

Ademir Batista dos Santos Neto

**Combinação de modelos preditivos de séries temporais
através de funções de cópulas**

Recife

17 de agosto de 2023



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOMETRIA E ESTATÍSTICA APLICADA

Combinação de modelos preditivos de séries temporais através de funções de cópulas

Tese julgada adequada para obtenção do título de Doutor em Biometria e Estatística Aplicada, defendida e aprovada por unanimidade em 30/06/2023 pela comissão examinadora

Área de concentração: Biometria e Estatística Aplicada

Orientador: Professor Dr. Tiago Alessandro Espínola Ferreira
Coorientador: Professor Dr. Paulo Renato Alves Firmino

Recife

17 de agosto de 2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

- C731c Neto, Ademir Batista dos Santos
 Combinação de modelos preditivos de séries temporais através de funções de cópulas / Ademir Batista dos Santos
 Neto. - 2023.
 93 f. : il.
- Orientador: Tiago Alessandro Espindola Ferreira.
 Coorientador: Paulo Renato Alves Firmino.
 Inclui referências e apêndice(s).
- Tese (Doutorado) - Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Biometria e
 Estatística Aplicada, Recife, 2023.
1. Previsão. 2. Aprendizado de Máquina. 3. Funções de Cópulas. 4. Séries Temporais. I. Ferreira, Tiago Alessandro
 Espindola, orient. II. Firmino, Paulo Renato Alves, coorient. III. Título

CDD 519.5

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOMETRIA E ESTATÍSTICA APLICADA
Combinação de modelos preditivos de séries temporais através de funções de
cópuas

Ademir Batista dos Santos Neto

Tese julgada adequada para obtenção do título de Doutor em Biometria e Estatística Aplicada, defendida e aprovada por unanimidade em 30/06/2023 pela comissão examinadora

Orientador:

Professor Dr. Tiago Alessandro
Espínola Ferreira
Orientador

Banca examinadora:

Professor Dr. Paulo Renato Alves
Firmino
Universidade Federal do Cariri

Professor Dr. Jader da Silva Jale
Universidade Federal Rural de
Pernambuco

Professor Dr. Paulo Salgado Gomes
de Mattos Neto
Universidade Federal de Pernambuco

Professor Dr. Francisco de Sousa
Ramos
Universidade Federal de Pernambuco

*Este trabalho é dedicado
ao meu filho Matheus*

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Agradeço primeiramente a Deus por ter me concedido todas as condições necessárias para que eu pudesse concluir esse trabalho e me iluminar em todo o caminho que trilhei durante o meu doutorado.

Agradeço muito aos meu pais. A minha mãe Elisângela por representar tudo para mim e sempre ter me dado amor e carinho por toda a minha vida. Ao meu pai Ademir Junior que constantemente me mostrou os valores corretos e me orientou com muita sabedoria nas minhas decisões.

Agradeço a toda minha família pelo amor e carinho incondicionais. Em especial a meus irmãos: Eduarda, Thaís, Hugo e Marcella. Agradeço muito ao meu padrasto Mauricio que sempre me guiou no caminho dos estudos. Agradeço também aos meus avós: Maria José, Ademir e Maria Bernadete que sempre me deram muito carinho e afeto.

Agradeço de forma especial a duas pessoas que sem elas não poderia ter concluído esse trabalho. Agradeço muito ao meu orientador, professor Tiago, que me orientou em diversos aspectos, acadêmicos ou não, com toda paciência e dedicação necessária nesse processo. E meu coorientador o professor Paulo Renato que teve participação fundamental no meu desenvolvimento acadêmico e profissional.

Agradeço aos meus professores que sempre me transmitiram conhecimentos para que eu pudesse me tornar uma pessoa melhor.

Agradeço aos meus amigos e colegas por serem essas pessoas tão especiais em minha vida, que sempre colaboram para o meu desenvolvimento.

Por fim, agradeço especialmente a uma pessoa muito importante na minha vida, a minha esposa Rhayane Vaz. Obrigado por todo carinho, apoio e dedicação que você teve comigo durante todos esses anos que estamos juntos.

*“No one can forecast the economy with certainty.”
(Jamie Dimon)*

Resumo

O mercado financeiro é um ambiente muito dinâmico, o que ocasiona alta volatilidade nos dados inerentes a esses processos. Logo, torna-se um grande desafio modelar e prever séries temporais provenientes desses mercados. No entanto, entender o comportamento das séries temporais financeiras desempenha um papel fundamental na tomada de decisões mais eficientes no mundo dos negócios. Portanto, vários estudos têm como objetivo a elaboração de metodologias sofisticadas voltadas para a previsão de séries, com foco principal nas séries temporais financeiras. Dentre esses estudos, destacam-se aqueles em que mais de um modelo é utilizado para realizar as previsões. A combinação de modelos de previsão de séries temporais tem levado a resultados mais precisos do que os modelos aplicados individualmente, como comprovado por vários trabalhos na literatura. Assim, desde o século passado, diversas técnicas que promovem a combinação de modelos de previsão foram introduzidas. A busca por modelos de combinação mais precisos tem sido objeto de estudo de muitas pesquisas. Neste trabalho, evidenciou-se o potencial do uso de funções de cópulas para combinar previsões de modelos de aprendizado profundo de máquina dedicados a séries temporais financeiras. Nessa Tese, alguns modelos consolidados de previsão individuais, como ARIMA, Redes Neurais Artificiais clássicas (RNA) e redes recorrentes de aprendizado profundo (*long short term memory* - LSTM) são empregados para prever cinco séries temporais financeiras. Os resultados são comparados por métricas de desempenho como: a raiz do erro quadrático médio - *Rooted Mean Squared Error* - RMSE, Erro Absoluto Percentual Médio - *Mean Absolute Percentage Error* - MAPE, Previsão de Mudança de Direção - *Prediction of Change in Direction*- POCID e U de Theil. As técnicas de *ensemble* utilizadas foram: média simples, mediana, funções de cópulas e MLPs (RNAs do tipo *multi-layer perceptron*). Os resultados obtidos mostraram que a combinação por cópulas utilizando abordagens de aprendizado profundo apresentaram melhores resultados em comparação com outras abordagens presentes na literatura. No geral, concluímos que, no que diz respeito às séries temporais financeiras, combinando técnicas de aprendizado profundo usando funções de cópulas leva a melhores previsões.

Palavras-chave: Previsão; Aprendizado de Máquina; Funções de Cópulas; Séries Temporais.

Abstract

The financial market is a highly dynamic environment, characterized by significant volatility in the data associated with its processes. Consequently, modeling and predicting time series derived from these markets pose substantial challenges. However, comprehending the behavior of financial time series plays a pivotal role in making more informed decisions in the business domain. Consequently, numerous studies aim to develop sophisticated methodologies for forecasting series, with a particular emphasis on financial time series prediction. Notably, studies that employ multiple models to perform forecasts have garnered attention. The combination of time series forecasting models has consistently yielded more accurate results than individual models, as demonstrated by several works in the literature. As a result, numerous techniques promoting the combination of forecasting models have been introduced since the previous century. Research efforts have focused on devising accurate combination models that effectively weight all the involved models. In this study, we showed the potential of utilizing copula functions to combine deep learning techniques for predicting financial time series. Specifically, we employ established individual forecasting models, such as ARIMA, Artificial Neural Networks (ANN), and recurrent deep learning networks (Long Short Term Memory - LSTM), to predict five financial time series. Performance metrics, including Rooted Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Prediction of Change in Direction (POCID), are used to compare the results. The ensemble techniques employed in this article include simple mean, simple median, copula functions, and MLPs. The findings of this study demonstrate that combining copula functions with deep learning approaches yields superior results compared to other approaches documented in the literature. Overall, we conclude that, in the context of financial time series, combining deep learning techniques using copula functions generally leads to more accurate predictions in terms of accuracy.

Key-words: Forecast; Machine Learning; Copulas function; Time series.

Lista de figuras

Figura 1 – Série temporal do fechamento das ações da empresa Google de julho até dezembro de 2008	7
Figura 2 – Série dos valores de fechamento da S&P 500 entre os anos de 2013 e 2022	9
Figura 3 – Função de auto correlação da série da Google apresentada na Figura 1.	11
Figura 4 – Função de auto correlação parcial da série da Google apresentada na Figura 1.	12
Figura 5 – Analogia entre modelos biológico e computacional de neurônios. Fonte: https://www.researchgate.net/figure/Figura-19-Neuronio-biologico-esquerda-e-neuronio-artificial-direita_fig18_300015903 , 2022	13
Figura 6 – Diagrama do Perceptron.	14
Figura 7 – Representação da função logística.	16
Figura 8 – Camadas de uma rede neural MLP.	17
Figura 9 – Esquema de conexões de uma rede neural recorrente.	20
Figura 10 – Detalhamento do <i>loop</i> de um nó da RNR.	20
Figura 11 – Representação de uma rede <i>deep learning</i> . Fonte: https://shre.ink/1hA , 2023	22
Figura 12 – Representação de um nó de uma rede LSTM. Fonte: https://bit.ly/3m1kRTS , 2015	23
Figura 13 – Representação de um nó de uma rede LSTM. Fonte: https://bit.ly/3m1kRTS , 2015	24
Figura 14 – Representação de um nó de uma rede LSTM. Fonte: https://bit.ly/3m1kRTS , 2015	25
Figura 15 – Representação de um nó de uma rede LSTM. Fonte: https://bit.ly/3m1kRTS , 2015	26
Figura 16 – Representação de um nó de uma rede LSTM. Fonte: https://bit.ly/3m1kRTS , 2015	27
Figura 17 – Esquema da representação de modelos de combinação de previsões <i>via</i> cópulas.	31
Figura 18 – Esquema da representação da combinação <i>via</i> MLP.	35
Figura 19 – Série e Lagplot PB.	40
Figura 20 – Série e Lagplot GG.	40
Figura 21 – Série e Lagplot SP.	41

Figura 22 – Série e Lagplot NQ.	42
Figura 23 – Série e Lagplot GU.	43
Figura 24 – Representação gráfica do conjunto de treinamento da série PB.	46
Figura 25 – Conjuntos da série PB e previsões dos modelos individuais.	51
Figura 26 – Resíduos dos modelos individuais para a previsão da série PB.	52
Figura 27 – Representação gráfica do processo de estimação da previsão de cópulas.	54
Figura 28 – Esquema da MLP para a combinação dos três modelos individuais.	55
Figura 29 – Análise residual estudenterizada para a série NQ a respeito das previsões dos modelos individuais.	62
Figura 30 – Análise residual estudenterizada para a série NQ a respeito das previsões dos modelos combinados.	64
Figura 31 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo ARIMA	77
Figura 32 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo ANN	77
Figura 33 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo LSTM	78
Figura 34 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo Média	78
Figura 35 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo Mediana	79
Figura 36 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo Cópula	79
Figura 37 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo MLP	80
Figura 38 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo ARIMA	81
Figura 39 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo ANN	81
Figura 40 – Análise residual estudenterizada para a série sp a respeito das previsões do modelo LSTM	82
Figura 41 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo Média	82
Figura 42 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo Mediana	83
Figura 43 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo Cópula	83

Figura 44 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo MLP	84
Figura 45 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo ARIMA	85
Figura 46 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo ANN	85
Figura 47 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo LSTM	86
Figura 48 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo Média	86
Figura 49 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo Mediana	87
Figura 50 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo Cópula	87
Figura 51 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo MLP	88
Figura 52 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo ARIMA	89
Figura 53 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo ANN	90
Figura 54 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo LSTM	90
Figura 55 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo Média	91
Figura 56 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo Mediana	91
Figura 57 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo Cópula	92
Figura 58 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo MLP	92
Figura 59 – Análise residual estudenterizada para a série NQ a respeito das previsões do modelo MLP	93

Lista de tabelas

Tabela 1 – Resumo dos trabalhos relacionados a tese. Onde na segunda coluna da tabela entende-se CL combinação linear, MHP modelo híbrido de previsão, CnL combinação não linear.	38
Tabela 2 – Tabela com as estatísticas básicas das séries avaliadas.	43
Tabela 3 – Parâmetros da função <code>auto.arima()</code>	45
Tabela 4 – Resumo do modelo ARIMA para o conjunto de treinamento da série PB.	46
Tabela 5 – Argumentos da função <code>nntear()</code>	47
Tabela 6 – RMSE da validação cruzada e Ljung-Box p valor.	48
Tabela 7 – Parâmetros das distribuições multivariadas dos resíduos dos modelos.	53
Tabela 8 – Parâmetros da função <code>nntear()</code>	55
Tabela 9 – Tabela com o resultado das métricas das previsões dos modelos para o conjunto de teste.	59
Tabela 10 – Tabela com a proporção de <i>outliers</i> para cada análise residual estuden- tizada.	61

Lista de símbolos

\mathbb{R}	Conjunto dos números Reais
μ	Média
σ^2	Variância
γ	Covariância
Ω	Espaço amostral
\mathcal{F}	sigma-álgebra
P	Função de probabilidade
ϕ	Parâmetros do modelo auto regressivo
θ_p	Parâmetro do modelo de médias móveis
ϵ_t	Ruído branco
∇y_t	Diferenciação da séries no instante y_t
$\rho(\tau)$	Função de autocorrelação
$X_1 \dots X_n$	Entradas da rede neural
$W_1 \dots W_n$	Pesos das rede neural
Σ	Agregador Linear
$g(\cdot)$	Função de ativação
Θ	Viés
u	Sinal de ativação
y	Saída da rede
\tanh	Tangente hiperbólica
X_{t-1}	Entrada anterior na rede

h_t	Saída atual
C_t	Estado da célula na rede LSTM
MS_t	Média Simples
$C(\cdot)$	Função de Cópulas
F_Y	Função de distribuição acumulada multivariada
$\phi(t)$	Função geradora da cópula de Clayton
$\phi(vi)$	Função geradora da cópula de Gumbel
f_c	Combinador não linear

Sumário

1	Introdução	1
1.1	Justificativa	1
1.2	Objetivos	4
1.2.1	Objetivo geral	4
1.2.2	Objetivos específicos	4
1.3	Estrutura do documento	4
2	Modelagem e previsão de séries temporais	6
2.1	Séries temporais	6
2.1.1	Séries temporais financeiras	8
2.2	Modelos de previsão de séries temporais	10
2.2.1	Modelos ARIMA	10
2.2.2	Redes neurais artificiais	12
2.2.2.1	Arquiteturas de redes neurais artificiais	14
2.2.2.2	Redes neurais de múltiplas camadas	17
2.2.2.3	Redes neurais recorrentes	19
2.2.2.4	Redes neurais em previsão de séries temporais	21
2.2.3	<i>Long Short Term Memory</i>	21
3	Combinação de modelos de previsão	28
3.1	Combinação de modelos	28
3.1.1	Combinadores lineares	28
3.1.2	Combinadores não lineares	29
3.1.2.1	Funções de cópulas	29
3.1.2.2	Combinadores baseados em RNA	34
4	Trabalhos relacionados	36
5	Séries temporais avaliadas	39
5.1	Processamento das séries avaliadas	39
5.1.1	Petrobras (PB)	39
5.1.2	Google (GG)	40
5.1.3	S&P500 (SP)	41
5.1.4	Nasdaq (NQ)	41
5.1.5	Taxa de câmbio GBP para USD (GU)	42
5.1.6	Estatísticas das séries avaliadas	43

6	Metodologia	44
6.1	Modelos de previsão	44
6.1.1	ARIMA	44
6.1.2	ANN	47
6.1.3	LSTM	48
6.1.4	Média simples e mediana	50
6.1.5	Cópulas	50
6.1.6	Redes neurais	54
6.2	Métricas de avaliação dos modelos preditivos	56
7	Resultados	58
7.1	Avaliações dos modelos	58
7.2	Análise estudentizada	60
8	Conclusão	66
	Referências bibliográficas	68
	Apêndices	76
APÊNDICE A	Petrobras (PB)	77
APÊNDICE B	S&P500 (SP)	81
APÊNDICE C	Google (GG)	85
APÊNDICE D	Taxa de câmbio GBP para USD (GU)	89
APÊNDICE E	Nasdaq (NQ)	93

1 Introdução

1.1 Justificativa

O conhecimento acerca do mercado financeiro é de extrema importância para a tomada de decisões estratégicas. Entretanto, esse ambiente é altamente dinâmico e a compreensão dos modelos financeiros torna-se um desafio à medida que diversas fontes de informações aumentam a volatilidade dos modelos, dificultando a generalização dos preditores. A fim de obter maior precisão em relação à acurácia das previsões, muitos autores têm proposto o desenvolvimento de novas metodologias para o estudo e previsão de séries temporais, especialmente aquelas relacionadas ao mercado financeiro ([MATSUMOTO et al., 2019](#)). Embora não exista uma fórmula universal para a previsão livre de ruídos, o esforço constante de aprimoramento das metodologias de previsão contribui para o desenvolvimento de técnicas cada vez mais sólidas em relação à previsão de séries temporais.

Uma série temporal é qualquer conjunto de observações a respeito de um fenômeno, organizadas em relação a algum índice de ordenamento. A característica que diferencia uma série temporal previsível de uma amostra aleatória qualquer de dados é a presença de correlação estatística ou dependência sequencial nos dados observados ([EHLERS, 2007](#)). Essa característica permite que seja possível elaborar modelos para prever, por exemplo, qual será o próximo valor observado na série temporal ([KIRCHGÄSSNER; WOLTERS; HASSLER, 2012](#)). Os objetivos em analisar séries temporais podem ser vários, como por exemplo: descrever as propriedades da série, explicar o comportamento de uma série através da análise de outra, previsão dos pontos futuros da série e estudar o fenômeno que controla a série. Contudo, um dos principais objetivos no estudo de uma série é a sua previsão. Para isso devem ser consideradas muitas variáveis intrínsecas à série e também fenômenos do ambiente externo que possam estar relacionados a ela. Existem diversos fenômenos geradores de séries temporais, dentre eles existem os relativos ao mercado financeiro. Séries temporais de mercado envolvem problemas do mundo real. E se tratando do mundo real, as relações que existem entre os dados geralmente são não lineares. De acordo com Xie *et al.* ([XIE et al., 2021](#)), embora existam técnicas mais sofisticadas de previsão de séries temporais, muitos profissionais de mercado financeiro ainda recorrem a abordagens tradicionais baseadas em modelos lineares, como a regressão linear múltipla. No entanto, o uso de modelos lineares pode levar a previsões imprecisas, especialmente em

mercados financeiros altamente voláteis e complexos. Séries de mercado apresentam muitas vezes características que dificultam a sua predição, como: irregularidades, volatilidade e sazonalidades (ZIVOT; WANG, 2013). Nessa pesquisa, serão utilizadas séries temporais relacionadas ao mercado financeiro, como por exemplo: bolsa de valores, flutuação de preço de ações, cotação de moedas, entre outros.

Ao longo do tempo, foram desenvolvidas várias metodologias para a predição de séries temporais. Talvez uma das mais comentadas seja o *Auto Regressive Integrated Moving Average* (ARIMA) (BOX et al., 2015). Esse é um modelo linear que reúne dois outros modelos de previsão desenvolvidos pelos mesmos autores que são o de médias móveis e o auto regressivo além de um processo de integração desses dois modelos. O modelo auto regressivo (AR) para a variável de interesse é baseado em uma combinação linear dos valores passados da própria variável, que nesse caso são os pontos da série. O modelo de médias móveis (MA) ao invés de usar os valores passados da série temporal para gerar uma regressão, utiliza os resíduos passados do preditor para formar essa regressão (BOX et al., 2015).

Apesar da metodologia ARIMA ser vastamente utilizada na literatura, a maioria dos problemas do mundo real são não-lineares, o que gera uma perda na acurácia da previsão (LJ; NETO; FERREIRA, 2009). Com vista nesse problema, várias outras metodologias de previsão de séries temporais foram propostas no campo da inteligência artificial. Dentre elas, uma que merece destaque são as redes neurais artificiais (RNAs). RNAs comprovaram sua capacidade de previsão de séries temporais em diversos trabalhos da literatura ((MOODY; DARKEN, 1989)(ZHANG; PATUWO; HU, 1998),(ZHANG, 2003),(JUNIOR; FERREIRA, 2008),(FERREIRA; VASCONCELOS; ADEODATO, 2008), (SHAH et al., 2016), (FATHI, 2019),(TORRES et al., 2021)). Esses modelos têm a capacidade de representar o complexo relacionamento não linear que existe entre os dados desconsiderando qualquer premissa inicial sobre a natureza dessas relações (GHIASSI; SAIDANE; ZIMBRA, 2005). Uma rede neural artificial é inspirada no funcionamento do cérebro humano que automaticamente classifica e extrai padrões para representar a informação desejada, caracterizando o aprendizado (BISHOP; NASRABADI, 2006).

