permite que a máquina avalie suas hipóteses e as revise, se necessário (HAYKIN, 2008).

Na IA clássica, é dada ênfase na construção de representações simbólicas, que são presumivelmente assim chamadas porque representam algo. Do ponto de vista da cognição, a IA assume a existência de representações mentais e ela modela a cognição como o processamento sequencial de representações simbólicas (NEWELL; SIMON, 1972).

Por outro lado, nas redes neurais a ênfase está no desenvolvimento de modelos de processamento paralelamente distribuídos. Estes modelos assumem que o processamento de informação acontece através da interação de muitos neurônios onde cada neurônio envia sinais para outros neurônios da rede (RUMELHART; MCCLELLAND, 1986).

O termo Machine Learning ML ou Aprendizado de Máquina AM, é uma parte da Inteligência Artificial, que trata da aplicação de modelos matemáticos que buscam aprender com os dados e se tornam mais inteligentes à medida que mais dados são processados e inseridos, sendo uma imitação do sistema de aprendizagem humano, por exemplo, uma pessoa experimentou mais e se expôs mais, ela conseqüentemente tende a aprender com isso e adaptar seu comportamento (HAYKIN, 2008; ALPAYDIN, 2014).

No Aprendizado de Máquina, as máquinas são capazes de aprender, isso significa que a resposta para um determinado problema será o resultado desta aprendizagem e não um código pré-estabelecido que um programador detalhou e planejou.

De acordo com Simon Haykin (2008), Aprendizagem de Máquina AM envolve dois tipos bastante diferentes de processamento de informação: o *indutivo* e o *dedutivo*. No processamento de informação *indutivo*, padrões gerais e regras são determinados a partir dos dados brutos e da experiência. Por outro lado, no processamento de informação *dedutivo* são utilizadas regras gerais para determinar fatos específicos. A aprendizagem baseada em similaridade utiliza indução, enquanto a prova de um teorema é uma dedução baseada em axiomas conhecidos e em outros teoremas existentes. A aprendizagem baseada em explanação utiliza-se tanto de indução como dedução.

Em resumo a Inteligência Artificial IA é a teoria e o desenvolvimento de sistemas de computação capazes de realizar tarefas que normalmente requerem inteligência humana.

2.4 REDES NEURAIS ARTIFICIAIS – RNA

As Redes Neurais Artificiais RNA, foram desenvolvidas por volta dos anos 1940, pelo matemático Walter Pitts e o neurofisiologista Warren McCulloch, com o objetivo de propor um modelo computacional cujo funcionamento seguisse os fundamentos de neurônios biológicos (LUDWIG; COSTA, 2007). As RNAs simulam sinapses estruturadas com o objetivo de reproduzir o funcionamento do cérebro humano. As RNAs possuem importantes propriedades como o aprendizado através de treinamento, tolerância a falhas e a habilidade de generalizar sistemas complexos (HAYKIN, 2008).

Em 1948, foi publicado o livro *Cybernetics* de Wiener, apresentando alguns conceitos importantes sobre o controle, comunicação e processamento estatístico de sinais. A segunda edição do livro foi publicada em 1961, com a adição de um novo material sobre aprendizagem e auto-organização, somente depois de mais de 30 anos que Hopfield conseguiu consumir a ligação entre a mecânica estatística e os sistemas de aprendizagem (HAYKIN, 2008).

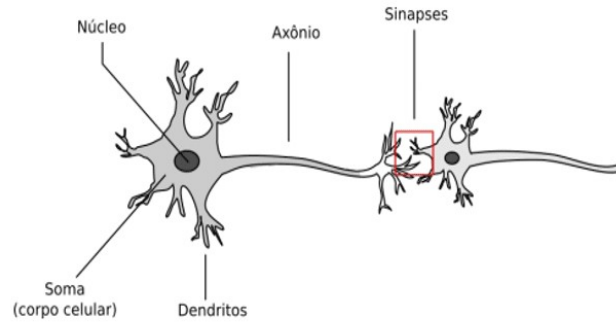
De forma simples, pode se observar uma Rede Neural como uma estrutura que conecta pequenas unidades, os neurônios, de forma organizada. Através desta organização, a combinação das operações unitárias simples realizadas por cada neurônio levará a soluções de problemas complexos (BO K WONG; YAKUP SELVI, 1998)

A estrutura das RNAs se baseiam no cérebro humano, que possui em média 86 bilhões de neurônios e cada neurônio realiza cerca de sete mil conexões com outros neurônios, ou seja, sete mil sinapses com outros neurônios, com uma complexibilidade enorme, onde a informação é enviada através de impulsos elétricos, quando um neurônio está enviando informações através de um impulso elétrico para outro neurônio, denomina-se como neurônio ativado, por outro lado, quando um neurônio está desativado, o mesmo não envia nenhuma informação para outros neurônios. E, um mesmo neurônio pode estar recebendo e enviando impulsos simultaneamente (LENT et al, 2012).

A partir dessas conexões de troca de informações se evidencia e estabelece reconhecimento de padrões na rede biológica, mesmo funcionamento do cérebro humano, baseado nessas conexões que se fez a criação de Redes Neurais Artificiais, uma RNA também contém neurônios, porém eles são artificiais, onde cada neurônio realiza conexões com outros neurônios, tendo a capacidade de realizar operações matemáticas simples, de soma e

multiplicação, com os dados que recebem e enviando o resultado para outros neurônios, portanto, do mesmo modo que em uma rede neural biológica, nas redes artificiais a grande quantidade de neurônios e conexões possibilitarão que operações individuais simples, quando somadas, levem a resoluções de problemas de grande complexidade (HAYKIN, 2008).

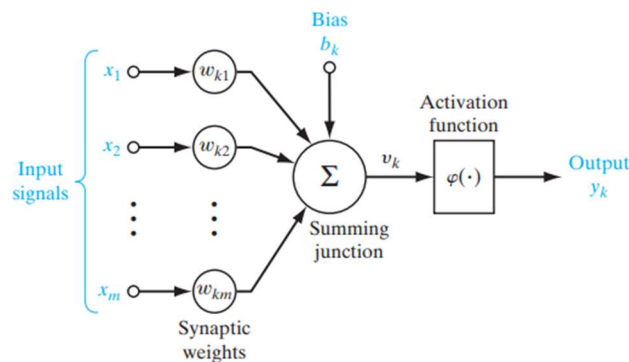
FIGURA 2 - MODELO DE NEURÔNIO BIOLÓGICO



Fonte: Haykin, 2008

O elemento principal de uma RNA é o neurônio, sendo uma estrutura lógica-matemática que simula o comportamento e funções do neurônio biológico, sendo uma unidade de processamento de informação fundamental para a operação de uma RNA, conforme apresentado na Figura 3, podemos visualizar um modelo de neurônio base para o projeto de Redes Neurais Artificiais.

FIGURA 3 - MODELO NÃO LINEAR DE UM NEURÔNIO, ROTULADO K



Fonte: Haykin, 2008.

Matematicamente podemos descrever um neurônio k da seguinte forma:

$$u_k = \sum_{j=1}^m \omega_{k,j} x_j \quad (2)$$

e

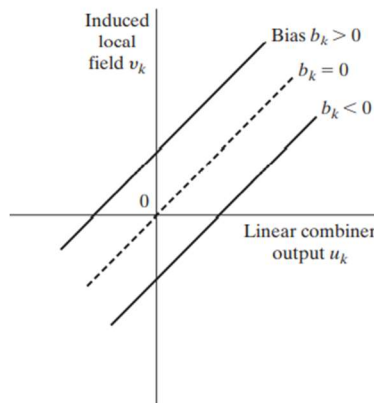
$$y_k = \varphi(u_k + b_k) \quad (2.1)$$

Onde $x_1, x_2, \dots, x_m$ representam os sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{km}$, são os pesos sinápticos do neurônio k ; u_k (não está na figura 3) é a saída do combinador linear, devido aos sinais de entrada; b_k é o *bias*; $\varphi(\cdot)$ é a função de ativação; e y_k é o sinal de saída do neurônio. O uso do bias b_k tem o efeito de aplicar uma transformação afim à saída u_k do combinador linear, conforme a equação 2.2.

$$v_k = u_k + b_k \quad (2.2)$$

Em particular dependendo se o bias b_k for positivo ou negativo, a relação entre o campo local induzido ou potencial de ativação v_k do neurônio k e a saída do combinador linear u_k é modificada conforme figura 4, daqui em diante, esses dois termos são usados de forma equivalente. Observe que, como resultado dessa transformação afim, o gráfico de v_k versus u_k não passa mais pela origem.

FIGURA 4- TRANSFORMAÇÃO AFIM PRODUZIDA PELA PRESENÇA E UM VIÉS



Fonte: Haykin, 2008.

O bias b_k é um parâmetro externo do neurônio k . Podemos explicar sua presença como na equação (2.1). De forma equivalente, podemos formular a combinação das equações (2) a (2.2) da seguinte forma:

$$v_k = \sum_{j=0}^m \omega_{k,j} x_j \quad (2.3)$$

$$y_k = \varphi(v_k) \quad (2.4)$$

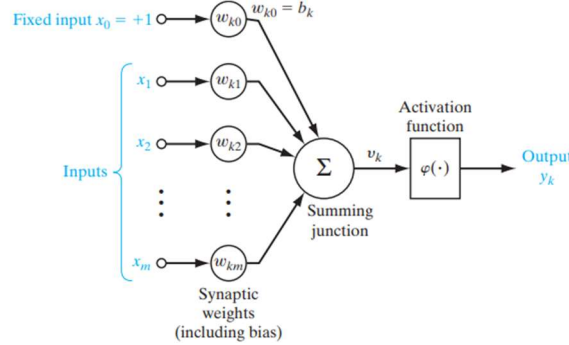
Assim, na equação 1.4, adicionamos uma nova sinapse. A sua entrada é:

$$x_0 = +1 \quad (2.5)$$

E o seu peso é

$$w_{k0} = b_k \quad (2.6)$$

Podemos, portanto, reformular o modelo do neurônio k como mostrado na Figura 5.

FIGURA 5- OUTRO MODELO NÃO LINEAR DE UM NEURÔNIO; LEVA EM CONTA O VIÉS b_k 

Fonte: Haykin, 2008.

Nesta figura 5, o efeito do viés é contabilizado realizando dois acontecimentos:

- (1) adicionando um novo sinal de entrada fixado em +1 e;
- (2) adicionando um novo peso sináptico igual ao viés b_k .

Embora os modelos das Figuras 3 e 5 sejam diferentes na aparência, são matematicamente equivalentes.

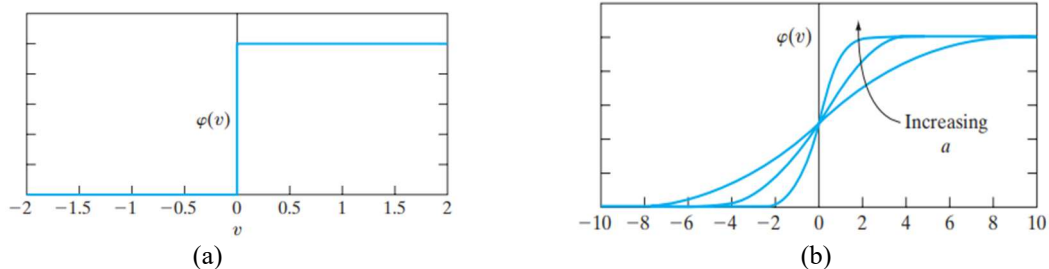
2.4.1 Funções de Ativação

No momento de escolha da arquitetura da Rede Neural Artificial, se faz necessário escolher o tipo de funções de ativação que se deseja em cada local, as funções de ativação são aplicadas em cada nó da rede, antes de passar para a próxima camada.

A função de ativação, denotada por v_k , define a saída de um neurônio em termos do campo local induzido v .

Função Limiar: para esse tipo de função de ativação, descrito na figura 6 a seguir temos:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (2.7)$$

FIGURA 6 – (A) FUNÇÃO LIMITE, (B) FUNÇÃO SIGMOIDE PARA PARÂMETRO DE INCLINAÇÃO VARIÁVEL a 

Fonte: Haykin, 2008.

Na engenharia, essa forma de função limite é comumente chamada de função *Heaviside*. Correspondentemente, a saída do neurônio k empregando tal função de limiar é expressa como:

$$y_k = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases} \quad (2.8)$$

Onde v_k é o campo local induzido do neurônio;

$$v_k = \sum_{j=1}^m \omega_{k,j} x_j + b_k \quad (2.9)$$

Na literatura, esse neurônio é chamado de *modelo de McCulloch-Pitts*, em reconhecimento ao trabalho pioneiro realizado por McCulloch e Pitts (1943). Nesse modelo, a saída de um neurônio assume o valor 1 se o campo local induzido desse neurônio for não negativo e 0 caso contrário. Esta declaração descreve a propriedade tudo ou nada do modelo McCulloch-Pitts (HAYKIN, 2008).

A razão de se utilizar as funções de ativação é que elas permitem que o modelo ajuste problemas mais complexos com menos nós, pois adicionam não linearidade ao cálculo (GRAVES, 2012; NG, 2017).

A informação de entrada é ponderada e processada por uma função de soma e esse novo sinal passa pela função de ativação, que corresponde à diferença de potenciais no núcleo da célula, através disso, os neurônios artificiais são baseados em uma função matemática que pode ser demonstrada na equação 3 (MEENAL; SELVAKUMAR, 2018; HAYKIN, 2008).

$$y_i = \varphi_i(\sum_{j=1}^m x_j \omega_{i,j} + b_k) \quad (3)$$

Onde x_j é a j -ésima entrada do neurônio i , $\omega_{i,j}$ é o peso associado, φ_i é a função de ativação, e b_k é o *bias*, uma constante, componente que gera a variância em relação a entrada.

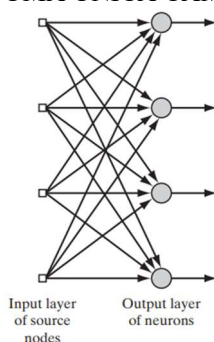
2.4.2 Arquitetura de Rede Neural

A maneira pela qual os neurônios de uma rede neural são estruturados está intimamente ligada ao algoritmo de aprendizado usado para treinar a rede. Podemos, portanto, falar de algoritmos de aprendizado (regras) usados no projeto de redes neurais como sendo estruturados (HAYKIN, 2008).

Segundo os autores Haykin (2008) e Meenal e Selvakumar (2018), uma rede neural em camadas, os neurônios são organizados na forma de camadas. Na forma mais simples de uma rede em camadas, RNA rasa, temos uma camada de entrada de nós de origem que se projeta diretamente em uma camada de saída de neurônios (nós computacionais), mas não vice-versa.

Em outras palavras, essa rede é estritamente do tipo *feedforward*, na Figura 7 podemos observar a estrutura de RNA que é representada por quatro nós de entrada como na de saída, denominada *rede de camada única*, não se conta a camada de entrada de nós de entrada, porque não é realizado nenhum tipo de performasse computacional.

FIGURA 7 - REDE FEEDFORWARD COM UMA ÚNICA CAMADA DE NEURÔNIOS

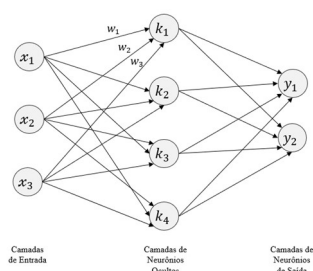


Fonte: Haykin, 2008.

A segunda classe de uma rede neural *feedforward* se distingue pela presença de uma ou mais camadas ocultas, cujos nós de computação são correspondentemente chamados de neurônios ocultos ou unidades ocultas; o termo “oculto” refere-se ao fato de que esta parte da rede neural não é vista diretamente da entrada ou saída da rede. A função dos neurônios ocultos é intervir entre a entrada externa e a saída da rede de alguma maneira útil (HAYKIN, 2008). Isso significa que se tem apenas uma camada oculta de neurônios entre os dados de entrada e saída da rede. Assim a habilidade dos neurônios ocultos de extraírem estatísticas de ordem elevada é muito valiosa quando o tamanho da camada de entrada é grande.

Por concisão, a rede na figura 8 é referida como uma rede 3-4-2, porque ela possui 3 neurônios de entrada x , 4 neurônios ocultos k e 2 neurônios de saída y , os dados de entradas x_1 , x_2 e x_3 serão passados a cada neurônio da camada oculta k_1 , k_2 , k_3 e k_4 sendo que cada dado de entrada terá um peso específico para cada neurônio que o receber w_1 , w_2 e w_3 .

FIGURA 8 - REDE FEEDFORWARD TOTALMENTE CONECTADA COM UMA CAMADA OCULTA E UMA CAMADA DE SAÍDA



Fonte: Adaptado pelo autor, Haykin, 2008

Desta forma, teremos que o valor de cada neurônio será igual ao valor de cada dado de entrada, multiplicado pelo seu respectivo peso, somado com as demais multiplicações, e, também, com um valor constante final, chamado de bias, conforme apresentado anteriormente na equação 3.

Uma vez que todos os neurônios de saída tenham seus valores definidos, aquele que tiver o maior valor será o neurônio mais ativado, indicando assim que a resposta para o problema buscado é equivalente ao resultado que este neurônio de saída representa (em um problema de classificação). A precisão desta rede neural será definida pelo ajuste dos pesos e bias de cada neurônio.

Diz-se que a rede neural está totalmente conectada no sentido de que todos os nós em cada camada da rede estão conectados a todos os outros nós da camada direta adjacente. Se, no entanto, alguns dos links de comunicação (conexões sinápticas) estiverem faltando na rede, dizemos que a rede está parcialmente conectada (HAYKIN, 2008).

O funcionamento básico das RNAs incide em aprender a relação existente entre os dados por meio de vários neurônios conectados em diferentes camadas (ADHIKARI; AGRAWAL, 2013). De acordo com Silva et al. (2010) potenciais aplicações dos métodos de Redes Neurais Artificiais podem ser descritas como: aproximação de funções; controle de processos; classificação e reconhecimento de padrões; agrupamento de dados; modelos de previsão de variáveis e processos de otimização. As principais arquiteturas de RNAs podem ser agrupadas em: redes *feedforward* de camadas simples; redes *feedforward* com múltiplas camadas; redes recorrentes e redes reticuladas (SILVA et al., 2010).

Silva (2010), demonstra que para se construir uma RNA, em geral, se faz a divisão dos dados disponíveis de entrada em duas amostras: a primeira amostra geralmente de 60% a 90% da amostra original serve para realização do treinamento e a segunda parte da amostra entre 40% e 10% da amostra original, compreende para a realização de testes e análises do modelo que foi treinado.

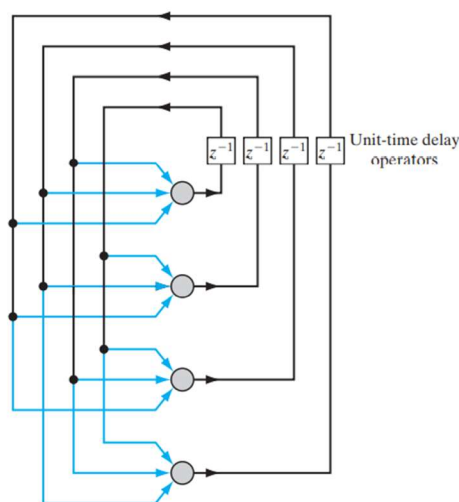
Segundo Chen, Daouk e Leung (2001), as RNAs proporcionam resultados promissores para a previsão financeira, além da abordagem de previsões econométricas.

Redes neurais podem ser treinadas com o algoritmo *backpropagation*. Ele permite ajustar os pesos de uma rede neural com base no erro (isto é, perda) obtida na iteração anterior. O ajuste adequado dos pesos garante erros mais baixos, tornando o modelo confiável, aumentando sua generalização (NG, 2017).

2.5 RECURRENT NEURAL NETWORKS – RNN

Uma Rede Neural Recorrente RNN se distingue de uma rede neural feedforward por ter pelo menos um laço de realimentação. Por exemplo, uma rede recorrente pode consistir em uma única camada de neurônios com cada neurônio alimentando seu sinal de saída de volta para as entradas de todos os outros neurônios, conforme ilustrado na arquitetura da Figura 9.

FIGURA 9 - REDE RECORRENTE SEM LOOPS DE REALIMENTAÇÃO E SEM NEURÔNIOS OCULTOS

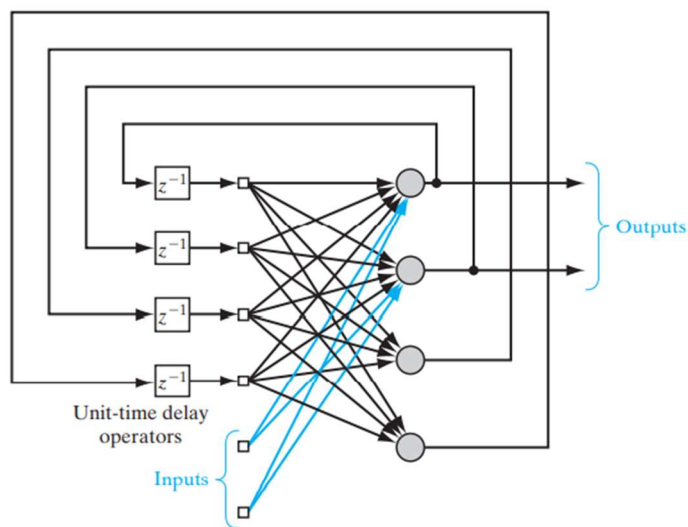


Fonte: Haykin, 2008.

Na estrutura representada figura acima, não há laços de auto realimentação na rede; auto realimentação refere-se a uma situação em que a saída de um neurônio é realimentada em sua própria entrada. A rede recorrente ilustrada na Figura 9 também não possui neurônios ocultos.

Outra classe de redes recorrentes com neurônios ocultos, pode ser observada na Figura 10 onde as conexões de realimentação apresentadas originam-se dos neurônios ocultos, bem como dos neurônios de saída. A presença de laços de realimentação, seja na estrutura recorrente da Figura 9 ou na Figura 10, tem um impacto profundo na capacidade de aprendizado da rede e em seu desempenho. Além disso, os laços de realimentação envolvem o uso de ramos particulares compostos por *elementos de atraso unitário de tempo* (denotados por z^{-1}), resultando em um comportamento dinâmico não linear, assumindo que a rede neural contenha unidades não lineares.

FIGURA 10 - REDE RECORRENTE COM NEURÔNIOS OCULTOS



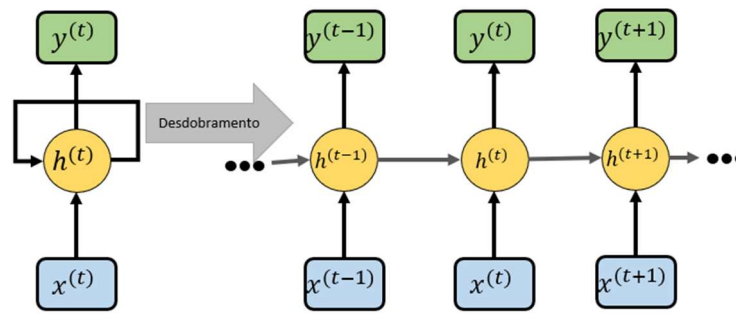
Fonte: Haykin, 2008.

RNN são uma classe de RNA onde a saída da etapa anterior é alimentada como entrada para a etapa atual junto com a entrada normal. RNAs feedforward, todas as entradas e saídas são independentes umas das outras, mas em casos como quando é necessário prever a série temporal, os valores anteriores são necessários e, portanto, é necessário lembrar os valores anteriores (LIPTON ET AL., 2015), (GRAVES, 2012). É descoberto que RNN sofre de problema de gradiente de desaparecimento.

À medida que propagamos o erro pela rede, ele tem que passar pelo loop temporal, as camadas ocultas conectadas a si mesmas no tempo por meio de pesos perdidos. Como esse peso é aplicado muitas e muitas vezes sobre si mesmo, isso faz com que o gradiente diminua rapidamente. Como resultado, os pesos das camadas na extrema esquerda são atualizados muito mais lentamente do que os pesos das camadas na extrema direita. Isso cria um efeito dominó porque os pesos das camadas da extrema esquerda definem as entradas para as camadas da extrema direita. Portanto, todo o treinamento da rede sofre, e isso é chamado de problema do gradiente de desaparecimento.

Redes neurais recorrentes resolvem esse problema. São redes com laços, permitindo que as informações persistam, as RNN são projetadas para sequências e são capazes de lembrar informações passadas. Na Figura 11, um esboço de uma rede neural recorrente de camada única, onde recebe a entrada $x^{(t)}$, entrada de x no tempo t , passando pela camada oculta de $h^{(t)}$ e produzindo um valor de saída em $y^{(t)}$, o resultado dessa camada no tempo t irá retroalimentar a próxima entrada, esse laço permite que as informações sejam passadas de uma etapa da rede para a próxima.

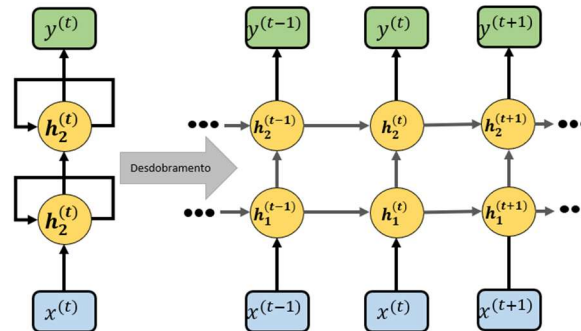
FIGURA 11 - ESTRUTURA DE UMA RNN COM UMA ÚNICA CAMADA OCULTA



Fonte: Adaptado pelo autor, Graves, 2012.

