Predição do valor de fechamento do Fundo de Índice (BOVA11) utilizando redes nurais e o método box jenkins

Rodrigo Gomes Dutra[1]

[1] [2]Faculdade de Engenharia Elétrica e Biomédica, Universidade Federal do Pará,
Av. Augusto Correa 01, Belém, Pará 66075-090, Brasil

Autor Correspondente[1]: dutra.rgdgd@gmail.com

Resumo

O artigo detalha a construção e o uso de 3 estruturas para a predição de séries temporais, as duas primeiras estruturas são baseadas em estruturas de redes neurais, enquanto a terceira é baseada no modelo estatístico ARIMA. A primeira estrutura trata-se no modelo de rede Perceptron de múltiplas camadas(MLP) aplicada diretamente na série temporal , a segunda estrutura trata-se de uma MLP aplicada para prever a diferença da série temporal, a terceira estrutura basea-se no algorítimo auto-regressivo com médias móveis integrativo. Essas estruturas foram aplicadas na regressão da série temporal do fundo de indice BOVA11, visando-se assim comparar o desempenho de todas as topologias no problema de proposto. Todas as estruturas foram construídas utilizando software Python [1] juntamente com a bibliotecas Keras [2] e Statsmodels [3]

Palavras-chave: Redes Neurais Artificiais. ARIMA .Séries temporais.Predição. BOVA11.Python. Keras.

1 INTRODUÇÃO

Vários estudos aplicados na predição de valores de ações já foram conduzidos e propostos, comparando-se métodos de aprendizado de máquina, como as redes neurais artificiais(RNA) e métodos estatísticos como o ARIMA, alguns desses estudos foram conduzidos analizando ações do mercado estrangeiro, [4], [5] e algumas outras no mercado brasileiro de ações(B3), [6] e [7].

As Redes Neurais Artificiais (RNA) tratam-se de modelos matemáticos construídos a partir do conhecimento do neurônio biológico, tendo como objetivo mimetizar o comportamento da células nervosa [8] . Assim, simulando a característica de um sistema nervoso, de forma a ter capacidade de processar múltiplas entradas, reconhecer e classificar padrões. Dessa forma, RNA tem grande capacidade de aplicação em vários tipos de contexto.

O modelo ARIMA trata-se de um modelo estatístico proposto por box e jenkins, que consiste na junção de 3 processos, um auto regressivo, um baseado em médias móveis e outro integrativo. [9]

Este artigo busca a implementação comparativa de métodos propostos anteriormente e mais um modelo hibrido no qual consiste em prever as diferenças da série temporal com uma RNA, de forma a aplicar o operador de diferenciação discreta que será explanado mais a frente nesse artigo. Assim a rede neural será alimentada por uma série temporal que será mais estacionária em relação a série temporal original, assim sendo uma aplicação diferente a rede neural que recebe a série temporal sem algum tratamento.

A aplicação proposta nesse artigo trata-se de regressão de séries temporais. O fundo de indice BOVA11 consiste em um ETF(*Exchange Traded Funds*), que busca mimetizar o comportamento do Indice Ibovespa, assim refletindo a performances de desempenho do indice.

O indice IBOVESPA é um importante termômetro do mercado Brasileiro, representando um indicador de desempenho das principais ações comercializadas na bolsa de valores brasileira(B3), assim refletindo diretamente a performance financeira do país.

1.1 Metodologia

Para o fim proposto, foi usado o Software Python(3.65) [1] juntamente com a bibliotecas Keras [2], Scikit [10], Statsmodels [3], para respectivamente criar as RNA, manipulação da base de dados e para a criação de processos ARIMA.

1.2 Objetivos

O objetivo desse trabalho é primeiramente validar as topologias de redes apresentadas neste, aplicando-as em um problema de regressão de séries temporais.Feito isso, o trabalho também tem como objetivo comparar o desempenho das 2 estruturas de RNA e uma estrutura estatística para o problema proposto.