Chatfield (CHATFIELD, 2000) afirma que uma boa parte da literatura voltada a séries temporais assume que existe um único modelo intrínseco à série. Contudo, o autor afirma que mesmo que exista um único modelo, ele será raramente conhecido *a priori* e mesmo assim não há garantias que ele se ajuste à série selecionada. Na mesma linha Hyndman e Athanasopoulos (HYNDMAN; ATHANASOPOULOS, 2018) questionam a suposição de um único modelo intrínseco para séries temporais. Os autores destacam o conceito de "hierarquia de previsão", onde múltiplos modelos podem ser necessários

para capturar os padrões e dinâmicas subjacentes em diferentes níveis de uma série temporal, como componentes sazonais, de tendência e irregulares. Autores como Neuman (NEUMAN, 2003) inclusive comentam que utilizar um único modelo pode conduzir a vieses estatísticos e à subestimação da real incerteza subjacente à série temporal. Baseado nessas argumentações, autores como Wichard e Ogarzaleke (WICHARD; OGORZALEK, 2004) afirmam que a combinação de modelos tende a conduzir a resultados mais acurados. É importante ressaltar que ao combinar modelos, deve-se selecionar os modelos que mais se ajustem à série temporal analisada para que se possa de fato aumentar a acurácia da previsão (FIRMINO; NETO; FERREIRA, 2014). Uma das maneiras mais intuitivas de se pensar em combinação linear de modelos é a média ponderada dos preditores, onde os pesos de um dado modelo estão diretamente relacionados com a covariância dos erros das previsões (MENEZES; BUNN; TAYLOR, 2000). Existem também outras maneiras de combinar preditores como estimadores de mínima variância (MENEZES; BUNN; TAYLOR, 2000) e funções de Cópulas (OLIVEIRA; TORGO, 2015). Uma cópula pode ser definida como uma função que associa distribuições de probabilidade marginais para formar uma distribuição de probabilidade conjunta multivariada, com a finalidade de juntar todas as distribuições marginais sem perda de informação (FIRMINO; NETO; FERREIRA, 2014). Essas funções medem as dependências entre as variáveis (NELSEN, 2007).

No mesmo sentido de compreender as dependências e relações entre as variáveis em um conjunto de dados, as técnicas de *Deep learning* são arcabouço de metodologia de aprendizagem de máquina baseado em representação de dados através de aprendizado. Esse aprendizado pode ser ou não supervisionado (SCHMIDHUBER, 2015). A ideia é alcançar uma representação dos padrões existentes entre as informações de forma similar ao sistema nervoso biológico, mapeando as relações que existem entre várias respostas neurais (BENGIO; GOODFELLOW; COURVILLE, 2017). Técnicas de *deep learning* visam criar sistemas que consigam entender essas relações em larga escalas com grande quantidade de informação. Geralmente, essa quantidade de processamento exige arquiteturas que consigam trabalhar com processamento massivo de dados. Algumas das arquiteturas baseadas em *deep learning* são *deep neural networks*, *deep belief networks* e *recurrent neural networks* (RNN). Em especial, RNN envolve conexões entre os nós que formam um grafo direcionado ou não direcionado ao longo de uma sequência temporal (BENGIO, 2009). Assim como as redes neurais mais tradicionais, redes RNN podem modelar as complexas relações não lineares. A arquitetura da RNN gera um modelo composto onde o objeto é expresso como uma composição de camadas. As camadas extras possibilitam a composição de características das camadas mais iniciais, o que potencialmente modela a complexidade entre os dados de forma mais eficiente do que os modelos de "redes neurais rasas". Com isso, é possível avaliar que o uso dessas técnicas pode ser muito interessante para a predição

de séries de mercado onde as relações não lineares entre os dados estão muito presentes. Um exemplo de RNN é a rede *Long Short Term Memory* (LSTM). A LSTM é capaz de aprender dependências de ordem em problemas de previsão de sequência, o que é uma característica necessária e importante em situações complexas, como a previsão de séries temporais.

1.2 Objetivos

Esta tese tem como objetivo estudar a precisão da combinação de arquiteturas de *deep learning* utilizando funções de cópulas na previsão de séries temporais do mercado financeiro. O estudo busca empregar um conjunto de séries de previsão de modelos de mercado para capturar a relação não linear presente nos dados dessas séries, combinando-os por meio de funções de cópulas. Além disso, serão utilizadas métricas para avaliar a precisão das previsões tanto individualmente quanto quando combinadas. A seguir, serão apresentados o objetivo geral e os objetivos específicos desta tese

1.2.1 Objetivo geral

Estudar a acurácia de combinação de arquiteturas de *deep learning* através de funções de cópulas na predição de séries temporais relacionadas ao mercado financeiro.

1.2.2 Objetivos específicos

- Estudo e coleta de séries temporais financeiras relevantes para o desenvolvimento do trabalho;
- Estudar modelos lineares e não lineares de acoplamento de preditores de séries temporais;
- Avaliar a acurácia de modelos de preditores acoplados e ponderar a sua representatividade em reduzir a magnitude do erro da previsão;
- Experimentar a combinação de diferentes modelos de cópulas

1.3 Estrutura do documento

Esse trabalho de tese de doutorado encontra-se dividido nos seguintes capítulos. O Capítulo 2 promove uma revisão de literatura a respeito de alguns temas pertinentes à modelagem e previsão de séries temporais. O Capítulo 3 apresenta um arcabouço

teórico sobre combinação de modelos a partir de técnicas que serão utilizadas nessa tese. É apresentado no Capítulo 4 os trabalhos relacionados aos estudo realizado nessa tese. Apresenta-se no Capítulo 5 as séries temporais utilizadas nas predições. No Capítulo 6 é abordada a metodologia utilizada para a previsão das séries temporais desse trabalho. O Capítulo 7 contém os resultados encontrados na tese de doutorado. Por fim, é apresentado no Capítulo 8 a conclusão da tese.

2 Modelagem e previsão de séries temporais

2.1 Séries temporais

Uma série temporal é um conjunto de dados indexados, mais comumente pelo tempo. Uma série temporal discreta é uma sequência de pontos igualmente espaçados no tempo. Logo, é uma sequência de dados em tempo discreto. Apesar dos estudos de séries temporais terem raízes históricas que remontam ao século XVIII, quando Abraham de Moivre estudou a mortalidade em diferentes idades, um dos primeiros trabalhos a formalizar o estudo de modelos de séries temporais foi o trabalho de Yule (YULE, 1927) onde o autor analisa séries temporais periódicas e discute a periodicidades em séries temporais, com foco na série de manchas solares. Desde então as séries temporais passaram a ser usadas em várias áreas como economia (bolsa de valores) (MORETTIN, 2017), medicina (eletrocardiograma) (MALETZKE et al., 2011), meteorologia (temperaturas mensais) (SORJAMAA et al., 2007), etimologia (casos de COVID19 em determinada região) (VOKÓ; PITTEK, 2020), fenômenos naturais (manchas solares)(SHENSHI; ZHIQIAN; JITAI, 1999) e outras mais. Formalmente, considere Y uma série temporal em que:

$$Y = \{y_t \in \mathbb{R} | t = 1, 2, 3, \dots, N\}, \quad (2.1)$$

onde y_t é uma observação coletada no instante t e N é o número de observações da série. Para exemplificar séries temporais, será apresentada a série dos valores de fechamento das ações da empresa Google. Apresenta-se na Figura 1 essa série onde o eixo horizontal representa o tempo e o eixo vertical representa o valor diário de fechamento das ações.

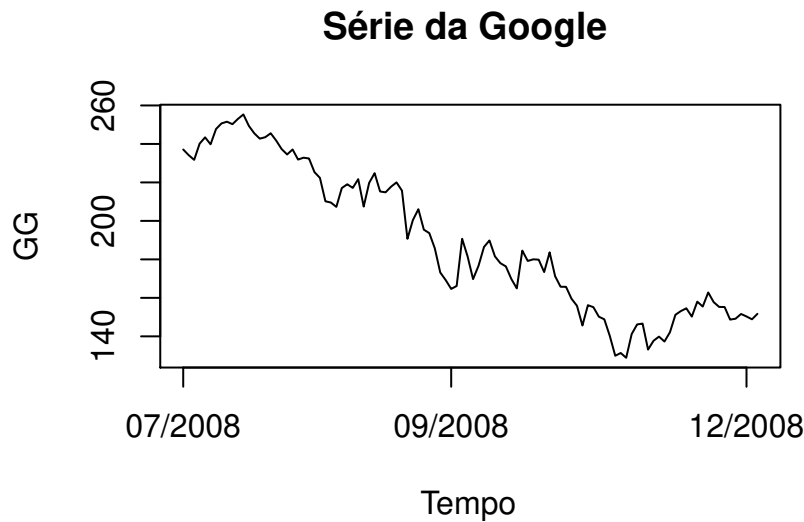


Figura 1 – Série temporal do fechamento das ações da empresa Google de julho até dezembro de 2008

Algumas propriedades de séries temporais são muito importantes, especialmente no que se diz respeito à sua predição. A sazonalidade de uma série é um certo padrão que tende a se repetir a cada determinado período de tempo. Outra característica de uma série temporal é a tendência. Uma série pode exibir uma tendência de crescimento ou decrescimento de seus valores. Observando a série temporal da Google, é possível inferir que as ações da empresa nesse período de tempo apresenta uma leve tendência decrescente ao longo dos anos. Uma característica muito importante de uma série temporal é a sua estacionariedade. Esse é um atributo elementar no que se diz respeito à previsão da série. Um processo é dito estacionário se a sua média e variância permanecem constantes ao longo do tempo. Logo, sendo Y_t um ponto futuro e ainda não observável da série, μ a média, σ^2 a variância e γ a covariância do processo, ele é considerado estacionário (estacionariedade fraca) se: $E[Y_t] = \mu_t = \mu$; média constante para qualquer tempo t , sua variância é constante, representado por:

$$E[(Y_t - \mu)^2] = E[(Y_{t-s} - \mu)^2] = \sigma_y^2 \quad (2.2)$$

e a sua covariância também é constante, descrita por

$$E[Y_t, Y_s] = [(Y_t - \mu_t)(Y_s - \mu_s)] = \gamma_{(|x-s|)} \quad (2.3)$$

(KIRCHGÄSSNER; WOLTERS; HASSLER, 2012). Para uma série que apresente tendência ou sazonalidade, não há estacionariedade porque esses componentes vão afetar o valor da média ou mesmo da variância da série temporal em diferentes períodos (HYNDMAN; ATHANASOPOULOS, 2014)

2.1.1 Séries temporais financeiras

O mercado financeiro se baseia na negociação de ações de empresas. Uma ação corresponde a menor parcela de um capital de uma empresa. Os indivíduos que adquirem ações de uma determinada empresa passam a ser sócios dessa empresa adquirindo também os deveres e direitos inerentes a sociedade. Uma bolsa de valores é o ambiente em que ocorrem as negociações de compra e venda das ações (FURGANG, 2010). As ações da bolsa sofrem flutuação nos preços pelos mais diversos fatores, como por exemplo: políticas externas, administração da companhia, mudança de tendência de consumo e outros mais. A prática da especulação de compra e venda das ações das empresas na bolsa de valores é o que gera a movimentação no mercado financeiro. Toda essa movimentação é o que gera o conjunto de dados das séries temporais financeiras.

Uma série temporal é dita determinística se os valores futuros da série podem ser descritos exatamente por alguma função matemática. Já se esses valores futuros puderem ser descritos apenas em termos de uma distribuição de probabilidade, então a série temporal é chamada de não determinística ou estocástica. Um processo estocástico é composto por um conjunto de variáveis aleatórias (ROSS, 1995). Esse processo estocástico é uma sequência de variáveis aleatórias que estão definidas em um conjunto Ω , denominado espaço de estados. Um processo estocástico pode ser definido por: $Y = \{Y_{t,\theta}; t \in T; \Theta \in \Omega\}$ onde Y é um conjunto de variáveis aleatórias definidas sobre um espaço de probabilidade $\{\Omega, \mathcal{F}, P\}$ e indexadas pelos elementos do conjunto de parâmetros T onde $T \in \mathbb{R}$ e para cada $\Theta \in \Omega$ fixado tem-se uma série temporal $Y_{t,\theta}$.

As séries financeiras são frequentemente modeladas como processos estocásticos devido à natureza incerta e imprevisível dos preços dos ativos ao longo do tempo. Segundo Tsay (TSAY, 2010), “O preço de um ativo financeiro evolui ao longo do tempo e forma um processo estocástico. Os preços observados são uma realização do processo estocástico subjacente”. Ao considerar que os preços observados são uma realização do processo estocástico subjacente, é reconhecido que cada ponto de dados é uma amostra de um conjunto infinito de possíveis trajetórias que o preço poderia seguir. Isso implica que a previsão precisa dos preços futuros é desafiadora, pois envolve a análise das propriedades estocásticas do processo subjacente. No geral, não existe uma grande diferença entre avaliar séries temporais financeiras das derivadas de outras áreas como epidemiologia, oceanografia, meteorologia, saúde dentre outros, desde que partindo-se do princípio que as séries sejam estocásticas. Contudo, existe uma característica muito marcante nas séries financeiras, em especial as que são derivadas da flutuação de ativos financeiros, que é convencionalmente chamada de volatilidade (MORETTIN, 2017). A volatilidade representa a variância condicional, que caso não se repita para toda série, não coincide com a variância

global da série observada.

Uma característica marcante de séries temporais financeiras é que em geral elas não são linearmente dependentes, mas existe dependência entre as observações da série. Isso implica dizer que em geral modelos lineares como os da família ARIMA não são apropriados para estudar os modelos dessas séries. Além do mais, as séries financeiras da vida real apresentam características como tendência e sazonalidade que dificultam a adequação dos modelos lineares.

A fim de exemplificar uma série temporal financeira, apresenta-se na Figura 2 a representação da séries da S&P 500 com a flutuação dos valores dos retornos das ações entre os períodos de 2013 a 2022. A série S&P 500 é um índice composto por quinhentos ativos cotados nas bolsas de NYSE ou NASDAQ, qualificados devido ao seu tamanho de mercado, sua liquidez e sua representação de grupo industrial.

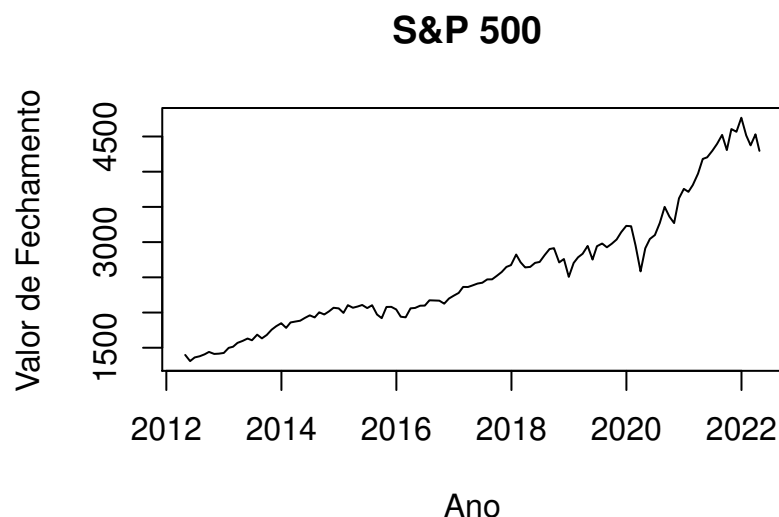


Figura 2 – Série dos valores de fechamento da S&P 500 entre os anos de 2013 e 2022

É possível observar que na série que representa o S&P500 é apresentada uma tendência crescente do valor das ações. Essa tendência pode ser constatada através do teste de Mann-Kendall (MCLEOD, 2005), que aplicado a essa série apresenta valor $\tau = 0.913$ e p -valor $\leq 2.22e^{-16}$. Como o p -valor é menor que 0.05, infere-se que a série apresenta tendência.

2.2 Modelos de previsão de séries temporais

2.2.1 Modelos ARIMA

O *Auto Regressive Integrated Moving Average* (ARIMA) é um modelo estatístico linear para a previsão de séries temporais formalizado pelos autores no livro de Box e Jenkins (BOX et al., 2015). Esse é um dos modelos mais comentados na literatura para predição de séries temporais. O modelo ARIMA é uma integração de dois modelos, o auto regressivo e o de médias móveis. O modelo geral é classificado em ARIMA(p,d,q), onde:

- p é o número de termos auto regressivos;
- d é o número de diferenciações necessárias para estacionariedade;
- q é o número de termos de médias móveis.

O modelo auto regressivo para a variável de interesse é baseado em uma combinação linear dos valores passados da variável, que nesse caso são os pontos da série. Então um modelo auto regressivo de ordem p pode ser definido como:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t. \quad (2.4)$$

Onde ε_t representa o ruído branco da série, ϕ_p os parâmetros auto regressivos e y_t é um determinado ponto na série temporal.

O modelo de médias móveis, ao invés de usar os valores passados da série temporal para gerar uma regressão, utiliza os resíduos da previsão. O modelo de médias móveis de ordem q pode ser definido como:

$$y_t = \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \quad (2.5)$$

em que ε_t representa o ruído branco da série, θ_q são os parâmetros do modelo de médias móveis e y_t é um determinado ponto na série temporal.

Algumas séries são não estacionárias, mas apresentam um comportamento homogêneo na sua estacionariedade, por conta disso é necessário haver uma quantidade de diferenciações d para que a série passe a apresentar um comportamento estacionário. A primeira diferenciação é representada por:

$$\nabla y_t = y_t - y_{t-1}, \quad (2.6)$$

em que ∇y_t é a derivação de determinado ponto da série em uma primeira aproximação.

Uma função muito importante para estimar os parâmetros dos modelos ARIMA é a Função de Auto Correlação. A Função de Auto Correlação utiliza os coeficientes de auto covariância para gerar uma função que revela a correlação entre quaisquer janelas de valores da série (BOX et al., 2015). Essa função é definida por:

$$\rho(\tau) = \frac{\gamma_\tau}{\gamma_0} = \frac{E[(y_t - \mu)(y_{t+\tau} - \mu)]}{E[(y_t - \mu)^2]}. \quad (2.7)$$

A Figura 3 apresenta a função de auto correlação para a série temporal da Google.

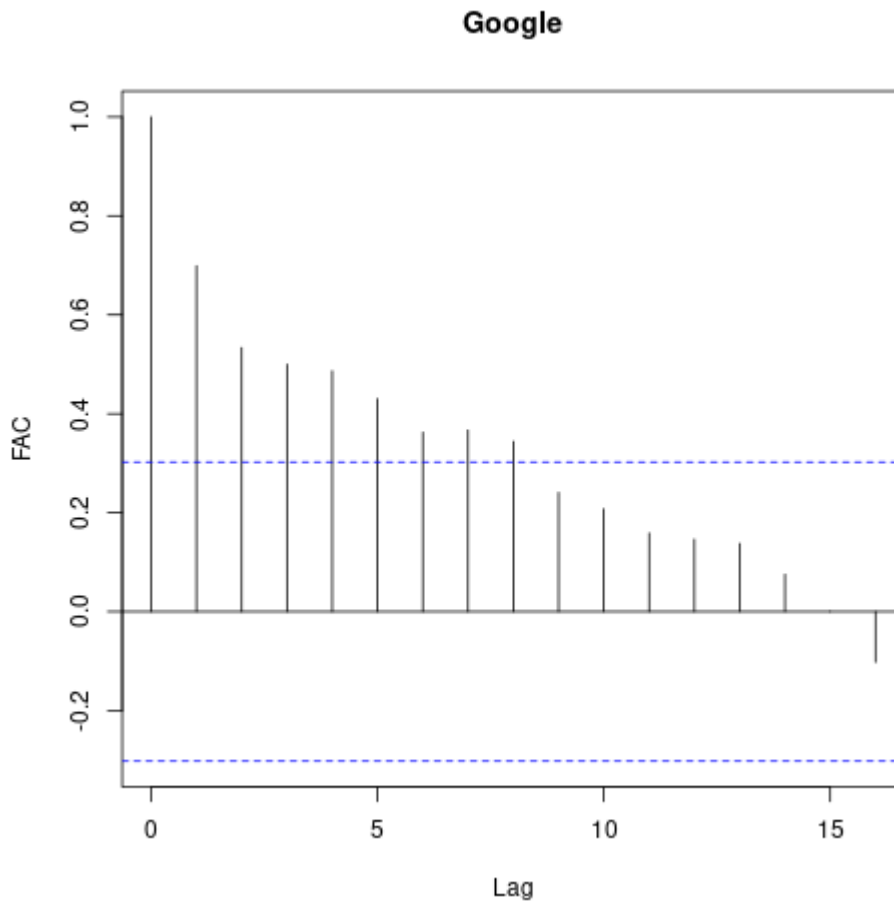


Figura 3 – Função de auto correlação da série da Google apresentada na Figura 1.

Outra função utilizada para estimar os modelos da previsão é a Função de Auto Correlação Parcial. A Função de Auto Correlação Parcial, entre quaisquer dois pontos da série, é a correlação que permanece se o impacto de todas os outros pontos da série fosse eliminado. Sendo ϕ_{kj} o j -ésimo coeficiente em uma função auto regressiva de ordem k e ϕ_{kk} o último elemento, então de acordo com (BOX et al., 2015) tem-se:

$$\rho_j = \phi_{k1}\rho_{j-1} + \dots + \phi_{k(k-1)}\rho_{j-k+1} + \phi_{kk}\rho_{j-k}; \text{ com } j = 1, 2, 3, \dots, k \quad (2.8)$$

As equações de Yule-Walker (ESHEL, 2003) levam a essa expressão, que depois de resolvidas para um caso específico de um modelo AR(2) por exemplo resulta em:

$$\phi_{22} = \frac{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & \rho_2 \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{vmatrix}} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2} \quad (2.9)$$

A Figura 4 apresenta a Função de Auto Correlação Parcial da série temporal da Google (CLETTE et al., 2014).