Essa é a principal diferença da RNN para uma RNA, onde uma RNA simples recebe a informação e repassa adiante, uma estrutura de RNN possui o laço, gerando a retroalimentação, o mesmo ocorre para RNN com mais de uma camada oculta, conforme Figura 12.

FIGURA 12 - ESTRUTURA DE UMA RNN COM UMA DUAS CAMADAS OCULTAS



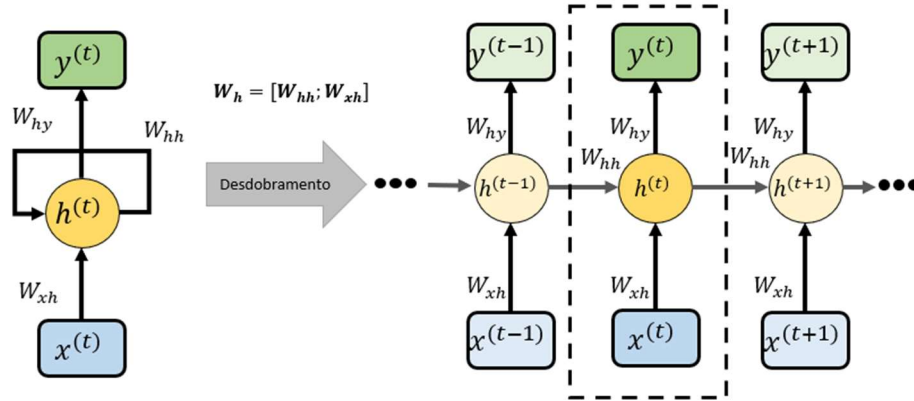
Fonte: Adaptado pelo autor, Graves, 2012.

Conforme podemos observar na figura 12, as camadas ocultas $h_1^{(t)}$ e $h_2^{(t)}$ recebem tanto as informações de entrada $x^{(t)}$ como a saída da camada oculta de $h_2^{(t-1)}$, $h_1^{(t-1)}$ e $h_1^{(t)}$, e assim por diante, portanto, uma RNN, pode possuir uma única camada ou várias camadas ocultas recorrentes com realimentação.

Pode-se observar que eles não são tão diferentes de uma rede neural normal. Uma rede neural recorrente pode ser considerada como várias cópias da mesma rede, cada uma passando uma mensagem a um sucessor. Ao desenrolarmos o laço, essa natureza em cadeia revela que as redes neurais recorrentes estão intimamente relacionadas a sequências e listas.

Na computação da ativação da RNN, cada aresta direcionada (as conexões entre as caixas) na representação da RNN está associada a uma matriz de peso. Esses pesos não dependem do tempo t , portanto, eles são compartilhados no eixo do tempo.

FIGURA 13 - COMPUTAÇÃO DE UMA RNN



Fonte: Adaptado pelo autor, Graves, 2012

Podemos observar a primeira matriz sendo W_{xh} sendo o peso entre a entrada de $x^{(t)}$ para a camada oculta de $h^{(t)}$, uma segunda matriz de W_{hh} que esta associada a aresta recorrente da realimentação da entrada da camada oculta $h^{(t)}$, e por último, a matriz W_{hy} , que é matriz de peso associada da camada oculta $h^{(t)}$ para a saída $y^{(t)}$, partindo do principio de uma matriz concatenada entre $W_h = [W_{hh}; W_{xh}]$.

A computação das ativações é semelhante aos outros tipos de rede neural feedforward. Para a camada oculta, a entrada $Z_h^{(t)}$ (pré ativação) é computada através de uma combinação linear, ou seja, se realiza o cálculo da soma das multiplicações das matrizes de peso com os vetores correspondentes e se adiciona a unidade de polarização.

$$Z_h^{(t)} = W_{xh}x^{(t)} + W_{hh}h^{(t-1)} + b_h \quad (4)$$

Aplicando a função de ativação nas unidades ocultas na etapa t :

$$h^{(t)} = \varphi(Z_h^{(t)}) \quad (4.1)$$

Assim, temos a saída de $h^{(t)}$, sendo uma função de ativação aplicada $Z_h^{(t)}$.

Como primeira formulação temos:

$$h^{(t)} = \varphi_h(W_{xh}x^{(t)} + W_{hh}h^{(t-1)} + b_h) \quad (4.2)$$

E como segunda formulação:

$$h^{(t)} = \varphi_h(W_h [x^{(t)}; h^{(t-1)}]^T + b_h) \quad (4.3)$$

Resultado da saída final:

$$y^{(t)} = \varphi_y(W_{hy}h^{(t)} + b_y) \quad (4.4)$$

2.6 PROCESSO DE APRENDIZAGEM

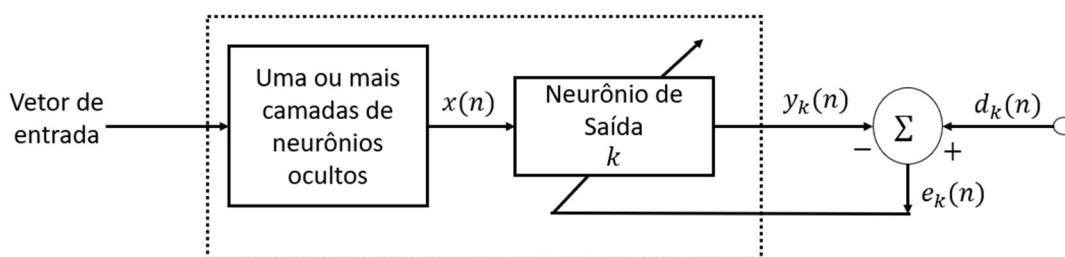
Assim como existem diferentes maneiras pelas quais nós mesmos aprendemos com nossos próprios ambientes, o mesmo acontece com as redes neurais. Para uma Rede Neural é primordial a habilidade de aprender a partir de seu ambiente e de melhorar seu desempenho através da aprendizagem. Uma RNA aprende acerca do seu ambiente através de um processo iterativo de ajustes aplicados a seus pesos sinápticos e níveis de bias.

Mendel e McClaren (1970), definem aprendizagem de redes neurais como sendo um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre.

2.6.1 Aprendizagem por Correção de Erro

De acordo com Haykin (2008), considerando o caso simples de um neurônio k que constitui o único nó computacional da camada de saída de uma rede neural alimentada adiante como apresentado na Figura 14. O Neurônio k é acionado por um vetor de sinal $x(n)$ produzido por uma ou mais camadas de neurônios ocultos, por sua vez, acionadas por um vetor de entrada aplicado aos nós de fonte da rede neural.

FIGURA 14 - DIAGRAMA EM BLOCOS DE UMA RNA



Fonte: Adaptado pelo auto, Haykin 2008.

O argumento n representa o instante de tempo discreto, ou mais precisamente, o passo de tempo de um processo iterativo envolvido no ajuste dos pesos sinápticos do neurônio k . O sinal de saída do neurônio k é representado por $y_k(n)$. Este sinal, representando a única saída da rede neural, é comparado com uma resposta desejada ou saída alvo, representada por $d_k(n)$. Consequentemente, é produzido um sinal de erro por $e_k(n)$ (HAYKIN, 2008).

Por definição matemática, temos:

$$e_k(n) = d_k(n) - y_k(n) \quad (5)$$

O Sinal de erro $e_k(n)$ aciona um mecanismo de controle, cujo propósito é aplicar, uma sequência de ajustes corretivos aos pesos sinápticos do neurônio k . Os ajustes corretivos são projetados para aproximar passo a passo o sinal de saída $y_k(n)$ da resposta desejada $d_k(n)$. Este objetivo é alcançado minimizando-se uma função de custo ou índice de desempenho $\mathcal{E}(n)$, definindo em termos do sinal de erro $e_k(n)$ como:

$$\mathcal{E}(n) = \frac{1}{2} e_k^2(n) \quad (5.1)$$

Com isso, $\mathcal{E}(n)$ é o valor instantâneo da energia do erro. Os ajustes passo a passo dos pesos sinápticos do neurônio k continuam até o sistema atingir um estado estável. Neste ponto, o processo é encerrado.

2.6.2 Aprendizagem Baseada em Memória

Na aprendizagem baseada em memória, todas as (ou a maioria das) experiências passadas são armazenadas explicitamente em uma grande memória de exemplos de entrada-saída classificados corretamente: $\{(x_i, d_i)\}_{i=1}^N$, onde x_i representa um vetor de entrada e d_i representa a resposta desejada correspondente. Sem perda de generalidade, restringimos a resposta desejada a ser um escalar. Em um problema de classificação de padrões binário, por exemplo, há duas classes/hipóteses a serem consideradas, representadas por $\mathcal{E}_1$ e $\mathcal{E}_2$. Neste exemplo a resposta desejada d_i assume o valor de 0 (ou -1) para a classe de $\mathcal{E}_1$ e o valor 1 para a classe de $\mathcal{E}_2$. Quando desejamos classificar um vetor de teste X_{teste} (não visto antes), o algoritmo responde buscando e analisando os dados de treinamento de uma vizinhança local de X_{teste} .

Todos os algoritmos de aprendizagem baseada em memória envolvem dois ingredientes essenciais:

- a) O critério utilizado para definir a vizinhança local do vetor de X_{teste}
- b) A regra de aprendizagem aplicada aos exemplos de treinamento na vizinhança de X_{teste} .

Os algoritmos diferem entre si na forma como estes dois ingredientes são definidos. Em um tipo simples, mas efetivo de aprendizagem baseada em memória conhecido como regra do vizinho mais próximo, a vizinhança local é definida como o exemplo de treinamento que se encontra na vizinhança imediata do vetor de teste X_{teste} . Em particular dizemos que o vetor

$$X'_N \in \{X_1, X_2, \dots, X_N\} \quad (5.2)$$

É o vizinho mais próximo de X_{teste} se:

$$\min d(X_i, X_{teste}) = d(X'_N, X_{teste}) \quad (5.3)$$

Onde $d(X_i, X_{teste})$ é a distância euclidiana entre os vetores X_i e X_{teste} . A classe associada com a distância mínima, ou seja, o vetor X'_N é apresentada como a classificação de X_{teste} . Esta regra é independente da distribuição fundamental responsável pela geração dos exemplos de treinamento.

De acordo com Cover e Hart (1967), formalmente a regra do vizinho mais próximo sendo uma ferramenta de classificação de padrões. A análise apresentada por eles é baseada em duas suposições:

- i) Os exemplos classificados (X_i, d_i) são independentemente e identicamente distribuídos (iid), de acordo com a distribuição de probabilidade conjunta do exemplo (X, d) .
- ii) O tamanho da amostra N é infinitamente grande.

Levando em consideração estas duas suposições, mostra-se que a probabilidade de erro de classificação pela regra do vizinho mais próximo é limitada acima pelo dobro da probabilidade de erro bayesiana, isto é, a mínima probabilidade de erro entre todas as regras de decisão.

2.7 APRENDIZADO PROFUNDO – DEEP LEARNING

Um dos principais ramos do Machine Learning é o Aprendizado Profundo, que como o próprio nome sugere, realiza uma aprendizagem profunda, com várias camadas de processamento, através das Redes Neurais Artificiais, que buscam um comportamento similar ao do cérebro humano. Outro exemplo de RNA, é uma RNA alimentada com múltiplas camadas, que se distingue pela presença de uma ou mais camadas ocultas, cujos nós computacionais são denominados de neurônios ocultos ou unidades ocultas.

O aprendizado profundo permite que modelos computacionais compostos de várias camadas de processamento aprendam representações de dados com vários níveis de abstração. Esses métodos melhoraram drasticamente o estado da arte em reconhecimento de fala, reconhecimento visual de objetos, detecção de objetos e muitos outros domínios, como descoberta de drogas e genômica. O aprendizado profundo descobre uma estrutura complexa em grandes conjuntos de dados usando o algoritmo de retro propagação para indicar como uma

máquina deve alterar seus parâmetros internos que são usados para calcular a representação em cada camada a partir da representação na camada anterior.

O termo Deep Learning começou a difundir a partir do ano de 2007, com a elaboração dos trabalhos de Ranzato, Poultney, Chopra, LeCun (2007), Bengio, Lamblin, Popovici, Larochelle (2007) e Hinton, Osindero, Teh (2006), através da aplicação de RNAs (GOODFELLOW et al., 2016).

Segundo Goodfellow (2016), deve-se desenvolver um desenho hierárquico do mundo real para problemas muito complexo e/ou intuitivos, onde redes abstratas e simples são definidas e interligadas entre si através de relações básicas, a partir dessa estruturação é possível a realização do aprendizado de forma automática de acontecimentos difíceis a partir de uma série de conceitos simples. O termo Deep Learning origina-se, portanto, deste empilhamento hierarquizado de muitos níveis de conceitos simples que leva ao aprendizado.

Com o avanço da tecnologia nos últimos anos e com o desenvolvimento de softwares e maior capacidade de processamento o aprendizado de máquina se tornou capaz de conseguir aprender qualquer coisa através de registros históricos baseado em várias informações auxiliando no processo de tomada de decisão, aplicado também no mercado acionário.

2.8 LONG SHORT-TERM MEMORY (LSTM)

Em um contexto neurobiológico, memória se refere as alterações neurais relativamente duradouras induzidas pela interação de um organismo com o seu ambiente (Teyler, 1986). Sem esta alteração não pode haver memória. Além disso, para que a memória seja útil, ela deve ser acessível ao sistema nervoso para poder influenciar o comportamento futuro.

De acordo com Arbib (1989), a memória se divide em memória de “curto prazo” e de “longo prazo”, dependendo da retenção. *Memória de curto prazo* se refere a uma compilação de conhecimento que representa o estado corrente do ambiente. Quaisquer discrepâncias entre o conhecimento armazenado na memória de curto prazo e um novo estado são usadas para atualizar a memória de curto prazo. *Memória de longo prazo*, se refere ao conhecimento armazenado por um longo período ou permanentemente.

LSTM é uma rede RNN proposta pela primeira vez por Sepp Hoch Reiter e Jürgen Schmidhuber em 1997 para resolver o problema de gradiente de desaparecimento em RNNs. Comparado com o RNN comum, o LSTM adiciona porta de entrada e porta de esquecimento para resolver o problema de desaparecimento e explosão de gradiente, para que informações de

longo prazo possam ser capturadas e possa ter melhor desempenho em texto de sequência longa. Assim LSTM usa as seguintes portas para resolver o problema:

- Input Gate (Porta de Entrada): escolhe qual valor da entrada atualizará o estado da memória.
- Output Gate (Porta de Saída): escolhe qual será a saída da célula com base na entrada e na memória da célula.
- Forget Gate (Porta de Esquecimento): se definido como true, a célula esquece as informações provenientes das camadas anteriores.

Essa porta de esquecimento não fazia parte inicialmente da rede LSTM, mas foi proposta por Gers (2000) para permitir que a rede reinicie seu estado, dessa forma a célula lembra valores em intervalos de tempo arbitrários e as três portas regulam o fluxo de informações associadas à célula.

As Redes Neurais denominadas Long Short-Term Memory (LSTM) fazem parte da arquitetura recorrente. Essas redes possuem o objetivo de aprender padrões complexos em estruturas com dependência temporal e vários estágios de processamento. Após várias interações nos treinamentos das redes em suas camadas ocultas, pode se perder do começo de uma série até suas fases intermediárias e finais (GREFF et al., 2016).

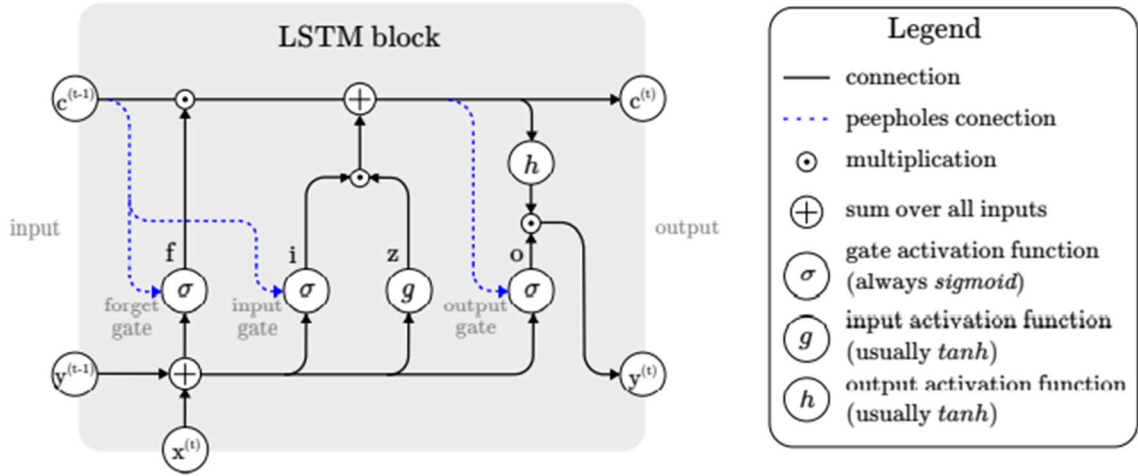
Resumindo, a arquitetura LSTM consiste em um conjunto de sub redes conectadas recorrentemente, conhecidas como blocos de memória. A ideia por trás do bloco de memória é manter seu estado ao longo do tempo e regular o fluxo de informações através de unidades de portas não lineares.

A Figura 15 mostra a arquitetura de um bloco LSTM, que envolve as portas, o sinal de entrada $x^{(t)}$, a saída $y^{(t)}$, as funções de ativação e as conexões Peephole⁴ (conexões do olho mágico). A saída do bloco é conectada repetidamente de volta à entrada do bloco e a todas as portas.

Com o objetivo de esclarecer como funciona o modelo LSTM, vamos supor uma rede composta por N blocos de processamento e M entradas. A passagem para a frente neste sistema neural recorrente é descrita abaixo.

⁴ As conexões Peephole foram originalmente introduzidas por Gers em 2000 para ajudar os LSTMs a aprender sincronizações precisas.

FIGURA 15 - ESTRUTURA DE LSTM



Fonte: GREFF, 2016.

Entrada de bloco: Esta etapa é dedicada à atualização do componente de entrada do bloco, que combina a entrada atual $x^{(t)}$ e a saída dessa unidade LSTM $y^{(t-1)}$ na última iteração. Isso pode ser feito conforme descrito abaixo:

$$z^{(t)} = g(W_z x^{(t)} + R_z y^{(t-1)} + b_z) \quad (6)$$

Onde W_z e R_z são os pesos associados com $x^{(t)}$ e $y^{(t-1)}$, respectivamente, enquanto b_z representa o vetor de peso de polarização.

Porta de Entrada: Nesta etapa, atualizamos a porta de entrada que combina a entrada atual $x^{(t)}$, a saída dessa unidade LSTM $y^{(t-1)}$, e o valor da célula $c^{(t-1)}$, na última iteração. A seguinte equação mostra este procedimento:

$$i^{(t)} = \sigma(W_i x^{(t)} + R_i y^{(t-1)} + p_i \odot c^{(t-1)} + b_i) \quad (6.1)$$

Onde $\odot$ denota a multiplicação pontual de dois vetores, W_i , R_i e p_i são os pesos associados com $x^{(t)}$, $y^{(t-1)}$ e $c^{(t-1)}$ respectivamente, enquanto b_i representa para o vetor de polarização associado com este componente.

Nas etapas anteriores, a camada LSTM determina quais informações devem ser retidas nos estados de célula da rede $c^{(t)}$. Isso incluiu a seleção dos valores candidatos $z^{(t)}$ que poderiam ser potencialmente adicionados aos estados da célula e os valores de ativação $i^{(t)}$ das portas de entrada.

Forget Gate (Porta de Esquecimento): Nesta etapa, a unidade LSTM determina quais informações devem ser removidas de seus estados de célula anteriores $c^{(t-1)}$. Portanto, os valores de ativação $f^{(t)}$ das portas de esquecimento na etapa de tempo t são calculados com base na entrada atual $x^{(t)}$, nas saídas $y^{(t-1)}$ e no estado $c^{(t-1)}$ das células de memória na etapa

de tempo anterior ($t - 1$), as conexões peephole e os termos de polarização b_f das portas de esquecimento. Isso pode ser feito da seguinte forma:

$$f^{(t)} = \sigma(W_f x^{(t)} + R_f y^{(t-1)} + p_f \odot c^{(t-1)} + b_f) \quad (6.7)$$

Onde W_f , R_f e p_f são os pesos associados com $x^{(t)}$, $y^{(t-1)}$ e $c^{(t-1)}$ respectivamente, enquanto b_f denota para o vetor de peso de polarização.

Célula: Esta etapa calcula o valor da célula, que combina a entrada do bloco $z^{(t)}$, a porta de entrada $i^{(t)}$ e os valores da porta de esquecimento $f^{(t)}$, com o valor da célula anterior. Isso pode ser feito conforme descrito abaixo:

$$c^{(t)} = z^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)} \quad (6.8)$$

Porta de saída: Esta etapa calcula a porta de saída, que combina a entrada atual $x^{(t)}$ a saída dessa unidade LSTM $y^{(t-1)}$ e o valor da célula $c^{(t-1)}$ na última iteração. Isso pode ser feito conforme descrito abaixo:

$$o^{(t)} = \sigma(W_o x^{(t)} + R_o y^{(t-1)} + p_o \odot c^{(t)} + b_o) \quad (6.9)$$

Onde W_o , R_o e p_o são os pesos associados com $x^{(t)}$, $y^{(t-1)}$ e $c^{(t-1)}$ respectivamente, enquanto b_o denota para o vetor de peso de polarização.

Saída do bloco: Finalmente, calculamos a saída do bloco, que combina o valor da célula atual $c^{(t)}$ com o valor da porta de saída atual da seguinte forma:

$$y^{(t)} = g(c^{(t)}) \odot o^{(t)} \quad (7.0)$$

Nas etapas acima, σ , g e h denotam funções de ativação não lineares pontuais. O sigmoide logístico $\sigma(x) = \frac{1}{1+e^{1-x}}$ é usado como uma função de ativação do portão, enquanto a tangente hiperbólica $g(x) = h(x) - \tanh(x)$ é freqüentemente usada como a função de ativação de entrada e saída do bloco.

A funcionalidade desta arquitetura inspirou os autores Li, F., Zhang, M., Tian, B., Chen, B., Fu, G., Ji, D. (2018) com o trabalho “*Recognizing irregular entities in biomedical text via deep neural networks*” a aprimorar o treinamento de redes muito profundas. O mecanismo de bloqueio foi empregado nas chamadas redes de rodovias para permitir um fluxo desimpedido de informações através de muitas camadas. Isso poderia ser considerado como mais uma prova de conceito, mostrando o funcionamento,

As redes LSTM possuem as mesmas características das redes recorrentes convencionais, porém possuem a capacidade de gravar informações por extensos períodos ao processar uma sequência temporal. Os pontos de memória de uma rede LSTM são denominadas células. Tais células são capazes de carregar informações até o final de uma sequência ou

identificar as informações que devem ser esquecidas pela rede a partir de alguma etapa de processamento.

Em 2019, Rodrigues, F., Markou, I. e Pereira, F.C., usaram LSTM para modelar as observações da série temporal e, posteriormente, melhorar as previsões combinando isso com a entrada de dados de texto. Usando essa técnica, eles tentaram prever a demanda de táxi em Nova York, embora o método proposto também seja generalizável para outras aplicações. Este é um benefício importante da arquitetura, pois foi mostrado empiricamente que o erro de previsão pode ser bastante reduzido com esta abordagem.

Devido a recursos complexos, como não linearidade, não estacionariedade e correlação de sequência, os dados financeiros representam um grande desafio de previsão. Foi mostrado por Fischer e Krauss, no entanto, que a rede LSTM supera os benchmarks mais tradicionais: a floresta aleatória, uma rede neural profunda padrão e uma regressão logística padrão.

2.9 REDES NEURAIAS NA PREVISÃO DO MERCADO FINANCEIRO

Quando se trata de sequências temporais em dados, os dados de séries temporais vêm imediatamente à mente. Ainda assim, este é um conceito amplo. No sentido mais literal das previsões de séries temporais, o modelo LSTM foi aplicado às previsões do mercado financeiro em, por exemplo, Fisher e Krauss (2018) e Yan e Ouyang (2019).

Sagheer e Kotb (2019) confirmaram esta descoberta de que o LSTM supera as abordagens padrão ao prever a produção de petróleo. Em sua pesquisa, eles empilharam várias camadas de blocos LSTM umas sobre as outras de maneira hierárquica. Isso aumentou a capacidade do modelo de processar tarefas temporais e permitiu capturar melhor a estrutura das sequências de dados.

Na tentativa de prever o preço de mercado do petróleo, Cen e Wang (2019) chegaram às mesmas conclusões. O modelo de predição proposto, composto pela arquitetura vanilla LSTM, mostrou-se superior.

Liu 2019 comparou o LSTM com outros modelos, como máquinas de vetores de suporte. Esta pesquisa na estimativa da volatilidade das ações financeiras não apenas mostrou que era relativamente fácil calibrar o LSTM, mas também que resulta em previsões precisas até mesmo para grandes intervalos de tempo.

Wu (2019) usaram a decomposição do modo empírico do conjunto para selecionar as funções do modo intrínseco adequado que servem como entrada para o modelo LSTM para

prever os movimentos do preço do petróleo bruto. Essa metodologia provou ser muito eficaz, mesmo quando o número de resultados de decomposição varia.

Historicamente, o mercado financeiro se concentrou em modelos estocásticos e matemáticos, com isso, as RNAs surgem no mercado financeiro como um auxílio para os investidores realizarem suas previsões, baseados em dados históricos na perspectiva de aumentar os seus retornos e ganhos. Segundo Kumar e Thenmozhi (2006, p. 13), baseado nas RNAs, a previsão do mercado acionário tem como alvo reduzir os riscos de investimentos e proteger os retornos dos investimentos contra a volatilidade do mercado.