1.3 Organização do Trabalho

A organização do artigo foi estruturado da seguinte forma: A seção 2 detalha o banco de dados e seu uso dentro do artigo; Na seção 3 encontra-se a base teórica de RNA e a apresentação dos 2 modelos utilizados;A seção 4 detalha os processos ARIMA e sua aplicação na base de dados proposta; A seção 5 detalha o desenvolvimento e aplicação da MLP;A seção 6 detalha o desenvolvimento e aplicação da rede neural integrativa ;A seção 7 apresenta os resultados; A seção 8 dispõe a conclusão.

2 Banco de Dados

O Índice Bovespa (IBOV) é o principal indicador de desempenho das ações negociadas na bolsa de valores brasileira(B3).Reunindo as empresas mais influentes do mercado de capitais brasileiro, este índice é composto pelas ações de companhias listadas na B3 que atendem a determinados critérios. Os papéis que compõe o Ibovespa correspondem a cerca de 80% do número de negócios e do volume financeiro do mercado brasileiro de capitais.

A base de banco de dados do fundo de índice BOVA11 foi obtida através do *yahoo finance* [11], onde foi extraídas as informações preço de fechamento, preço de abertura, preço de máxima, preço de mínima e o volume de ações negociadas. Esses dados são relativos ao período de 6/12/2010 até o dia 14/11/2019, com intervalo diário.