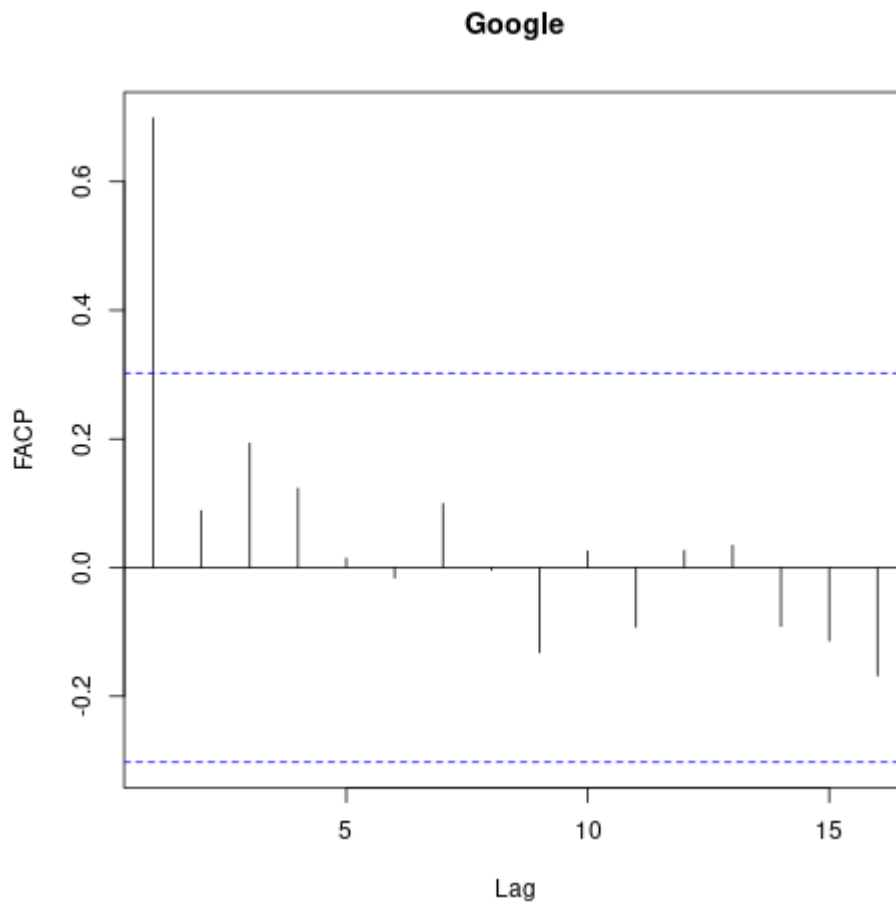


Figura 4 – Função de auto correlação parcial da série da Google apresentada na Figura 1.

2.2.2 Redes neurais artificiais

Redes neurais artificiais (RNAs) são modelos de computador baseados no sistema nervoso humano. Eles têm a capacidade de adquirir e preservar conhecimento (baseados em informações) e podem ser definidos como uma coleção de unidades de processamento

representadas por neurônios artificiais, interligadas por um grande número de interconexões (sinapses artificiais), e implementadas por vetores e matrizes de peso sináptico (SILVA et al., 2017).

As redes neurais artificiais são uma tecnologia baseada em estudos do cérebro e do sistema nervoso, como ilustrado na Figura 5. Essas redes emulam uma rede neural biológica, embora utilizem um conjunto simplificado de conceitos dos sistemas neurais biológicos. Os modelos de redes neurais simulam a atividade elétrica do cérebro e do sistema nervoso. Os elementos de processamento, conhecidos como “perceptrons” (neurônios), estão conectados a outros elementos de processamento. Geralmente, os neurônios são organizados em camadas ou vetores, onde a saída de uma camada serve como entrada para a próxima. Cada neurônio pode estar conectado a todos ou a um subconjunto de neurônios na camada seguinte, simulando as conexões sinápticas do cérebro. Os sinais de dados ponderados que entram em um neurônio simulam a excitação elétrica de uma célula nervosa e, conseqüentemente, a transferência de informações dentro da rede. Os valores de entrada para um elemento de processamento são multiplicados por um peso de conexão, $w_{n,m}$, que simula o fortalecimento das vias neurais no cérebro. É através do ajuste dos pesos que aprendizagem é emulada em RNAs (WALCZAK, 2018).

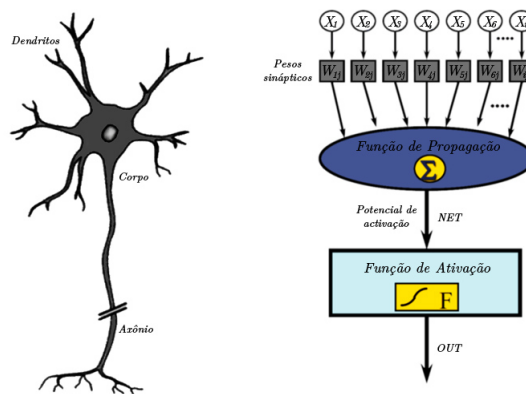


Figura 5 – Analogia entre modelos biológico e computacional de neurônios. Fonte: https://www.researchgate.net/figure/Figura-19-Neuronio-biologico-esquerda-e-neuronio-artificial-direita_fig18_300015903 , 2022

A ideia de combinar vários elementos de processamento em uma rede remonta ao trabalho de McCulloch e Pitts no início da década de 1940 (MCCULLOCH; PITTS, 1943), enquanto Hebb, em 1949, é creditado como o primeiro a definir uma regra de aprendizado para explicar o comportamento das redes neurais (HEBB, 1949).

No final da década de 1950, Rosenblatt desenvolveu o primeiro neurônio artificial chamado de *perceptron* (ROSENBLATT, 1958). No entanto, em 1969, Minsky e Papert

publicaram o livro “Perceptrons”, demonstrando as limitações computacionais das redes neurais de camada única que estavam em uso na época (Minsky & Papert, 1969).

Essas limitações teóricas resultaram em uma diminuição no financiamento e na pesquisa em redes neurais. No entanto, em 1986, McClelland e Rumelhart, publicaram o livro “Processamento distribuído e paralelo” (RUMELHART et al., 1986). Nesse trabalho, eles desenvolveram o algoritmo de retropropagação, que possibilitou o treinamento de múltiplas camadas de perceptrons e a introdução das camadas ocultas nas redes neurais artificiais. Isso marcou o surgimento das MLPs (*perceptron* de múltiplas camadas).

A descoberta das MLPs e do algoritmo de retropropagação revitalizou os esforços de pesquisa e desenvolvimento em redes neurais artificiais. Desde então, tem havido uma renovação significativa no campo das redes neurais, impulsionada por essas descobertas.

2.2.2.1 Arquiteturas de redes neurais artificiais

O modelo mais simples de rede neural desenvolvido e que inclui as principais características de um rede neural é o modelo de Rosenblatt (ROSENBLATT, 1958) chamado de *perceptron*. Esse modelo é constituído de apenas um neurônio que pode resolver problemas com dados linearmente separáveis. Esse perceptron é representado na Figura 6

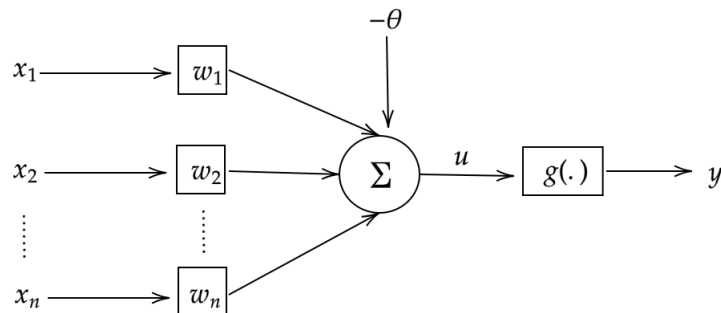


Figura 6 – Diagrama do Perceptron.

Como apresentado na Figura 6, os múltiplos sinais de entrada provenientes do ambiente externo (entrada) são representados pelo conjunto $\{x_1, x_2, x_3, \dots, x_n\}$, análogo aos impulsos elétricos externos recolhidos pelos dendritos no neurônio biológico. A pesagem realizada pelas junções sinápticas da rede são implementadas no neurônio artificial como um conjunto de pesos sinápticos $\{w_1, w_2, \dots, w_n\}$. Analogamente, a relevância de cada uma das entradas do neurônio $\{x_i\}$ é calculada multiplicando-se eles por seu peso sináptico correspondente $\{w_i\}$, ponderando assim todos os informações externas que chegam ao

neurônio. Assim, é possível verificar que a saída do neurônio, denotado por u , é a soma ponderada de suas entradas. Considerando a Figura 6, é possível ver que o neurônio artificial é composto por sete elementos básicos:

- Os sinais de entrada (x_1, x_2, \dots, x_n) são os sinais ou amostras provenientes do ambiente externo e representam os valores assumidos pelas variáveis de entrada; Os sinais de entrada são normalmente transformados (conforme função de ativação) para aumentar a eficiência computacional de algoritmos de aprendizado.
- Os pesos sinápticos (w_1, w_2, \dots, w_n) são os valores usados para ponderar cada uma das variáveis de entrada, o que permite quantificar sua relevância com respeito à funcionalidade do neurônio;
- O agregador linear (Σ) reúne todos os sinais de entrada ponderados pela sináptica de pesos para produzir um sinal de ativação;
- Limiar de ativação ou viés (θ) é uma variável usada para especificar o limiar que o resultado produzido pelo agregador linear deve ter para gerar um valor de disparo para a saída do neurônio;
- O potencial de ativação (u) é o resultado produzido pela diferença entre o agregador linear e o limiar de ativação. Se este valor for positivo, ou seja, se $u \leq \theta$, então o neurônio produz um potencial excitatório; caso contrário, será inibitório;
- Função de ativação (g) cujo objetivo é limitar a saída do neurônio dentro de um intervalo razoável de valores, assumidos por sua própria imagem funcional;
- Saída (y) consiste no valor final produzido pelo neurônio dado um determinado conjunto de sinais de entrada, e também pode ser usado como entrada para outros neurônios.

A expressão a seguir expressa o funcionamento do perceptron:

$$y = g(u), \quad (2.10)$$

em que u é expresso por:

$$u = \sum_{i=1}^n w_i \cdot x_i - \theta \quad (2.11)$$

Assim, a operação do neurônio artificial pode ser resumida pelas seguintes etapas:

- Apresentação um conjunto de valores ao neurônio, representando as variáveis de entrada;
- Multiplicação de cada entrada do neurônio pelo seu peso sináptico correspondente;
- Obtenção do potencial de ativação produzido pela soma ponderada da entrada de sinais e subtrair o limiar de ativação;
- Aplicação de uma função de ativação adequada para limitar a saída do neurônio.
- Por fim, compilação da saída empregando a função de ativação neural no potencial de ativação.

A função de ativação mais comum utilizada é a função logística (GOODFELLOW; BENGIO; COURVILLE, 2016). O resultado de saída produzido pela função logística sempre assumirá valores reais entre zero e um. Sua expressão matemática é dada por:

$$g(u) = \frac{1}{1 + e^{-\beta \cdot u}} \quad (2.12)$$

em que β é uma constante real associada à inclinação da função em seu ponto de inflexão. É ilustrado na Figura 7 a função de ativação logística:

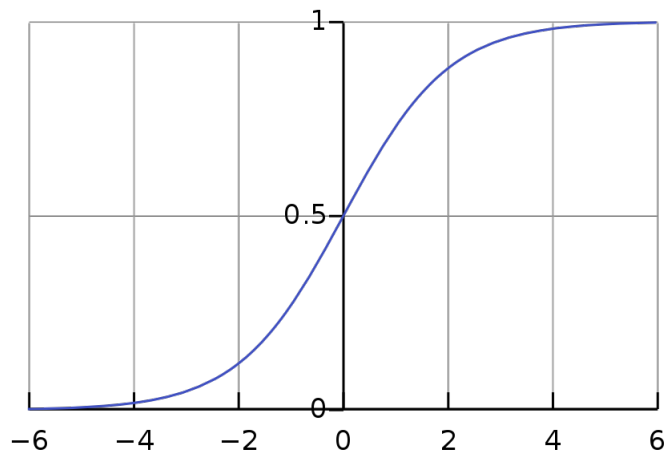


Figura 7 – Representação da função logística.

Uma agregação de perceptrons divididos em camadas forma um tipo de rede neural muito citada em diversos trabalhos que é a rede neural de múltiplas camadas ou mais popularmente conhecida por MLP (*multi layer perceptron*)

2.2.2.2 Redes neurais de múltiplas camadas

Uma rede neural de múltiplas camadas é composta basicamente de um conjunto de *perceptrons* divididos em três camadas: entrada, oculta e saída (SILVA et al., 2017). Essas camadas são definidas como:

- Camada de entrada: Esta camada é responsável por receber informações (dados), sinais, características ou medições do ambiente externo. Essas entradas (amostras ou padrões) são transformadas para valores dentro dos limites produzidos pela funções ativação. Essa transformação resulta em uma melhor precisão numérica para o operações matemáticas realizadas pela rede;
- Camada oculta: Essas camadas são compostas por neurônios que são responsáveis por extrair padrões associados ao processo ou sistema que está sendo analisado. Essas camadas realizam a maior parte do processamento interno de uma rede neural;
- Camada de saída: Essa camada é responsável por produzir e apresentar as saídas finais da rede, que resultam do processamento realizada pelos neurônios nas camadas anteriores.

A Figura 8 apresenta um exemplo de rede MLP totalmente conectada.

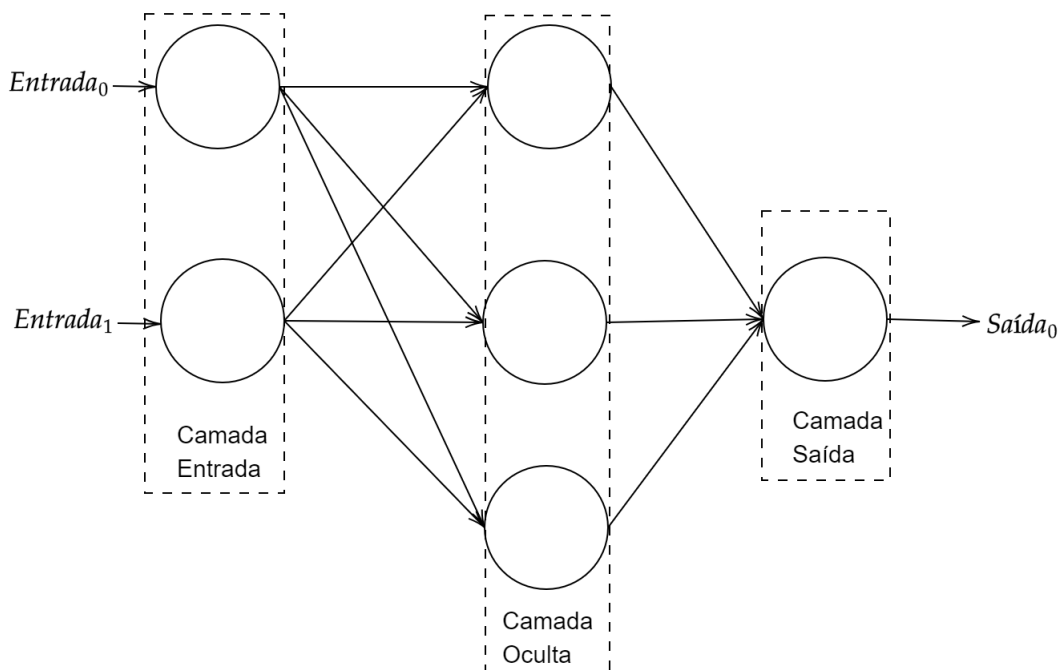


Figura 8 – Camadas de uma rede neural MLP.

Dentro da arquitetura das redes neurais, existem diversos fatores a respeito de sua arquitetura. Talvez um dos mais relevantes esteja relacionado ao treinamento das redes neurais. Basicamente, o treinamento de modelos de redes neurais consiste em buscar os melhores resultados através de várias combinações de topologias de rede. Esse ajustes nos pesos das redes podem ser feitos de maneira supervisionada ou não. Geralmente, no que diz respeito ao treinamento das redes, esse processo costuma consumir muito tempo e processamento até que seja encontrado o modelo de rede (TISSOT; CAMARGO; POZO, 2012). Existem duas categorias de treinamento em redes neurais artificiais: os supervisionados e os não supervisionados.

A estratégia de aprendizado supervisionado é baseada na disponibilidade de um conjunto de saídas desejadas para um determinado conjunto de entradas. Para isso se faz necessário uma tabela com dados de entrada/saída, que contém os atributos e valores, que representam o processo e seu comportamento. É a partir dessa informação que as estruturas neurais vão formular o seu balizamento sobre os ajustes dos pesos da rede e consequentemente o aprendizado da rede neural. Já a estratégia não supervisionada não necessita de nenhum conhecimento prévio a respeito do conjunto de saída. Ao invés disso, a rede se organiza procurando particularidades entre os elementos que compõem todo o conjunto amostral, identificando subconjuntos que apresentam semelhanças. O algoritmo de aprendizado ajusta os pesos sinápticos e limites da rede para refletir esses subconjuntos dentro da sua topologia (SILVA et al., 2017).

Um dos algoritmos mais populares de treinamento para redes neurais é o *backpropagation*. Esse algoritmo consistem um conjunto de etapas estruturadas a fim de ajustar os pesos da rede (CILIMKOVIC, 2015). Rojas (ROJAS, 2013) afirma que após os pesos das camadas intermediárias serem escolhidos aleatoriamente o algoritmo de *backpropagation* realiza as seguintes 4 etapas:

- Processamento das informações pelas camadas da rede;
- *Backpropagation* para a camada de saída;
- *Backpropagation* para a a camada intermediária;
- Atualização dos pesos;

Esse processo é repetido até que os ajustes dos pesos ocasionem uma redução na *função de erro* (é uma medida quantitativa que avalia o quão bem um modelo de está realizando uma determinada tarefa) da rede, de modo que ela se torne suficientemente pequena.

2.2.2.3 Redes neurais recorrentes

As redes neurais recorrentes, ou recursivas, são modelos de redes neurais semelhantes aos modelos MLP, mas que possuem a característica da retroalimentação em seus nós, o que acaba promovendo algumas características como:

- Comportamento dinâmico;
- Habilidade de memorizar relacionamentos;
- Possibilidade de guardar informação nos seus nós intermediários.

As redes neurais recorrentes (RNR) surgiram inicialmente do trabalho de Hopfield (HOPFIELD, 1982), que se baseou nas arquiteturas recorrentes, sistemas dinâmicos e física estatística para formular o seu modelo (SILVA et al., 2017). Durante os anos foi havendo um aprimoramento das redes neurais recorrentes adaptadas às técnicas de *deep learning* como é o caso da LSTM.

Em uma rede recorrente, a ideia do processamento unidirecional é substituída pela de ciclos entre as unidades das redes. As camadas da rede podem se conectar com camadas anteriores e até com elas mesmas. Assim, a informação deixa de fluir em um único sentido e a saída da rede passa a não depender unicamente da entrada *a priori* da rede, mas também de entradas anteriores. Essa mudança de paradigma acaba ocasionando um ganho de “memória” na rede (NELSON, 2017). A RNR consegue, através dessa capacidade de armazenar padrões em sua memória, resolver problemas de regressão de maneira mais acurada pelo fato de conseguir armazenar padrões de curto e longo prazo na sua arquitetura. Contudo, o principal problema dessa arquitetura é o seu processo de treinamento, que costuma ser desafiador por conta do problema do acúmulo de informação ocasionado o que é chamado de *gradient boosting*. É apresentado na Figura 9 um modelo de RNR.

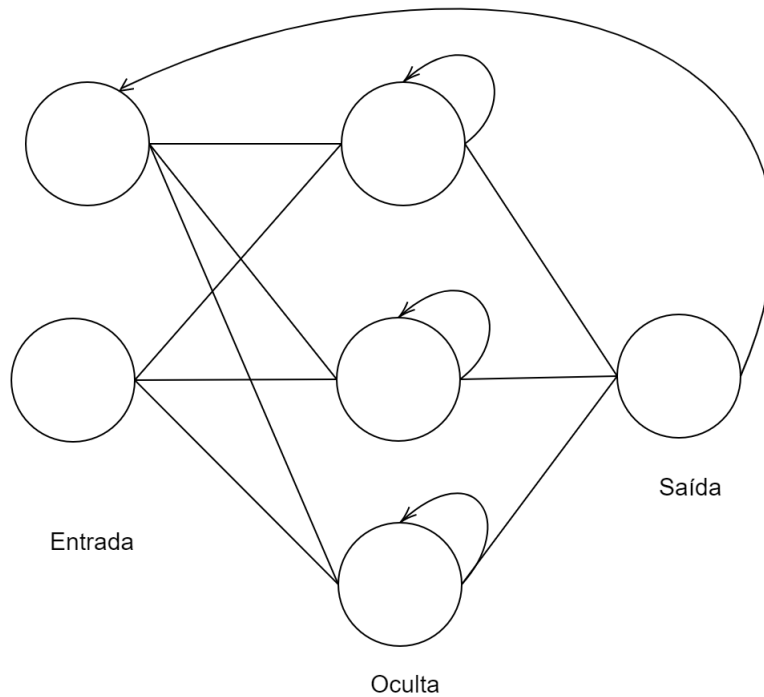


Figura 9 – Esquema de conexões de uma rede neural recorrente.

Observando apenas o neurônio da camada de saída de rede da Figura 9, é possível ver a recorrências existentes na arquitetura da rede, o que promove o armazenamento de memória de curto prazo. A Figura 10 apresenta um recorte detalhado das interações recursivas da rede neural.