Diversos são os estudos que utilizam redes neurais artificiais como ferramenta de previsão do mercado financeiro. Sobre essa questão, Guresen (2011) descrevem RNAs como uma das melhores técnicas para modelar o mercado de ações. Segundo os autores, uma rede neural não contém um padrão de fórmula e pode ser facilmente adaptada às mudanças que ocorrem no mercado devido, principalmente, à sua capacidade de aprendizagem e de fazer alterações de acordo com o que foi aprendido.

Mueller (1996), aplicou RNA na capacidade de previsão de preços das ações das empresas brasileiras, onde obteve resultados de que a RNA proporcionava um melhor sincronismo, se adequando com mais eficiente e rapidez as mudanças.

White (1988), foi um dos precursores na utilização de modelos de Redes Neurais para previsão do mercado de ações, em seu trabalho “*Proceedings of the IEEE International Conference on Neural Networks*”, o autor utilizou-se de uma rede neural sem realimentação para avaliar as ações da IBM, verificando e analisando alterações nos preços e os retornos diários, com o objetivo de avaliar a teoria do mercado eficiente proposta por Fama em 1970. Apesar de que os resultados descobertos por White (1988) não tenham alcançado níveis altos de previsão, a pesquisa destacou a potencialidade existente para este tipo de análise.

Nobre e Zárate (2013), apresentaram um modelo fundamentado em RNA, que permitiu realizar a previsão do desempenho dos preços das ações da Petrobras no curto prazo, combinando análise técnica, fundamentalista e de séries temporais, onde conseguiram uma assertividade acima de 84% nos modelos construídos.

Sob a mesma ótica, o estudo de Franco e Steiner (2014) utilizou modelos de redes neurais artificiais para previsão de séries temporais financeiras. Foi feito um experimento com o intuito de testar as redes em várias situações, principalmente no que diz respeito ao número de neurônios na camada oculta e às variações dos pesos iniciais. Para as simulações foram escolhidas as dez ações de maior peso na composição do Índice Bovespa e os dados de entrada da rede foram as cotações históricas das ações escolhidas. Os autores concluíram que as redes

mostradas estão aptas a fazer previsões de razoável acuidade, mesmo em momentos de volatilidade no mercado.

Sumarti e Wahydudi (2014) estudaram o processo de tomada de decisão na negociação de ações e opções com a utilização de redes neurais para uma carteira simulada. A partir de seus resultados, os autores concluíram que o método apresenta uma boa capacidade em prever o preço dos ativos, e que a carteira simulada obteve retornos consideráveis. Nessa mesma linha, a pesquisa de Dash, Dash e Bisoi (2015) utilizou redes neurais na previsão e modelagem precisa de dados financeiros. O modelo proposto pelos autores foi comparado com outras técnicas de previsão de séries temporais e os resultados indicaram que redes neurais oferecem melhorias significativas no desempenho da previsão de volatilidade.

2.10 MERCADO DE AÇÕES

As Bolsas de Valores são os centros de negociações das ações e garantem a operacionalização das negociações. Estas são organizadas como associações com funções de interesse público. Portanto, a Bolsa de Valores é o local onde as ações das empresas são compradas e vendidas. No Brasil, este local é a Bolsa de Valores, Mercadorias e Futuros de São Paulo, conhecida como a BM&FBovespa.

As operações de compra e venda de ações na BMF&Bovespa evoluíram de Pregão Viva Voz (PVV), onde os ativos eram negociados verbalmente por operadores humanos, para o conceito de Pregão Eletrônico (PE), onde os ativos são negociados a distância sem intermediário humano, tendo apenas um sistema eletrônico como ferramenta de negociação. Para detalhes de funcionamento do PE e as tecnologias utilizadas recomenda-se consultar o site oficial BMF&Bovespa, na seção regulamentos e manuais (BMFBOVESPA, 2017).

Pode-se classificar o investidor de acordo com a frequência de transações de compra e venda de ações (MURPHY, 1999; ELDER, 1993). Essa classificação determina o perfil do investidor.

Buy and Hold - São operações em que o prazo de retorno está previsto no período de anos;

Position Trader - As operações de compra e a venda de ações se dão no prazo de semanas e meses;

Swing Trader - São operações que acontecem no prazo de 2 a 6 dias;

Day Trader - As operações de compra e venda de ações acontecem no mesmo dia. O investidor que realiza esse tipo de operação espera pequenas variações no preço em curto prazo, aproveitando apenas uma parte do movimento e realizando diversas operações ao longo do dia.

Todas as formas de operar na Bolsa de Valores apresentadas acima têm uma característica em comum: podem ser feitas por pessoas através do PE, ou seja, as ordens de compra e de venda podem ser disparadas por seres humanos. Recentemente, uma outra forma de operar na bolsa foi introduzida, as HFT, que são negociações realizadas por computadores e algoritmos (ALDRIDGE, 2013) e visam obter lucro através de um grande volume de negócios com uma pequena margem de rentabilidade em cada transação. Nesse caso, as transações são realizadas completamente por computadores, sem qualquer interação humana.

Diversas abordagens com evidências empíricas abordam os efeitos das decisões de investimentos em ações. Em meio a elas, aparecem: a influência do quociente Q de Tobin como importante indicador de oportunidades de novos investimentos rentáveis para as empresas (BLOSE; SHIEH, 1997; CHUNG; WRIGHT; CHAROENWONG, 1998); a questão dos investimentos em pesquisa e desenvolvimento (CHAN; KENSINGER; MARTIN, 1990; TSETSEKOS; ZANTOUT, 1994).

3 METODOLOGIA

Nesta seção reporta-se como foram coletadas e tratadas as series históricas de preços e retornos de ações selecionadas do mercado de ações do Brasil, as variáveis a serem analisadas e os métodos estatísticos e de Machine Learning aplicados, no sentido de cumprir os dois objetivos específicos desta dissertação.

Para a comparação do desempenho dos modelos de Redes Neurais como ferramentas de previsão e amparo na tomada de decisões, foi realizado um experimento avaliando as métricas de R², MAE, MSE e RMSE, no qual os modelos dispostos para avaliação foram os modelos RNN-U, RNN-M, LSTM-U e LSTM-M.

Tal experimento utilizou-se da linguagem de programação *Python*⁵, como estrutura foi utilizado Keras, que é uma API de rede neural de alto nível escrita em Python, Keras não pode funcionar sozinho, ele precisa de um back-end para operações de baixo nível, portanto, instalamos uma biblioteca de software dedicada, TensorFlow do Google. Como ambiente de desenvolvimento, usamos o Anaconda Distribution e o Jupyter Notebook. Usamos Matplotlib para visualização de dados, Numpy para várias operações de Array e Pandas para análise de dados.

Para avaliação do poder preditivo fora da amostra das várias arquiteturas de redes neurais profundas, RNN-U, RNN-M, LSTM-U e LSTM-M, se utiliza um estudo de caso das ações que ABEV3, PETR3, PETR4, ITUB4, a quais compõe o índice IBOVESPA (Anexo A).

Justifica-se o estudo de casos de ações selecionadas pelo seu grau de liquidez e relevância do ponto de vista de investidores institucionais, e considerando-se que as ações representam firmas e setores heterogêneas.

3.1 TESTE DA RAZÃO DA VARIÂNCIA

O primeiro objetivo específico foi verificar se os retornos de ações da ABEV3, ITUB4, PETR3 e PETR4, seguem um processo estocástico Random Walk ou a validade da HME na forma sua forma fraca.

Uma das questões da econometria financeira é se os preços dos ativos financeiros são previsíveis. Talvez por causa da óbvia analogia entre investimentos financeiros e jogos de azar,

⁵ Python é uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991.

os modelos matemáticos de preços de ativos têm uma história extraordinariamente rica que antecede praticamente todos os outros aspectos da análise econômica. O fato de muitos matemáticos e cientistas proeminentes terem aplicado suas consideráveis habilidades para prever os preços dos títulos financeiros é uma prova do fascínio e dos desafios desse problema. De fato, a economia financeira moderna está firmemente enraizada nas primeiras tentativas de "vencer o mercado", um esforço que ainda é de interesse atual, discutido e debatido em artigos de jornais, conferências (LO ANDREW; CRAIG, 1988)

Ao decorrer dos anos diversos testes estatísticos foram desenvolvidos com o objetivo de testar a hipótese de que as séries financeiras seguem um passeio aleatório: se as séries financeiras como o preço da ação não seguem um passeio aleatório é uma evidência de que pode haver uma dependência temporal nas mesmas e justifica a utilização de alguma estratégia de predição. Neste cenário se destaca o teste da razão da variância, proposto por Lo Andrew W e MacKinlay A. Craig (1988).

Vários estudos foram realizados buscando confrontar a *Hipótese de Passeio Aleatório* (HPA) nas séries financeiras por meio destes testes (MESQUITA; OLIVEIRA; PEREIRA, 2019).

Com base nos trabalhos de Mesquita et al (2019), para se analisar a relação entre o desempenho da previsão dos modelos de Redes Neurais com os resultados de diferentes testes da razão da variância em séries, a hipótese sugerida para este trabalho é que é possível previamente selecionar as ações para a aplicação de redes neurais RNN e LSTM e obter resultados superiores na previsão fora da amostra, para ações que refutam a HPA.

As comparações entre a variância dos retornos de um período com múltiplos períodos é utilizada para testar a HPA. Vários testes tentam explorar quaisquer divergências desta predição, sendo um dos mais famosos o teste da razão da variância proposto por Lo Andrew W. e MacKinlay A. Craig (1988).

Supondo que o processo gerador de retornos é estacionário, a variância de um período é $V(1) = \text{var}(r_t)$ e a variância de dois períodos consecutivos e a soma das suas variâncias e covariâncias:

$$V(2) = \text{var}(r_t + r_{t+1}) = \text{var}(r_t) + \text{var}(r_{t+1}) + 2\text{cov}(r_t + r_{t+1}) = (2 + 2\rho_1)V(1)$$

Com ρ_1 sendo o primeiro *lag* da autocorrelação de um período das séries, a razão de dois períodos é então definida como:

$$VR(2) = \frac{V(2)}{2V(1)} = 1 + \rho_1 \quad (8)$$

O termo da autocorrelação é zero quando HPA se aplica e então a razão da variância é igual a 1. Caso contrário, a hipótese de HPA é falsa e a razão da variância pode ser tanto maior ou menor do que 1.

Considerando um período de N retornos, sendo N um inteiro maior ou igual a 2. Quando a hipótese de HPA é verdadeira,

$$V(N) = \text{var}(r_t + r_{t+1} + \dots + r_{t+N-1}) = \text{var}(r_t) + \text{var}(r_{t+1}) + \dots + \text{var}(r_{t+N-1}) = NV(1)$$

e assim a razão variância é 1 para todo N :

$$V(N) = \frac{V(N)}{NV(1)} = 1 \quad (8.1)$$

Quando a hipótese de HPA é falsa, $V(N)$ é igual a $NV(1)$ mais os termos de covariância entre todos os pares de retornos distintos, sendo assim:

$$V(N) = NV(1) + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^N \text{cov}(r_{t+i-1}, r_{t+j-1}) \quad (8.2)$$

$$V(N) = V(1) + [N + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^N p_{j-i}] \quad (8.3)$$

$$VR(N) = 1 + \frac{2}{N} \sum_{j=1}^{N-1} (N-j) p_j \quad (8.4)$$

O teste rejeita a hipótese nula de HPA quando o valor da razão se distânciava de 1. Isto acontece quando uma função linear das $(N-1)$ primeiras autocorrelações

$$(N-1)\rho_1 + (N-2)\rho_2 + (N-3)\rho_3 + \dots + 2\rho_{N-2} + \rho_{N-1} \quad (8.5)$$

se distanciam de zero.

Os autores Lo Andrew W. e MacKinlay A. Craig (1988) desenvolveram duas estatísticas para o teste da razão da variância: M1 e M2. De acordo com C. Amelie e D. Olivier (2009) se uma série temporal r_t é independente e identicamente distribuída (i.i.d.), sob a hipótese nula de $VR(N) = 1$ para um valor fixo de N e $T \rightarrow \infty$, então M1 é igual a:

$$M1(N) = \frac{VR(r,N)-1}{\phi(N)^{0.5}} \quad (9)$$

A estatística de teste M1(N) segue uma distribuição normal assintoticamente sob a hipótese nula de que $VR(N)=1$. A variância $\phi(N)$ é dada por:

$$\phi(N) = \frac{2(N-1)(N-1)-1}{3NT} \quad (9.1)$$

Já a estatística M2 é utilizada quando a série exibe heterocedasticidade condicional, e é igual a:

$$M2(N) = \frac{VR(r,N)-1}{\phi^*(N)^{0.5}} \quad (9.2)$$

M2 segue uma distribuição normal assintoticamente sob a hipótese nula de que $VR(N) = 1$, em que:

$$\phi(N) = \sum_{j=1}^{N-1} \left[\frac{2(n-j)}{N} \right]^2 \delta(j) \quad (9.3)$$

e

$$\delta(j) = \{\sum_{t=j+1}^T (r_t - \hat{u})^2 (r_{t-j} - \hat{u})^2\} \div \{[\sum_{t=1}^T (r_t - \hat{u})^2]^2\} \quad (9.4)$$

Lo e MacKinlay 1990 apresentaram que mais da metade da autocorrelação positiva do índice é atribuível a efeitos cruzados positivos baseado nas aplicações abordadas por eles. Eles também observam que efeitos cruzados positivos podem explicar a aparente lucratividade de estratégias de investimento contrárias, estratégias que são contrárias à direção geral do mercado. Essas estratégias, baseadas na noção de que os investidores tendem a reagir exageradamente à informação, consistem em vender "vencedores" e comprar "perdedores". Vender os vencedores e comprar os perdedores gerará lucros esperados positivos na presença de correlação serial negativa, porque os perdedores atuais provavelmente se tornarão vencedores futuros e os vencedores atuais provavelmente se tornarão perdedores futuros.

3.1.1 Coleta e Tratamento de Dados

Para cumprir este objetivo específico foram coletadas as séries históricas de preços com frequência diária do site do Yahoo Finance, depois calculados os retornos das ações da ABEV3, ITUB4, PETR3 e PETR4, e, por último, foi aplicado o teste da razão da variância, por meio do Script R do Anexo B.

Para definição das variáveis explicativas, foi baseado como arcabouço teórico propostos nos trabalhos de Fischer e Krauss (2018) e Zhang e colaboradores (2018), que apresentam modelos de RNA do tipo LSTM realizando a previsão do preço dos ativos, índices em vários mercados do mundo como Europa, Estados Unidos, Austrália, China e Brasil.

No desenvolvimento deste trabalho, os dados obtidos para a realização foram coletados através da biblioteca *yfinance* onde o conjunto de dados possui 2.463 amostras, uma para cada dia de operação das negociações, correspondendo ao período de 01/01/2012 até 31/12/2021 (10 anos), cada registro de amostra retorna o valor de fechamento do mercado, representados diariamente por dados de Abertura, Máximo, Mínimo, Fechamento e Volume. O Script para buscar as informações como exemplo da Petrobras (PETR3), foi escrito da seguinte forma:

```
# Loading the dataset
df = yf.download(tickers='PETR3.sa', start="2012-01-01", end="2021-12-31")
df.drop(df[df['Volume']==0].index, inplace = True)
df['Close'].plot()
```

A Figura 16 apresenta o gráfico com o período selecionado das ações PETR3.

FIGURA 16 - AÇÃO PETR3 DE 01/01/2012 A 31/12/2021



Fonte: O autor, <https://finance.yahoo.com/>, 2022.

O modelo de RNN e LSTM estabelece uma formatação de dados sequenciais, para isso, tais informações foram armazenadas em um *DataFrame*⁶ tendo como campo-chave o código da ação e a data, conforme exposto na Figura 17.