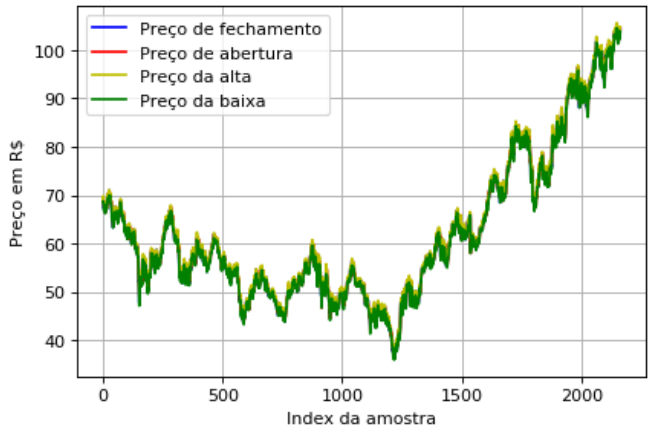

Figura 1. Comportamento da série temporal

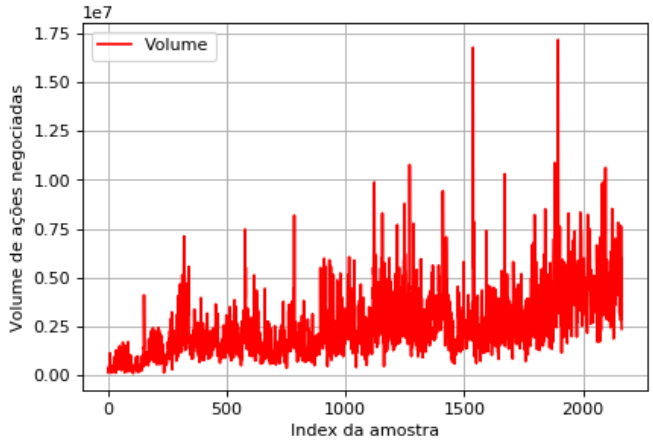

Figura 2. Comportamento da variável volume de ações negociadas

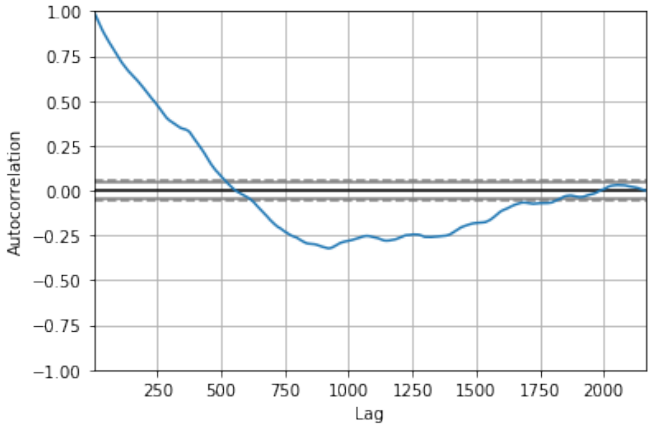

Figura 3. Autocorrelação da variável preço de fechamento

A variável de interesse desse artigo é o preço de fechamento, dessa forma é importante notar o comportamento não estacionário da variável, o qual pode ser observado na Figura 1, onde a variável apresenta Várias oscilações e mudanças na tendência da série. Outro forte indicio da não estacionaridade da série é o comportamento da autocorrelação da série mostrada na Figura 3, onde autocorrelação apresenta um decaimento lento e gradativo, não se limitando nos limites de confiabilidade do gráfico.

Dessa forma, como a base de dados apresenta uma série temporal de caráter não estacionário será testado topologias que tratam este comportamento através do operador da diferenciação :

$$diff[n] = x[n] - x[n-d] \tag{1}$$

Onde d é a ordem da diferenciação.

A base de dados foi normalizada por coluna utilizando-se da equação :

$$x'_i = \frac{(x_i - x_{min}) * (L_{max} - L_{min})}{x_{max} - x_{min}} + L_{min} \tag{2}$$

A divisão da base de dados em dados de treino, validação e teste foi realizada com a biblioteca Scikit[7], a qual randomiza a seleção de índex para a divisão do banco de dados em dada proporção previamente programada. A proporção utilizada foi de 70% dos dados para treino , 15% para validação e 15% para teste.

3 Estruturas de RNA

RNA podem apresentar vários tipos de estrutura, dado sua capacidade de ligar neurônios artificiais. Como estruturas de sistemas nervosos biológicos, cada estrutura distinta de RNA pode ser especializada em problemas específicos. Nesse sentido vale-se a discussão que não necessariamente uma rede com uma quantidade grande de neurônios e ligações será melhor que uma rede com poucas ligações e menor números de neurônios dado alguma aplicação especifica.

Um neurônio artificial é modelado tendo terminais de entradas$x$ e um terminal de saída.O comportamento das sinapses é simulado através dos pesos aplicados nas entradas do neurônio, a saída do neurônio é dada por :

$$Sj = \theta(\sum_{i=0}^{m}(wij * xi + bj)) \tag{3}$$

Onde θ representa a função de ativação, bj corresponde ao *bias*, wij são os pesos sinápticos[5].

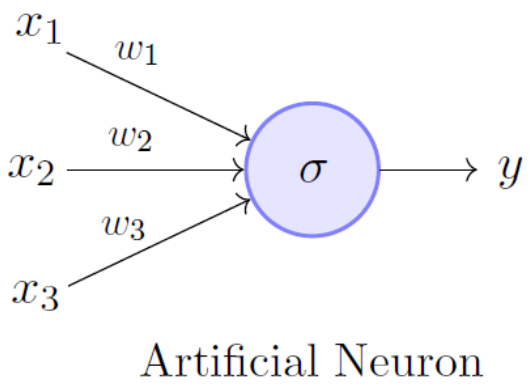

Figura 4. Estrutura de um neurônio artificial

De forma geral uma rede neural é constituída de unidades simples de processamento denominadas de neurônios e ligações massivas entre essas unidades.[5] Mais a frente nessa seção será detalhado a constituição de Redes MLP aplicadas no problema proposto.

3.1 Perceptrom de Múltiplas Camadas

A rede perceptron de múltiplas camadas como o próprio nome indica apresenta várias camadas em sua estrutura,tendo como composição mínima uma camada de entrada, uma camada de saída e uma camada intermediária. Segundo o modelo descrito por [8], essas camadas são ligadas por sinapses com pesos(*pij*) intrínsecos, assim compondo uma rede vasta de interconexões regidas por pesos e funções de ativação.