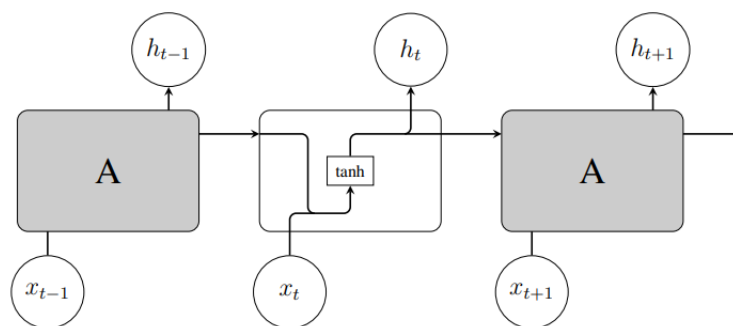


Figura 10 – Detalhamento do *loop* de um nó da RNR.

Suponha x um conjunto de valores provenientes de uma série temporal. Para observar o valor da série no instante t , x_t , através da Figura 10, necessita-se de processos realizados no instante $t - 1$. Por sua vez, x_t terá posteriormente seu processo utilizado para inferências do instante $t + 1$. Assim, tem-se que a interação na RNN pode ser definida por $x_t \cdot h_{t-1}$, passada por uma função de ativação \tanh (tangente hiperbólica).

2.2.2.4 Redes neurais em previsão de séries temporais

RNAs são utilizadas em diversas áreas do conhecimento para tratar de muitas problemáticas que são baseadas em classificação e previsão. Redes neurais tem sido usados em vários problemas de classificação, como por exemplo, classificação: de tipos de matérias baseados em vidro (EL-KHATIB; ABU-NASSER; ABU-NASER, 2019), tipos de câncer cerebral (JOSHI; RANA; MISRA, 2010), imagens biomédicas (BARZEKAR; YU, 2022), mensagens de correio eletrônico (ALGHOUL et al., 2018), sinais ultrassônicos (CHEN et al., 2022), degradação de pastagens (CHAGAS et al., 2009), qualidade dos frutos da manga (NAIK; DESAI, 2022) entre outros.

A maior parte dos modelos de séries temporais do mundo real apresenta um comportamento não linear (TEALAB, 2018), o que acaba prejudicando o desempenho dos modelos lineares. Devido ao fato de RNA conseguirem detectar melhor os relacionamentos não lineares presentes nos diversos modelos de séries temporais, esses modelos têm sido muito exploradas, alcançando resultados atraentes em comparação aos modelos lineares tradicionalmente utilizadas para previsão de séries temporais ((FERREIRA; VASCONCELOS; ADEODATO, 2008), (ZHANG; QI, 2005), (TEALAB; HEFNY; BADR, 2017), (JAIN; KUMAR, 2007), (YOLCU; EGRIOGLU; ALADAG, 2013), (CHENG et al., 2022), (GHEYAS; SMITH, 2009)). Além disso, a versatilidade dos modelos baseados em redes neurais permite que a rede neural, além de modelo de previsão, possa também ser utilizada como um combinador de modelos de previsão (DONALDSON; KAMSTRA, 1996).

2.2.3 *Long Short Term Memory*

Deep Learning (DL) é uma área do aprendizado de máquina que surgiu da interseção das áreas de redes neurais, inteligência artificial, modelagem baseada em grafos e otimização (LECUN; BENGIO; HINTON, 2015). Com o aperfeiçoamento dos processadores e a redução do custo das memórias dos computadores as teorias que embasavam o aprendizado profundo de máquina puderam ser aplicadas empiricamente resultando em novas metodologias. Uma das arquiteturas de DL é a *Deep Neural Network* (DNN). A característica de uma DNN se assemelha bastante ao de uma MLP, como na Figura 8, com camadas de entrada e saída, mas com a diferença de uma grande quantidade de camadas escondidas e neurônios conectados por camadas (BENGIO, 2009). Essas várias camadas escondidas têm a vantagem de aproximar funções extremamente complexas. Pode-se entender uma DNN como uma combinação de modelos individuais de regressão. Esses modelos (neurônios) são associados para fornecer combinações mais flexíveis de saídas. Esta flexibilidade permite que a rede se adéque a qualquer função. A Figura 11 apresenta uma representação de uma rede neural DNN, exemplificando a quantidade de conexões que podem haver em uma rede neural

desse tipo.

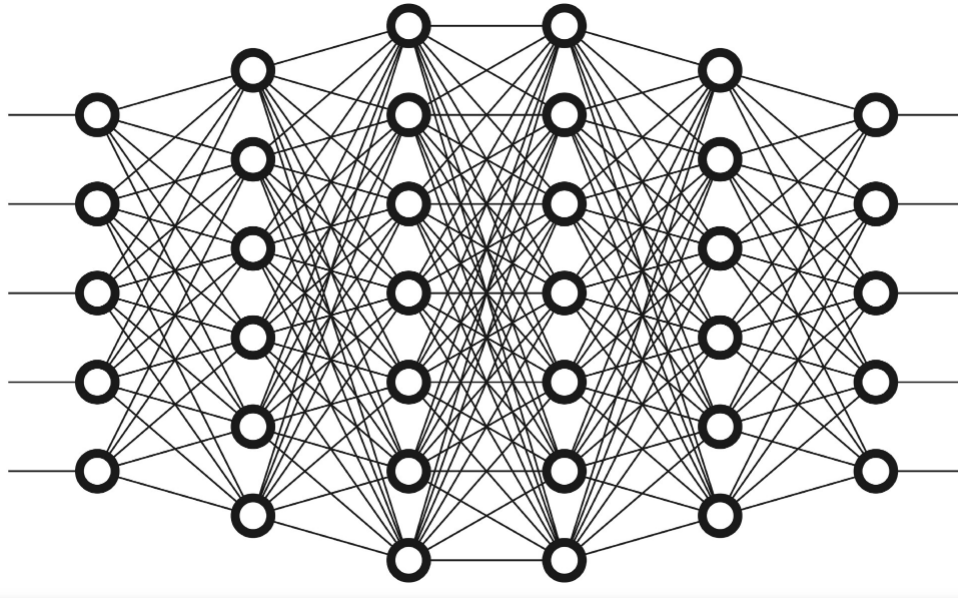


Figura 11 – Representação de uma rede *deep learning*. Fonte: <https://shre.ink/1hA>, 2023

No arcabouço das DNN, existem varias outras arquiteturas de DL como por exemplo a *Convolutional Neural Network* (CNN). A CNN é um modelo de rede DL baseado no córtex visual animal, muito utilizada para processamento e análise de imagens digitais. A CNN é uma rede neural composta por várias camadas que implementam a extração de recursos e depois a classificação do objeto (imagem). Cada uma dessas camadas tem a função de dividir, filtrar, extrair padrões das imagens e classificá-las (O'SHEA; NASH, 2015). Uma outra arquitetura muito utilizada para reconhecimento de padrões e problemas de regressão pela sua capacidade de armazenar memórias de curto e longo prazo é a LSTM.

A rede LSTM é uma rede neural recorrente capaz de armazenar dependências de longo prazo. Inicialmente apresentada no trabalho de Hochreiter e Schmidhuber (HOCHREITER; SCHMIDHUBER, 1997) foram refinadas e popularizadas para a resolução de uma grande quantidade de problemas. Esse tipo de arquitetura de rede neural é especialmente desenhada para evitar problemas muito comuns de RNN que é o armazenamento de muitas dependências de longo prazo. Habitualmente, as redes neurais recorrentes como apresentadas na Figura 9 apresentam uma estrutura de laço com uma única camada *than*. Já as LSTM, além dessa cadeia de estrutura, possui outras 3 camadas interagindo de uma maneira muito particular (BROWNLIE, 2017). A Figura 12 apresenta um nó de uma rede LSTM com suas quatro “portas” e suas interações.

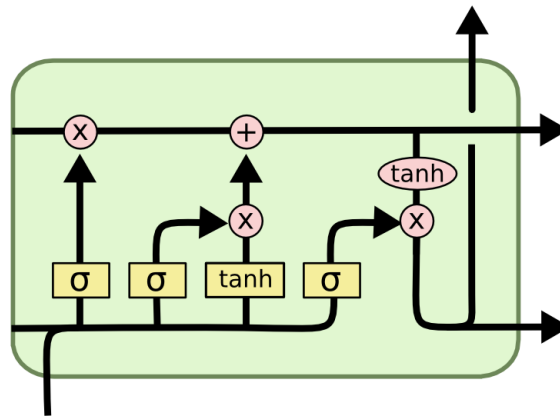


Figura 12 – Representação de um nó de uma rede LSTM. Fonte: <https://bit.ly/3m1kRTS>, 2015

Na Figura 12 os retângulos amarelos representam as camadas (portas da rede), os círculos vermelhos representam operações e as setas indicam vetores com o fluxo de informação da rede. O principal elemento da LSTM é a chave de estado, que é a linha horizontal no topo do diagrama. O estado da célula percorre toda o nó com apenas algumas interações lineares simples. É usual que a informação flua ao longo dela inalterada. A rede LSTM adiciona ou remove informações através de "portas". As portas são uma maneira opcional de deixar passar informações. Eles são compostos de uma camada sigmoide (σ) e uma operação de multiplicação pontual (x). A camada sigmoide assume valores entre 0 e 1 onde 0 significa a transmissão de nenhuma informação e 1 significa que toda a informação deve seguir o fluxo.

O primeiro estágio é aonde se decide que informações serão deletadas do estado da célula. Essa decisão é tomada por camada sigmoide (σ) "forget gate". Essa camada observa a entrada h_{t-1} e x_t e retorna um valor entre 0 e 1 para cada estado da célula C_{t-1} . A Figura 13 apresenta o recorte dessa dinâmica.

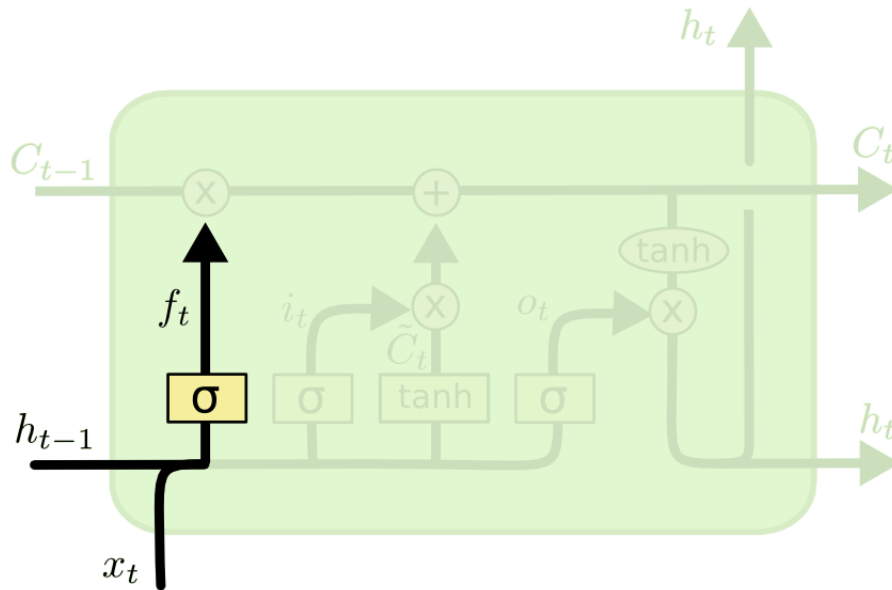


Figura 13 – Representação de um nó de uma rede LSTM. Fonte: <https://bit.ly/3m1kRTS>, 2015

Onde:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.13)$$

O segundo estágio é decidir a respeito de quais informações serão armazenadas na célula. Primeiro uma camada sigmóide (σ) chamada de *input gate* decide que valores serão atualizados e depois a camada *tanh* cria um vetor \tilde{C}_t de novos valores que podem ser adicionados a célula. A Figura 14 apresenta o recorte dessa dinâmica.

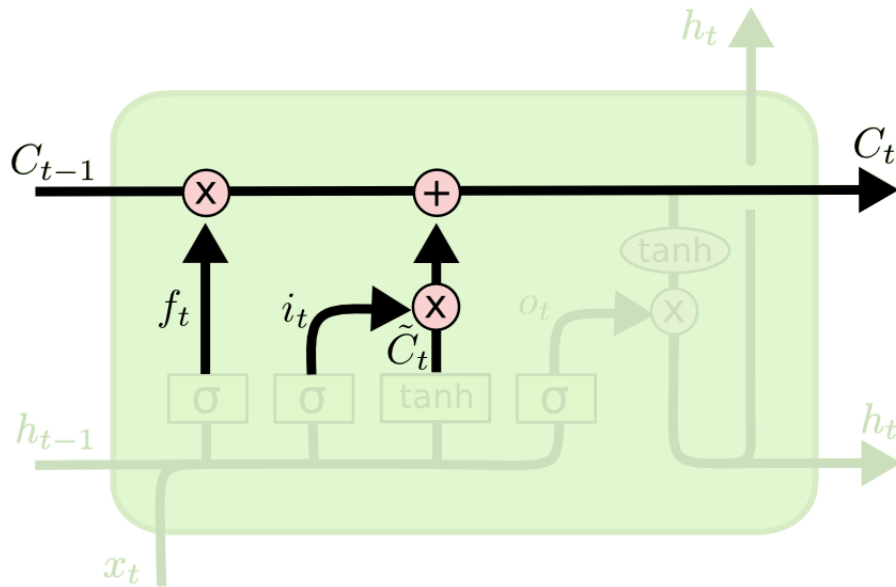


Figura 14 – Representação de um nó de uma rede LSTM. Fonte: <https://bit.ly/3m1kRTS>, 2015

Onde:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.14)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.15)$$

Na terceira etapa é atualizado o estado da antiga célula C_{t-1} para a célula C_t . Para isso, é multiplicado o antigo estado f_t por $i_t \cdot \tilde{C}_t$. Esses são os novos valores candidatos, dimensionados por quanto decidimos atualizar cada valor de estado. Essa camada é conhecida como *output gate*. É apresentado na figura 15 o recorte dessa dinâmica.

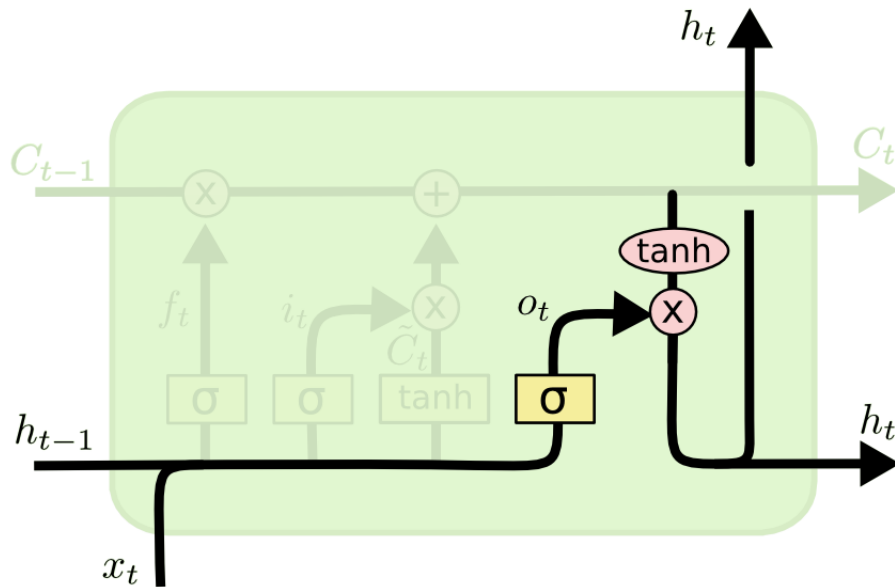


Figura 15 – Representação de um nó de uma rede LSTM. Fonte: <https://bit.ly/3m1kRTS>, 2015

Onde:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

Por fim, é apresentado a saída da célula baseado em seu estado. Primeiro, executa-se uma camada sigmoide (σ) que decide quais partes do estado da célula serão produzidos. Em seguida, o estado da célula em \tanh (para normalizar os valores entre -1 e 1) e são multiplicados pela saída da porta sigmoide, para que apenas sejam produzidos as partes que tenham sido calculados. A Figura 16 apresenta o recorte dessa dinâmica.

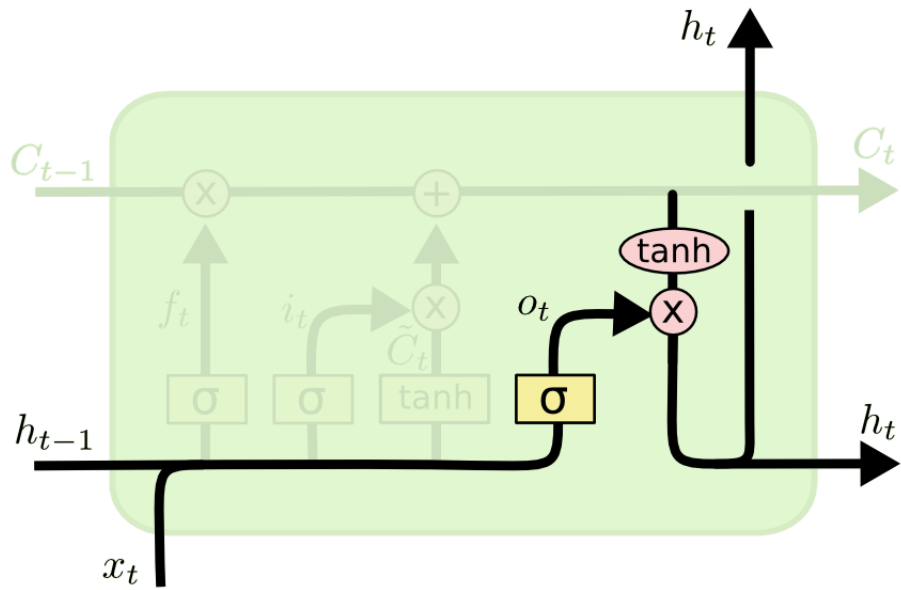


Figura 16 – Representação de um nó de uma rede LSTM. Fonte: <https://bit.ly/3m1kRTS>, 2015

Onde:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.16)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (2.17)$$

3 Combinação de modelos de previsão

3.1 Combinação de modelos

Existem diversos modelos para previsão de séries temporais. Contudo, a maior parte desses modelos analisa os dados utilizando apenas um modelo de previsão, o que em muitos casos pode não ser suficiente para utilização eficiente da informação disponível (CAVALERI; RIBEIRO, 2011). Chatfield (CHATFIELD, 2000) afirma que mesmo considerando que exista um único modelo que consiga se ajustar à série em questão, raramente esse modelo será conhecido *a priori* e que não há garantias que ele se ajustará otimamente a série observada. Além disso, trabalhos como Neuman (NEUMAN, 2003) e Mayer e Yang (MAYER; YANG, 2023) afirmam que a utilização de apenas um modelo tem a tendência de conduzir a vieses estatísticos. Uma abordagem muito promissora para a previsão de séries temporais é uma agregação entre diversos modelos de previsão.

Alguns tipos de modelos de previsão, requerem muitos recursos computacionais para estruturar o modelo (FIRMINO; NETO; FERREIRA, 2014), já outros podem ser elementares, inespecíficos e, às vezes, incompletos, apesar de amplamente aplicados na prática (WEISS; RAVIV; ROETZER, 2018). Assim, a combinação de diversos modelos torna-se um boa solução para obter melhores previsões. A agregação de modelos de previsão é uma estratégia interessante pois, em alguns instâncias da previsão, alguns modelos são mais precisos do que outros e vice-versa. Da mesma forma, analogamente a literatura de finanças, onde para minimizar o risco, a estratégia de diversificação da carteira de ativos é adotada (HOLLAUER; ISSLER; NOTINI, 2008). Os modelos combinadores podem ser divididos em duas classes: os modelos de combinação linear e não linear. A diferença entre esses dois tipos de combinadores é basicamente que em um modelo linear, os pesos dos combinadores são explícitos e até interpretáveis o que não ocorre em um combinador não linear.

3.1.1 Combinadores lineares

A maneira mais usual de pensar em uma combinação de modelos é de forma linear, em que a previsão combinada é o resultado de uma soma ponderada dos modelos individuais. A média simples e a mediana certamente são as alternativas de combinação mais simples da literatura (MAKRIDAKIS; HIBON, 2000). Em se tratando de k preditores,

a média atribui peso de $1/k$ para cada preditor combinado. É apresentado na Equação 3.1 o combinador média simples para a previsão de k modelos, digam-se \hat{y}_{ti} , para o valor de uma dada série temporal no instante t , diga-se y_t .

$$MS_t = \frac{\sum_{i=1}^k \hat{y}_{ti}}{k}. \quad (3.1)$$

Por outro lado, a estimativa de combinação via mediana para y_t , MD_t , considera os valores centrais das estimativas dos k modelos, organizados em ordem crescente (ou decrescente). Assim, a mediana promove uma competição interessante entre os modelos para alcançar as posições centrais na previsão.

Ambos MS e MD não requerem uma fase de treinamento e são ditos não paramétricos, uma vez que fornecem estatísticas de localização descritiva dos dados.

3.1.2 Combinadores não lineares

Os combinadores não lineares têm recebido grande destaque na literatura como é possível observar em trabalhos como o de Hajirahimi e Khashei ([HAJIRAHIMI; KHASHEI, 2019](#)). Esse reconhecimento, em parte, pelo fato de combinadores não lineares considerarem as relações não lineares entre os preditores.