FIGURA 17 - CONJUNTO DE DADOS DA AÇÃO PETR3

```
In [6]: df
```

```
Out[6]:
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2012-01-02	23.129999	23.610001	22.740000	23.209999	16.632437	3758200
2012-01-03	23.209999	24.020000	23.209999	24.020000	17.345385	5402100
2012-01-04	23.799999	24.280001	23.570000	24.170000	17.453705	4588400
2012-01-05	24.100000	24.309999	23.760000	24.020000	17.345385	7905400
2012-01-06	24.100000	24.350000	23.910000	24.000000	17.330944	3454500
...	...	...	...	...	...	...
2021-12-23	30.350000	30.680000	30.180000	30.440001	30.440001	6243500
2021-12-27	30.440001	31.230000	30.240000	31.040001	31.040001	17868100
2021-12-28	31.260000	31.469999	31.000000	31.059999	31.059999	8078000
2021-12-29	31.010000	31.340000	30.750000	30.950001	30.950001	11460950
2021-12-30	30.950001	31.129999	30.549999	30.700001	30.700001	34157000

2463 rows x 6 columns

Fonte: O autor, 2022.

Assim, ao observar as ações, os dados de abertura (Open) são referentes ao preço do ativo quando o mercado é aberto para as atividades das operações financeiras compra e venda,

⁶ Pandas DataFrame é uma estrutura de dados bidimensional com os dados alinhados de forma tabular em linhas e colunas, mutável em tamanho e potencialmente heterogênea, semelhantemente a uma pasta de trabalho do MS-EXCEL

os dados de fechamento (Close) são referentes ao preço do ativo no momento do encerramento das operações financeiras e fechamento do mercado, os dados de fechamento ajustado (Adj Close) se referem ao preço de fechamento após todos os desdobramentos e distribuições de dividendos aplicáveis, os dados são ajustados usando multiplicadores apropriados de divisão e dividendos, aderindo aos padrões do Centro de Pesquisa em Preços e Títulos (CRSP), além disso durante as operações do dia de compra e venda são coletados os dados de máximo (High) e mínimo (Low) que indicam, os valores de máximo e mínimo de determinado ativo atingiu durante as variações que ocorram naquela data, e por fim os dados de Volume, que representam o volume de transações (compra e venda) daquele determinado dia.

Conforme Ashwood (2013), Bao, Yue e Rao (2017) e Fischer e Krauss (2018), o conjunto de dados foi no primeiro momento particionado em três subconjuntos, a saber: treino, validação e teste. O tamanho das sequências de treino, validação e teste se definiu com base nas observações de cada ação sendo 2.464 dias, se dividiu em 1.864 dias para treino, 300 para validação e os últimos 300 para teste e com base nesse treinamento.

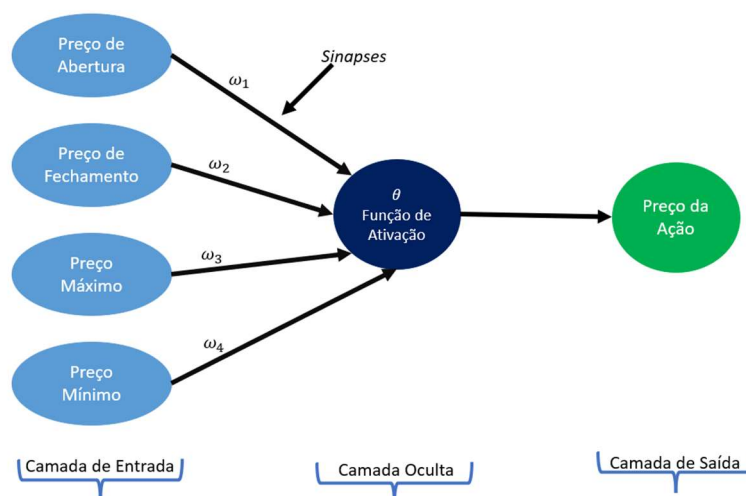
3.2 AVALIAÇÃO DO PODER PREDITIVO DE MODELOS DE REDES NEURAI RNN E LSTM

O segundo objetivo específico (avaliar o poder preditivo de modelos de redes neurais de memória curta e/ou longa), por sua vez, foi atingido por meio da aplicação do método do gradiente descente estocástico para avaliar e especificar os hiper parâmetros de modelos de redes neurais RNN e RNN LSTM, e por último, pela aplicação da técnica de validação cruzada para séries temporais e das métricas estatísticas para comparar o poder preditivo de modelos de redes neurais com memória curta e/ou longa: MAE, MSE, RMSE e R2 por meio do Script Python disponível em https://github.com/pinheirotj/LSTM_UEPG.

3.2.1 Modelo de Redes Neurais

Conforme apresentado no referencial teórico na equação 1, para se realizar o treinamento de uma Rede Neural, inicialmente será fornecido uma série de dados de entrada, de forma que a Rede Neural irá adquirir cada uma dessas informações, encontrar um padrão que caracteriza cada informação, de forma que toda vez que for introduzida uma nova informação que ela não tenha observado ainda, deverá identificar o padrão de comportamento, realizando todo esse processo de aprendizado de forma autônoma.

FIGURA 18 - EXEMPLO DE ESTRUTURA DE UMA RNA



Fonte: O autor, 2022..

A função sigmoide é uma função que calibra a partir de uma entrada entre 0 e 1, ou seja, se a entrada é muito negativa ela terá um valor próximo de 0, se a entrada é muito positiva ela terá um valor próximo de 1, em resumo, ela coloca o resultado da saída entre 0 e 1, como exemplo o treinamento de um neurônio.

$$\text{Preço da Ação} = \varphi_i(\text{Preço de Abertura}_1\omega_{1,0} + \text{Preço de Fechamento}_2\omega_{2,0} + \text{Preço Máximo}_3\omega_{3,0} + \text{Entrada}_n\omega_{n,0} + \theta_i)$$

Se faz necessário a utilização do uso de Sistema Computacional para realizar a calibração dos pesos. Cada escolha de peso é realizada de forma aleatória, e então conforme é realizada o treinamento e validação os pesos vão sendo calibrados e ajustados.

Para se realizar a calibração, cria-se uma função, essa função ira somar todo o erro que essa Rede Neural apresentou, sendo a soma de todos os erros elevado ao quadrado, retornando o tamanho do erro total, a intenção é deixar esse erro o mais próximo de zero, ou seja, momento em que a rede está mais calibrada, para diminuir esse erro pode se iniciar ajustando os pesos, alterando o primeiro peso e verificando o que ocorreu com erro, e então se realiza o recalcule somente com aquela alteração, com base no resultado, se identifica uma redução se realiza o aumento do valor do peso, o inverso também é verdadeiro. Essa etapa se repete para cada peso, realizando os testes e comparações até que a função seja reduzida. Na fase final a rede neural realiza o comparativo com a informação de saída e a correta que deveria ser obtida.

A melhor estratégia é pegar todos os valores de entrada e calibrar os pesos para todas as entradas, ou seja, se utiliza de todas amostras e informa ao mesmo tempo, entrando com a primeira amostra, calculando o peso e o valor da função de erro, na sequência outra amostra com outros pesos calculados e outra função erro, com isso pode ser realizado a soma das funções

de erros de cada amostra, obtendo uma função de erro total, sendo o somatório de todos os erros individuais, dessa forma ocorre a calibração para todos os casos de uma única vez para todas as entradas, corrigindo as saídas individuais.

Após a função erro ser calibrada, se realiza o processo de teste, ou seja, utiliza-se as outras informações de amostras que a Rede Neural não observou, amostras diferentes do treinamento, são as amostras que se sabe o resultado de saída, mas não foram informadas para a RNA, com essas novas amostras é realizada a conferência e verificação de saída se foi a esperada ou não e estima-se a eficácia de quanto ela foi assertiva.

No Aprendizado Profundo, ou seja, muitas camadas e por conta dessa quantidade de camadas é denominado *deep learning*, onde todos os neurônios estão conectados uns aos outros, o cálculo de número de variáveis se torna muito maior, de tal forma:

$$N_{c1} = \varphi_i(\sum E\omega_e + \theta_e) \quad (10)$$

Onde N_{c1} representa o neurônio da camada oculta 1, E representa as entradas, realizando assim o somatório de todas as entradas, multiplicando pelos seus respectivos pesos ω_e da entrada, mais a constante θ_e bias.

$$N_{c2} = \varphi_i(\sum N_{c1}\omega_{c1} + \theta_{c1}) \quad (10.1)$$

O Mesmo processo se repete para o próximo Neurônio N_{c2} , cada um deles vai ser calculado com base no valor do neurônio da camada c_1 , sendo N_{c1} neurônio da camada 1, multiplicado pelo peso da camada 1 w_{c1} , até todos os neurônios realizarem todas as somas.

$$N_s = \varphi_i(\sum N_{c2}\omega_{c2} + \theta_{c2}) \quad (10.2)$$

Os neurônios da camada de saída N_s vão depender dos neurônios da camada 2 N_{c2} multiplicando pelos respectivos pesos da camada 2 w_{c2} . E assim se repete de acordo com o número de camadas ocultas disponíveis no modelo.

A principal vantagem de se ter mais variáveis e que dessa forma se obtém de muitos mais valores para ajustar, com isso possibilita uma precisão mais assertiva, tendo uma quantidade de ajuste maior, dado que na rede neural simples, chegamos ao ponto de limitação de desempenho, não se consegue reduzir o valor da função erro, neste caso por ter mais variáveis obtém-se valor da função erro mais baixo, tornando a rede neural mais precisa ainda.

Além disso, pode-se fragmentar as entradas pelas camadas ocultas, onde cada neurônio está identificando cada fração, ou seja, observando para cada “pedaço” da amostra e não como o conjunto todo, ele está calibrando e verificando se o neurônio está ativado ou não, e outra camada estará analisando uma camada macro, por ser o acúmulo das ativações das camadas anteriores e assim se identifica para ativação ou não o neurônio, dessa forma os neurônios de

saída irão atrelar todos os neurônios das camadas anteriores e ativar os neurônios de saída. Cada camada fica responsável por identificar um padrão diferente, considerado um modelo de caixa preta, pois não se consegue visualizar na prática e compreender como está funcionando, com isso conseguimos obter o melhor resultados pois é subdividido as tarefas, é quanto mais profundo (mais camadas), mais precisão pode se ter, em teoria.

A arquitetura de LSTM necessita de diversos testes e cenários para conseguir alcançar a melhor e ideal forma, para tal, os principais atributos que influenciam no desempenho e desenvolvimento do modelo são a quantidade de camadas, quantidade de neurônios por camadas e as funções de ativação dos neurônios ocultos e da saída de rede (RUBIM, 2019).

Se realizou o treinamento sobre dois modelos de aprendizado profundo, RNN e LSTM, sendo que para o modelo LSTM se empregou da sua versão multivariada denominado como LSTM-M e univariada denominada como LSTM-U, e por fim o modelo RNN univariado e multivariado denominado como RNN-U e RNN-M.

Nos modelos RNN-U e LSTM-U realizou-se o treinamento apenas na série de preços de fechamento ajustado do conjunto de dados, portanto, sendo um modelo univariado, para os modelos LSTM-M e RNN-M, treinou-se o modelo na série de registros contendo preço de abertura, fechamento, baixa e alta, tendo como alvo o preço de fechamento ajustado.

3.2.2 Validação Cruzada em Série Temporal

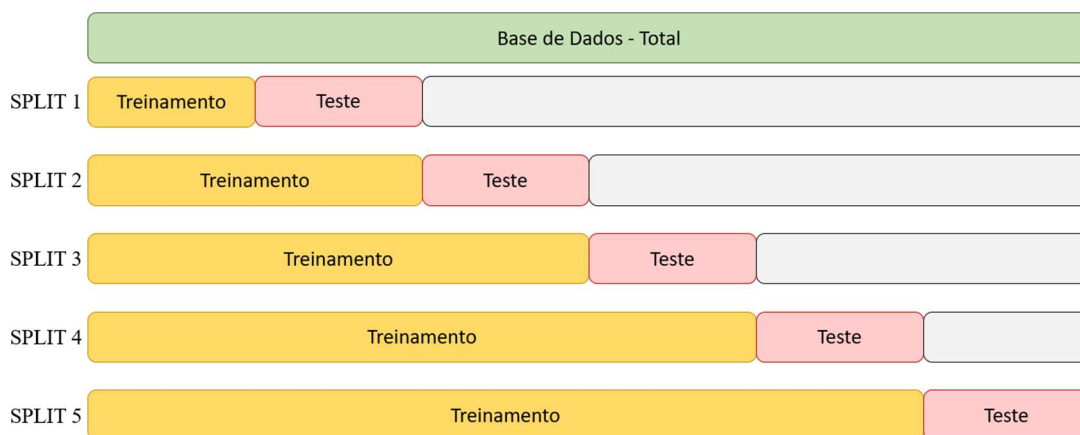
Para melhorar a acuracidade das amostras, se realizou o tratamento dos dados, removendo dados redundantes e ruídos, como os registros com volume igual a zero e os registros que são idênticos ao registro anterior. Para unificar o intervalo de dados, aplicou-se a normalização mínima e máximo e se mapeou os valores para um intervalo entre 0 e 1.

A validação cruzada é frequentemente usada para fornecer uma avaliação objetiva do desempenho de um modelo. Na validação cruzada, os dados são divididos arbitrariamente em duas partes: conjunto de treinamento e conjunto de teste. O desempenho do modelo é avaliado pela média do erro quadrado sob validação em todas as tentativas do experimento, assim o modelo deve ser treinado K vezes, onde $1 < K \leq N$ (KOHAVI, 1995).

De acordo com os autores Hyndman e Athanasopoulos (2021), no sentido de reduzir o overfitting e identificar o padrão de comportamento, se faz necessário a divisão do conjunto de dados em determinadas partes.

Neste trabalho utilizou-se da validação cruzada de forma contínua proposta por Hyndman e Athanasopoulos (2021), realizando-se a quebra do conjunto de dados em cinco partições denominadas splits conforme apresenta a figura 18.

FIGURA 19 - COMPOSIÇÃO DOS 5 SPLITS PARA VALIDAÇÃO CRUZADA



Fonte: O autor, 2022.

Foi seguida também as duas principais regras propostas:

- 1) Cada conjunto de dados de teste deve conter observações únicas;
- 2) As observações do conjunto de treinamento ocorrem antes de seu conjunto de teste correspondente.

Conforme podemos observar na Figura 18, a base de dados total foi dividida em 5 splits, com isso se realizou uma validação cruzada de 5 vezes, onde o tamanho de cada split corresponde a 300 dias, como resultado temos 5 pares de conjuntos de treinamento/teste:

- Split 1: Treinamento dos primeiros 900 dias e Teste dos 300 dias posteriores;
- Split 2: Treinamento dos primeiros 1.200 dias e Teste dos 300 dias posteriores;
- Split 3: Treinamento dos primeiros 1.500 dias e Teste dos 300 dias posteriores;
- Split 4: Treinamento dos primeiros 1.800 dias e Teste dos 300 dias posteriores;
- Split 5: Treinamento dos primeiros 2.100 dias e Teste dos 300 dias posteriores;

Com base nos resultados de cada split, se armazenou as seguintes métricas estatísticas: MAE; MSE; RMSE e R^2 , assim após o último teste do split 5, se realizou a médias dessas métricas estatísticas.

A principal vantagem e contribuição desse método de validação cruzada contínua é que se utiliza de vários conjuntos de dados para se realizar o procedimento de treinamento e em seguida o teste, tal procedimento busca verificar se os resultados irão contribuir de forma significativa na otimização do *overfitting*, e obtenção de maior robustez dos modelos avaliados.

3.2.3 Gradiente Descendente Estocástico

Métodos iterativos de otimização são muito utilizados em Aprendizado de Máquina. Para este trabalho se utilizou do método de gradiente descendente estocástico. No aprendizado profundo, a função objetivo é comumente a média das funções de perda para cada exemplo no conjunto de dados de treinamento. Matematicamente podemos definir uma função objetiva:

$$f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \quad (11)$$

Em que $f_i(x)$ é a função perda do conjunto de dados de treinamento com n exemplos, um índice de i é um vetor de parâmetro de x .

O gradiente da função objetivo em x é calculado como

$$\nabla f(x) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x) \quad (11.1)$$

Se o gradiente descendente for utilizado, o custo de computação para cada iteração de variável independente é $\mathcal{O}(n)$, que cresce linearmente com n . Portanto, quando o conjunto de dados de treinamento do modelo é grande, o custo de descida do gradiente para cada iteração será muito alto.

A descida do gradiente estocástica (SGD) reduz o custo computacional a cada iteração realizada. A cada iteração de descida do gradiente estocástico, amostramos uniformemente um índice $i \in \{1, \dots, n\}$ para exemplos de dados aleatoriamente e se calcula o gradiente $\nabla f_i(x)$ para atualizar x :

$$x \leftarrow x - \eta \nabla f_i(x) \quad (11.2)$$

Em que η é a taxa de aprendizado, pode se observar que o custo computacional de cada interação cai de $\mathcal{O}(n)$ da descida do gradiente para constante $\mathcal{O}(1)$. Deve-se mencionar que o gradiente estocástico $\nabla f_i(x)$ é a estimativa imparcial do gradiente $\nabla f(x)$.

$$\mathbb{E}_i \nabla f_i(x) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x) = \nabla f(x) \quad (11.3)$$

Isso significa que, em média, o gradiente estocástico é uma boa estimativa do gradiente.

Os hiper parâmetros não são aprendidos pela Rede Neural durante o treinamento e devem ser ajustados manualmente. Nos modelos propostos por gradiente descendente, podemos distinguir cinco hiper parâmetros:

(i) Timesteps: Quantidade de dias que o modelo de rede irá avaliar para dados históricos para realizar a previsão;

- (ii) hl: Quantidade de camadas e número de neurônios, para todos os modelos se utilizou de duas camadas sendo que para cada camada da rede neural, a quantidade de neurônios foi escolhida para atingir o mínimo da função;
- (iii) lr: Taxa de aprendizado;
- (iv) Batch size: Tamanho dos lotes de dados que vai trabalhar, foi utilizado para os modelos lotes de 32 e 64 amostras;
- (v) num_epochs: Quantidade de interações que a rede vai realizar para aprender, com foco em minimizar o erro;