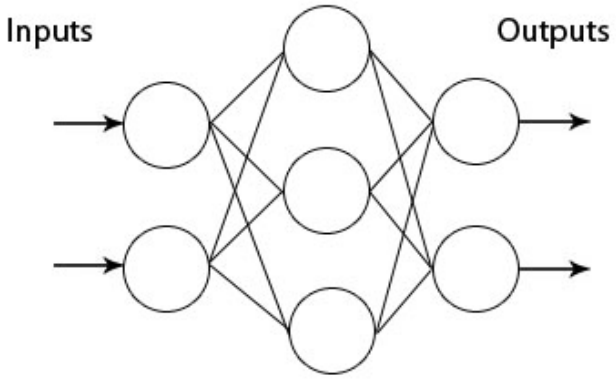

Figura 5. Estrutura de um neurônio artificial

O aprendizado da MLP consiste em 2 passos, o passo direto e o passo inverso. O passo direto ou *forward pass* consiste em passar as entradas pela rede, assim aplicando os pesos associados as sinapses, e obtendo assim a saída. O segundo passo consiste na retro propagação do erro, é calculado o gradiente da função de perda na camada de saída, posteriormente esse gradiente é utilizado para aplicar recursivamente a regra da cadeia para atualizar os pesos de toda a rede. Esse algorítimo é chamado de *backpropagation*[1].

3.2 Redes Neural integrativa

Esta Rede Neural rede neural recebe essa denominação por conta que os dados de entrada são diferenciados em uma ordem d conforme descrito na Equação 2, dessa forma os dados alimentados na rede neural estão mais perto da estacionariedade descrita no sentido amplo, conforme [9]. Dessa forma essa estrutura pode alcançar mais exito do que uma estrutura na qual não se aplique esse tipo de tratamento a estacionaridade.

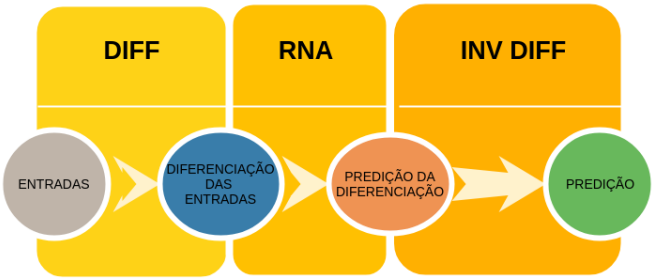

Figura 6. Estrutura da RNA integrativa

A Figura 5 ilustra o processo da rede, de forma que as entradas passam pelo processo da diferenciação (DIFF), transformando as entradas (preço de fechamento, abertura, máxima mínima e volume de ações) nas suas respectivas versões diferenciadas. Após isso esses dados são alimentados a uma RNA, onde a rede neural recebera amostras passadas das entradas diferenciadas e tentara prever o próximo passo da diferença de ordem d do preço de fechamento. A saída da rede é alimentada a equação de inversão da diferenciação, de forma que a predição final seja do próximo preço de fechamento e não a diferença do preço de fechamento de ordem d.

O processo de inversão da diferenciação de ordem d consiste em aplicar uma operação análoga a equação 2:

$$x[n] = diff[n] + x[n-p] \tag{4}$$

Dessa forma os dados diferenciados são somados aos atrasos de ordem d de cada uma das entradas.

A ordem d foi escolhida de forma que após a aplicação da equação 2 seria notório a redução do comportamento estacionário, e a autocorrelação do preço de fechamento ainda apresentaria autocorrelação alta com relação aos primeiros atrasos, porem reduzindo drasticamente conforme a ordem dos atrasos fossem aumentando.