Dessa maneira, os combinadores não lineares considerando a previsão de séries temporais onde o modelo gerador seja estocástico tende a apresentar melhores resultados. Serão apresentados dois modelos de combinação não linear: funções de cópulas e MLP.

3.1.2.1 Funções de cópulas

Cópula pode ser entendida como uma função $C(\cdot)$ que combina distribuições univariadas para construir uma distribuição multivariada combinando as interdependências entre essas distribuição univariadas. A teoria da cópula foi proposta pela primeira vez por Sklar ([SKLAR, 1959](#)), que postulou que a função de distribuição acumulada multivariada $F_Y(y_1, \dots, y_i, \dots, y_k)$ para um conjunto qualquer de variáveis aleatórias $Y = (Y_1, \dots, Y_i, \dots, Y_k)$ pode ser escrita como:

$$F_Y(y_1, \dots, y_i, \dots, y_k) = C_V(v_1, \dots, v_i, \dots, v_k), \quad (3.2)$$

em que $v_i = F_{Y_i}(y_i)$ e o vetor V é composto por variáveis aleatórias $V_i \sim U[0, 1]$. Logo, v_i trata-se de uma instância de V_i , $i = 1, \dots, k$. Dessa forma, pode-se definir uma cópula C da seguinte forma: Para o vetor de variáveis aleatórias $Y = (Y_1, \dots, Y_i, \dots, Y_k)$, com função de distribuição acumulada conjunta F_Y e distribuições acumuladas marginais F_{Y_i}

($i = 1, 2, \dots, k$), a cópula C é uma função acumulada k -dimensional, que pode ser obtida pela álgebra:

$$\begin{aligned} F_Y(y_1, \dots, y_i, \dots, y_k) &= P(Y_1 \leq y_1, \dots, Y_i \leq y_i, \dots, Y_K \leq y_k) = \\ C_V(v_1, \dots, v_i, \dots, v_K) &= P(V_1 \leq v_1, \dots, V_i \leq v_i, \dots, V_K \leq v_k) = \\ &P(V_1 \leq F_{Y_1}(y_1), \dots, V_i \leq F_{Y_i}(y_i), \dots, V_K \leq F_{Y_k}(y_k)). \end{aligned} \quad (3.3)$$

Se $V_i \sim U[0, 1]$ então C é uma função única. As distribuições marginais de entrada da cópula são uniformemente distribuídas. Dessa maneira, a instância v_i é o valor da distribuição marginal dada pela variável Y_i em uma determinada instância y_i . Então, funções de cópulas promovem um modelo multivariado para $(Y_1, \dots, Y_i, \dots, Y_k)$ pelo mapeamento de $(y_1, \dots, y_i, \dots, y_k)$ para $(v_1, \dots, v_i, \dots, v_k)$. Dessa forma, para a função de distribuição de probabilidade (FDP) conjunta de Y , pode-se demonstrar que:

$$\begin{aligned} p_Y(y_1, \dots, y_i, \dots, y_k) &= \frac{\partial F_Y(y_1, \dots, y_k)}{\partial y_1 \dots \partial y_n} = \\ &c(F_{Y_1}(y_1), \dots, F_{Y_n}(y_n)) \prod_{i=1}^n p_{Y_i}(y_i). \end{aligned} \quad (3.4)$$

Em uma perspectiva de combinação de modelos de previsão de séries temporais, cópulas podem ser entendidas como uma função de combinação multivariada que agrega as distribuições marginais univariadas dos erros dos modelos individuais de previsão. A Figura 17 apresenta um esquema resumido da combinação de modelos de previsão *via* cópulas.

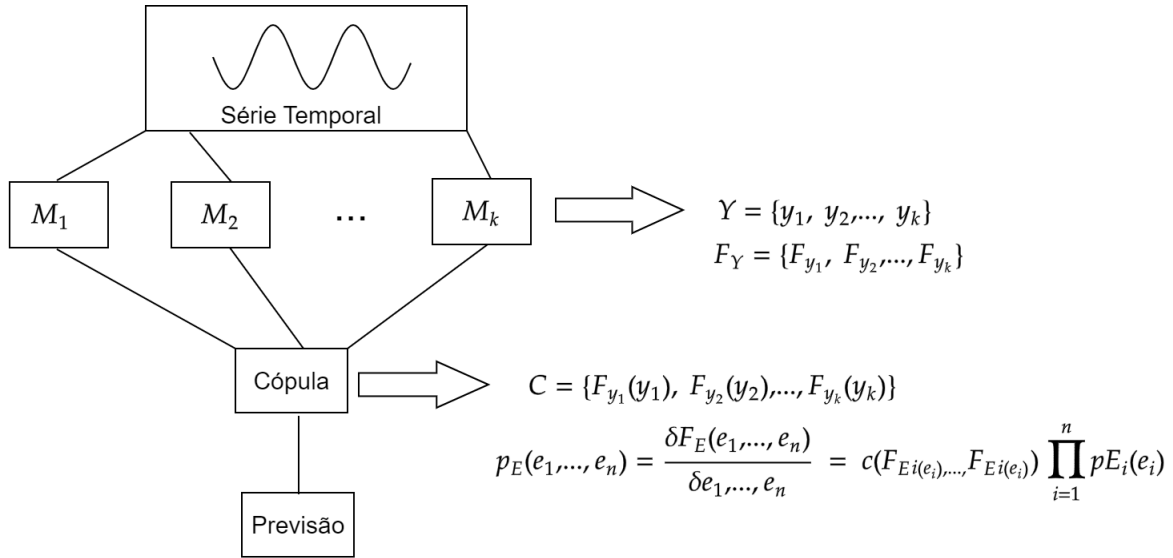


Figura 17 – Esquema da representação de modelos de combinação de previsões *via* cópulas.

Seja M o conjunto de modelos individuais e T o vetor de valores reais da série, então pode-se afirmar que através dos modelos individuais $M_{1,\dots,n}$ e do vetor T , podemos calcular os resíduos (E), onde:

$$E = M_n - T_n \quad (3.5)$$

A partir de E é possível calcular as distribuições acumuladas conjuntas $v_i \sim [0, 1]$

$$v_i = F_{E_i}(e_i) \quad (3.6)$$

Então calculado os v_i , calcula-se a FDP conjunta de E :

$$p_E(e_1, \dots, e_n) = \frac{\partial F_E(e_1, \dots, e_n)}{\partial e_1 \dots \partial e_n} = \underbrace{c(F_{E_1}(e_1), \dots, F_{E_n}(e_n))}_{\text{Função densidade de Cópulas}} \underbrace{\prod_{i=1}^n p_{E_i}(e_i)}_{\text{Função densidade marginal}} \quad (3.7)$$

Existem diversos tipos de famílias de cópulas (NELSEN, 2007); dentre elas serão destacadas aqui as famílias Arquimediana e Elíptica. As cópulas Elípticas são distribuições multivariadas que possuem diversas propriedades similares à distribuição normal multivariada. Estas se mostram atrativas por ser possível mensurar estruturas de dependência não normais (LEAL, 2010). As cópulas arquimedianas são uma classe associativa de cópulas. As cópulas arquimedianas mais comuns admitem uma fórmula explícita, algo impossível, por

exemplo, para a cópula Elíptica (NELSEN, 2007). Na prática, as cópulas de Arquimedes são populares porque permitem modelar a dependência em dimensões arbitrariamente altas com apenas um parâmetro, governando a força da dependência.

A família de cópulas Arquimediana possui propriedades matemáticas interessantes com grande aplicação na modelagem de problemas econômicos, estatísticos e atuariais. Esta gama de aplicações se deve a sua lei de formação simples e das propriedades desejáveis que possui. Dentro da família arquimediana, podem-se citar as cópulas: Clayton (CUVELIER; NOIRHOMME-FRAITURE, 2005), Frank (GENEST, 1987) e Gumbell (OLIVEIRA et al., 2013). Uma cópula arquimediana pode ser definida por:

$$C(v_1, \dots, v_n) = \phi^{-1}(\phi(v_1), \dots, \phi(v_n)), \quad (3.8)$$

em que $\phi(\cdot)$ é a função geradora da cópula e $\phi^{-1}(\cdot)$ é a sua função inversa.

A cópula de Clayton tem sua função geradora definida por:

$$\phi(t) = \frac{1}{\Theta(t^{-\theta} - 1)}. \quad (3.9)$$

Substituindo a Equação 3.9 na Equação 3.8, obtém-se a cópula de Clayton, dada por:

$$C(v_1, \dots, v_k) = \max((v_1^{-\theta} + \dots + v_k^{-\theta} + n - 1)^{-\frac{1}{\theta}}, 0), \quad (3.10)$$

em que Θ é o parâmetro de dependência, com $\Theta \sim (0, \infty)$. Com Θ tendendo a 0 as marginais tendem a se tornar independentes

A cópula de Frank tem função geradora dada pela fórmula:

$$\phi(t) = -\ln \left(\frac{\exp(-\theta t) - 1}{\exp(-t) - 1} \right). \quad (3.11)$$

Substituindo a Equação 3.11 na Equação 3.8, obtém-se a cópula de Frank, dado por:

$$C(v_1, \dots, v_k) = -\ln \left(1 + \frac{(\exp(-\theta v_1) - 1) \dots (\exp(-\theta v_k) - 1)}{\exp(-\theta) - 1} \right), \quad (3.12)$$

em que $\theta > 0$ é o parâmetro que mede o grau de dependência entre as variáveis.

A cópula de Gumbel tem função geradora dada pela fórmula

$$\phi(v_i) = (-\ln v_i)^\theta, \quad (3.13)$$

com $\theta \geq 1$. Substituindo a Equação 3.13 na Equação 3.8, obtem-se a cópula de Gumbel, dada por:

$$C(v_1, \dots, v_k) = \exp[-((-\ln v_1)^\theta + \dots + (-\ln v_k)^\theta)^{\frac{1}{\theta}}] \quad (3.14)$$

As cópulas Elípticas não apresentam uma representação explícita devido a sua formulação ser dada implicitamente por:

$$C(v_1, \dots, v_k) = H(F_{x_1}^{-1}(v_1), \dots, F_{x_k}^{-1}(v_k)), \quad (3.15)$$

em que F_{X_1}, \dots, F_{X_k} são contínuas. O exemplo mais comentado na literatura de cópulas elípticas é a cópula normal ou Gaussiana (NELSEN, 2007). Ela generaliza a distribuição normal multivariada, por permitir marginais não-normais.

A cópula Gaussiana é simétrica e modela a dependência por uma matriz definida positiva, em que os elementos descrevem a dependência entre as variáveis duas a duas. Nesta cópula, a dependência entre pares de variáveis é dada por uma matriz de correlações $k \times k$ (geralmente envolvendo Correlações de Pearson), τ , cujos valores podem variar de -1 a 1, onde $\tau(X_i, X_j) = 0$ indica independência entre as variáveis (X_i, X_j) de índices i e j , $\tau(X_i, X_j) = -1$ implica dependência perfeita negativa, $\tau(X_i, X_j) = 1$ reflete a dependência perfeita positiva. A cópula Normal é definida por (MEYER, 2013):

$$C(v_1, \dots, v_k) = \Phi(\rho^{-1}(v_1), \dots, \rho^{-1}(v_k)|\rho), \quad (3.16)$$

em que $\rho^{-1}(\cdot)$ é a inversa da distribuição acumulada marginal da normal padrão e $\Phi(\cdot)$ a distribuição acumulada conjunta da normal multivariada com vetor de médias zero e matriz de correlações igual a ρ .

Assim a, função densidade de cópulas pode ser escrita por:

$$\begin{aligned} c(v_1, \dots, v_k) &= (\rho^{-1}(v_1), \dots, \rho^{-1}(v_k)|\rho) \\ &= \frac{1}{\sqrt{|\rho|}} \exp \left(-\frac{1}{2} (\rho^{-1}(v_1), \dots, \rho^{-1}(v_k)) (\rho^{-1} - \mathbb{1}) \begin{pmatrix} \rho^{-1}(v_1) \\ \vdots \\ \rho^{-1}(v_k) \end{pmatrix} \right) \end{aligned} \quad (3.17)$$

onde $\mathbb{1}$ é a matriz identidade.

Para o caso em que $\rho^{-1}(v_i)$ é a função inversa da distribuição normal univariada, a cópula normal gera uma distribuição de probabilidades conjunta normal. No entanto, é

suficiente que pelo menos uma das distribuições marginais não seja normal para que a distribuição de probabilidades conjunta resultante difira desta específica cópula Normal.

3.1.2.2 Combinadores baseados em RNA

Um valor de previsão obtido da combinação de previsões é uma saída de uma função que consiste na soma de previsões ponderadas produzidas por diferentes modelos de previsão. Esses pesos ponderam a contribuição dos modelos individuais de previsão correspondentes usados na combinação das previsões. Esses pesos podem ser determinados levando-se em consideração algumas premissas ou usando métodos de otimização (ALADAG; EGRIOGLU; YOLCU, 2010). Essa descrição de combinação de modelos remete ao conceito de RNA onde a camada de entrada passa a ser as previsões dos modelos individuais, a ponderação citada é o ajuste dos pesos que o processo de treinamento da rede promove e a saída da rede seria o resultado do processo de combinação, isto é, o valor da série temporal sob predição. RNA são usadas para determinar os melhores valores dos pesos na combinação de previsão. Quando esta configuração de redes neurais artificiais é usada, valores ótimos dos pesos podem fornecer o melhor mapeamento não linear. Assim, tanto a função de combinação quanto os valores ótimos dos pesos podem ser determinados usando RNA.

A função de combinação não linear pode ser escrita considerando a seguinte combinação para dois modelos de previsão adicionado o fator não linear:

$$f_c = w_0 + w_1 f_1 + w_2 f_2 + w_{nl} f_1 f_2 \quad (3.18)$$

Na Equação 3.18 f_1 e f_2 representam as predições para os seus respectivos modelos e w_{nl} é o peso para o termo não linear que pode ser encontrado minimizando a soma dos erros quadráticos. Avaliando a Equação 3.18, quando $w_0 = 0$ e $w_{nl} = 0$ ela se torna uma combinação linear como na Equação 3.1.

Usualmente, na implementação do combinador baseado em RNA, um neurônio é empregado na camada oculta para não perder a capacidade de generalização da rede neural. É apresentado na Figura 18 a representação de uma combinação via RNA.

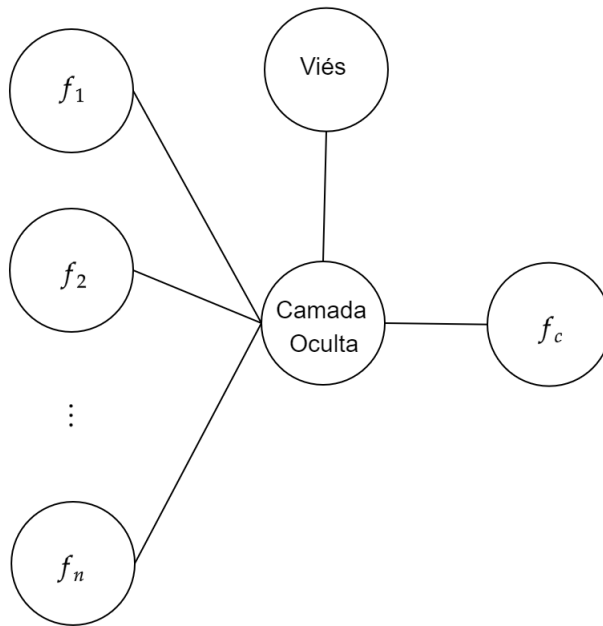


Figura 18 – Esquema da representação da combinação *via* MLP.

Para os neurônios das camadas ocultas é utilizada a função logística descrita na Equação 2.12 já para a camada de saída é utilizada uma função linear. O modelo matemático da RNA apresenta na Figura 18 pode ser descrito por:

$$f_c = \left(\left(\sum_{n=1}^n w(1, i) \frac{1}{1 + e^{(-f_i)}} \right) + w_1 \right) w(2, 1) + w_2 \quad (3.19)$$

onde f_i é o valor da previsão obtido pelo modelo i , $w(1, i)$ são os pesos entre as entradas e as camadas ocultas, w_1 é o peso entre o viés e a camada oculta. $w(2, i)$ é o peso entre o neurônio da camada oculta e o da camada de saída. Portanto, o modelo expresso na Equação 3.19 tem uma forma não linear. Quando a rede neural, cuja modelo matemático é expresso pela Equação 3.19, é treinado, os pesos calculados são os pesos de combinação de previsão e as saídas obtidas são as previsões combinadas.

4 Trabalhos relacionados

A combinação de modelos de previsão já é uma área bastante consolidada na literatura e por tanto possui várias abordagens nos mais diversos trabalhos como mostrado na pesquisa de Hajirahimi e Khashei (HAJIRAHIMI; KHASHEI, 2019) que apresentaram mais de 150 trabalhos relacionados com modelos híbridos de previsão de séries temporais. Um dos trabalhos pioneiros na área de combinação de previsão foi o de Bates e Granger (BATES; GRANGER, 1969) e de Reid (REID, 1969) e desde então diversos trabalhos com as mais diferentes abordagens de combinação foram propostos na área.

Com relação à combinação linear pode-se citar o trabalho de Lin *et al.* (LIN *et al.*, 2014) que combinou modelos de ELM (*Extreme Learning Machines*) utilizando a média aritmética das saídas dos modelos para a previsão de índices de deslizamentos de terra. Wang *et al.* (WANG; ZHANG; LU, 2019) propuseram uma nova metodologia de combinação linear para treinamento de redes neurais aplicadas a previsão de séries de velocidade de ventos. Andrawis *et al.* (ANDRAWIS; ATIYA; EL-SHISHINY, 2011) utilizaram a média aritmética para combinar as previsões de um conjunto de redes neurais, modelos de regressão Gaussiana e linear, em uma série temporal desenhadas para competição. A mediana e a média winsorizada foram utilizadas no trabalho de Adhikari e Agrawal (ADHIKARI; AGRAWAL, 2012). Qiu *et al.* (QIU *et al.*, 2021) utilizaram modelos ARIMA com adição de sazonalidade (SARIMA) para realizar a previsão de séries de casos de varíola. Adhikari (ADHIKARI, 2015) propôs em seu trabalho uma forma de combinação linear que determina os pesos dos modelos através da análise de padrões nos dados nas previsões de uma base de dados de validação. O Softmax foi uma proposta de combinação de um conjunto de redes neurais a partir de validação cruzada (DONATE *et al.*, 2013).

Já em relação a combinadores não lineares é possível citar os trabalhos de Ferreira *et al.* (FERREIRA; VASCONCELOS; ADEODATO, 2008) que propõe uma nova metodologia híbrida de previsão de séries temporais baseada na combinação de algoritmos genéticos e RNA. Gheyas e Smith (GHEYAS; SMITH, 2009) elaboraram um conjunto de modelos híbridos de redes neurais e regressão linear onde a saída de um conjunto de modelos é utilizado para o treinamento de uma segunda etapa da modelagem. Nóbrega e Oliveira (NÓBREGA; OLIVEIRA, 2014) combinaram ELM e modelos de SVR (*supported vector regression*) utilizando filtros de Kalman para a previsão de séries temporais financeiras. Oliveira *et al.* (OLIVEIRA *et al.*, 2017) utilizou funções de cópulas para combinação

de previsões de séries temporais e as compara com os combinadores lineares. Neto *et al.* (NETO *et al.*, 2021) utilizam RNA, em específico MLPs para a combinação de modelos na previsão de séries relacionadas ao nível de poluição do ar. Em outro trabalho, Neto *et al.* (de Mattos Neto *et al.*, 2021) utiliza modelos lineares de *deep learning* para realizar a previsão combinada para séries relacionadas a velocidades de ventos. Já na linha de sistemas híbridos de previsão de séries temporais pode-se citar o trabalho de Oliveira e Lurdemir (OLIVEIRA; LUDERMIR, 2016) traz uma abordagem com modelos ARIMA, RNA e algoritmos de dispersão de partículas (*particle swarm optimization* (PSO)) para combinação de preditores.

Já se é consolidado na literatura que modelos combinados apresentam resultados mais acurados que os modelos individuais (WELCH; GOYAL, 2008). Em especial, aumentando o grau de diversidade dos modelos com diferentes arquiteturas, diferentes tipos de pré-processamento, diversas variáveis de entrada, possibilitam que modelos combinadores robustos de previsões sejam encontrados (OLIVEIRA; TORGO, 2015). Contudo, uma outra discussão que tem ganhado força no campo da combinação de modelos de predição é a quantidade de modelos a serem utilizados. Enquanto autores como Armstrong (ARMS-TRONG, 2001) afirmam que existe a necessidade de ao menos 5 modelos individuais de predição para que um grau mínimo de diversidade seja alcançado e assim favoreça a previsão combinada, outros trabalhos afirmam que a utilização de mais de 5 modelos na combinação da previsão pode surtir efeito negativo para a acurácia do modelo combinador (MAKRIDAKIS; WINKLER, 1983)

É apresentado na Tabela 1 apresenta uma síntese de alguns trabalhos apresentados nessa Seção.