A taxa de aprendizado define o tamanho dos passos que daremos em direção ao mínimo em cada iteração. Se esses passos forem muito pequenos, é quase garantido que chegaremos ao ponto de mínimo da função, mas para isso talvez precisemos de muitas iterações de treino, tornando o algoritmo desnecessariamente lento. Por outro lado, se atribuído uma taxa de aprendizado muito alta, pode acontecer de ter valores acima da função custo e ir cada vez mais longe do mínimo, resultando em uma falha completa de aprendizado. Isso acontecerá quando o passo inserido for tão grande que pulará o ponto de mínimo e chegará em um ponto na função custo mais alto do que de onde saímos. Nesse novo ponto, o gradiente será ainda maior, aumentando mais ainda o passo seguinte e arremessando ainda mais longe do ponto de mínimo a cada iteração (NESTEROV; VIAL, 2000; HAZAN et al., 2008).

Destaca-se que a taxa de aprendizado não deve ser nem tão grande, nem tão pequena. Como sugestão de ajustamento desse hiper parâmetro é iniciar com 0.01 e explorar os pontos em volta dez vezes maior/menor (isto é, 0.1 e 0.001). Na maioria dos casos, uma boa taxa de aprendizado será algum dos seguintes valores: 1, 0.1, 0.01, 0.001, 0.0001, 0.00001.

Com uma boa taxa de aprendizado, selecionar o número de iterações de treino é uma tarefa fácil. Mesmo assim, recomenda-se plotar o valor da função custo a cada iteração de treino, dessa forma conseguimos identificar se a função custo já chegou em uma região em que o seu valor não diminui ou diminui pouco a cada iteração.

Para fornecer respostas e um direcionamento estatístico e matemático, durante o processo de modelagem utilizou-se os parâmetros mais utilizados para descrever matematicamente a relação entre alguns preditores e a variável alvo. Utilizamos as métricas de R² (R ao quadrado), MSE (Erro Quadrático Médio), RMSE (Raiz do Erro Quadrático Médio), MAE (Erro Absoluto Médio) e MAPE (Erro Percentual Absoluto Médio).

4 RESULTADOS E DISCUSSÕES

Nesta seção se apresenta os principais resultados obtidos com os modelos de aprendizado profundo: RNN-U, RNN-M, LSTM-U e LSTM-M, ambos desenvolvidos nesse primeiro momento para as ações PETR3, PETR4, ITUB4 e ABEV3 sendo seus atributos os dados de abertura “Open”, máximo “High”, mínimo “Low” e Fechamento Ajustado “Adj Close”.

4.1 TESTE DA RAZÃO DA VARIÂNCIA PARA RANDOM WALK

As Tabelas 1, 2 e 3 reportam respectivamente os valores das estatísticas M1 e M2 do teste da razão da variância para os valores de $N=2$ (período diário), $N=5$ (período semanal) e $N=15$ (período proxy mensal). No caso das variáveis aleatórias independentes e identicamente distribuídas i.i.d do Random Walk, se utiliza a M1 e para o caso dos retornos independentes, mas com heterocedasticidade se utiliza o M2.

Como no teste da razão da variância, o numerador de M1 e M2 é igual a $VR-1$ e como $VR < 1$ representa a presença de autocorrelação serial negativa e $VR > 1$ a presença de autocorrelação positiva; então se os M1 e M2 estimados forem maiores ou menores do que 0, pode-se inferir que há autocorrelação serial positiva ou negativa, respectivamente.

Os testes de M1 e M2 para as ações da PETR3, PETR4, ITUB4 e ABEV, que foram realizadas no software R podem ser observadas na Tabela 1 para período igual a 2 diário.

TABELA 1 - TESTE DA RAZÃO DA VARIÂNCIA PARA $N=2$, PERÍODO 01/01/2012 A 31/12/2021

Ação	M1	M2
PETR3	-2.329286**	-1.100138**
PETR4	-2.51858***	-1.20063***
ABEV3	-3.90826***	-1.95699***
ITUB4	-2.76718***	-1.81681***

Nota: (*) Nível de significância de 10%; (**) Nível de significância de 5%; (***) Nível de significância de 1%.
Fonte: O autor, 2022..

Conforme pode-se observar nos resultados da razão da variância, todas as ações rejeitaram a hipótese nula, ou seja, refutam a HME na forma fraca, em ambas as estatísticas de teste M1 e M2 as ações apresentam uma certa dependência temporal mostrando que a série de retornos das ações não apresentam uma Random Walk, uma vez que apresentaram autocorrelação negativa a níveis significativos estatisticamente de 5% e 1%.

Observa-se que para o período semanal, as ações ainda apresentam uma correlação negativa, conforme os resultados da tabela a seguir:

TABELA 2 - TESTE DA RAZÃO DA VARIÂNCIA PARA N=5, PERÍODO 01/01/2012 A 31/12/2021

Ação	M1	M2
PETR3	-0.5067334	-0.2399211
PETR4	-0.5172683	-0.257729
ABEV3	-2.236484**	-1.138724**
ITUB4	-3.070363***	-2.117452***

Nota: (*) Nível de significância de 10%; (**) Nível de significância de 5%; (***) Nível de significância de 1%.
Fonte: O autor, 2022.

Nesse caso, observa-se que somente as ações da ABEV3 e ITUB4, rejeitam a hipótese nula a níveis de significância estatística de 5% e 1%, por outro lado as ações da PETR3 e PETR4 não rejeitaram a hipótese nula.

Por fim a Tabela 3, retorna os resultados obtidos para o período de 3 semanas de N=15.

TABELA 3 - TESTE DA RAZÃO DA VARIÂNCIA PARA N=15, PERÍODO 01/01/2012 A 31/12/2021

Ação	M1	M2
PETR3	1.588621	0.7961932
PETR4	1.6354	0.8576313
ABEV3	-1.963313**	-1.16748**
ITUB4	-1.834996*	-1.348344*

Nota: (*) Nível de significância de 10%; (**) Nível de significância de 5%; (***) Nível de significância de 1%.
Fonte: O autor, 2022.

Verifica-se que as ações da PETR3 e PETR4 para períodos mais longo, apresentam uma autocorrelação positiva, porém não rejeitam a hipótese nula, por fim as ações da ABEV3 e ITUB4, continuam apresentando uma autocorrelação negativa e rejeitam a hipótese nula a níveis de 5% e 10%.

Estes resultados empíricos sugerem que aplicações de modelos de previsões de redes neurais recorrentes de aprendizado profundo com memória curta e longa como LSTM-U ou LSTM-M tendem a obter resultados melhores do que modelos de previsão sem memória ou com memória curta que não captam não linearidades como RNN-U ou RNN-M, no caso das ações da ABEV3 e ITUB4.

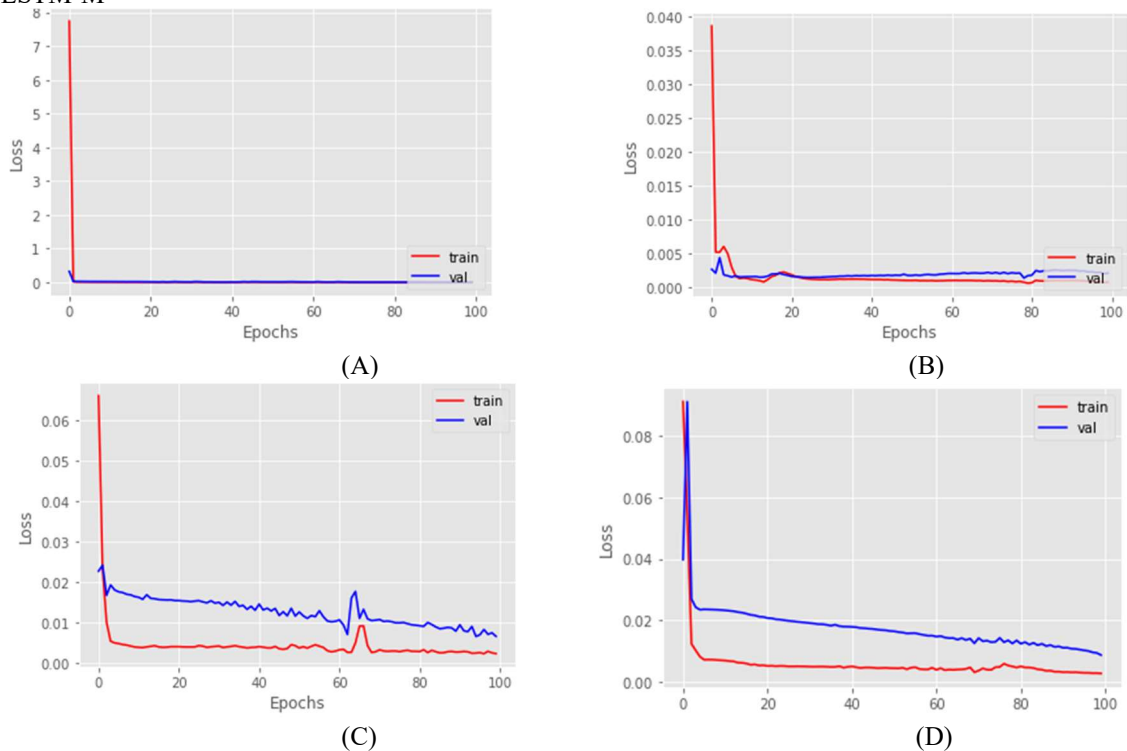
4.2 AVALIAÇÃO DOS HIPER PARÂMETROS DOS MODELOS

Como apresentado na metodologia, para a escolha dos hiper parâmetros das redes neurais RNN-U, RNN-M, RNN LSTM-U e RNN LSTM-M, neste trabalho foi aplicado a função perda do gradiente descendente: a seleção de hiper parâmetros destas redes neurais artificiais buscou minimizar o erro da função perda do gradiente descendente.

4.2.1 Resultados da Ação PETR3

Para o estudo de caso da ação PETR3, realizou-se o processo da definição dos hiper parâmetros para avaliação do gradiente descendente, o qual se obteve os seguintes resultados.

FIGURA 20 - GRADIENTE DESCENTE PARA AÇÃO PETR3: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M



Fonte: O autor, 2022.

Conforme podemos observar na Figura 20, para a ação da PETR3, para os modelos RNN-U e RNN-M foi necessário a interação de aproximadamente 20 vezes da rede para que ela conseguisse aprender e minimizar o erro de aprendizado, por outro lado, para os modelos LSTM-U e LSTM-M se fez necessário a interação de mais de 60 vezes para se minimizar o erro de aprendizado, os hiper parâmetros utilizados para cada Modelo podem ser observados a seguir:

TABELA 4 – HIPER PARÂMETROS PARA TREINAMENTO DA AÇÃO PETR3

PETR3	RNN-U	RNN-M	LSTM-U	LSTM-M
Timesteps	25	25	25	25
hl	[50; 35]	[50; 35]	[50; 35]	[50; 35]
lr	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
batch_size	64	64	64	64
num_epochs	100	100	100	100

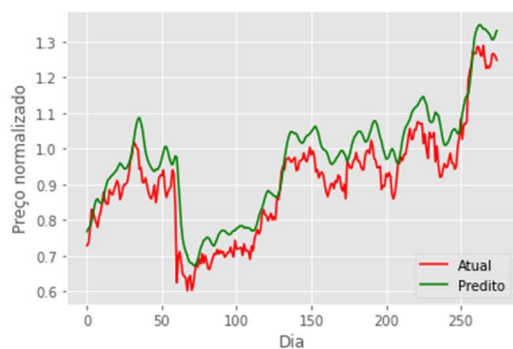
Fonte: O autor, 2022..

Os resultados com base nesses hiper parâmetros podem ser observados na Figura 21.

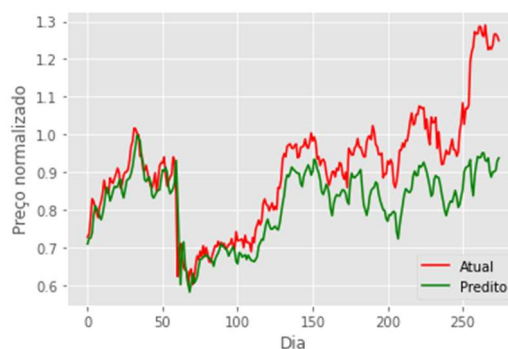
FIGURA 21 - MÉTRICAS ESTATÍSTICAS AÇÃO PETR3: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M

MAE = 0.06495175869706855
MSE = 0.005691455834380278
RMSE = 0.07544173801272262
R2-Score = 0.74637667860092

MAE = 0.09275178265696972
MSE = 0.01597897960610492
RMSE = 0.12640798869575023
R2-Score = 0.28794275870153074



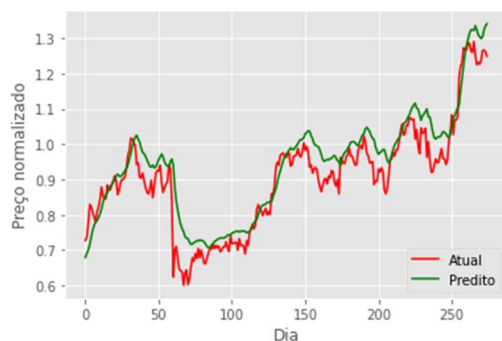
(A)



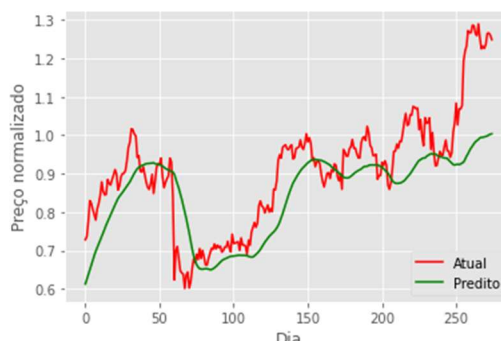
(B)

MAE = 0.04847631423766091
RMSE = 0.0618777381614187
MSE = 0.003828854479973092
R2 = 0.8293781419378385

MAE = 0.08380194086501591
MSE = 0.012283485440141927
RMSE = 0.1108308866703769
R-Squared Score = 0.4526217713680587



(C)



(D)

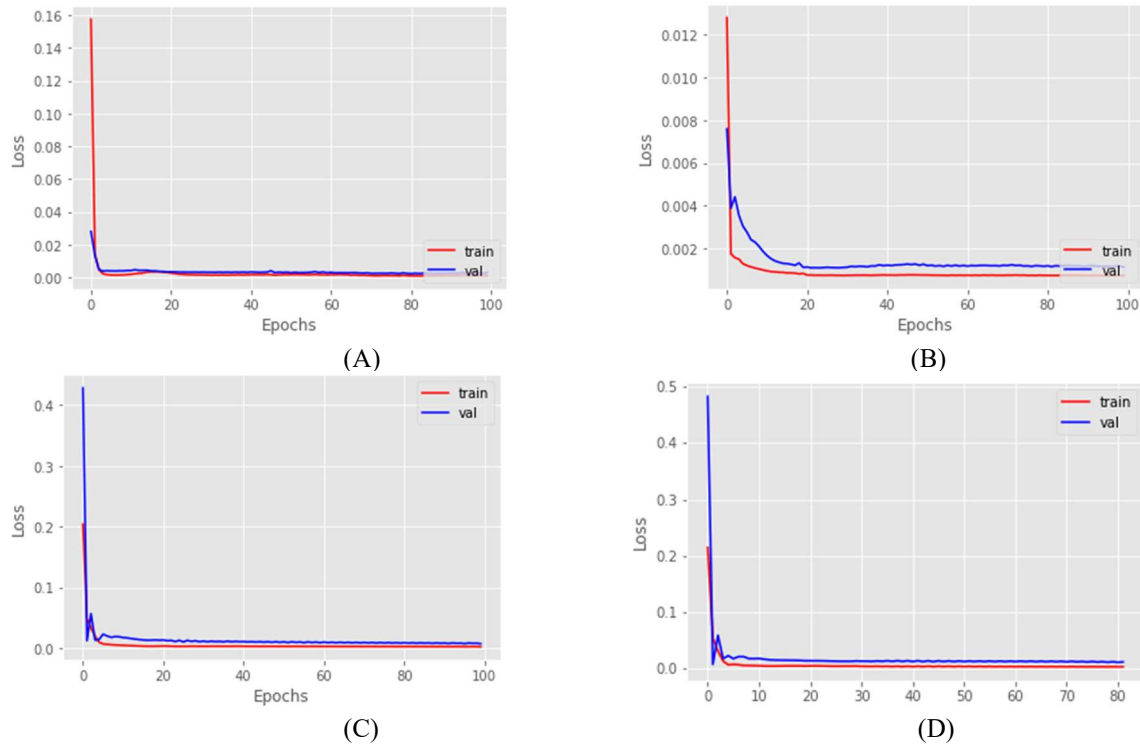
Fonte: O autor, 2022.

Para cada modelo obteve-se as métricas estatísticas de MAE, MSE, RMSE e R2, onde pode se observar que o modelo que melhor apresentou resultados baseado na métrica do coeficiente de determinação R2, foi o modelo LSTM-U, que retornou um R2 de 0.82, seguido pelo modelo RNN-U com R2 de 0.74, modelo LSTM-M com R2 de 0.45 e por fim o modelo de RNN-M apresentou um valor de R2 de 0.28.

4.2.2 Resultados da Ação PETR4

Para o estudo de caso da ação PETR4, realizou-se o processo da definição dos hiper parâmetros para avaliação do gradiente descendente, o qual se obteve os seguintes resultados.

FIGURA 22 - GRADIENTE DESCENTE PARA AÇÃO PETR4: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M



Fonte: O autor, 2022.

Conforme podemos observar na Figura 22, para a ação da PETR4, todos os modelos RNN-U, RNN-M, LSTM-U e LSTM-M foi necessário a interação de aproximadamente 20 vezes da rede para que ela conseguisse aprender e minimizar o erro de aprendizado, os hiper parâmetros utilizados para cada Modelo podem ser observados na tabela a seguir:

TABELA 5 – HIPER PARÂMETROS PARA TREINAMENTO DA AÇÃO PETR4

PETR4	RNN-U	RNN-M	LSTM-U	LSTM-M
Timesteps	10	10	10	10
hl	[30; 25]	[30; 25]	[30; 25]	[30; 25]
lr	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
batch_size	64	64	64	64
num_epochs	100	100	100	100

Fonte: O autor, 2022.

Os resultados com base nesses hiper parâmetros podem ser observados na Figura 23.

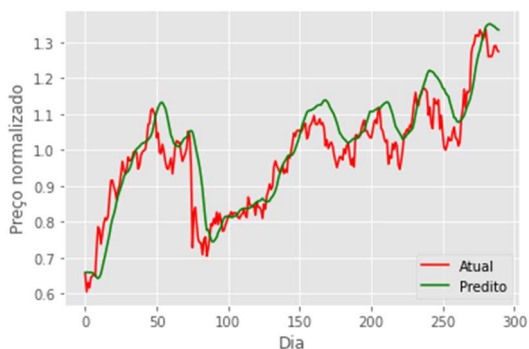
FIGURA 23 - MÉTRICAS ESTATÍSTICAS AÇÃO PETR4: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M

MAE = 0.03903533998954843
MSE = 0.002710182132905438
RMSE = 0.052059409648068794
R2-Score = 0.8752151355734519



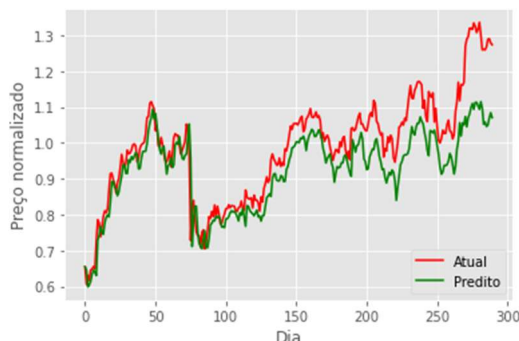
(A)

MAE = 0.05458440401873771
RMSE = 0.0733499732780428
MSE = 0.005380218579889592
R2 = 0.7522786766533711



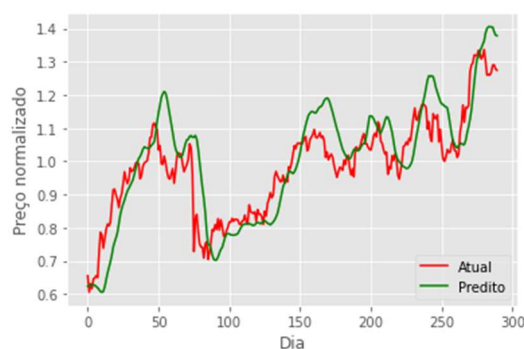
(C)

MAE = 0.06631374213377779
MSE = 0.007400841646237986
RMSE = 0.08602814450072713
R2-Score = 0.6592432983456262



(B)

MAE = 0.0735395652233312
MSE = 0.008901974057700048
RMSE = 0.09435027322535981
R-Squared Score = 0.5901266883574476



(D)

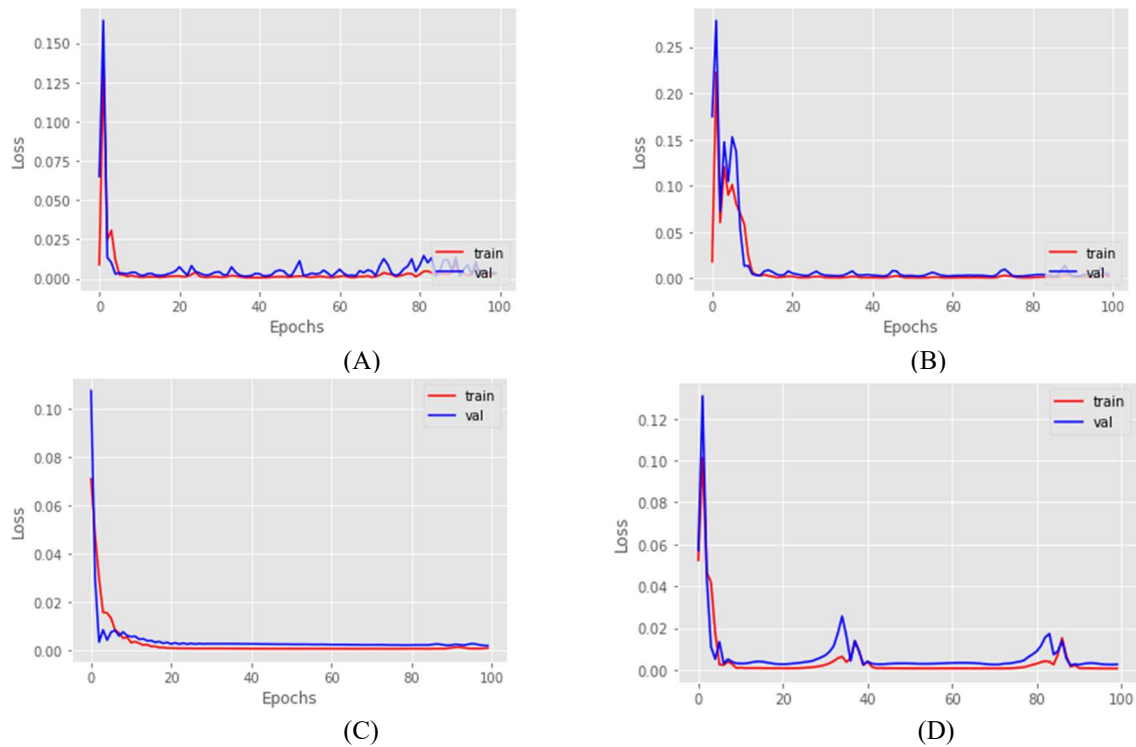
Fonte: O autor, 2022.

Para cada modelo obteve-se as métricas estatísticas de MAE, MSE, RMSE e R2, onde pode se observar que o modelo que melhor apresentou resultados baseado na métrica do coeficiente de determinação R2, foi o modelo RNN-U que retornou um R2 de 0.87, seguido pelo modelo LSTM-U com R2 de 0.75, o modelo RNN-M com R2 de 0.65 e por fim o modelo de LSTM-M apresentou um valor de R2 de 0.59.

4.2.3 Resultados da Ação ITUB4

Para o estudo de caso da ação ITUB4, realizou-se o processo da definição dos hiperparâmetros para avaliação do gradiente descendente, o qual se obteve os seguintes resultados.

FIGURA 24 - GRADIENTE DESCENTE PARA AÇÃO ITUB4: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M



Fonte: O autor, 2022.

Conforme podemos observar na Figura 24, para a ação da ITUB4, para os modelos RNN-U; RNN-M e LSTM-U foi necessário a interação de aproximadamente 20 vezes da rede para que ela conseguisse aprender e minimizar o erro de aprendizado, por outro lado o modelo LSTM-M a rede começou a minimização dos erros a partir da interação 40, os hiper parâmetros utilizados para cada Modelo podem ser observados na tabela a seguir:

TABELA 6 – HIPER PARÂMETROS PARA TREINAMENTO DA AÇÃO ITUB4

ITUB4	RNN-U	RNN-M	LSTM-U	LSTM-M
Timesteps	10	10	10	10
hl	[40; 35]	[40; 35]	[40; 35]	[40; 35]
lr	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
batch_size	64	64	64	64
num_epochs	100	100	100	100

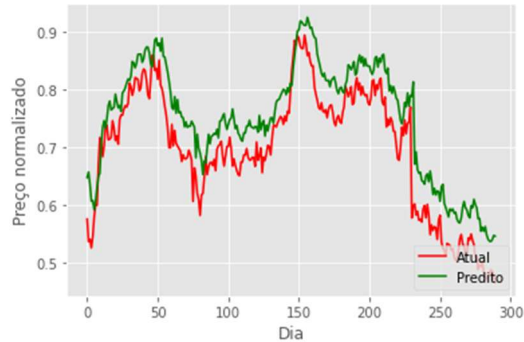
Fonte: O autor, 2022.

Os resultados com base nesses hiper parâmetros podem ser observados na Figura 25.

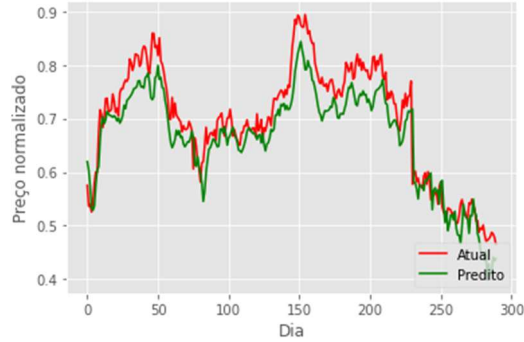
FIGURA 25 - FIGURA 25 – MÉTRICAS ESTATÍSTICAS AÇÃO ITUB4: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M

MAE = 0.054853268637986216
MSE = 0.003790326539623935
RMSE = 0.061565627907331014
R2-Score = 0.6627466058647082

MAE = 0.03778735843131389
MSE = 0.00204167887149718
RMSE = 0.045184940760137994
R2-Score = 0.8183365471727476



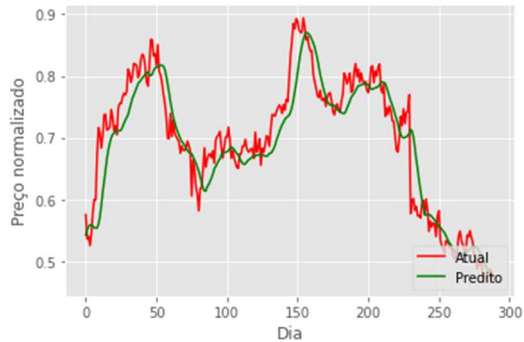
(A)



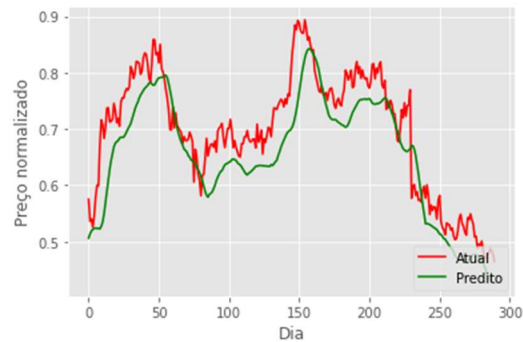
(B)

MAE = 0.03076675300708403
RMSE = 0.040651083032965274
MSE = 0.0016525105517530372
R2 = 0.852963816917484

MAE = 0.051108273996007675
MSE = 0.0036762262669331014
RMSE = 0.060631891500538736
R-Squared Score = 0.6728990895596667



(C)



(D)

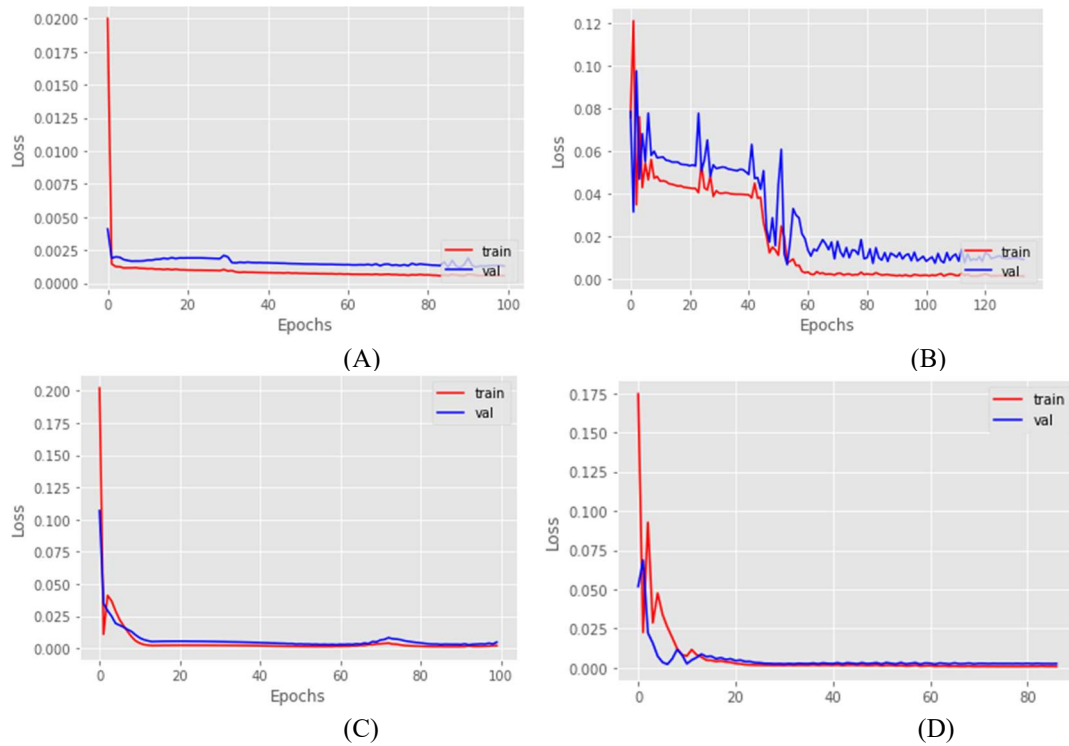
Fonte: O autor, 2022.

Para cada modelo obteve-se as métricas estatísticas de MAE, MSE, RMSE e R2, onde pode se observar que o modelo que melhor apresentou resultados baseado na métrica do coeficiente de determinação R2, foi o modelo LSTM-U, que retornou um R2 de 0.85, seguido pelo modelo RNN-M com R2 de 0.81, o modelo LSTM-M com R2 de 0.67 e por fim o modelo de RNN-U com R2 de 0.66.

4.2.4 Resultados da Ação ABEV3

Para o estudo de caso da ação ABEV3, realizou-se o processo da definição dos hiper parâmetros para avaliação do gradiente descendente, o qual se obteve os seguintes resultados.

FIGURA 26 - GRADIENTE DESCENTE PARA AÇÃO ABEV3: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M



Fonte: O autor, 2022.

Conforme podemos observar na Figura 26, para a ação da ABEV3, para os modelos RNN-U, LSTM-U e LSTM-M foi necessário a interação de aproximadamente 20 vezes da rede para que ela conseguisse aprender e minimizar o erro de aprendizado, por outro lado o modelo RNN-M a rede começou a minimização dos erros a partir da interação 100, os hiper parâmetros utilizados para cada Modelo podem ser observados na tabela 7 a seguir:

TABELA 7 – HIPER PARÂMETROS PARA TREINAMENTO DA AÇÃO ABEV3

ABEV3	RNN-U	RNN-M	LSTM-U	LSTM-M
Timesteps	10	10	10	10
hl	[50; 35]	[50; 35]	[50; 35]	[50; 35]
lr	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
batch_size	64	64	64	64
num_epochs	100	100	100	100

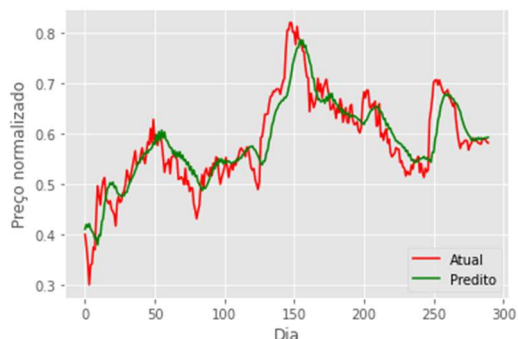
Fonte: O autor, 2022.

Os resultados com base nesses hiper parâmetros podem ser observados na Figura 27.

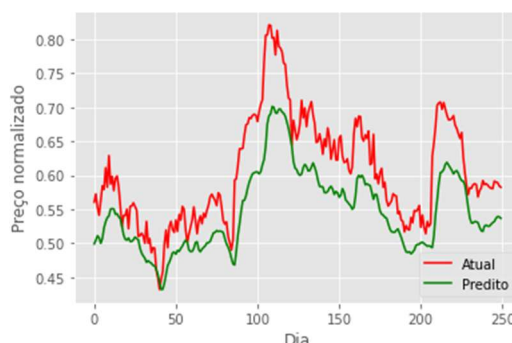
FIGURA 27 - MÉTRICAS ESTATÍSTICAS AÇÃO ABEV3: (A) RNN-U; (B) RNN-M; (C) LSTM-U; (D) LSTM-M

MAE = 0.03198219471676629
MSE = 0.0017812939131787884
RMSE = 0.04220537777557249
R2-Score = 0.7860821205088628

MAE = 0.05758833254051054
MSE = 0.0042229621230474934
RMSE = 0.06498432213270747
R2-Score = 0.3277174810154837



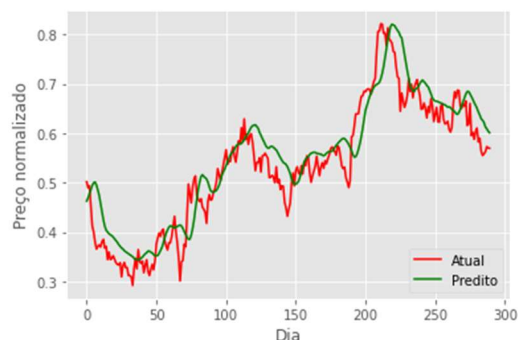
(A)



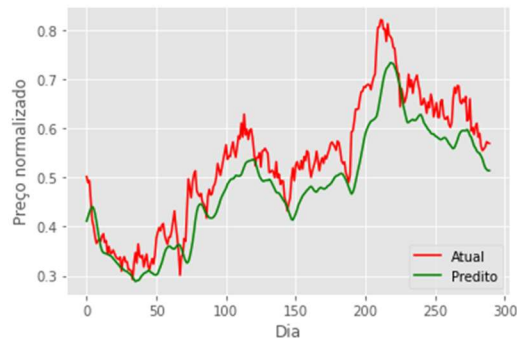
(B)

MAE = 0.04023686896923212
RMSE = 0.04991905180793081
MSE = 0.0024919117334028804
R2 = 0.8453492747123511

MAE = 0.05353040293814257
MSE = 0.004068637895201621
RMSE = 0.06378587535812001
R-Squared Score = 0.7474959080031022



(C)



(D)

Fonte: O autor, 2022.

Para cada modelo obteve-se as métricas estatísticas de MAE, MSE, RMSE e R2, onde pode se observar que o modelo que melhor apresentou resultados baseado na métrica do coeficiente de determinação R2, foi o modelo LSTM-U, que retornou um R2 de 0.84, seguido pelo modelo RNN-U com R2 de 0.78, o modelo LSTM-M com R2 de 0.74 e por fim o modelo de RNN-M com R2 de 0.32.

4.3 AVALIAÇÃO DOS MODELOS POR VALIDAÇÃO CRUZADA

Partindo da metodologia apresentada de validação cruzada, se realizou alguns ensaios para produzir o design dos modelos RNN-U, RNN-M, LSTM-M e LSTM-U, para as ações PETR3, PETR4, ITUB4 e ABEV3 onde para ambos se utilizou de duas camadas ocultas compostas por neurônios do tipo recorrentes, ou seja, que possuem retroalimentação.