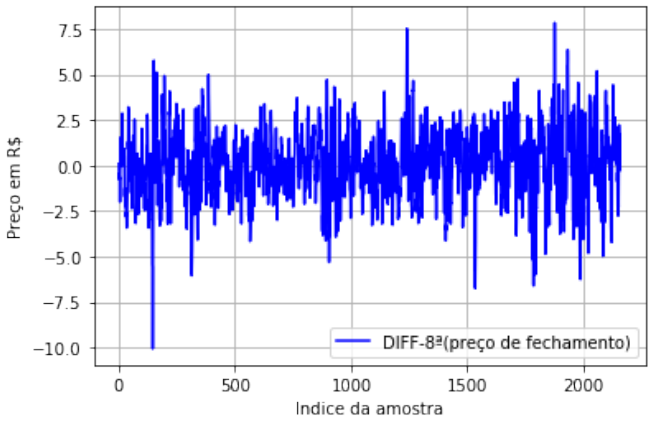

Figura 7. Diferença de ordem 8 do preço de fechamento

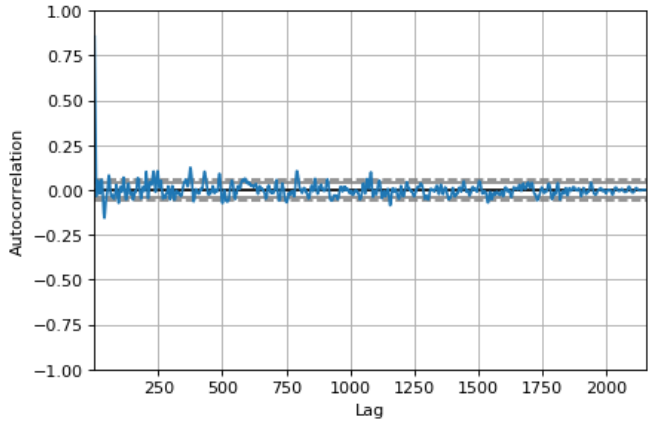

Figura 8. Autocorrelação da diferença de ordem 8 do preço de fechamento

4 Desenvolvimento da Estrutura ARIMA

Um processo ARIMA, consiste em sumo em 3 processos, o processo AR(p), o processo MA(q), e o processo I(d). Assim, o primeiro processo indica que a variável de interesse é auto-regressada baseado nas amostras passadas, o segundo processo indica que o erro de predição é uma combinação dos termos de erro que aconteceram em amostras da regressão passada, o ultimo processo é responsável por aplicar a a diferenciação na variável de interesse, com o intuito de torna-la mais estacionária possível [9].Dessa maneira, o modelo ARIMA busca realizar uma regressão para ajustar-se aos dados fornecidos na melhor forma possível.

O processo I(d) é o processo responsável por realizar a operação da equação 2 para um processo ARMA(p,q) então assumir a regressão da variável de interesse. Dessa forma o modelo ARIMA(p,d,q) possui ordem $d = 8$, semelhantemente como foi aplicado na estrutura da rede neural integrativa. O processo ARMA(p,q) pode ser escrito como :

$$X_t - \alpha_1 * X_{t-1} ... - \alpha_p * X_{t-p} = \epsilon_t + \theta_1 * \epsilon_{t-1} + ... + \theta_q * \epsilon_{t-q} \quad (5)$$

Onde ϵ_t, θ_t e α_t representam respectivamente, os termos de erro, os parâmetros da média móvel, e os parâmetros auto-regressivos. Definida a ordem d da estrutura ARIMA o processo foi alimentado com a mesma base de dados que fora aplicado nas estruturas MLP e assim foi realizado o fit do processo ARIMA e retornados os dados de BIC e AIC, os quais representam respectivamente: Critério de Informação Bayesiano (BIC) e Critérios de informação de Akaike(AIC), os quais são largamente utilizados para a classificação de modelos estatísticos.