Autor	Área	Ano	Metodologia
LIN et al.	CL	2014	Média Aritiméticas na combinação de ELMs
WANG; ZHANG; LU	CL	2019	Modelagem linear para combinação de RNA
ANDRAWIS; ATIYA; EL-SHISHINY	CL	2011	Média Aritmética para combinar RNA e modelos de regressão Gaussiana
ADHIKARI; AGRAWAL	CL	2012	Mediana e da média winsorizada na combinação de RNA
QIU et al.	CL	2021	Modelos SARIMA para previsão e combinação
ADHIKARI	CL	2015	Modelagem de combinação linear para ajuste de pesos dos modelos através de análise de padrões nos dados
DONATE et al.	CL	2013	Combinação de modelos via validação cruzada
FERREIRA; VASCONCELOS; ADEODATO	MHP	2008	Modelagem Híbrida de previsão utilizando algoritmos genéticos e RNA
GHEYAS; SMITH	MHP	2009	Modelos híbridos de RNA e regressão linear
OLIVEIRA; TORGO	MHP	2016	Combinação de RNA, PSO e algoritmos genéticos
NÓBREGA; OLIVEIRA	CnL	2014	Filtros de Kalman para combinação de ELM e SVM
OLIVEIRA et al.	CnL	2017	Funções de cópulas
NETO et al.	CnL	2021	RNAs
de Mattos Neto et al.	MHP	2021	LSTM e RNA

Tabela 1 – Resumo dos trabalhos relacionados a tese. Onde na segunda coluna da tabela entende-se CL combinação linear, MHP modelo híbrido de previsão, CnL combinação não linear.

5 Séries temporais avaliadas

5.1 Processamento das séries avaliadas

Nesse trabalho de tese foram avaliadas 5 séries temporais financeiras acerca do mercado de ações: Petrobras (PB), Google (GG), Nasdaq (NQ), S&P500 (SP) e taxa de câmbio GBP para USD (GU). Essas séries foram extraídas do repositório do *Yahoo! Finance website* ((MSFT), 2020). Nas próximas subseções serão analisadas mais detalhadamente cada uma das séries. As análises realizadas aqui foram feitas no ambiente *R Studio*, versão *R 3.6.3* (R Core Team, 2020). O intervalo das séries desse estudo foi definido com base em outros trabalhos da literatura para favorecer a comparação adequada dos resultados obtidos no Capítulo 7, que aborda os resultados dessa tese. As séries especificamente avaliadas nesse trabalho se encontram disponibilizadas no seguinte repositório de acesso público <https://github.com/ademirNeto/FinancialSeries>.

5.1.1 Petrobras (PB)

A série PB apresenta os valores de fechamento diários das ações da Petrobras entre 05 de janeiro de 2000 e 09 de junho de 2020, num total de 5005 observações.

A Petrobras é uma empresa brasileira de capital aberto em que o governo brasileiro é o acionista majoritário sendo assim uma grande empresa estatal de capital misto. A Petrobras é uma empresa do segmento de energia que atua em 14 países na exploração, produção, refino, comercialização e transporte de petróleo, gás natural e seus derivados. As ações da Petrobras são negociadas na B3 que é a bolsa de valores brasileira.

É apresentado na Figura 19a a representação gráfica dos valores de fechamento das ações da Petrobras ao longo do período estudado. A Figura 19b apresenta a análise de Lags da série PB obtidas através da função `gglagplot` do pacote `forecast`.

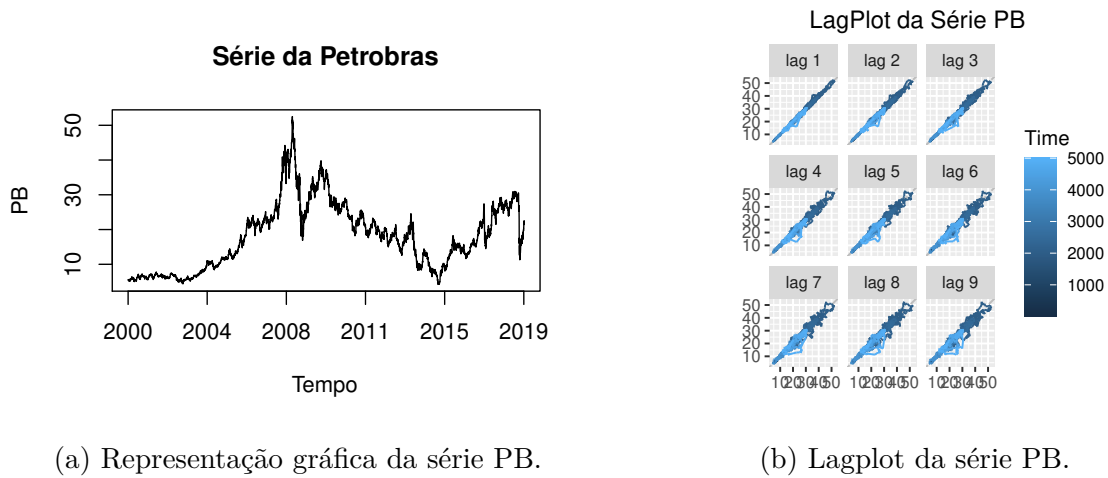


Figura 19 – Série e Lagplot PB.

5.1.2 Google (GG)

A série GG apresenta os valores de fechamento diários das ações da Google entre 30 de julho de 2008 e 30 de dezembro de 2008, num total de 107 observações.

A Google é uma empresa de tecnologia multinacional americana de capital aberto que atua majoritariamente na área da inteligência artificial, especificamente com mecanismo de busca, publicidade online, computação em nuvem.

A Figura 20a apresenta a representação gráfica dos valores de fechamento das ações da Google ao longo do tempo. A Figura 20b apresenta a análise de Lags da série da Google obtidas através da função `gglagplot` do pacote `forecast`.

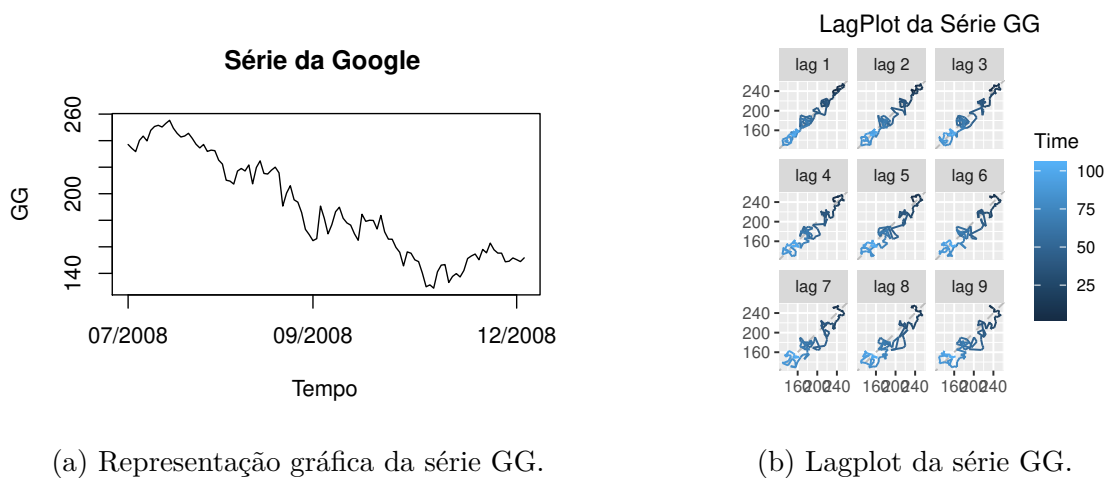


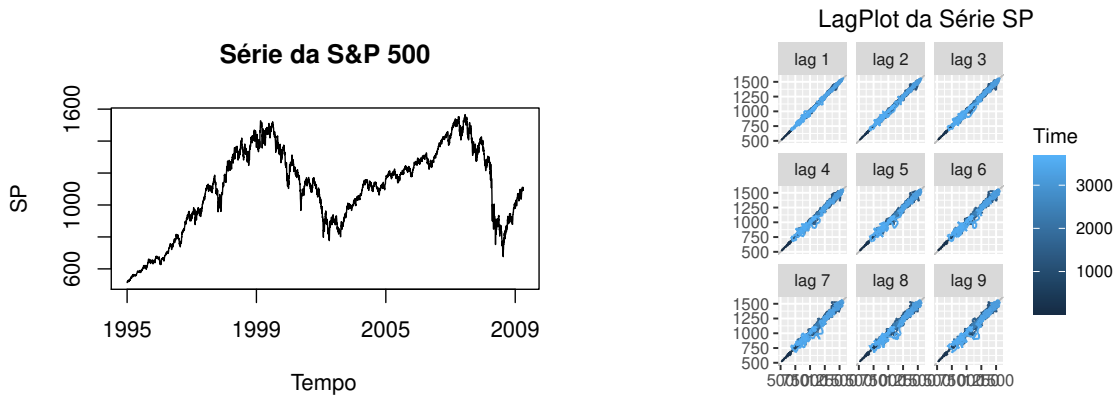
Figura 20 – Série e Lagplot GG.

5.1.3 S&P500 (SP)

A série S&P500 apresenta os valores de fechamento diários desse índice composto entre 01 de maio de 1995 e 01 de dezembro de 2009, num total de 3673 observações.

A S&P500 é um índice composto por 500 ativos cotados na bolsa da NASDAQ. Esses ativos são qualificados conforme seu tamanho, liquidez e representação industrial. A S&P500 é um índice ponderado e o peso de cada ativo é definido pelo seu valor no mercado.

A Figura 21a apresenta a representação gráfica das observações da série S&P500. A Figura 21b apresenta a análise de Lags da série S&P500 obtidas através da função `gglagplot` do pacote `forecast`.



(a) Representação gráfica da série SP.

(b) Lagplot da série SP.

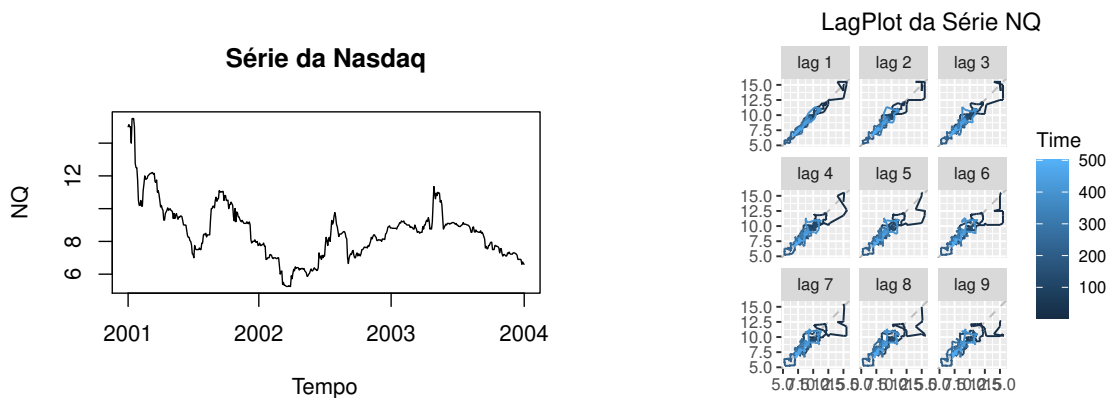
Figura 21 – Série e Lagplot SP.

5.1.4 Nasdaq (NQ)

A série da Nasdaq apresenta os valores de fechamento diários dessa bolsa entre 20 de fevereiro de 2001 e 25 de junho de 2004, num total de 840 observações.

O Nasdaq é um mercado de ações automatizados norte-americano onde estão listadas mais de 2800 empresas. É o segundo maior mercado de ações em capitalização de mercado do mundo. O NASDAQ caracteriza-se por reunir empresas de alta tecnologia em eletrônica, informática, telecomunicações, biotecnologia e outras áreas similares.

A Figura 22a apresenta a representação gráfica das observações da série Nasdaq. A Figura 22b apresenta a análise de Lags da série NQ obtidas através da função `gglagplot` do pacote `forecast`.



(a) Representação gráfica da série NQ.

(b) Lagplot da série NQ.

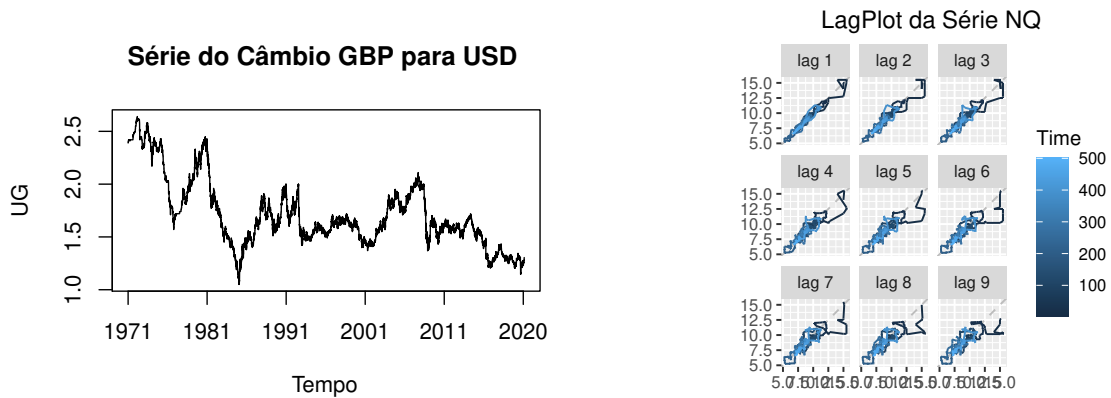
Figura 22 – Série e Lagplot NQ.

5.1.5 Taxa de câmbio GBP para USD (GU)

A série GU apresenta os valores de proporção entre o dólar americano e a Libra esterlina entre 04 de março de 1971 de 2000 e 30 de julho de 2020, num total de 12630 observações.

A libra esterlina (GBP) é a moeda oficial do Reino Unido desde de fevereiro de 1971. O dólar americano (USD) é a moeda oficial dos Estados Unidos da América desde 1792. Ambas as moedas são muito valorizadas no mercado global. O índice de câmbio GBP/USD trás uma proporção do valor do dólar em relação a libra. Sendo que por exemplo, se o valor do câmbio estiver em 1.25, são necessários 1.25 USD para a aquisição de 1 GPB.

A Figura 23a apresenta a representação gráfica dos valores da comparação entre GBP e USD ao longo do tempo. A Figura 23b apresenta a representação gráfica dos valores da comparação entre GBP e USD ao longo do tempo.



(a) Representação gráfica da série GG.

(b) Lagplot da série GU.

Figura 23 – Série e Lagplot GU.

5.1.6 Estatísticas das séries avaliadas

É apresentado na Tabela 2 algumas estatísticas básicas relacionadas as séries avaliadas nesse estudo.

Estatística	PB	GG	SP	NQ	GU
Quantidade de Pontos	5005	107	3673	840	12630
Valor Mínimo	4.2	128.8	514.3	15.5	1.05
Valor Máximo	52.51	255.3	1565.2	5.25	2.64
Média	17.94	189.3	1104.4	8.55	1.71
Mediana	17.51	181.5	1131.8	8.25	1.63
Desvio Padrão	9.49	37.79	256.96	1.72	0.32

Tabela 2 – Tabela com as estatísticas básicas das séries avaliadas.

6 Metodologia

6.1 Modelos de previsão

A metodologia abordada nessa tese realiza a comparação da acurácia da previsão de 7 modelos de previsão. Desses 7 modelos, 3 são individuais e os outros 4 são modelagens de combinação para os modelos individuais. Os três modelos individuais utilizados são das classes ARIMA, redes neurais auto-regressivas (do inglês *Neural Network Auto Regression* (NNAR)), denominado doravante nessa tese de ANN, e LSTM. Para a combinação desses modelos, foram abordados combinadores lineares (Média e Mediana), funções de cópulas e MLP. Para todos os modelos de previsão, as séries temporais avaliadas foram divididas em um conjunto de treinamento que corresponde às primeiras 75% das observações e o conjunto restante, chamado aqui de teste, que corresponde às últimas 25% observações da série. A metodologia de previsões de séries temporais desenvolvida aqui para previsão de séries temporais com combinação de modelos de cópulas foi registrada junto ao Instituto Nacional de Propriedade Industrial - INPI - sob o número BR512021001615-7 ([NETO; FIRMINO; FERREIRA, 2021](#)).

6.1.1 ARIMA

As previsões realizadas do modelo ARIMA foram obtidas utilizando o pacote `forecast` do R. A obtenção das previsões do modelo ARIMA foram divididas em duas etapas: ajuste do modelo à série de treinamento e previsão em si. A função que realiza a modelagem da série é a `auto.arima()`, que utiliza uma variação do algoritmo de Hyndman-Khandaka ([HYNDMAN; ATHANASOPOULOS, 2014](#)). Essa função combina testes de raiz unitária, minimização de critério de informação (como o AIC) e maximização da função de verossimilhança. É apresentado na Tabela 3 os parâmetros da função `auto.arima()` os quais foram utilizados na modelagem ARIMA das séries.

Parâmetro	Descrição
y	Série temporal
max p	Valor máximo para o termo autoregressivo p
max q	Valor máximo para o termo de médias móveis q
max P	Valor máximo para o termo P (sazonal)
max Q	Valor máximo para o termo Q (sazonal)
seasonal	Considera também modelos sazonais
stepwise	Se marcado como "False" avalia todos os modelos gerados da série
nmodels	Número máximo de modelos avaliados
parallel	Permite a busca em paralelo para os modelos
num.cores	Especifica quantas unidades de processamento podem ser utilizados
allowdrift	Quando tem valor "True" permite modelos com termo de derivação
allowmean	Quando tem valor "True" permite modelos com média não nula
ic	Critério de informação para seleção do modelo

Tabela 3 – Parâmetros da função `auto.arima()`.

Após a realização dos testes estatísticos para a identificação da estacionariedade da série, é realizado o ajuste do modelo ARIMA através da seguinte função: `auto.arima(train, max.p =20, max.q =20, max.P =20, max.Q =20, seasonal=TRUE, stepwise = FALSE, parallel=TRUE, num.cores=NULL, nmodels=5000, start.p = 4, start.q = 4, allowdrift = FALSE, allowmean = FALSE , ic = "aic")`.

A função `auto.arima` retorna o modelo escolhido (conforme os valores de p, d e q), os termos autorregressivos (ar) e de média móvel (ma), de “*drift*” (termo constante em modelo que tem $d > 0$ devido à sazonalidade) e erro padrão (s.e. – “*standard error*”) de cada valor estimativa apresentada. O erro padrão mostra o quanto de variação ou “dispersão” existe do estimador em relação à sua média (ou valor esperado), se todos os pontos estiverem exatamente sobre a linha de regressão, então esta regressão explicaria toda a variação, ou seja, seria igual a zero. A função também imprime os critérios de informação AIC e BIC, assim como a variância (σ^2), que é definida como o quadrado do desvio padrão.

É apresentado na Figura 24 o conjunto de treinamento da série PB que vai da observação 1 até a 3753 de um total de 5005 observações. Aplicado a função `autoarima` a esse conjunto de dados o modelo resultante é apresentado na Tabela 4.

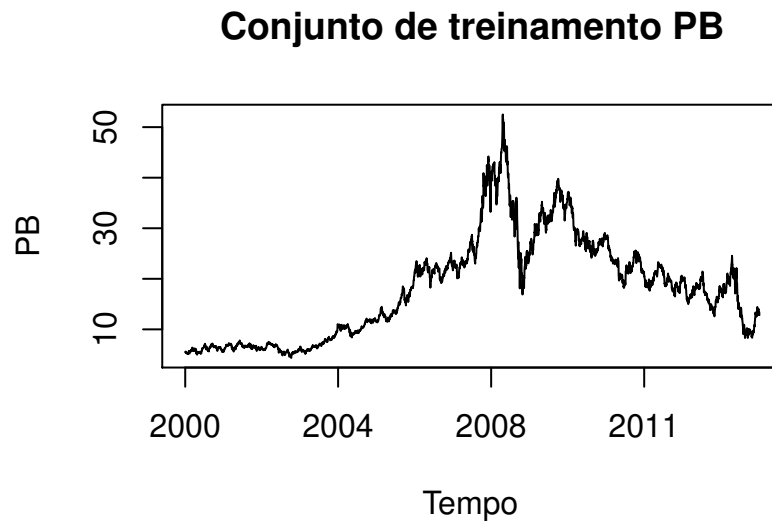


Figura 24 – Representação gráfica do conjunto de treinamento da série PB.

Indicador	Valor
Modelo	ARIMA(2,1,3)
ar1	-0.8391
ar2	-0.6871
ma1	0.8775
ma2	0.705
sigma2	0.2632
AIC	5646.67
BIC	5684.05

Tabela 4 – Resumo do modelo ARIMA para o conjunto de treinamento da série PB.

O número de linhas dos regressores, obtido na função `auto.arima()`, deve ser o mesmo da série temporal. O modelo criado por `arima()` e `auto.arima()` com um argumento `xreg` é uma regressão linear com erros ARMA:

$$y_t = \beta x_t + \eta_t \quad (6.1)$$

$$\eta_t = \phi_t \eta_{t-1} + \dots + \phi_p \eta_{t-p} - \Theta_1 Z_{t-1} - \dots - \Theta_q Z_{t-q} + Z_t \quad (6.2)$$

Onde y_t é o valor da projeção no instante t , x_t é a covariância no instante t e β é seu coeficiente; ϕ é o polinômio ligado ao operador autorregressivo de ordem p ; Θ é o polinômio ligado ao operador de média móvel de ordem q ; z_t é o ruído branco no instante t .

Com o modelo definido então é usado a função `fitted(Arima(modelo))`, que realiza previsão do modelo ARIMA a partir do modelo ajustado ao conjunto de treinamento.