Os hiper parâmetros utilizados para cada modelo e ação foram os que apresentam os melhores retornos de minimização da curva do gradiente descendente, tais hiper parâmetros utilizados foram demonstrados no item anterior, por se tratar de Redes Neurais, não existe um valor padronizado para se utilizar e assumir como padrão para os modelos testados, pois cada modelo e arquitetura possui características e comportamentos diferentes, devido principalmente ao tipo do papel que está sendo negociado, volatilidade e outros impactos.

Se realizou o processo de validação cruzada, com 5 splits para cada uma das ações do modelo, a fim de verificar o comportamento das curvas de preço previsto comparando com o preço realizado, pois os metodos de avaliação podem gerar valores diferentes, uma vez que, ao utilizar o método onde se define o range de treinamento, teste e validação como fixos, pode-se cair no viés de se realizar o teste em uma amostragem simples e de fácil aprendizado e/ou de aprendizado muito complexo, podendo acarretar em resultados viesados, por outro lado, ao se utilizar a validação cruzada, pode-se obter um maior range de amostras e treinamento e se estimar os valores com maior precisão.

4.3.1 Validação Cruzada PETR3

Com base nos resultados dos hiper parâmetros obtidos através da curva da função perda do gradiente descendente da ação da PETR3 apresentado na seção 4.2.1, se realizou o treinamento para os modelos de RNN-U, RNN-M, LSTM-M e LSTM-U utilizando o método de validação cruzada com 5 splits para a ação da PETR3, se cujos resultados obtidos são reportados na Tabela 8.

TABELA 8 - RESULTADOS AÇÃO PETR3 – VALIDAÇÃO CRUZADA

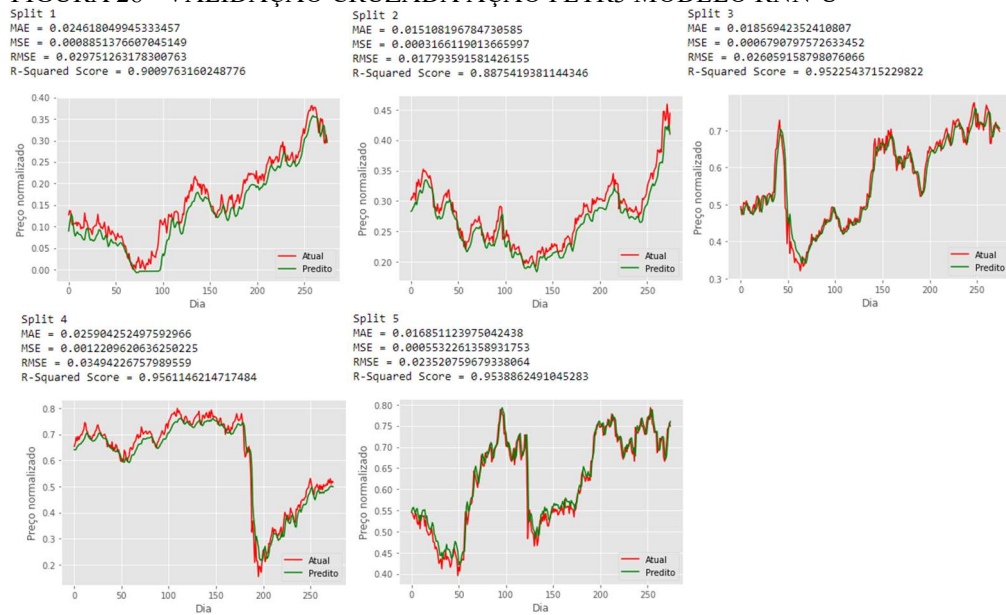
PETR3	RNN-U	RNN-M	LSTM-U	LSTM-M
MAE	0.02021	0.08290	0.02777	0.08458
MSE	0.00073	0.00916	0.00152	0.00931
RMSE	0.02641	0.08862	0.03616	0.09091
R2_Score	0.93015	0.26748	0.88465	0.18780

Fonte: O autor, 2022.

Pode-se verificar, por meio da validação cruzada e das métricas R2, MAE, MSE, RMSE de que cada modelo, tem sua particularidade de desempenho, podendo se destacar para este estudo de caso da PETR3, que os modelos com melhor adaptação e coeficiente de determinação são do tipo RNN-U e LSTM-U.

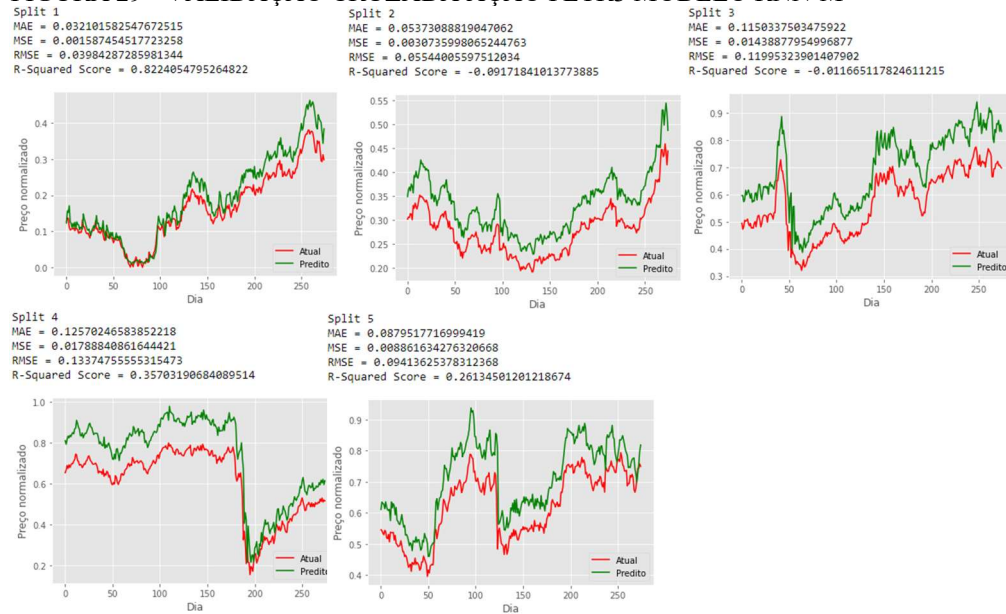
Os resultados dos splits para cada modelo realizado pode ser observado na sequência:

FIGURA 28 - VALIDAÇÃO CRUZADA AÇÃO PETR3 MODELO RNN-U



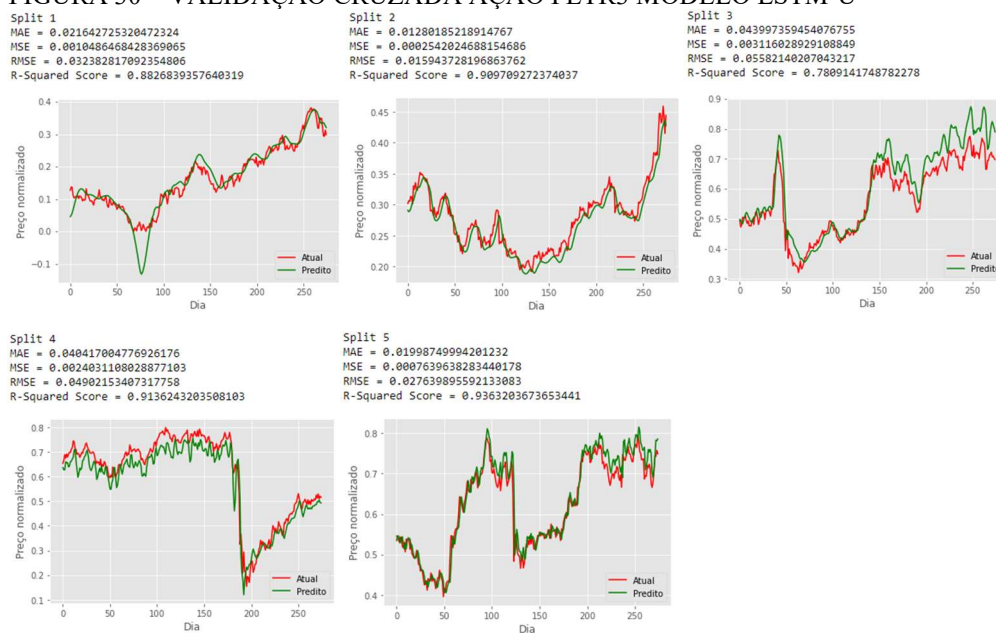
Fonte: O autor, 2022.

FIGURA 29 - VALIDAÇÃO CRUZADA AÇÃO PETR3 MODELO RNN-M



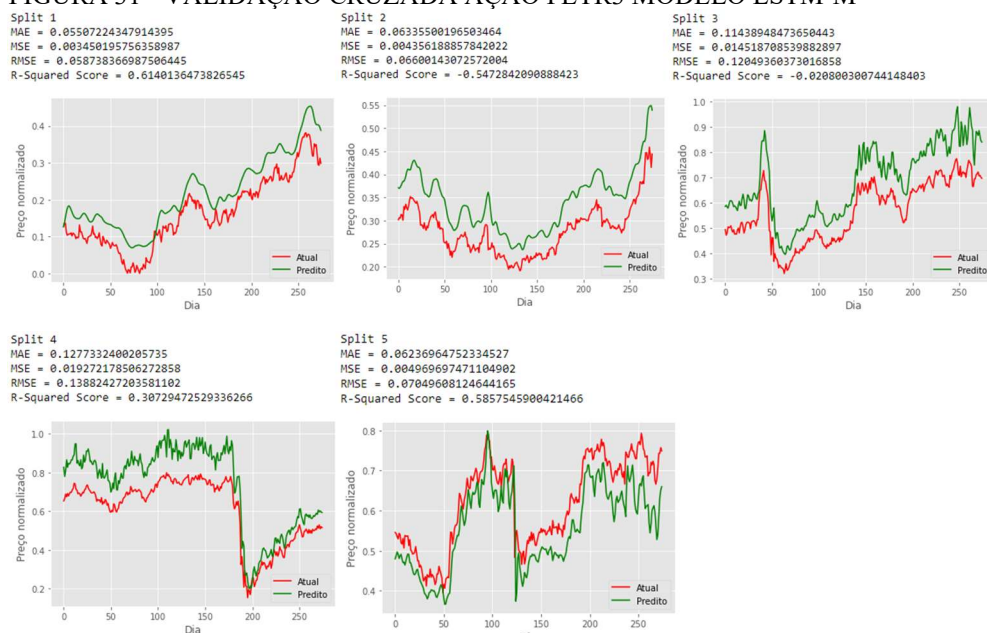
Fonte: O autor, 2022.

FIGURA 30 – VALIDAÇÃO CRUZADA AÇÃO PETR3 MODELO LSTM-U



Fonte: O autor, 2022.

FIGURA 31 - VALIDAÇÃO CRUZADA AÇÃO PETR3 MODELO LSTM-M



Fonte: O autor, 2022.

Conforme, pode se observar pela visualização gráfica e baseado nos resultados obtidos para a ação da PETR3 descritos na Tabela 8, identifica-se que o modelo com maior assertividade foi o modelo de RNN-U, que retornou um R2 de 0.93015, ou seja, o modelo RNN-U conseguiu explicar cerca de 93,01% do valor previsto, demonstrando que o modelo com memória curta do tipo univariado para esta ação analisada apresentou os índices de assertividade melhores em relação aos demais modelos.

4.3.2 Validação Cruzada PETR4

Com base nos resultados dos parâmetros obtidos através da curva da função perda do gradiente descendente da ação da PETR4 apresentado na seção 4.2.2, se realizou o treinamento para os modelos de RNN-U, RNN-M, LSTM-M e LSTM-U utilizando método de validação cruzada com 5 splits para a ação da PETR4, cujos resultados são reportados na Tabela 9.

TABELA 9 - RESULTADOS AÇÃO PETR4 – VALIDAÇÃO CRUZADA

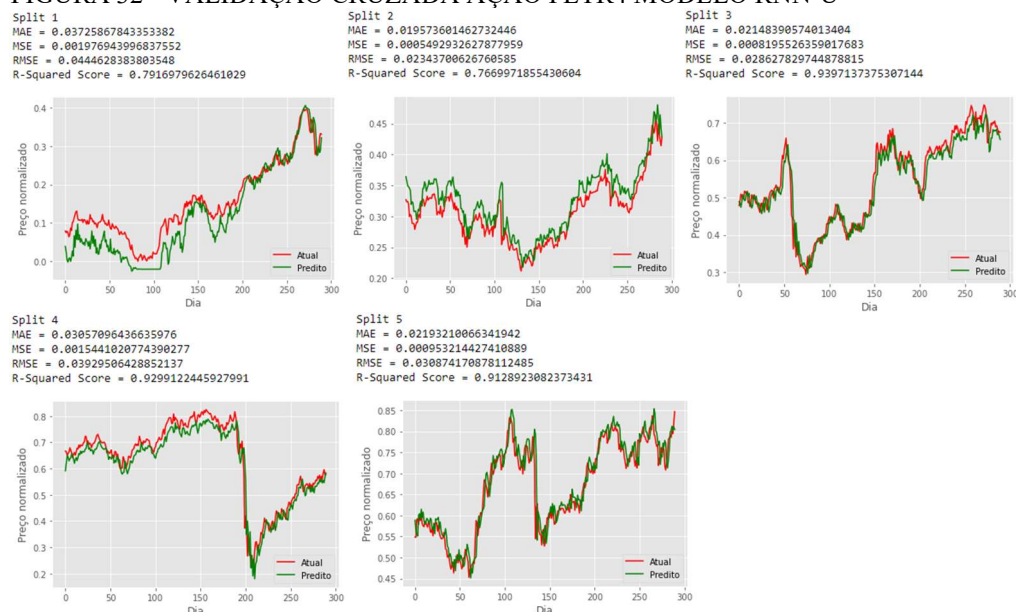
PETR4	RNN-U	RNN-M	LSTM-U	LSTM-M
MAE	0.02616	0.06963	0.02692	0.06512
MSE	0.00117	0.00635	0.00136	0.00661
RMSE	0.03334	0.07523	0.03544	0.07546
R2_Score	0.86824	0.29493	0.87323	0.32145

Fonte: O autor, 2022.

Pode-se identificar, por meio da validação cruzada e da métrica do coeficiente de determinação R2, de que cada modelo, tem sua particularidade de desempenho, pode-se destacar para o caso da PETR4 que os modelos com melhor adaptação e coeficiente de determinação foram do tipo RNN-U e LSTM-U.

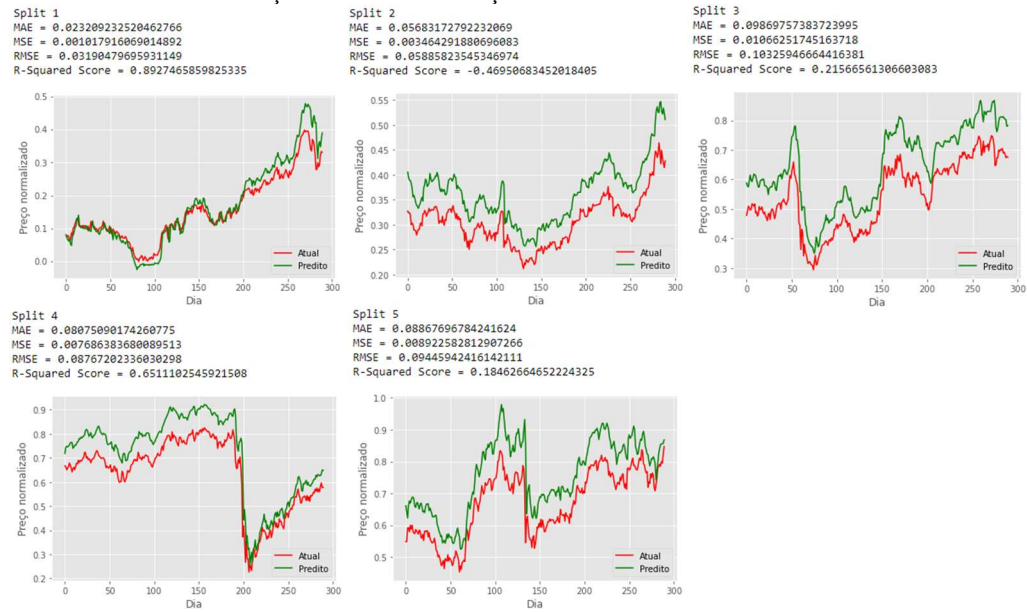
Os resultados dos splits para cada modelo realizado pode ser observado na sequência:

FIGURA 32 - VALIDAÇÃO CRUZADA AÇÃO PETR4 MODELO RNN-U



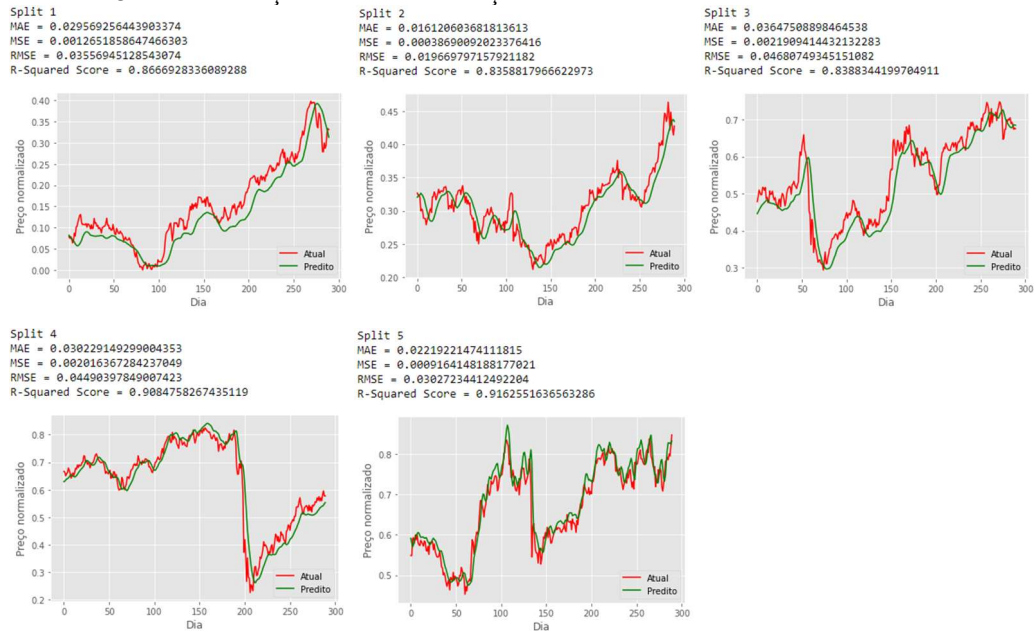
Fonte: O autor, 2022.

FIGURA 33 - VALIDAÇÃO CRUZADA AÇÃO PETR4 MODELO RNN-M



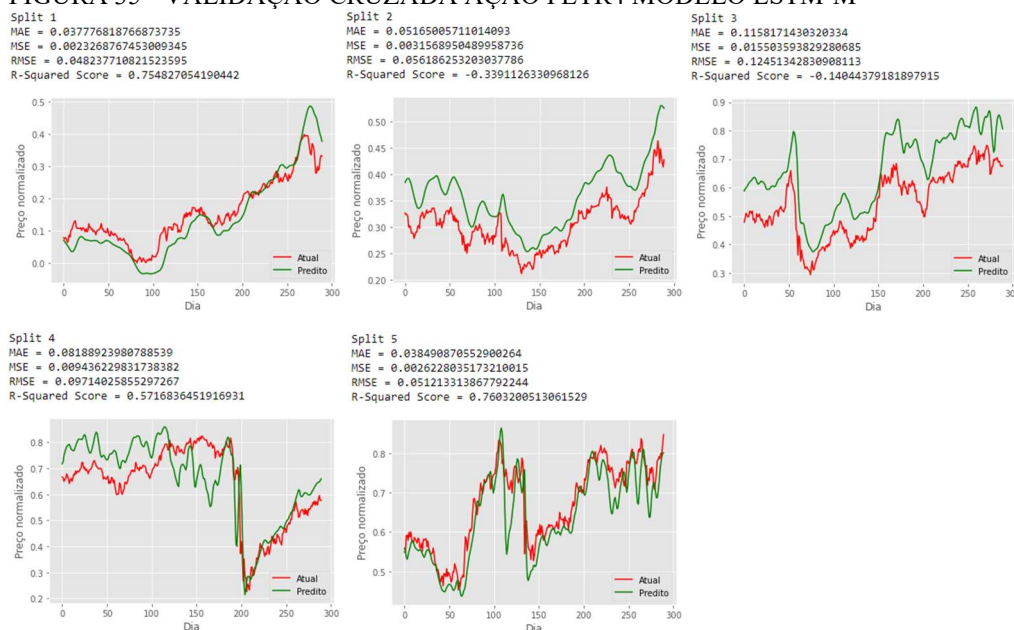
Fonte: O autor, 2022.

FIGURA 34 - VALIDAÇÃO CRUZADA AÇÃO PETR4 MODELO LSTM-U



Fonte: O autor, 2022.

FIGURA 35 - VALIDAÇÃO CRUZADA AÇÃO PETR4 MODELO LSTM-M



Fonte: O autor, 2022.

Conforme, pode se observar pela visualização gráfica e baseado nos resultados obtidos para a ação da PETR4 descritos na Tabela 9, identifica-se que o modelo com maior assertividade foram os modelos de RNN-U e LSTM-U, que retornaram R2 de 0.86824 e R2 de 0.87323, ou seja, os modelos RNN-U e LSTM-U conseguiram explicar cerca de 86,82% e 87,32% respectivamente do valor previsto, demonstrando que nesse caso modelo com memória curta e longa do tipo univariado para esta ação analisada apresentaram os índices de assertividade melhores em relação aos demais modelos.

4.3.3 Validação Cruzada ITUB4

Com base nos resultados dos parâmetros obtidos através da curva função perda do gradiente descendente da ação ITUB4 apresentado em 4.2.3, se realizou o treinamento para os modelos de RNN-U, RNN-M, LSTM-M e LSTM-U utilizando método de validação cruzada com 5 splits para a ação da ITUB4, cujos resultados obtidos são reportados na Tabela 10.

TABELA 10 - RESULTADOS AÇÃO ITUB4 – VALIDAÇÃO CRUZADA

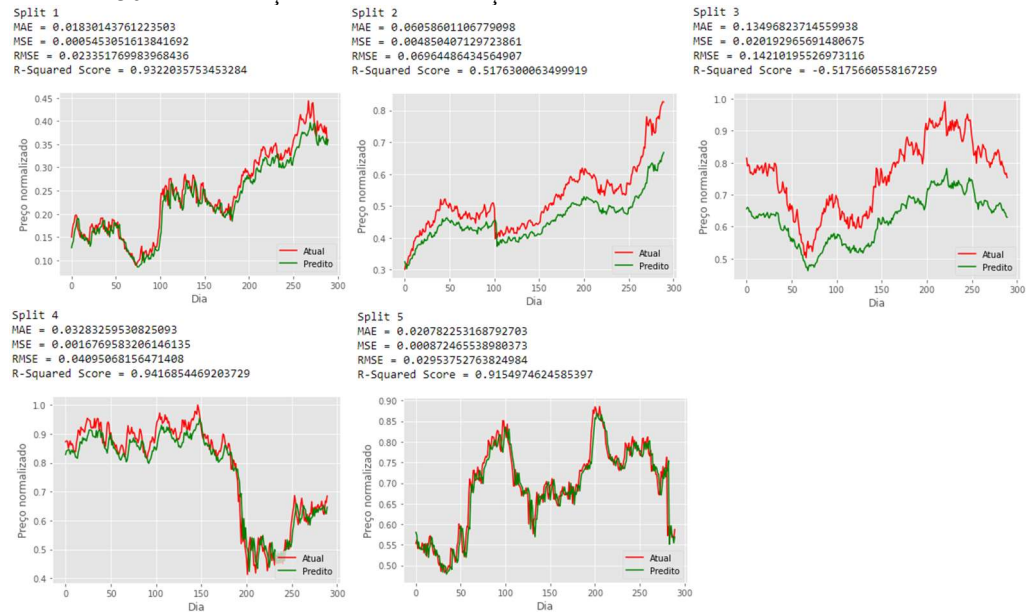
ITUB4	RNN-U	RNN-M	LSTM-U	LSTM-M
MAE	0.05349	0.05671	0.02963	0.03149
MSE	0.00563	0.00411	0.00165	0.00164
RMSE	0.06112	0.06280	0.03597	0.03875
R2_Score	0.55789	0.65627	0.90481	0.87730

Fonte: O autor, 2022.

Pode-se identificar, por meio da validação da métrica do coeficiente de determinação R^2 , de que cada modelo, tem sua particularidade de desempenho, podendo ressaltar para o caso da ITUB4 que os modelos com melhor adaptação e coeficiente de determinação são do tipo LSTM-U e LSTM-M.

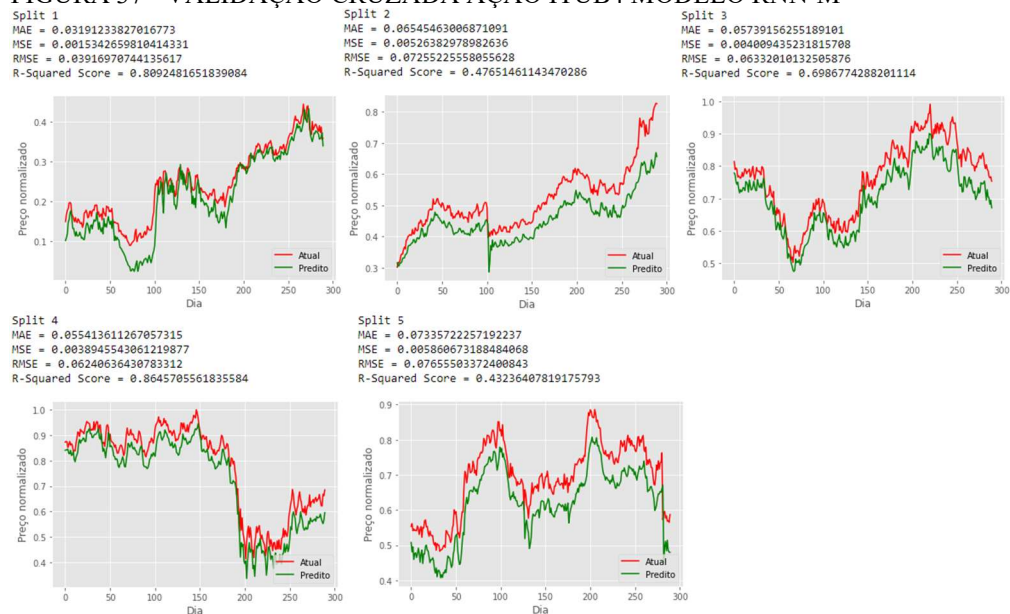
Os resultados dos splits para cada modelo realizado pode ser observado na sequência:

FIGURA 36 - VALIDAÇÃO CRUZADA AÇÃO ITUB4 MODELO RNN-U



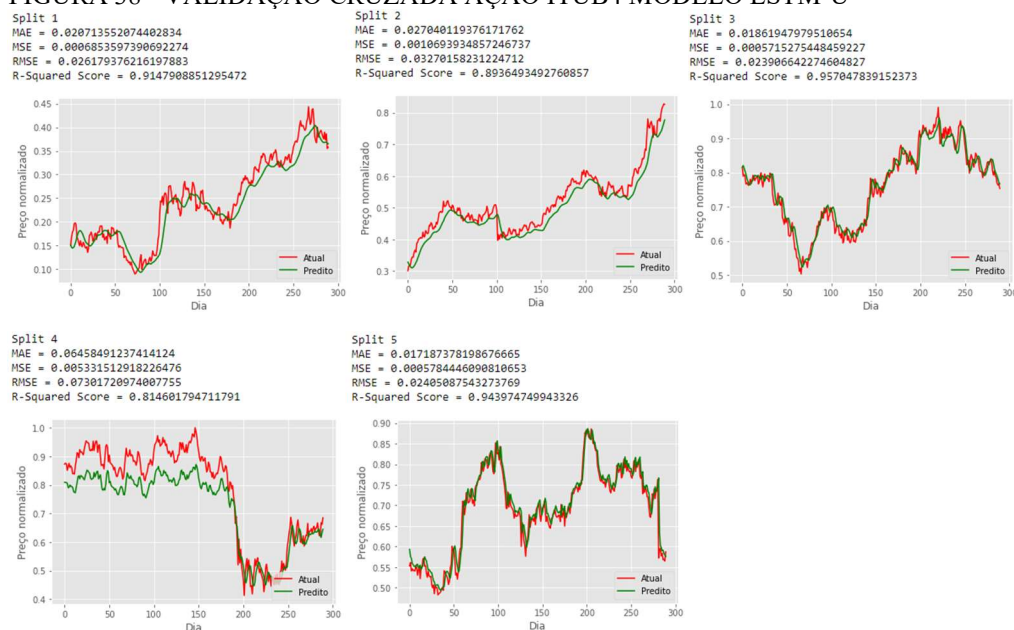
Fonte: O autor, 2022.

FIGURA 37 - VALIDAÇÃO CRUZADA AÇÃO ITUB4 MODELO RNN-M



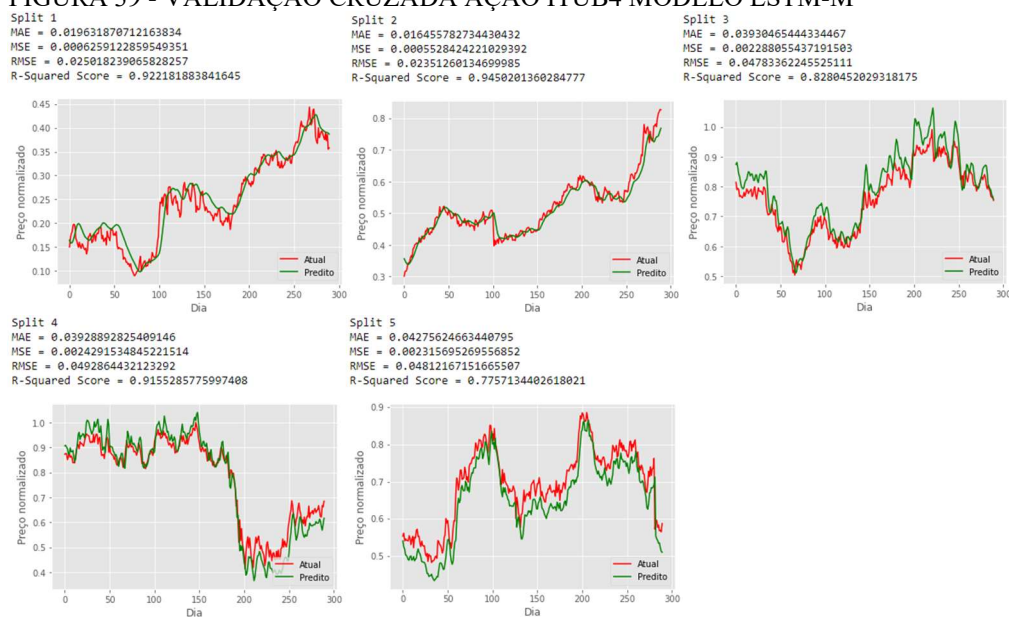
Fonte: O autor, 2022.

FIGURA 38 - VALIDAÇÃO CRUZADA AÇÃO ITUB4 MODELO LSTM-U



Fonte: O autor, 2022.

FIGURA 39 - VALIDAÇÃO CRUZADA AÇÃO ITUB4 MODELO LSTM-M



Fonte: O autor, 2022.

Conforme, pode se observar com base na Tabela 10 e da visualização gráfica para a ação ITUB4, verifica-se que os modelos de redes neurais com maior assertividade foram os modelos de LSTM-U e LSTM-M com R2 de 0.90481 e 0.87730, respectivamente, ou seja, conseguiram explicar cerca de 90,48% e 87,32% do valor previsto. A memória curta e longa destas redes neurais explica e justifica este maior poder preditivo.

4.3.4 Validação Cruzada ABEV3

Com base nos resultados dos hiper parâmetros obtidos através da curva da função perda do gradiente descendente da ação da ABEV3 apresentado na seção 4.2.4, se realizou o treinamento para os modelos de RNN-U, RNN-M, LSTM-M e LSTM-U utilizando método de validação cruzada com 5 splits para a ação da ITUB4, cujos resultados obtidos são reportados na Tabela 11.

TABELA 11 - RESULTADOS AÇÃO ABEV3 – VALIDAÇÃO CRUZADA

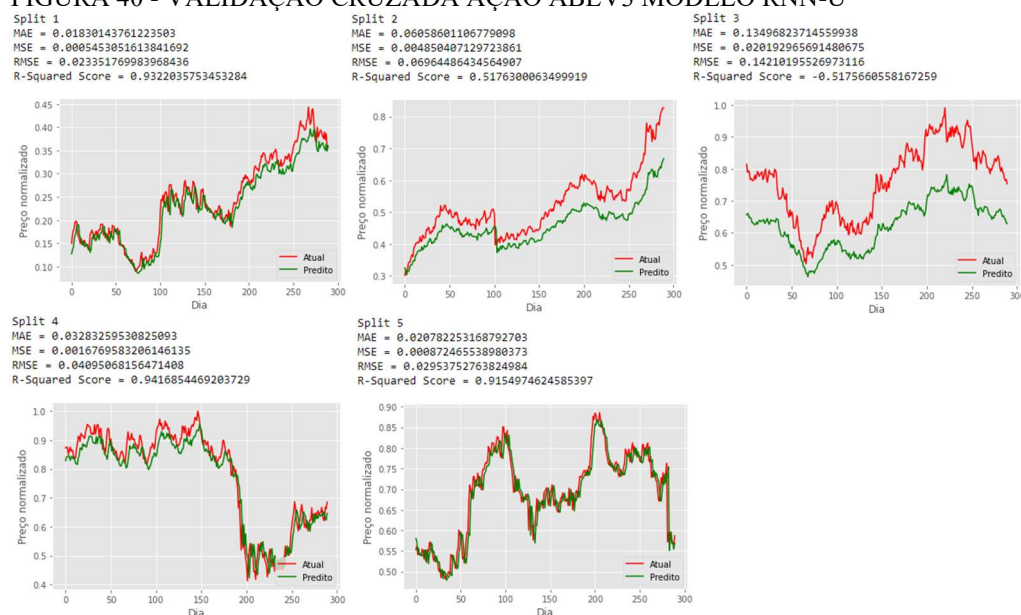
ABEV3	RNN-U	RNN-M	LSTM-U	LSTM-M
MAE	0.03172	0.05753	0.02633	0.05088
MSE	0.00183	0.00484	0.00123	0.00393
RMSE	0.03799	0.06499	0.03171	0.05729
R2_Score	0.42546	0.64177	0.63936	0.72352

Fonte: O autor, 2022.

Pode-se verificar, por meio da validação da métrica do coeficiente de determinação R2, de que cada modelo, tem sua particularidade de desempenho, destarte pode-se verificar para o caso da ABEV3 que os modelos com melhor performance nas métricas usadas são do tipo LSTM-U e LSTM-M.

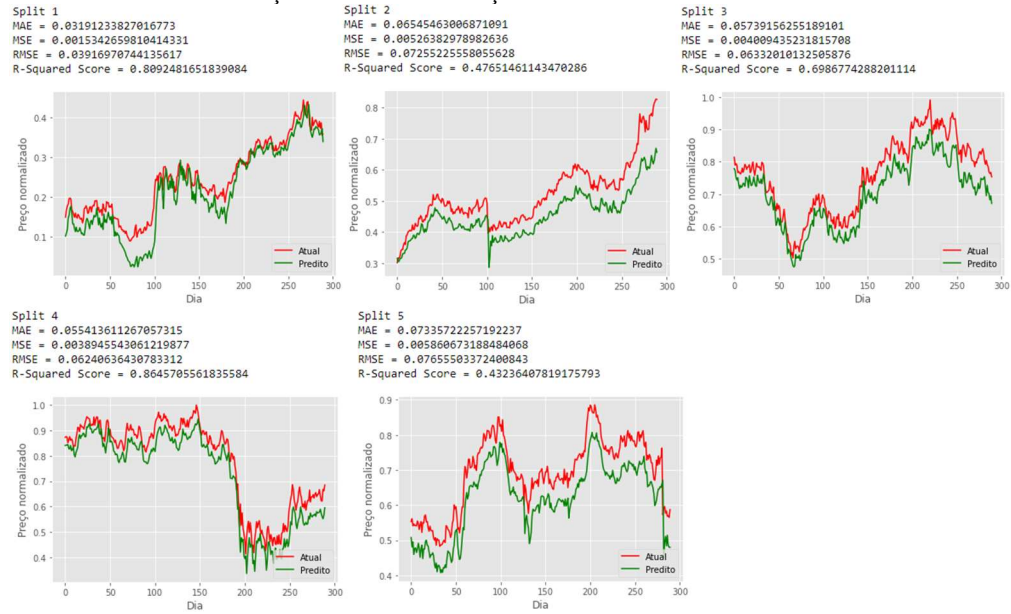
Os resultados dos splits para cada modelo realizado pode ser observado na sequência:

FIGURA 40 - VALIDAÇÃO CRUZADA AÇÃO ABEV3 MODELO RNN-U



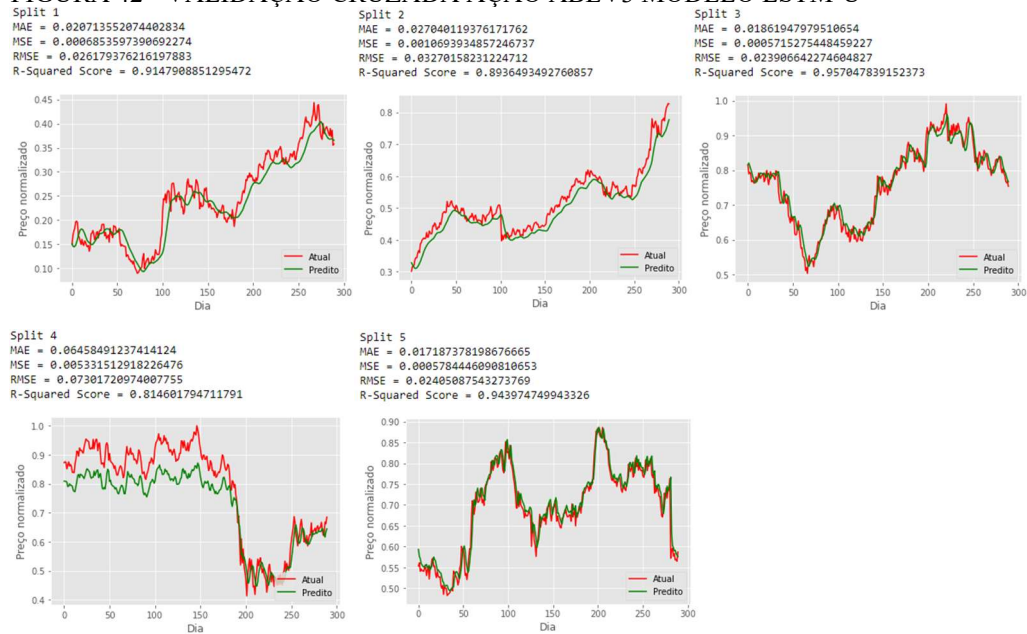
Fonte: O autor, 2022.

FIGURA 41 - VALIDAÇÃO CRUZADA AÇÃO ABEV3 MODELO RNN-M



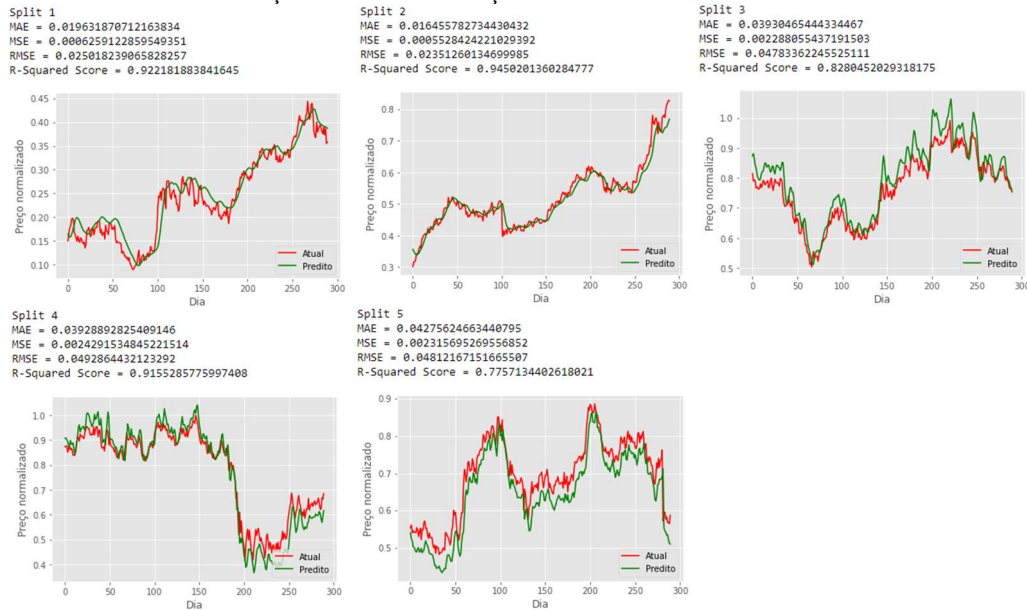
Fonte: O autor, 2022.

FIGURA 42 - VALIDAÇÃO CRUZADA AÇÃO ABEV3 MODELO LSTM-U



Fonte: O autor, 2022.

FIGURA 43 - VALIDAÇÃO CRUZADA AÇÃO ABEV3 MODELO LSTM-M



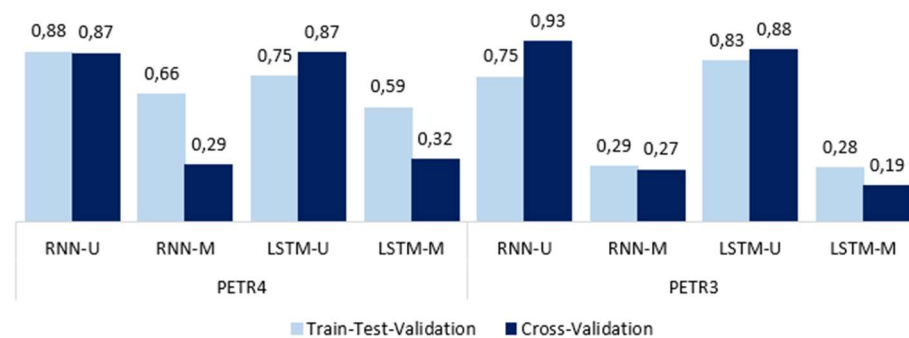
Fonte: O autor, 2022.

Conforme, pode se observar da Tabela 11 e da visualização gráfica para a ação ABEV3, verifica-se que os modelos de redes neurais com maior assertividade foram os modelos de RNN-M e LSTM-M com R2 de, respectivamente 0,64177 e 0,72352.

4.4 COMPARATIVO DOS RESULTADOS

Com o intuito de analisar e observar o comportamento das variações entre os modelos de validação cruzada e modelo de Treino-Teste-Validação que foi o primeiro utilizado no momento da definição dos hiper parâmetros, se coletou as informações do coeficiente de determinação R2 dos modelos aplicando ambas as metodologias, Validação Cruzada e Treino-Teste-Validação.

FIGURA 44 - COMPARATIVO DE R2 ENTRE VALIDAÇÃO CRUZADA E TREINO-TESTE-VALIDAÇÃO: PETR3 E PETR4.

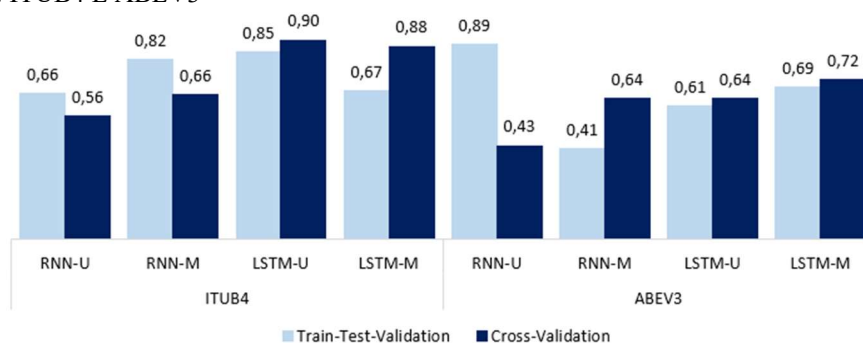


Fonte: O autor, 2022.

Para as ações da PETR3 e PETR4, observa-se uma variação maior entre os dois métodos de aprendizado, como exemplo no modelo LSTM-U, o método de validação cruzada apresentou um melhor retorno comparado com o primeiro método, tal variação pode ser explicada pois na validação cruzada se utiliza do espaçamento do tempo em diversos períodos e se faz a validação e treinamento para cada etapa, o que nesse caso específico pode ter ocorrido, foi de que no modelo Treino-Teste-Validação, por se utilizar períodos fixos de amostragem, treinamento e validação, pode ter refletido os impactos das ações do governo, desde que antes do ano de 2016, os preços da Petrobras eram controlados pelo governo, podendo ter causado assim, um aprendizado de maneira incorreta pela Rede Neural.

Ao se analisar as ações do ITUB4 e ABEV3, observa-se que nesse caso os modelos de LSTM-U e LSTM-M, apresentaram um coeficiente de determinação maior quando aplicado o método de validação cruzada.

FIGURA 45 - COMPARATIVO DE R2 ENTRE VALIDAÇÃO CRUZADA E TREINO-TESTE-VALIDAÇÃO: ITUB4 E ABEV3



Fonte: O autor, 2022.

Já para o modelo de RNN-U para ambas as ações o modelo de validação cruzada se mostrou com R2 inferior, tal comportamento pode ser refletido devido ao modelo de RNN-U utilizar-se de memória curta e ser univariado e por conta disso, conforme o teste de validação cruzada retornou para essas ações ambas rejeitavam a hipótese de Random Walk para períodos diários, semanal e mensal e apresentavam uma autocorrelação negativa.

5 CONSIDERAÇÕES FINAIS

O objetivo desta dissertação foi propor e aplicar uma metodologia para avaliar o poder preditivo de modelos de redes neurais RNN-U, RNN-M, LSTM-U e LSTM-M, por meio do teste da razão da variância para a hipótese de Random Walk, da técnica de validação cruzada para séries temporais e de um estudo de caso de ações do mercado de ações brasileiro: PETR3, PETR4, ITUB4 e ABEV3.

Foi proposto uma metodologia para se realizar o treinamento dos modelos preditivos com a visão de se obter a melhor probabilidade de valorização e assertividade das ações PETR3, PETR4, ITUB4 e ABEV3.

Dos resultados do teste da razão da variância para Random Walk pode-se inferir que os retornos das ações da PETR3 e PETR4 possuem uma autocorrelação serial negativa de primeira ordem; enquanto os retornos das ações da ABEV3 e ITUB4, possuem uma autocorrelação serial negativa também para ordens superiores.

Com efeito, no teste de validação cruzada, os modelos de redes neurais LSTM, como possuem memória curta e memória longa, reportaram maior poder preditivo em termos das várias métricas usadas, do que os modelos RNN-U e RNN-M, no caso das ações da ABEV3 e ITUB4. A memória curta permite explorar a autocorrelação negativa de primeira ordem, enquanto a memória longa permite explorar as autocorrelações de ordem superiores. Por último, do ponto de vista teórico, o que explica e justifica a presença de autocorrelação serial negativa em ações da PETR3, PETR4, ABEV3 e ITUB4, são os limites para as operações de arbitragem de traders racionais e os vieses comportamentais de traders irracionais e/ou noise traders.

Para futuros trabalhos de avaliação de poder preditivo de redes neurais LSTM sobre o mercado de ações brasileiro, propõe-se aumentar a quantidade de ações no estudo de caso e usar outras variáveis publicamente disponíveis, além dos preços históricos da ação, como variáveis fundamentalistas: resultados de balanço, taxa de câmbio, taxa de juros, preço do barril do petróleo, e índices de ações nacionais e internacionais como S&P500, Dow Jones industrial, Nasdaq, Bovespa, dentre outros.

Ainda para trabalhos futuros, propõe-se aplicar a técnica de bootstrap ou reamostragem para dados dependentes em combinação com a técnica de validação cruzada para series temporais, para testar a hipótese de que modelos de redes neurais com memória curta e longa RNN LSTM possuem maior poder preditivo de preços e retornos de ações do que modelos de redes neurais com memória curta RNN, ao nível de significância estatística de 1%, 5% e 10%.

REFERÊNCIAS

- ADHIKARI, R.; AGRAWAL, R. K. **An introductory study on time series modeling and forecasting**. 2013. Disponível em: <https://arxiv.org/abs/1302.6613>. Acesso em: 09 de setembro de 2021.
- ASHWOOD, A. J. **Portfolio Selection Using Artificial Intelligence**. 2014. Thesis (Doctor of Business Administration) - Queensland University of Technology, Queensland, 2013.
- BACHELIER, L. Théorie de la spéculation. **Annales scientifiques de l'É.N.S.** 3e série, v.17, p. 21-86, 1900. Disponível em: http://www.numdam.org/item?id=ASENS_1900_3_17__21_0. Acesso em: 09 de setembro de 2021.
- BAI, S.; KOLTER, J. Z.; KOLTUN, V. **An empirical evaluation of generic convolutional and recurrent networks for sequence modeling**. 2018. Disponível em: <https://arxiv.org/abs/1803.01271>. Acesso em: 10 de novembro de 2021.
- BAO, W.; YUE, J.; RAO, Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. **PLOS ONE**, v. 12, n. 7, p. e0180944, 14 jul. 2017. Disponível em: <https://doi.org/10.1371/journal.pone.0180944>. Acesso em 14 de nov. 2021.
- BARBERIS, N.; THALER, R. A Survey of Behavioral Finance. **National Bureau of Economic Research**, 2002. Disponível em: https://www.nber.org/system/files/working_papers/w9222/w9222.pdf. Acesso em 14 de nov. de 2021.
- C. M. Mesquita, R. Oliveira, A. C. M. Pereira. **Utilização de uma rede neural LSTM e testes da razão da variância para previsões em séries de ativos da Bovespa**. Universidade Federal de Minas Gerais (UFMG), 2019.
- CEN, Z.; WANG, J. Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer. **Energy**, v. 169, p. 160–171, fev. 2019.
- CHO, K.; MERRIËNBOER, B. V.; DANAU, D. B.; BENGIO, Y. **On the properties of neural machine translation: Encoder-decoder approaches**, 2014. Disponível em: <https://aclanthology.org/W14-4012.pdf>. Acesso em: 09 de setembro de 2021