Tabela 1
Performance relativos aos parâmetros (p,d,q)

Parametros PDQ	BIC	AIC
1,8,0	**137.331046380407**	**132.26440801806518**
2,8,0	141.01974741114802	134.26422959469227
3,8,0	143.0318565924805	134.5874593219108
4,8,0	140.99693640602953	130.86365968134592
5,8,0	144.30087986625838	132.47872368746084
1,8,1	141.0196286691889	134.26411085273315
2,8,2	145.3769811552839	135.24370443060027
3,8,3	149.23035444155028	135.71931880863877
4,8,4	152.74377654008788	135.85498199894852
5,8,5	168.28708816991568	148.02053472054843
0,8,1	165.13470278862047	160.06806442627865
0,8,2	156.47128232656237	149.71576451010662

5 Desenvolvimento da Rede MLP

A rede MLP foi desenvolvida em Python [1] com o auxilio da bibliotecas Keras [2].A divisão de dados em treino validação foi de 70%,15% e 15% respectivamente, realizado pela biblioteca Scikit [10]

A escolha das entradas seguiu a ordem da autocorrelação da entrada na figura 3, de forma que é possivel escolher um número grande de atrasos das variáveis do banco de dados como entrada. Assim foram escolhidos 20 atrasos de cada uma das variáveis que compõe o banco de dados utilizado.

A estrutura da rede apresenta 3 camadas, na camada de entrada exite 100 neurônios para receber os dados de entrada(20 passos passados de cada uma das 5 variáveis do banco de dados), a camada de saída dispõe de apenas 1 neurônio. Para definir o numero óptimo de neurônios na camada intermediária foi o treinamento da rede por 5 repetições e extraído o valor do MSE(*Mean Square Error*) relativos a 500ª época de treinamento.

Tabela 2
MSE versus Nº de neurônios

Neuronios	MSE médio de 5 repetições
5	1.464291479979983
10	1.2809192212472051
15	1.2289928958652923
30	**1.1779625977048223**
60	1.45865400888905

Depois de definido o número de neurônios da camada escondida, os dados utilizados no treino validação e teste foram normalizados, de forma que cada coluna dos 5 atributos foi dividida pelo maior número da coluna, porém o valor da saída desejada não foi normalizada, dessa forma seguindo o comportamento da figura 1.

A função de ativação da camada escondida escolhida foi *selu*,a função de ativação da camada de saída foi linear, a escolha das funções foi feita depois de vários testes de desempenho relacionando o erro com a função de ativação escolhida.

6 Desenvolvimento da Rede MLP Integrativa

Assim como a Rede MLP, a rede integrativa foi desenvolvida utilizando os mesmo software e bibliotecas.A divisão dos dados de entrada em treino validação e teste ocorreu de forma idêntica a rede MLP construída anteriormente.

De forma semelhante a primeira MLP as entradas foram definidas de forma que hajam uma correlação alta entre as entradas selecionadas, dessa forma após a aplicação do operador de diferenciação foi checado a autocorrelação, e como descrito anteriormente essa autocorrelação decaia rapidamente, dessa maneira foi selecionado apenas 5 amostras passadas da diferença de cada variável do banco de dados.

Feito a seleção da quantidade de entradas foi realizado a normalização das entradas e deixando a saída de fora dessa operação, dessa forma a rede neural foi treinada variando-se o número de neurônios na camada escondida de forma análoga a primeira MLP.

Tabela 3
MSE versus Nº de neurônios

Neurônios	MSE médio de 5 repetições
5	1.2415570015492647
10	**1.2524923509692554**
15	1.2733771818765203
30	1.2963275257845104
60	1.257574004045925

A função de ativação escolhida visando a maior acurácia da rede foi *selu*, enquanto a da camada escondida foi linear quanto na camada de saída.