6.1.2 ANN

Os valores dos *lags* (janelas temporais) das séries temporais podem ser utilizadas como entrada de uma rede neural, assim como foram usados os valores das janelas temporais em modelos autoregressivos. Esse tipo de rede neural é chamada de rede neural autoregressiva (NNAR) (HYNDMAN; ATHANASOPOULOS, 2018). Na metodologia abordada no pacote `forecast` do R a rede neural utilizada é um modelo MLP de uma única camada escondida onde a notação é NNAR(p,k), onde p é a quantidade de janelas temporais para avaliar o modelo e k a quantidade de neurônios na camada escondida. Um modelo NNAR(p,0) é equivalente a um modelo AR(p).

As previsões realizadas do modelo ANN foram também obtidas utilizando o pacote `forecast` do R. A exemplo do modelo ARIMA, a previsão dos modelos ANN se divide em duas etapas: ajuste da rede ao conjunto de treinamento e previsão. A função que realiza a modelagem da série é a `nnetar()`. É apresentado na Tabela 5 os parâmetros da função `nnetar()` os quais foram utilizados na modelagem de previsões com redes neurais das séries.

Parâmetro	Valor
y	Série
p	Número de lags utilizado com entrada da rede
repeats	Número de redes que serão ajustadas com diferentes valores aleatórios para os pesos iniciais
lambda	Parâmetro de transformação Box-Cox. Se lambda="auto", uma transformação é selecionada automaticamente usando BoxCox
size	Número de neurônios na camada escondida

Tabela 5 – Argumentos da função `nnetar()`.

Após a realização dos testes estatísticos para a identificação do *lag* do modelo, é realizado o ajuste da rede NNAR através da seguinte função: `nnetar(serie , p=9, lambda='auto', repeats = 100, size=30)`. Aplicando a função `nnetar()` ao conjunto de treinamento presente na Figura 24, é obtido o modelo NNAR(9,5). Em média, 100 redes neurais foram aplicadas e o modelo resultante foi de 9-5-1 o que indica 9 lags, 5 neurônios na camada escondida e 1 informa que a série não apresenta comportamento sazonal. O parâmetro 9 para o valor do *lag* foi escolhida através da análise do `lagplot` da série.

A fim de evitar um modelo com *overfitting*, foi aplicada uma validação cruzada no conjunto de treinamento. A validação cruzada foi realizada utilizando a função `tsCV` do pacote `forecast`. Com esta validação espera-se que o RMSE dos resíduos do modelo ajustado seja menor do que o RMSE obtido através da validação cruzada de séries temporais. Mais detalhes sobre validação cruzada podem ser encontrados em Bergmeir *et al.* (BERGMEIR; HYNDMAN; KOO, 2018). Por outro lado, em relação ao *underfitting*, foram realizados testes de Ljung-Box (LJUNG; BOX, 1978), tomando-se 0,05 como nível de significância. É apresentado na Tabela 6 os valores de RMSE da validação cruzada, RMSE do modelo ajustado e valores de Ljung-Box p para todas as séries estudadas.

Tabela 6 – RMSE da validação cruzada e Ljung-Box p valor.

Serie	RMSE Cross	RMSE	Ljung-Box p valor
PB	0.5154	0.5056	0.2006
NQ	0.2853	0.2577	0.5979
GG	7.7020	6.8353	0.0769
SP	2.8152	2.4034	0.0849
GU	0.0027	0.0017	0.9111

Com o modelo definido então é usado a função `fitted(forecast(modelo))` que realiza as previsões dos modelos ANN a partir do conjunto de treinamento.

6.1.3 LSTM

As previsões dos modelos LSTM são realizadas através das funções presentes nos pacotes `tensorflow` e `keras`. O `tensorflow` é uma biblioteca de *software* de código aberto para computação numérica usando grafos de fluxo de dados. Os nós no grafo representam operações matemáticas, enquanto as arestas do grafo representam os *arrays* de dados multidimensionais (tensores) comunicados entre eles. O pacote `keras` fornece uma interface R para Keras. Essa é uma API (*Application Programming Interface*) de redes neurais de alto nível desenvolvida com foco em permitir experimentação rápida. Esse pacote permite dentre outras situações que o mesmo código seja executado na CPU (*Central Processing Unit*) ou na GPU (*Graphics Processing Unit*).

Para as explicações práticas das previsões do modelo LSTM será considerado o conjunto de treinamento da série temporal PB presente na Figura 24. Inicialmente a série temporal é diferenciada para remover os componentes de dependência temporal, facilitando a modelagem da série. A função `diff(Series, differences = 1)` é utilizada para diferenciar a série uma vez. A rede LSTM espera que os dados estejam em modo de aprendizado

supervisionado, ou seja, deve-se aplicar um *lag* k utilizando a série no tempo (tk) como entrada e valor no instante t como saída, para cada *lag* k do conjunto. Assim como em qualquer outro modelo de rede neural, o conjunto de dados é redimensionado para o intervalo da função de ativação. A função o padrão para LSTM é a sigmóide (Figura 7) cujo intervalo é $(-1, 1)$.

Após a preparação do conjunto de dados, é realizado então o ajuste do modelos LSTM para a série. O *batch size* limita o número de amostras a serem mostradas à rede antes que uma atualização de peso possa ser realizada. Essa mesma limitação é então imposta ao fazer previsões com o modelo de ajuste. No geral deve-se utilizar um *batch size* que seja divisível por 8 a exemplo do valor 64. Contudo sendo uma previsão do próximo valor da série, pode-se utilizar o *batch size* igual a 1 (YOU et al., 2019). Inicialmente é definido o argumento `stateful= TRUE` para que os estados internos obtidos após o processamento de um *batch* de amostras sejam reutilizados como estados iniciais para as amostras do próximo *batch*. Como a rede é *stateful*, deve-se fornecer o *batch* de entrada em uma matriz tridimensional da forma (amostras, passos, características) onde:

- Amostras: Uma sequência é uma amostra. Um *batch* é composto por uma ou mais amostras.
- Passos: Um passo de tempo é um ponto de observação na amostra.
- Características: Uma característica é uma observação em um passo de tempo. Para um caso univariado, Características = 1

A camada de entrada LSTM é especificada pelo argumento *input_shape* na primeira camada oculta da rede. Essa entrada da rede LSTM é feita transformando uma entrada unidimensional para uma entrada tridimensional com as características do *batch size* descritas. Por exemplo um *array* unidimensional com 10 elementos pode ser rearranjado para uma entrada tridimensional de uma LSTM de maneira que seja: uma amostra, 10 passos e uma característica (1-10-1).

Logo é definido os parâmetros do modelo `compile (loss = mean_squared_error , optimizer = optimizer_adam (lr= 0.02, decay= 1e-6), metrics=c(accuracy))`. Onde o erro médio quadrático (MSE) foi definido como função de perda da rede. O algoritmo de otimização foi *Adaptive Monument Estimation* (ADAM) com a taxa de aprendizado de 0,02 e uma queda dessa taxa de $1e^{-6}$ a cada interação. Após a definição dos parâmetros é ajustado o modelo a entrada da rede LSTM. Inicialmente, define-se a quantidade de iterações em que será executado o ajuste da rede. O Argumento `shuffle = FALSE` indica que o conjunto de treinamento não será embaralhado mantendo cronologia

entre x_i e x_{i+t} . A rede LSTM também requer a redefinição do estado da rede após cada época. Para conseguir isso, a cada execução do *loop* do ajuste da rede é redefinido os estados por meio do argumento `reset_states()`.

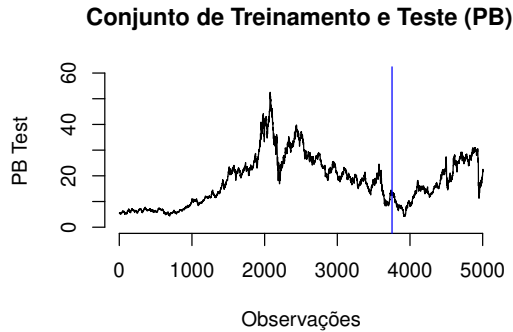
Para realizar a previsão da série *one step ahead* desse modelo é utilizado o modelo ajustado da série `model = predict (X, batch_size = batch_size)` onde o valor da previsão é informado. Por fim, esse valor é processado para ser revertido a escala aplicada para a adequação a função sigmoide e também é invertida a diferenciação realizada no conjunto de treinamento da série.

6.1.4 Média simples e mediana

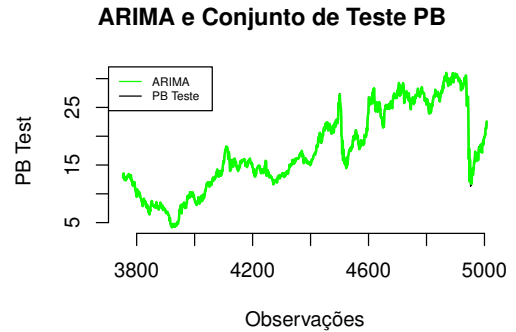
Os modelos combinados utilizam os vetores com as previsões dos três modelos individuais (ARIMA, ANN, LSTM) para formar a previsão do modelo combinado. Em relação aos modelos de combinação linear (média, mediana) são utilizadas as funções nativas `mean()` e `median()` do R.

6.1.5 Cópulas

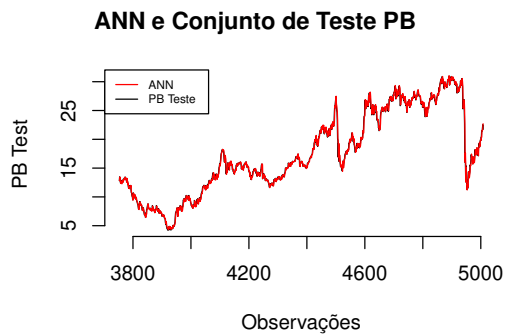
A previsão dos modelos de cópulas é realizada utilizando alguns pacotes do R: `forecast`, `rutil`, `copula`, `fGarch` e `Mass`. Para a demonstração da combinação via cópulas serão usados os modelos individuais das predições *via* ARIMA, ANN e LSTM para a realização da previsão do conjunto de teste da série PB. A Figura 25(A) apresenta os conjuntos de treinamento e teste separados por uma linha vertical e nas demais imagens uma sobreposição dos conjunto de teste e dos modelos individuais.



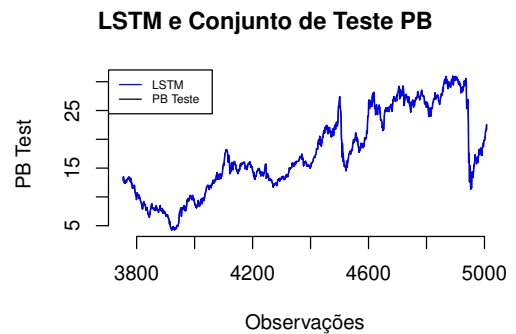
(a) Conjunto de testes e treinamento da série PB



(b) Previsões do modelo ARIMA para a série PB



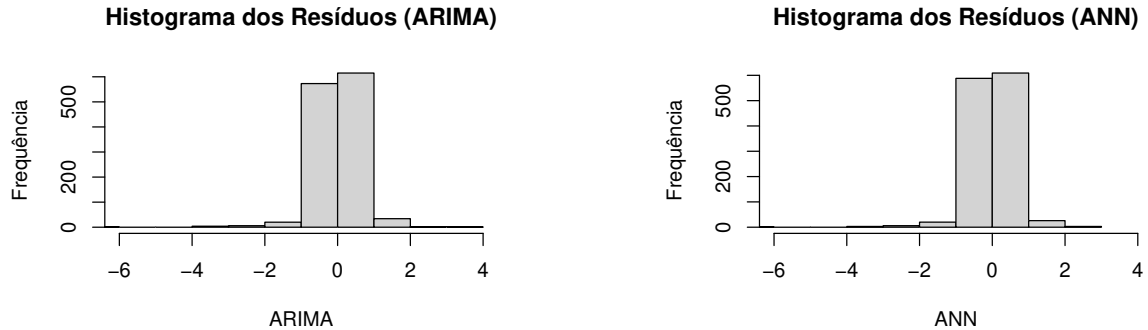
(c) Previsões do modelo ANN para a série PB



(d) Previsões do modelo LSTM para a série PB

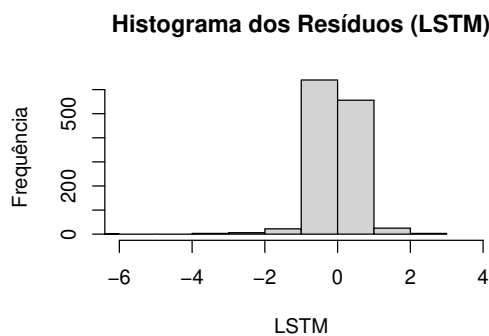
Figura 25 – Conjuntos da série PB e previsões dos modelos individuais.

Depois são extraídos os resíduos dos modelos individuais, isto é, das diferenças entre o real e o respectivo previsto. O histograma dos resíduos do modelo ARIMA está representado na Figura 26 (A) e o erro médio quadrático do modelo foi de 0,5659 BRL (moeda corrente do Brasil) O histograma dos resíduos do modelo ANN está representado na Figura 26 (B) e o erro médio quadrático do modelo foi de 0,5574. O histograma dos resíduos do modelo LSTM está representado na Figura 26 (C) e o erro médio quadrático do modelo foi de 0,5604.



(a) Resíduos do modelo ARIMA

(b) Resíduos do modelo ANN



(c) Resíduos do modelo LSTM

Figura 26 – Resíduos dos modelos individuais para a previsão da série PB.

Após calcular os resíduos dos modelos, o algoritmo carrega a lista de famílias de cópulas (Arquimediana, Elíptica) e de distribuições de de probabilidade (Clayton, Frank, Gumbel, Joe, normal). Após isso, são ajustadas as distribuições marginais (normal, cauchy, laplace, snormal¹) aos resíduos de cada modelo individual, baseando-se no critério de informação. Basicamente, são ajustados os modelos de cópulas aos dados e os que

¹ Snormal é a distribuição normal assimétrica, também conhecida como distribuição *Skew Normal*. Ela é uma distribuição estatística que apresenta assimetria em sua forma. A distribuição *Skew Normal* possui uma assimetria que pode ser positiva (cauda longa à direita) ou negativa (cauda longa à esquerda). Isso significa que a probabilidade de ocorrência de valores extremos é maior em uma direção do que na outra. Para maiores detalhes, consultar o livro de Azzalini (AZZALINI, 2013).

apresentarem o menor BIC são escolhidos. Esse processo é realizado para os três modelos individuais e na Tabela 7 são apresentados os parâmetros dos modelos ajustados.

	Número de Parâmetros	Marginal	BIC	Parâmetro m	Parâmetros
ARIMA	2	Laplace	1762.779	0.015	0.4
ANN	2	Laplace	1744.879	0.008	0.394
LSTM	2	Laplace	1733.543	-0.021	0.395

Tabela 7 – Parâmetros das distribuições multivariadas dos resíduos dos modelos.

Com as distribuições multivariadas dos erros são calculadas as funções de distribuições acumuladas (FDA) usando como parâmetros as distribuições multivariadas dos resíduos, essas FDA representam os v_i da fórmula 3.2. Após isso, é calculado a melhor distribuição multivariada baseada em cópula para os modelos. Essa função do algoritmo ajusta os modelos de cópulas nos FDA dos erros e calcula o BIC de cada cópula. Ao final a função que obtiver o menor BIC é selecionada para realizar a previsão. Para os modelos estudados nesse exemplo foi utilizado o modelo de cópula normal com os parâmetros $\rho = (0,983, 0,988 \text{ e } 0,99)$ para os as FDAs dos modelos ARIMA, ANN e LSTM respectivamente.

Com a melhor distribuição multivariada baseada em cópula calculada então é feita a previsão de cópulas combinando os modelos através da técnica de pontos espaçados uniformemente (*evelyn spaced points*). Primeiramente é criada sequencia de 1000 pontos cujo valor mínimo e máximo são definidos pelos valores previsto nos modelos. Essa sequência é espaçada identicamente por uma variável Δ que é definido por:

$$\Delta = \frac{\max - \min}{1000} \quad (6.3)$$

O eixo y dessa função é definida pela distribuição multivariada baseada em cópula dos resíduos dos modelos como está na Equação 3.4. O ponto de máximo dessa distribuição é a previsão do modelo de cópulas para cada ponto das distribuições calculadas. A Figura 27 contém uma representação de previsão por cópulas através da técnica pontos espaçados uniformemente.

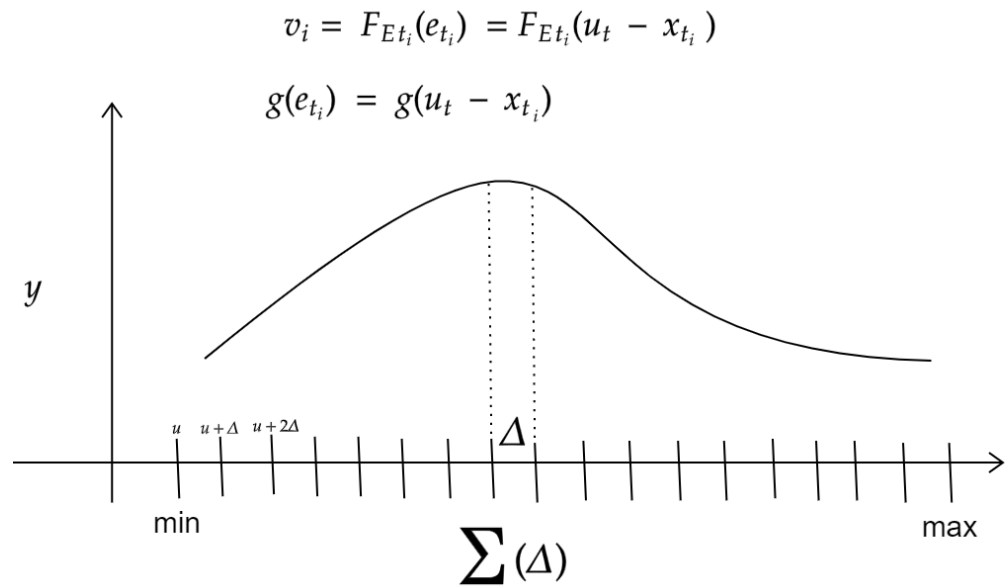


Figura 27 – Representação gráfica do processo de estimação da previsão de cópulas.

6.1.6 Redes neurais

A combinação através de redes neurais é feita a partir de um mapeamento dos modelos individuais como entrada de uma rede MLP que têm a sua saída a previsão combinada do modelo. Para as previsões com esse modelo foi utilizado o pacote `neuralnet` (FRITSCH; GUENTHER, 2019).

Inicialmente é gerado um modelos de rede neural utilizando a função `neuralnet` com os conjuntos de teste dos modelos individuais para treinar a MLP. Os parâmetros dessa função estão presentes na Tabela 8.

Parâmetro	Descrição
formula	uma descrição simbólica do modelo a ser ajustado.
data	número de lags utilizado com entrada da rede
hidden	um vetor de inteiros especificando o número de neurônios ocultos (vértices) em cada camada.
threshold	um valor numérico especificando o limite para as derivadas parciais da função de erro como critério de parada.
stepmax	os passos máximos para o treinamento da rede neural. Atingir esse máximo leva a uma parada do processo de treinamento da rede neural.
rep	o número de repetições para o treinamento da rede neural.
algorithm	string contendo o tipo de algoritmo para calcular a rede neural. Os seguintes tipos são possíveis: 'backprop', 'rprop+', 'rprop-', 'sag' ou 'slr'. 'backprop' refere-se a retropropagação,
act.fct	uma função diferenciável que é usada para suavizar o resultado do produto cruzado da co-variável ou neurônios e os pesos. Além disso, as strings, 'logística' e 'tanh' são possíveis para a função logística e tangente hiperbólica.
err.fct	uma função diferenciável que é usada para o cálculo do erro. Alternativamente, as strings 'sse' e 'ce' que representam a soma dos erros quadrados e a entropia cruzada podem ser usadas.

Tabela 8 – Parâmetros da função `nntrain()`.

A chamada da função para ajustar a rede é `neuralnet(serie~ANN+ARIMA+LSTM,data=df,hidden=3,act.fct = "logistic")`. O algoritmo de treinamento para a rede foi o de *back-propagation*, a função de ativação foi a logística e taxa de aprendizado foi de 0.01. A Figura 28 apresenta o modelo MLP de combinação das previsões individuais para o conjunto de treinamento da série PB.

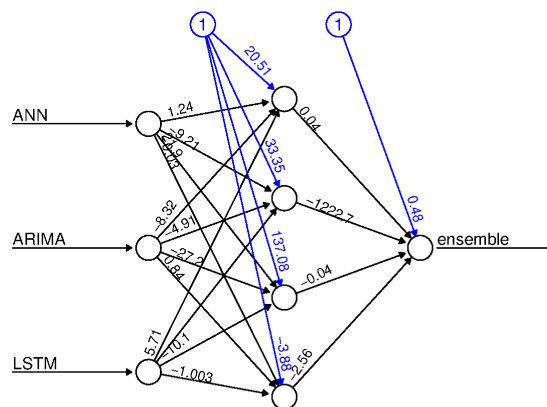


Figura 28 – Esquema da MLP para a combinação dos três modelos individuais.

Logo com modelo ajustado pelo treinamento da MLP, esse é utilizado para combinar

as entradas da rede que são os modelos individuais gerando assim a predição combinada pela MLP.