- CHUNG, J.; GULCEHRE, C.; CHO, K.; BENGIO, Y. **Empirical evaluation of gated recurrent neural networks on sequence modeling**. In: NIPS 2014 WORKSHOP ON DEEP LEARNING, 2014.
- CHUNG, J.; GULCEHRE, C.; CHO, K.; BENGIO, Y. **Gated feedback recurrent neural networks**, 2015. Disponível em: <https://arxiv.org/pdf/1502.02367.pdf>. Acesso em: 20 de agosto de 2021.
- COLAH. **Understanding LSTM Networks**. 2015. Disponível em: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acesso em: 20 de agosto de 2021.

CONT, R. Empirical properties of asset returns: stylized facts and statistical issues. **Quantitative Finance**, v. 1, n. 2, p. 223–236, fev. 2001.

FAMA, E. F. Efficient Capital Markets: a Review of Theory and Empirical Work. **The Journal of Finance**, v. 25, n. 2, p. 383–417, 1970.

FAMA, E. F. The Behavior of Stock-Market Prices. *The Journal of Business*, v. 38, n. 1, p. 34–105, jan. 1965.

FISCHER, T.; KRAUSS, C. Deep learning with long short-term memory networks for financial market predictions. **European Journal of Operational Research**, v. 270, n. 2, p. 654–669, out. 2018.

FISCHLER, M. A.; FIRSCHEIN, O. **Intelligence: the eye, the brain, and the computer**. Reading, Mass.: Addison-Wesley, 1987.

GER, F. **Long short-term memory in recurrent neural networks**. 2001 Thesis (PhD in physics), Beuth Hochschule für Technik Berlin, 2001.

GER, F. A.; SCHMIDHUBER, J. Recurrent nets that time and count. In Proceedings of the IEEE-INNS-ENNS. **International Joint Conference on Neural Networks**. Piscataway, NJ: IEEE, v. 3, p.189 – 194, 2000.

GER, F. A.; SCHMIDHUBER, E. LSTM recurrent networks learn simple context-free and context-sensitive languages. **IEEE Transactions on Neural Networks**, v. 12, n. 6, p. 1333–1340, 2001.

GER, F. A.; SCHRAUDOLPH, N. N.; SCHMIDHUBER, J. Learning precise timing with LSTM recurrent networks. **Journal of Machine Learning Research**, n. 3, p. 115–143, 2002.

GER, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to Forget: Continual Prediction with LSTM. **Neural Computation**, v. 12, n. 10, p. 2451–2471, out. 2000.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; **“Deep learning”** - vol. 1., Massachusetts, USA: MIT press, 2016.

GRAVES, A. **Supervised sequence labelling with recurrent neural networks**. Ph.D Thesis. 2008. Slides. Disponível em: <https://pdfs.semanticscholar.org/a97b/5db17acc731ef67321832dbbaf5766153135.pdf>. Acesso em: 01 out. 2021.

GRAVES, A. **Supervised sequence labelling with recurrent neural networks**. Berlin; Heidelberg: Springer, Copyright, 2012.

GRAVES, A. **Generating sequences with recurrent neural networks**. 2014. Disponível: [https://arxiv.org/pdf/1308.0850.pdf#:~:text=Recurrent%20neural%20networks%20\(RNNs\)%20are,and%20predicting%20what%20comes%20next](https://arxiv.org/pdf/1308.0850.pdf#:~:text=Recurrent%20neural%20networks%20(RNNs)%20are,and%20predicting%20what%20comes%20next). Acesso em: 21 de agosto de 2021.

GRAVES, A.; SCHMIDHUBER, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. **Neural Networks**, v. 18, n. 5-6, p. 602–610, jul. 2005.

GRAVES, A., FERNÁNDEZ, S.; SCHMIDHUBER, J. **Multi-dimensional recurrent neural networks**, 2007. Disponível em: <https://arxiv.org/pdf/0705.2011.pdf>. Acesso em: 11 de setembro de 2021.

GRAVES, A., JAITLEY, N.; MOHAMED, A. R. Hybrid speech recognition with Deep Bidirectional LSTM. **IEEE Workshop on Automatic Speech Recognition and Understanding**, p. 273–278, 2013.

GRAVES, A., WAYNE, G.; DANIHELKA, I. **Neural Turing machines**, 2014. Disponível em: <https://arxiv.org/pdf/1410.5401.pdf>. Acesso em: 12 de setembro de 2021.

GRAVES, A. et al. Hybrid computing using a neural network with dynamic external memory. **Nature**, v. 538, n. 7626, p. 471–476, out. 2016.

GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. LSTM: A search space odyssey. **IEEE Transactions on Neural Networks and Learning Systems**, n. 28, v.10, p. 2222–2232, 2016.

HAYKIN, S. **Redes neurais: princípios e práticas**. 2ª Ed., Porto Alegre: Editora Artmed, 2001.

HAYKIN, S. **Neural Networks and Learning Machines**. Third Edit., New Jersey, USA: Pearson Prentice Hall, 2008.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, nov. 1997. doi:10.1162/neco.1997.9.8.1735.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the National Academy of Sciences**, v. 79, n. 8, p. 2554–2558, 1 abr. 1982.

HYNDMAN, R. J.; ATHANASOPOULOS, G. **Forecasting: Principles and Practice**. 2. ed. Heathmont, Vic.: Otexts, 2018.

KAHNEMAN, D. Maps of Bounded Rationality: Psychology for Behavioral Economics. **American Economic Review**, v. 93, n. 5, p. 1449–1475, nov. 2003.

KAHNEMAN, D. **Rápido e Devagar: duas formas de pensar**. Tradução: Cássio de Abrantes Leite. Rio de Janeiro: Objetiva, 2012.608p.

KOHAVI, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. **Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence**, v. 2, p. 1137-1143, 1995.

KRAUSS, C., DO, X. A.; HUCK, N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. **European Journal of Operational Research**, n. 259, v. 2, p. 689–702, 2017.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. **Communications of the ACM**, v. 60, n. 6, p. 84–90, 2012.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep Learning. **Nature**, v. 521, n. 7553, p. 436–444, maio 2015.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. **Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)**, p. 253–256, 2010.

LEA, C. et. al. Temporal convolutional networks for action segmentation and detection. **IEEE International Conference on Computer Vision (ICCV)**. 2017. Disponível em: <https://arxiv.org/pdf/1611.05267.pdf>. Acesso em: 12 de setembro de 2021.

LENT, R. et al. How many neurons do you have? Some dogmas of quantitative neuroscience under revision. **European Journal of Neuroscience**, v. 35, n. 1, p. 1–9, 13 dez. 2011.

LI, F. et al. Recognizing irregular entities in biomedical text via deep neural networks. **Pattern Recognition Letters**, v. 105, p. 105–113, abr. 2018.

LI, S., LI, W., COOK, C., ZHU, C.; GAO, Y. Independently recurrent neural network: Building a longer and deeper RNN. **IEEE/CVF Conference on Computer Vision and Pattern Recognition**, p. 5457–5466, 2018.

LIPTON, Z. C., BERKOWITZ, J.; ELKAN, C. **A critical review of recurrent neural networks for sequence learning**. 2015. Disponível em: <https://arxiv.org/pdf/1506.00019.pdf>. Acesso em: 20 de setembro de 2021.

LIU, Y. Novel volatility forecasting using deep learning long short-term memory recurrent neural networks. **Expert Systems with Applications**, v. 132, p. 99–109, 2019.

LO, A. W.; MACKINLAY, A. C. Data-snooping biases in tests of financial asset pricing models. **Review of Financial Studies** v.3, n. 3 p. 431–67, 1990.

LO, A. W.; MACKINLAY, A. C. Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test. **Review of Financial Studies**, v. 1, n. 1, p. 41–66, jan. 1988.

MACCULLOCH, W.S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bull. Math. Biophys**, v. 5, p. 115–133, 1943.

MALKIEL, B. G. The Efficient Market Hypothesis and Its Critics. **Journal of Economic Perspectives**, v. 17, n. 1, p. 59–82, 2003.

MALKIEL, B. G. **A Random Walk Down Wall Street (Eleventh E)**. New York: W. W. Norton & Company, 2015.

MARTINS, G. DE A.; THEÓPHILO, C. R. **Metodologia da investigação científica para ciências sociais aplicadas**. 2ª edição, São Paulo - SP: Editora Atlas, 2009.

MCDERMOTT, J. Principles of artificial intelligence. **Artificial Intelligence**, v. 15, n. 1-2, p. 127–131, nov. 1980.

MEENAL, R.; SELVAKUMAR, A. I. Assessment of SVM, empirical and ANN based solar radiation prediction models with most influencing input parameters. **Renewable Energy**, v. 121, p. 324–343, jun. 2018.

MORITZ, B.; ZIMMERMANN, T. **Tree-based conditional portfolio sorts: The relation between past and future stock returns**. Working Paper, Ludwig Maximilian University of Munich, 2016.

NELSON, D. M. Q., PEREIRA, A. C. M.; OLIVEIRA, R. A. DE. Stock market's price movement prediction with LSTM neural networks. **Proceedings of the International Joint Conference on Neural Networks**, p. 1419–1426, 2017.

NEWELL, A.; SIMON, H. A. **Human Problem Solving**. Englewood Cliffs, NJ: Prentice-Hall, 1972.

NG, ANDREW. **Neural networks and deep learning**. Coursera. 2017. Disponível em: <https://www.coursera.org/learn/neuralnetworks-deep-learning>. Acesso em 14 de novembro de 2021.

OORD, A. V. D. et al. **Wavenet: A generative model for raw audio**. 2016. Disponível em: <https://arxiv.org/pdf/1609.03499.pdf>. Acesso em: 04 de dezembro de 2021.

OORD, A. van den et al. **Conditional image generation with pixelCNN decoders**, 2016. Disponível em: <https://arxiv.org/abs/1606.05328>. Acesso em: 04 de dezembro de 2021.

RABELO JUNIOR, T. S.; IKEDA, R. H. Mercados eficientes e arbitragem: um estudo sob o enfoque das finanças comportamentais. **Revista Contabilidade & Finanças**, v. 15, n. 34, p. 97–107, abr. 2004.

ROBERTS, H. V. **Statistical versus clinical prediction of the stock market**. University of Chicago, Chicago, 1967.

RODRIGUES, F., MARKOU, I., PEREIRA, F.C. Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. **Information Fusion**, v. 49, p. 120 - 129, 2019.

ROSS, S. A. The arbitrage theory of capital asset pricing. **Journal of Economic Theory**, v. 13, n. 3, p. 341–360, dez. 1976.

RUBIM, F. H. **Deep learning no mercado acionário brasileiro: fatores que possibilitam previsões consistentes para a tomada de decisão em condições de risco**. 2020. Dissertação (Mestrado em Administração) - Universidade de São Paulo, São Paulo 2019.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations of back-propagation errors. **Nature London**, v. 323, p. 533-536, 1986.

SAGHEER, A.; KOTB, M. Time series forecasting of petroleum production using deep LSTM recurrent networks. **Neurocomputing**, v. 323, p. 203–213, jan. 2019. Disponível em: <https://doi.org/10.1016/j.neucom.2018.09.082>. Acesso em 01 out. 2021.

SAGE, A. P.. **Concise encyclopedia of information processing in systems & [and] organizations**. Oxford: Pergamon Press, 1990.

SANG, C.; DI PIERRO, M. Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network. **The Journal of Finance and Data Science**, v. 5, n. 1, p. 1–11, mar. 2019.

STERNBERG, R. J.; STERNBERG, K. **Cognitive psychology**. 6. ed. [s.l.] Boston, Ma, Usa Cengage Learning Canada, 2017.

TVERSKY, A.; KAHNEMAN, D. Judgment Under Uncertainty: Heuristics and Biases. **Science**, v. 185, n. 4157, p. 1124–1131, 27 set. 1974.

WERNER F. M. DE B.; THALER, R. Does the Stock Market Overreact?. **The Journal of The American Finance Association**. v. 40, n. 3. P. 793-805. 1985. Disponível em: <https://doi.org/10.1111/j.1540-6261.1985.tb05004.x>. Acesso em 20 de nov. de 2021.

WONG, B. K.; SELVI, Y. Neural network applications in finance: A review and analysis of literature (1990–1996). **Information & Management**, v. 34, n. 3, p. 129–139, out. 1998. Disponível em: [https://doi.org/10.1016/S0378-7206\(98\)00050-0](https://doi.org/10.1016/S0378-7206(98)00050-0). Acesso em 16 de nov. de 2021.

WU, Y. X.; WU, Q. B.; ZHU, J. Q. Improved eemd-based crude oil price forecasting using LSTM networks. **Physica A: Statistical Mechanics and its Applications**, v. 516, p. 114–124, 2019.

YAN, H.; OUYANG, H. Financial time series prediction based on deep learning. **Wireless Personal Communications**, v. 102, p. 683–700, 2018.

YANG, J.; GUO, Y.; ZHAO, W. Long short-term memory neural network-based fault detection and isolation for electro-mechanical actuators. **Neurocomputing**, v. 360, p. 85-96, 2019.

ZHANG, J., CUI, S., XU, Y., LI, Q.; LI, T. A novel data-driven stock price trend prediction system. **Expert Systems with Applications**, v. 97, p. 60–69, 2018.

ANEXO A – ATIVOS QUE COMPÕEM A CARTEIRA IBOV DE SET A DEZ 2021

Código	Ação	Tipo		Qtde. Teórica	Part. (%)
ABEV3	AMBEV S/A	ON		4.358.814.864	3,135
ALPA4	ALPARGATAS	PN	N1	167.963.777	0,414
AMER3	AMERICANAS	ON	NM	546.064.815	0,993
ASAI3	ASSAI	ON	NM	794.116.430	0,626
AZUL4	AZUL	PN	N2	327.741.172	0,522
B3SA3	B3	ON	NM	6.079.530.858	3,71
BBAS3	BRASIL	ON	NM	1.420.530.937	1,797
BBDC3	BRADESCO	ON	EJ N1	1.378.842.305	1,149
BBDC4	BRADESCO	PN	EJ N1	4.691.427.537	4,518
BBSE3	BBSEGURIDADE	ON	NM	671.629.692	0,56
BEEF3	MINERVA	ON	NM	242.962.308	0,082
BIDI11	BANCO INTER	UNT	N2	477.617.336	1,322
BIDI4	BANCO INTER	PN	N2	186.303.277	0,164
BPAC11	BTGP BANCO	UNT	N2	1.301.655.996	1,551
BPAN4	BANCO PAN	PN	N1	341.124.068	0,257
BRAP4	BRADESPAR	PN	N1	222.604.507	0,605
BRDT3	PETROBRAS BR	ON	NM	1.164.602.565	1,296
BRFS3	BRF SA	ON	NM	811.759.800	0,838
BRKM5	BRASKEM	PNA	N1	264.642.296	0,768
BRML3	BR MALLS PAR	ON	NM	843.728.684	0,317
CASH3	MELIUZ	ON	NM	90.581.467	0,139
CCRO3	CCR SA	ON	NM	1.115.693.556	0,578
CIEL3	CIELO	ON	NM	1.144.359.228	0,129
CMIG4	CEMIG	PN	N1	1.114.214.662	0,659
COGN3	COGNA ON	ON	NM	1.847.994.874	0,244
CPFE3	CPFL ENERGIA	ON	ED NM	187.732.538	0,223
CPLE6	COPEL	PNB	N1	1.563.365.506	0,461
CRFB3	CARREFOUR BR	ON	NM	410.988.561	0,327
CSAN3	COSAN	ON	NM	1.169.237.744	1,124
CSNA3	SID NACIONAL	ON		642.398.790	0,971
CVCB3	CVC BRASIL	ON	NM	224.571.661	0,196
CYRE3	CYRELA REALT	ON	NM	281.609.283	0,231
DXCO3	DEXCO	ON	NM	272.062.873	0,239
ECOR3	ECORODOVIAS	ON	NM	339.237.914	0,144
EGIE3	ENGIE BRASIL	ON	NM	255.217.329	0,436
ELET3	ELETROBRAS	ON	N1	358.028.908	0,578
ELET6	ELETROBRAS	PNB	N1	242.987.127	0,399
EMBR3	EMBRAER	ON	NM	734.558.205	0,694
ENBR3	ENERGIAS BR	ON	NM	231.489.471	0,184
ENEV3	ENEVA	ON	NM	1.261.568.398	0,87
ENGI11	ENERGISA	UNT	N2	248.542.219	0,473
EQTL3	EQUATORIAL	ON	NM	1.010.511.085	1,126
EZTC3	EZTEC	ON	NM	101.618.236	0,115
FLRY3	FLEURY	ON	NM	303.005.900	0,313
GGBR4	GERDAU	PN	N1	1.097.534.498	1,305

GNDI3	INTERMEDICA	ON	NM	592.114.608	2,101
GOAU4	GERDAU MET	PN	N1	698.275.321	0,377
GOLL4	GOL	PN	N2	167.095.214	0,138
HAPV3	HAPVIDA	ON	NM	1.306.939.210	0,845
HGTX3	CIA HERING	ON	NM	117.608.748	0,186
HYPE3	HYPERA	ON	NM	411.803.359	0,617
IGTA3	IGUATEMI	ON	NM	85.748.277	0,121
IRBR3	IRBBRASIL RE	ON	NM	1.255.286.531	0,281
ITSA4	ITAUSA	PN	ED N1	4.515.559.175	2,206
ITUB4	ITAUNIBANCO	PN	ED N1	4.780.002.924	6,181
JBSS3	JBS	ON	NM	1.346.828.080	1,819
JHSF3	JHSF PART	ON	NM	305.915.142	0,083
KLBN11	KLABIN S/A	UNT	N2	812.994.397	0,909
LAME4	LOJAS AMERIC	PN	N1	853.524.256	0,223
LCAM3	LOCAMERICA	ON	NM	324.522.975	0,344
LREN3	LOJAS RENNER	ON	NM	889.725.318	1,434
LWSA3	LOCWEB	ON	NM	431.783.548	0,45
MGLU3	MAGAZ LUIZA	ON	NM	2.820.185.158	2,315
MRFG3	MARFRIG	ON	NM	348.234.011	0,323
MRVE3	MRV	ON	NM	294.643.879	0,169
MULT3	MULTIPLAN	ON	N2	273.506.645	0,24
NTCO3	GRUPO NATURA	ON	NM	843.870.809	1,859
PCAR3	P.ACUCAR-CBD	ON	NM	156.946.474	0,185
PETR3	PETROBRAS	ON	N2	3.426.385.188	4,063
PETR4	PETROBRAS	PN	N2	4.566.511.125	5,222
PETZ3	PETZ	ON	NM	258.407.844	0,299
PRI03	PETRORIO	ON	NM	839.159.130	0,662
QUAL3	QUALICORP	ON	NM	283.670.037	0,277
RADL3	RAIADROGASIL	ON	NM	1.071.076.905	1,214
RAIL3	RUMO S.A.	ON	NM	1.216.056.103	1,003
RDOR3	REDE D OR	ON	NM	320.214.380	0,953
RENT3	LOCALIZA	ON	NM	593.944.309	1,413
SANB11	SANTANDER BR	UNT		362.703.399	0,613
SBSP3	SABESP	ON	NM	340.001.934	0,526
SULA11	SUL AMERICA	UNT	N2	284.771.028	0,348
SUZB3	SUZANO S.A.	ON	NM	726.779.281	1,948
TAE11	TAESA	UNT	N2	218.568.234	0,356
TIMS3	TIM	ON	NM	808.619.532	0,418
TOTS3	TOTVS	ON	NM	480.960.677	0,818
UGPA3	ULTRAPAR	ON	NM	1.084.337.016	0,695
USIM5	USIMINAS	PNA	EDJ N1	514.680.651	0,364
VALE3	VALE	ON	NM	3.380.233.503	14,478
VIIA3	VIA	ON	NM	1.595.895.585	0,667
VIVT3	TELEF BRASIL	ON		443.933.052	0,812
WEGE3	WEG	ON	NM	1.484.859.030	2,338
YDUQ3	YDUQS PART	ON	NM	300.833.122	0,325
Total				90.778.488.381	100

ANEXO B – SCRIPT R TESTE RAZÃO DA VARIÂNCIA