7 Resultados

Após Realizar o treinamento das 3 estruturas de redes propostas, todas as estruturas foram colocadas a prova a fim de avaliar seus resultados com os mesmos dados de treino validação e teste, após escolher os melhores parâmetros para as 3 estruturas, conforme demonstrado nas tabelas 1,2 e 3 conforme exposto na seção 3 sobre o banco de dados.

7.1 Resultados MLP

Depois de selecionado o número ideal para teste dessa estrutura de RNA, foi predito 30 valores subsequentes da base de dados de teste.

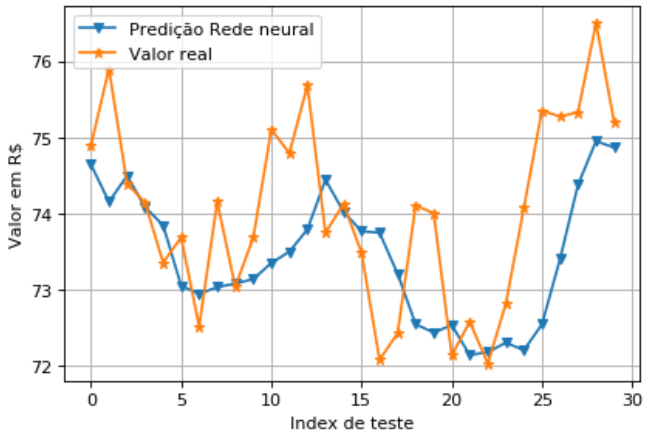

Figura 9. Resultado da MLP

7.2 Resultados RNA Integrativa

Feito a análise do melhor número de neurônios na camada escondida para a red MLP integrativa conforme a Tabela 3, foi predito 30 índices da base de dados de teste:

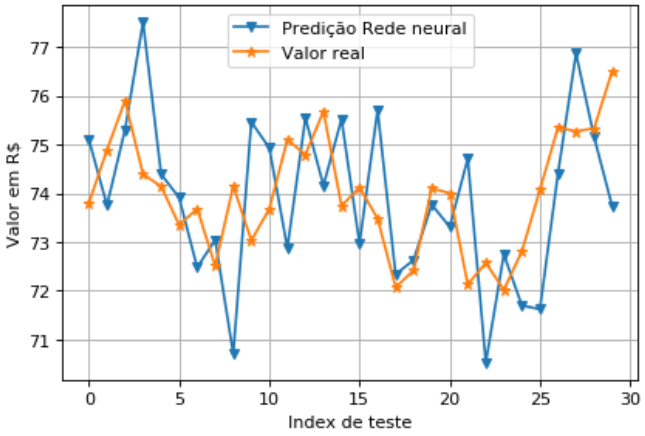

Figura 10. Resultado da predição MLP integrativa

7.3 Resultados do Modelo ARIMA

Feito a análise do melhor parâmetro para a estrutura ARIMA conforme a tabela 1, os resultados da predição de 30 index de teste estão a seguir:

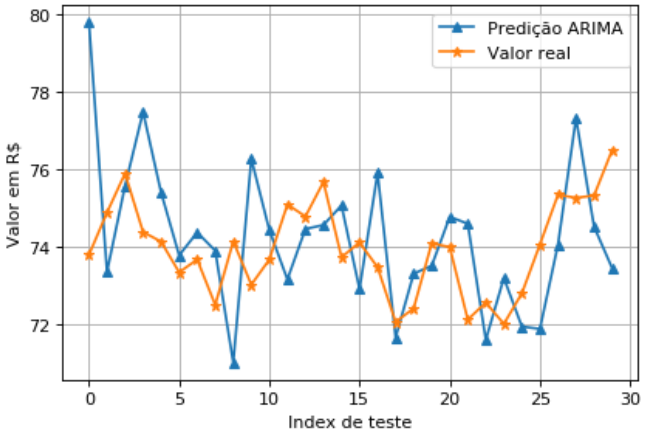

Figura 11. Resultado Predição Arima

7.4 Comparativo do desempenho

Após realizados os testes de desempenho das 3 estruturas,o comparativo de desempenho de ambas foi disposto na Tabela 4.