```
#1.POPRIEDADES ESTATÍSTICAS OU FATOS ESTILIZADOS DE AÇÕES DA PETR4
rm(list=ls())
library(quantmod)
library(ggplot2)
#1.1.PREÇO E RETORNOS AÇÕES Petrobras.
pbr <- getSymbols("PETR4.SA", src = "yahoo", from = "2012-01-01", to = "2021-12-31", auto.assign = FALSE)
pbrna<-na.omit(pbr)
summary(pbrna)

ggplot(pbrna, aes(x = index(pbrna), y = pbrna[,6])) + geom_line(color = "darkblue") +
  ggtitle("PETROBRÁS") +
  xlab("Datas") + ylab("PREÇO DA AÇÃO") + theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_date(date_labels = "%b %y", date_breaks = "30 months")

retornopbr<- diff(log(pbrna[,6]))
retornopbrna<-na.omit(retornopbr)
plot(retornopbrna)
summary(retornopbrna)

#1.2.FUNÇÃO AUTO-CORRELAÇÃO LINEAR E NÃO LINEAR DE RETORNOS
#Teste de Auto-Correlação de Retornos em Logs e Absolutos.
library(astsa)
acfl(retornopbrna)
absr<-Mod(ts(retornopbrna))
plot(absr)
library(astsa)
acfl(absr)
#Teste de Raiz Unitária de Retornos.
library("tseries")
TRUr<-adf.test(retornopbrna)
TRUR<-adf.test(absr)

#1.3.HISTOGRAMA E TESTE DE NORMALIDADE DE RETORNOS EM LOGS
histograma<-hist(retornopbrna$PETR4.SA.Adjusted, breaks = 200, col="grey", xlim=c(-
0.15,0.15),main="Petrobras Returns Histogram",xlab="Returns");
xfit = seq(min(retornopbrna$PETR4.SA.Adjusted), max(retornopbrna$PETR4.SA.Adjusted), length = 5000)
yfit = dnorm(xfit, mean = mean(retornopbrna$PETR4.SA.Adjusted), sd = sd(retornopbrna$PETR4.SA.Adjusted))
yfit = yfit * diff(histograma$mids[1:2]) * length(retornopbrna$PETR4.SA.Adjusted)
lines(xfit, yfit, col = "blue", lwd =0.1)
legend(x="topright", #posicao da legenda
       c("Returns","Normal"), #nomes da legenda
       col=c("grey","blue"), #cores
       lty=c(1,1,3), #estilo da linha
       lwd=c(2,2,4)) #grossura das linhas

#Teste de Normalidade de Jarque-Bera
library(tseries)
jarque.bera.test(retornopbrna)

#1.4.TESTE DA RAZÃO DE VARIANCIA DE LO, A. W., and A. C. MACKINLAY (1988):
library(vrtest)
r <- retornopbrna$PETR4.SA.Adjusted
kvec <- c(2)
Lo.Mac(r,kvec)
Wald(r,kvec)
```

```

kvec <- c(5)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(10)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(15)
Lo.Mac(r,kvec)
Wald(r,kvec)

```

#2.POPRIEDADES ESTATÍSTICAS OU FATOS ESTILIZADOS DE AÇÕES DA PETR3

```

rm(list=ls())
library(quantmod)
library(ggplot2)
#2.1.PREÇO E RETORNOS AÇÕES Petrobras.
pbr <- getSymbols("PETR3.SA", src = "yahoo", from = "2012-01-01", to = "2021-12-31", auto.assign = FALSE)
pbrna<-na.omit(pbr)
summary(pbrna)

```

```

ggplot(pbrna, aes(x = index(pbrna), y = pbrna[,6])) + geom_line(color = "darkblue") +
  ggtitle("PETROBRÁS") +
  xlab("Datas") + ylab("PREÇO DA AÇÃO") + theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_date(date_labels = "%b %y", date_breaks = "30 months")

```

```

retornopbr<- diff(log(pbrna[,6]))
retornopbrna<-na.omit(retornopbr)
plot(retornopbrna)
summary(retornopbrna)

```

#2.2.FUNÇÃO AUTO-CORRELAÇÃO LINEAR E NÃO LINEAR DE RETORNOS

#Teste de Auto-Correlação de Retornos em Logs e Absolutos.

```

library(astsa)
acfl(retornopbrna)
absr<-Mod(ts(retornopbrna))
plot(absr)
library(astsa)
acfl(absr)
#Teste de Raiz Unitária de Retornos.
library("tseries")
TRUr<-adf.test(retornopbrna)
TRUR<-adf.test(absr)

```

#2.3.HISTOGRAMA E TESTE DE NORMALIDADE DE RETORNOS EM LOGS

```

histograma<-hist(retornopbrna$PETR3.SA.Adjusted, breaks = 200, col="grey", xlim=c(-
0.15,0.15),main="Petrobras Returns Histogram ",xlab="Returns");
xfit = seq(min(retornopbrna$PETR3.SA.Adjusted), max(retornopbrna$PETR3.SA.Adjusted), length = 5000)
yfit = dnorm(xfit, mean = mean(retornopbrna$PETR3.SA.Adjusted), sd = sd(retornopbrna$PETR3.SA.Adjusted))
yfit = yfit * diff(histograma$mids[1:2]) * length(retornopbrna$PETR3.SA.Adjusted)
lines(xfit, yfit, col = "blue", lwd=0.1)
legend(x="topright", #posicao da legenda
      c("Returns","Normal"), #nomes da legenda
      col=c("grey","blue"), #cores
      lty=c(1,1,3), #estilo da linha
      lwd=c(2,2,4)) #grossura das linhas

```

#Teste de Normalidade de Jarque-Bera

```

library(tseries)
jarque.bera.test(retornopbrna)

```

#2.4.TESTE DA RAZÃO DE VARIANCIA DE LO, A. W., and A. C. MACKINLAY (1988):

```
library(vrtest)
r <- retornopbrna$PETR3.SA.Adjusted
kvec <- c(2)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(5)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(10)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(15)
Lo.Mac(r,kvec)
Wald(r,kvec)
```

#3.POPRIEDADES ESTATÍSTICAS OU FATOS ESTILIZADOS DE AÇÕES DA ABEV3

```
rm(list=ls())
library(quantmod)
library(ggplot2)
#3.1.PREÇO E RETORNOS AÇÕES AMBEV.
pbr <- getSymbols("ABEV3.SA", src = "yahoo", from = "2012-01-01", to = "2021-12-31", auto.assign = FALSE)
pbrna<-na.omit(pbr)
summary(pbrna)
```

```
ggplot(pbrna, aes(x = index(pbrna), y = pbrna[,6])) + geom_line(color = "darkblue") +
  ggtitle("ABEV3") +
  xlab("Datas") + ylab("PREÇO DA AÇÃO") + theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_date(date_labels = "%b %y", date_breaks = "30 months")
```

```
retornopbr<- diff(log(pbrna[,6]))
retornopbrna<-na.omit(retornopbr)
plot(retornopbrna)
summary(retornopbrna)
```

#3.2.FUNÇÃO AUTO-CORRELAÇÃO LINEAR E NÃO LINEAR DE RETORNOS

#Teste de Auto-Correlação de Retornos em Logs e Absolutos.

```
library(astsa)
acfl(retornopbrna)
absr<-Mod(ts(retornopbrna))
plot(absr)
library(astsa)
acfl(absr)
#Teste de Raiz Unitária de Retornos.
library("tseries")
TRUr<-adf.test(retornopbrna)
TRUR<-adf.test(absr)
```

#3.3.HISTOGRAMA E TESTE DE NORMALIDADE DE RETORNOS EM LOGS

```
histograma<-hist(retornopbrna$ABEV3.SA.Adjusted, breaks = 200, col="grey", xlim=c(-
0.15,0.15),main="AMBEV Returns Histogram",xlab="Returns");
xfit = seq(min(retornopbrna$ABEV3.SA.Adjusted), max(retornopbrna$ABEV3.SA.Adjusted), length = 5000)
yfit = dnorm(xfit, mean = mean(retornopbrna$ABEV3.SA.Adjusted), sd =
sd(retornopbrna$ABEV3.SA.Adjusted))
yfit = yfit * diff(histograma$mids[1:2]) * length(retornopbrna$ABEV3.SA.Adjusted)
lines(xfit, yfit, col = "blue", lwd =0.1)
legend(x="topright", #posicao da legenda
```

```

c("Returns","Normal"), #nomes da legenda
col=c("grey","blue"), #cores
lty=c(1,1,3), #estilo da linha
lwd=c(2,2,4)) #grossura das linhas

#Teste de Normalidade de Jarque-Bera
library(tseries)
jarque.bera.test(retornopbrna)

#3.4.TESTE DA RAZÃO DE VARIANCIA DE LO, A. W., and A. C. MACKINLAY (1988):
library(vrtest)
r <- retornopbrna$ABEV3.SA.Adjusted
kvec <- c(2)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(5)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(10)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(15)
Lo.Mac(r,kvec)
Wald(r,kvec)

#4.POPRIEDADES ESTATÍSTICAS OU FATOS ESTILIZADOS DE AÇÕES DA ITUB4
rm(list=ls())
library(quantmod)
library(ggplot2)
#3.1.PREÇO E RETORNOS AÇÕES ITAU.
pbr <- getSymbols("ITUB4.SA", src = "yahoo", from = "2012-01-01", to = "2021-12-31", auto.assign = FALSE)
pbrna<-na.omit(pbr)
summary(pbrna)

ggplot(pbrna, aes(x = index(pbrna), y = pbrna[,6])) + geom_line(color = "darkblue") +
  ggtitle("ITUB4") +
  xlab("Datas") + ylab("PREÇO DA AÇÃO") + theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_date(date_labels = "%b %y", date_breaks = "30 months")

retornopbr<- diff(log(pbrna[,6]))
retornopbrna<-na.omit(retornopbr)
plot(retornopbrna)
summary(retornopbrna)

#4.2.FUNÇÃO AUTO-CORRELAÇÃO LINEAR E NÃO LINEAR DE RETORNOS
#Teste de Auto-Correlação de Retornos em Logs e Absolutos.
library(astsa)
acfl(retornopbrna)
absr<-Mod(ts(retornopbrna))
plot(absr)
library(astsa)
acfl(absr)
#Teste de Raiz Unitária de Retornos.
library("tseries")
TRUr<-adf.test(retornopbrna)
TRUR<-adf.test(absr)

#4.3.HISTOGRAMA E TESTE DE NORMALIDADE DE RETORNOS EM LOGS

```

```

histograma<-hist(retornopbrna$ITUB4.SA.Adjusted, breaks = 200, col="grey", xlim=c(-0.15,0.15),main="ITAU
Returns Histogram ",xlab="Returns");
xfit = seq(min(retornopbrna$ITUB4.SA.Adjusted), max(retornopbrna$ITUB4.SA.Adjusted), length = 5000)
yfit = dnorm(xfit, mean = mean(retornopbrna$ITUB4.SA.Adjusted), sd = sd(retornopbrna$ITUB4.SA.Adjusted))
yfit = yfit * diff(histograma$mids[1:2]) * length(retornopbrna$ITUB4.SA.Adjusted)
lines(xfit, yfit, col = "blue", lwd =0.1)
legend(x="topright", #posicao da legenda
      c("Returns","Normal"), #nomes da legenda
      col=c("grey","blue"), #cores
      lty=c(1,1,3), #estilo da linha
      lwd=c(2,2,4)) #grossura das linhas

#Teste de Normalidade de Jarque-Bera
library(tseries)
jarque.bera.test(retornopbrna)

```

#4.4.TESTE DA RAZÃO DE VARIANCIA DE LO, A. W., and A. C. MACKINLAY (1988):

```

library(vrtest)
r <- retornopbrna$ITUB4.SA.Adjusted
kvec <- c(2)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(5)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(10)
Lo.Mac(r,kvec)
Wald(r,kvec)
kvec <- c(15)
Lo.Mac(r,kvec)
Wald(r,kvec)

```