Tabela 4
Performance relativos aos parâmetros (p,d,q)

Topologia	MSE	MAE
MLP	**1.393197**	**0.931671**
MLP Integrativa	2.740275	2.740275
ARIMA	3.966883	1.590634

Com base nos resultados apresentados na tabela 4, foi demonstrado que o melhor desempenho das 3 estruturas foi o da rede MLP sem o tratamento de estacionaridade proposto no modelo integrativo. Isto pode ser devido primeiramente ao comportamento especifico da base de dados, visto que o tratamento de estacionariadade é muitas vezes necessário para atingir marcas de desempenho mais relevantes.

8 Considerações finais

8.1 Discussão

Todas as estruturas apresentadas nesse artigo apresentaram bom desempenho para a predição de séries temporais não estacionárias, de forma que mesmo sem o tratamento da série temporal um modelo baseado em rede neural poderia aprender o comportamento não estacionário e prever os próximos passos da série temporal com valores baixos de MSE. Dessa forma, a estrutura MLP apresentou o melhor destaque, porém isso não significa que o tratamento através do operador de diferenciação não deva ser aplicado em outros estudos de caso ou em outras aplicações que envolvam a predição de séries temporais voltadas para o mercado financeiro.

Com relação a custo computacional a rede neural MLP que apresentou o melhor desempenho também apresentou o maior custo computacional, tendo o maior número de entradas e levando mais tempo para o treinamento e validação. Dessa forma, para aplicações nas quais deseja-se reduzir o valor de entradas e o custo computacional as outras 2 estruturas poderiam apresentar um desempenho versus custo computacional melhor para a aplicação.

8.2 Conclusão

Com base no problema de regressão baseado em séries temporais a rede MLP aplicada diretamente na série temporal, utilizando-se unicamente da normalização como tratamento dos valores da série, apresentou o melhor desempenho das 3 estruturas expostas nesse documento, com a ressalva do custo computacional. O maior número de entradas e tempo de execução poderia inviabilizar algum tipo de aplicação mais crítica com relação ao tempo. Dessa forma para aplicações onde a rapidez de treino seja crítica ou em aplicações que não dispõe de tanto poder computacional, como sistemas embarcados e IOT, a estrutura ARIMA ou a MLP integrativa poderiam ser soluções melhores.

Referências

[1] G. van Rossum, "Python tutorial, technical report cs-r9526, centrum voor wiskunde en informatica (cwi), amsterdam.","" 1995.

[2] F. Chollet *et al.*, "Keras: Deep learning library for theano and tensorflow.(2015)," *There is no corresponding record for this reference*, 2015.

[3] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, vol. 57, p. 61, Scipy, 2010.

[4] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of arima and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, vol. 2014, 2014.

[5] S. Chopra, D. Yadav, and A. Chopra, "Artificial neural networks based indian stock market price prediction: Before and after demonetization," *J Swarm Intel Evol Comput*, vol. 8, no. 174, p. 2, 2019.

[6] A. P. Miranda, P. S. Ceretta, and L. F. D. Lopes, "Estratégias de mercado acionário utilizando previsão de redes neurais em comparação com modelos autorregressivos," *Revista de Administração da Universidade Federal de Santa Maria*, vol. 8, no. 1, pp. 42–59, 2015.

[7] T. B. Papadopoulos, F. R. Bittencout, F. L. D. Leroy, T. C. B. Reis, and P. C. de Brito Neves, "Redes neurais artificiais na predição do preço futuro de fechamento de papéis com alta e baixa volatilidade negociados na bovespa e na bolsa de valores de nova iorque," *Blucher Marine Engineering Proceedings*, vol. 2, no. 1, pp. 873–884, 2016.

[8] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[9] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. springer, 2016.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[11] U. Data, "Yahoo! finance," 2012.