6.2 Métricas de avaliação dos modelos preditivos

A avaliação da acurácia dos modelos preditores das séries temporais é um fator crítico em sua análise (MAKRIDAKIS; HIBON, 2000). De fato, o uso de uma única métrica pode não ser suficiente para analisar o desempenho do modelo e talvez não revele muito claramente a qualidade da previsão (JUNIOR et al., 2010). Dessa forma serão usados para avaliar 3 métricas para avaliar a acurácia dos modelos apresentados. Uma das métricas mais utilizadas é a raiz do erro quadrático médio (*Root Mean Square Error* (RMSE)):

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (d_j - p_j)^2} \quad (6.4)$$

em que N representa o número de observações da série, d_j é o valor real da série no ponto j e p_j é o valor da predição. Quanto menor for o valor do RMSE melhor é o preditor.

Outra medida bastante utilizada é o *Mean Absolute Percentage Error* (MAPE). Essa medida é uma porcentagem do erro da previsão, definida por:

$$MAPE = \frac{100}{N} \sum_{j=1}^N \left| \frac{d_j - p_j}{d_j} \right| \quad (6.5)$$

Quanto menor for o valor do MAPE melhor é o preditor.

A medida *Prediction of Change in Direction* (POCID) é definida por:

$$POCID = 100 \frac{\sum_{j=1}^N D_j}{N} \quad (6.6)$$

em que

$$D_j = \begin{cases} 1, & \text{se } (d_j - d_{j-1})(p_j - p_{j-1}) > 0 \\ 0, & \text{caso contrário} \end{cases}$$

Nessa medida, D_j indica se a predição j acerta a tendência da série. Essa medida avalia a porcentagem de decisões corretas das estimativas em relação ao valor da série crescer ou decrescer no próximo ponto. Quanto maior for a porcentagem do POCID melhor é o preditor.

Por fim a medida U de Theil é uma medida de eficácia de previsão usada para avaliar o desempenho de um modelo de previsão em relação a um ruído branco. Ela fornece uma indicação de quão bem o modelo está realizando em comparação com uma abordagem simples de previsão, como a média dos valores observados.

A fórmula para calcular a métrica U de Theil é dada por:

$$U = \frac{\sqrt{\frac{1}{n} \sum_{t=1}^n \left(\frac{d_j - p_j}{y_t} \right)^2}}{\sqrt{\frac{1}{n} \sum_{t=1}^n \left(\frac{d_j - \bar{y}}{d_j} \right)^2}}$$

Um valor de U próximo a 1 indica que o modelo está performando tão bem quanto um *random walk*, enquanto valores menores que 1 indicam que o modelo está superando a previsão de um *random walk*.

7 Resultados

Nessa tese as séries estudadas foram modeladas e previstas com os três abordagens de previsão individuais (ARIMA, ANN, LSTM), e essas abordagens foram utilizadas nos modelos combinados (Média, Mediana, Cópulas e MLP) para combinarem as previsões dos modelos individuais. Essas previsões foram avaliadas pelas métricas RMSE, MAPE e POCID. Nas próximas subseções serão avaliadas as comparações entre os resultados obtidos nessas séries e uma avaliação comparativa com alguns trabalhos relacionados. Além disso, em outra subseção será apresentado uma análise estudentizada sobre o comportamento de cada modelo em relação a previsão feita para o conjunto de testes.

7.1 Avaliações dos modelos

É apresentado na Tabela 9 os resultados das métricas aplicadas aos modelos individuais e combinadores em relação ao conjunto de teste. Os melhores resultados por métrica para cada série estão destacados em negrito na tabela. É importante ressaltar que as séries avaliadas estão em diferentes escalas. Por exemplo, a série SP têm valor máximo de 1565.15 USD (moeda corrente americana) enquanto que para a série PB o seu valor máximo é 52.51 BRL. Essa diferença aponta resultados em diferentes escalas do RMSE. Contudo para as outras métricas, como elas apresentam os resultados normalizados (porcentagem), não ocorre a mesma situação.

Métrica	Modelo	PB	NQ	GG	SP	GU
RMSE	ARIMA	0.5662	0.03411	4.1984	0.9314	0.0099
	ANN	0.1709	0.0309	2.2599	2.3374	0.0098
	LSTM	0.0735	0.0188	1.7498	1.0144	0.0013
	COPULA	0.0532	0.0015	1.197	0.7899	0.2915
	SA	0.2065	0.0205	1.9019	1.0628	0.0066
	SM	0.131	0.0207	1.9343	0.9201	0.0097
	MLP	0.1229	0.0188	1.4451	0.8053	0.0015
MAPE	ARIMA	2.3386	0.2451	2.6426	0.0605	0.4461
	ANN	0.502	0.2163	1.2634	0.1399	0.4469
	LSTM	0.5154	0.2259	0.9213	0.0698	0.0744
	COPULA	0.2568	0.0153	0.7034	0.0375	0.0764
	SA	0.8174	0.1775	1.0713	0.0637	0.3016
	SM	0.4679	0.1841	1.0464	0.0608	0.4366
	MLP	0.2906	0.2074	0.8158	0.0423	0.078
POCID	ARIMA	44.8441	75.2	61.5384	99.4553	74.6056
	ANN	89.8481	76	80.7692	96.1873	74.5106
	LSTM	97.8417	76	76.923	99.6732	89.9366
	COPULA	94.5621	76	76.923	99.3846	88.9671
	SA	78.8169	76	73.0769	98.4749	79.8637
	SM	88.0095	76	69.2307	98.6928	74.9223
	MLP	93.7649	76	80.7692	99.1285	89.8416
U de THEIL	ARIMA	0.0791	0.338	0.886	0.432	2.11
	ANN	0.0779	0.333	0.586	0.78	1.84
	LSTM	0.0783	0.017	0.252	0.53	0.07
	COPULA	0.076	0.002	0.057	0.331	0.05
	SA	0.0782	0.223	0.493	0.578	1.33
	SM	0.0781	0.328	0.57	0.422	1.84
	MLP	0.0781	1.944	0.3	0.389	0.82

Tabela 9 – Tabela com o resultado das métricas das previsões dos modelos para o conjunto de teste.

Através de uma análise dos resultados apresentados na Tabela 9, dentre os modelos individuais, o resultados apresentados pelo modelo LSTM foram os mais bem avaliados. Esse resultado correspondeu com as expectativas criadas no delineamento do experimento, pois uma vez que a rede LSTM é um modelo relativamente mais sofisticado, algumas características dele apresentam um resultado superior aos demais modelos individuais. Além disso, entre os modelos combinados, as cópulas apresentaram melhores resultados neste experimento. Inclusive para a métrica U de Theil, os modelos de cópulas apresentaram resultados superiores em todas as séries avaliadas. Além do mais, avaliando a métrica u de Theil poucas amostras como a previsão dos modelos ARIMA para a série GU obteve

resultado superior a 1 indicando que o modelo é pior do que um *random walk*. Em geral, os modelos combinados apresentam melhores resultados que os modelos únicos, corroborando com muitos trabalhos na literatura (HOLLAUER; ISSLER; NOTINI, 2008). Esses resultados provaram que para as séries estudadas e utilizando a configuração dos modelos presentes neste trabalho, a combinação por meio de cópulas produziu resultados relevantes em termos de acurácia e com um custo computacional similar aos outros modelos uma vez que o seu tempo de execução não teve diferença relevante na escala de milissegundos em relação aos outros modelos.

Nas avaliações realizadas nesse trabalho, para as séries GG e NQ, os modelos LSTM e as funções de cópulas tiveram melhores resultados na métrica RMSE do que a maioria dos modelos apresentados em (FIRMINO; NETO; FERREIRA, 2014), onde algumas das séries temporais abordadas aqui também são avaliadas nesse trabalho. Além disso, na mesma metodologia, os resultados do MAPE da Copula e do LSTM foram superiores para a série NQ em comparação com outras metodologias (FERREIRA; VASCONCELOS; ADEODATO, 2008). Em relação à série GU, MLP, Copulas e LSTM tiveram melhor acurácia do que os preditores apresentados por outros autores (OLIVEIRA; LUDERMIR, 2016). Além disso, pela Tabela 9 é possível observar que os experimentos de cópulas apresentaram os melhores resultados, exceto para a série GU, onde o modelo individual LSTM alcançou o melhor desempenho de previsão. No entanto, de acordo com o Teste t-Student, é possível afirmar que com um intervalo de 95% de confiança para a série GU os resultados sobre LSTM e Cópulas não são estatisticamente diferentes.

O teste de Kolmogorv-Smirnov foi aplicado nos valores previstos sobre os modelos cópulas e LSTM, o p-valor foi de 1, o que significa que o teste não rejeitou a hipótese nula, que defende que a distribuição dos dois modelos são iguais. Portanto em relação à série GU os resultados sobre cópulas e LSTM são estatisticamente iguais.

7.2 Análise estudentizada

Como é possível observar através dos resultados apresentados nos Anexos: A, B, C, E e D, em muitas avaliações comparativas dos resultados realizados aqui nessa tese, os valores entre o conjunto de teste e as saídas dos modelos são muito próximos e, em muitas comparações, é difícil ver as diferenças entre o desempenho dos modelos. Portanto, os resíduos estudentizado (do inglês *studentized*) foram calculados usando a função `rstudent` do pacote `Rfit` (KLOKE; MCKEAN, 2012) do R.

Esta medida mostra quantos desvios padrão são observados em uma comparação entre o valor conjunto de teste e o modelo ajustado considerando todas as observações

(DUNEA; POHOATA; IORDACHE, 2015). Um resíduo estudentizado é calculado a partir da divisão de um resíduo pelo seu desvio padrão.

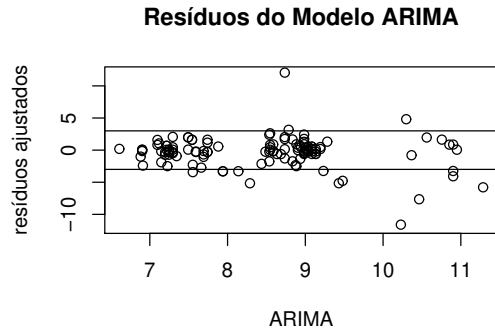
Neste trabalho, foi definido um valor absoluto de 3 para limitar os *outliers* na análise de resíduos estudentizados. O valor 3 foi baseado na regra empírica, que afirma que 99,7% dos dados observados, seguindo uma distribuição normal, estão dentro de 3 desvios padrão da média (LIND et al., 2006).

Devido ao número de modelos e séries observados neste artigo, os resultados da análise de resíduos estudentizados estão resumidos na Tabela 10. A tabela mostra uma proporção entre o número de *outliers* e o número de observações no conjunto de teste. Quanto menor a proporção de *outliers*, melhor é o modelo. Os melhores resultados para cada série temporal estão em negrito. Pode-se concluir que as séries PB, SP e GU são as séries mais desafiadoras, enquanto as séries GG e NQ foram as melhor ajustadas pelos modelos. O combinador baseado em cópulas foi o que obteve menor proporção de *outliers*, em média.

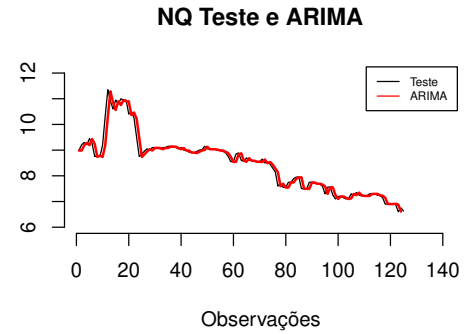
Modelo \ Série	PB	NQ	GG	SP	GU	Média Modelo
ARIMA	0.0214	0.088	0	0.0359	0.0133	0.0264
ANN	0.0199	0.072	0.0384	0.0566	0.0155	0.0289
LSTM	0.0111	0.08	0.0384	0.0185	0.0114	0.0265
COPULA	0.0143	0	0	0.0228	0.012	0.0081
SA	0.0238	0.056	0.0384	0.0511	0.0161	0.0309
SM	0.0119	0.116	0.0769	0.0185	0.0164	0.0438
MLP	0.0175	0.032	0.0384	0.0098	0.0123	0.0338
Média dos Modelos	0.0171	0.0634	0.0329	0.0304	0.0138	0.0286

Tabela 10 – Tabela com a proporção de *outliers* para cada análise residual estudentizada.

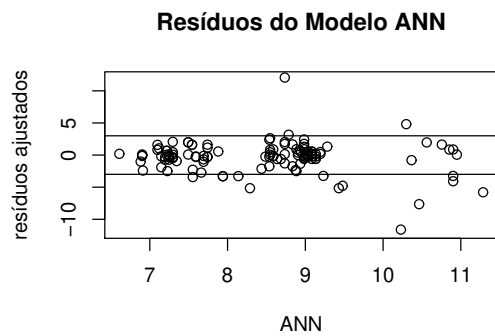
Para melhor representar a análise dos resíduos estudentizados, a Figura 29 mostra os resíduos estudentizados sobre os modelos individuais de previsão ajustados para a série NQ. O círculo que está fora das linhas delimitadas são os valores discrepantes no processo. Além disso, além do gráfico estudentizados, há um gráfico do conjunto de teste NQ com as previsões ARIMA e Cópulas.



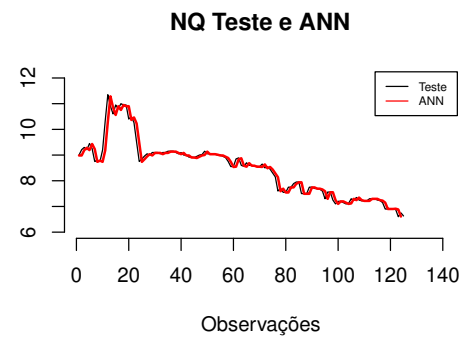
(a) Resíduos Estudentizados ARIMA



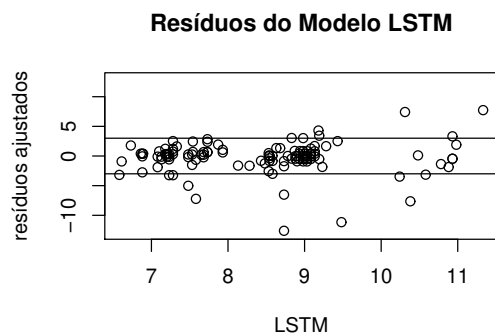
(b) Previsões do modelo ARIMA



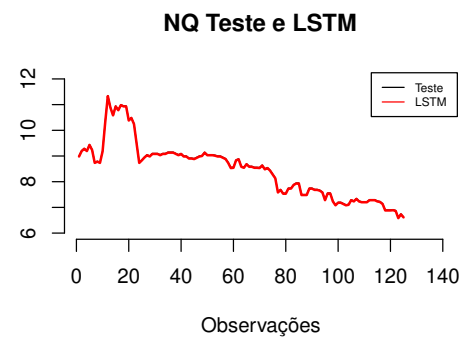
(c) Resíduos Estudentizados ANN



(d) Previsões do modelo ANN



(e) Resíduos Estudentizados LSTM

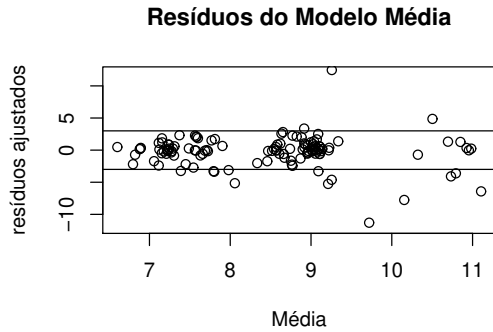


(f) Previsões do modelo LSTM

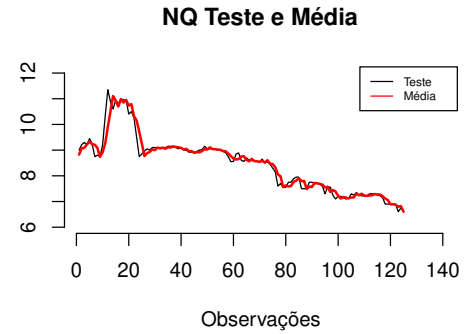
Figura 29 – Análise residual estudenterizada para a série NQ a respeito das previsões dos modelos individuais.

A Figura 30 mostra os resíduos estundetizados sobre os modelos combinadores de previsão ajustados para a série NQ.

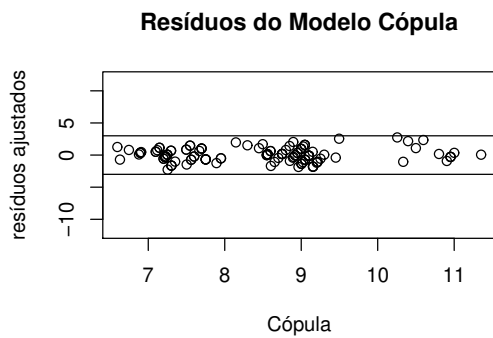
É possível observar que quanto mais alinhado ao centro das linhas delimitadas entre -3 e 3 melhor é o desempenho do modelo. Para a série NQ especificamente os desempenhos dos modelos combinados MLP e Cópulas tiveram um bom ajuste e promoveram melhores resultados em suas previsões.



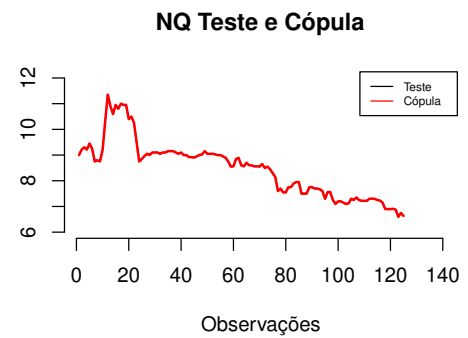
(a) Resíduos Estudentizados Média



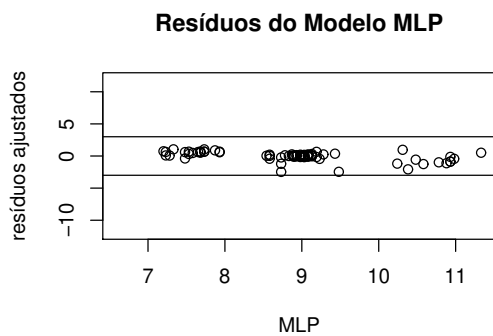
(b) Previsões do modelo Média



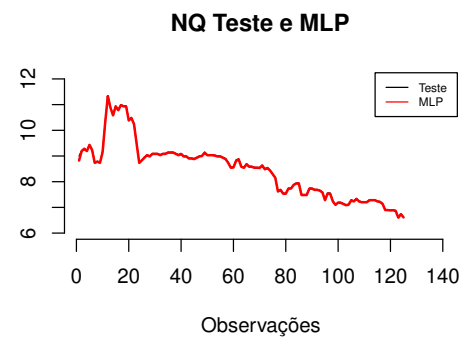
(c) Resíduos Estudentizados Cópulas



(d) Previsões do modelo Cópulas



(e) Resíduos Estudentizados MLP



(f) Previsões do modelo MLP

Figura 30 – Análise residual estudenterizada para a série NQ a respeito das previsões dos modelos combinados.

O apêndice dessa tese apresenta as análises detalhadas realizadas em NQ para as séries: PB Apêndice A, SP Apêndice A, GG Apêndice C e GU Apêndice D. Além disso os resultados da mediana para a série NQ E.

Em termos da previsão de modelos individuais os modelos LSTM têm se mostrado superiores a outros modelos individuais tradicionais na previsão de séries temporais. Comparados aos modelos ARIMA, os modelos LSTM podem capturar relações não lineares e padrões complexos, o que é particularmente útil para previsões em séries temporais não lineares. Além disso, os modelos LSTM são capazes de lidar com variações de longo prazo em dados temporais, o que os torna eficazes em previsões de séries com tendências de longo prazo. Comparados aos modelos de redes neurais tradicionais, que também são capazes de lidar com relações não lineares, os modelos LSTM podem capturar dependências de longo prazo e têm melhor desempenho na previsão de séries com ciclos de longa duração. Sendo assim o melhor desempenho individual nesse estudo ficou com o modelo LSTM. Já em termos de previsão dos modelos combinados as funções de cópulas se mostraram superiores aos modelos de média e mediana simples e aos de redes neurais MLP. Isso ocorreu porque as cópulas são capazes de modelar de forma mais precisa a dependência entre diferentes variáveis, o que é essencial para a previsão de séries temporais. Ao contrário dos modelos de média e mediana simples, que supõem que as variáveis são independentes umas das outras, as cópulas levam em consideração a dependência de longo prazo entre as variáveis, o que as torna mais eficazes na previsão de séries com relações complexas. Comparadas aos modelos de redes neurais como combinadores, as cópulas também apresentam vantagens, pois são menos suscetíveis a *overfitting*. Sendo assim como modelo individual, o LSTM teve destaque nesse estudo bem como o modelo das funções de cópulas para os modelos combinados.

8 Conclusão

Os modelos combinados de previsão têm sido amplamente explorados e estudados na área de séries temporais. Em particular, nas séries temporais financeiras, que frequentemente apresentam alta volatilidade, esses modelos desempenham um papel fundamental devido à capacidade de capturar relações não lineares presentes nesses processos estocásticos complexos. A volatilidade e imprevisibilidade dos mercados financeiros exigem abordagens sofisticadas que possam lidar com a natureza dinâmica e não linear dessas séries. Os modelos de previsão combinados se destacam como uma solução promissora, pois têm o potencial de combinar as vantagens de diferentes métodos de previsão, compensando suas limitações individuais. Ao integrar abordagens lineares e não lineares em uma estrutura conjunta, esses modelos oferecem uma perspectiva abrangente e mais precisa para a previsão de séries temporais financeiras. Portanto, o estudo e desenvolvimento de modelos combinados de previsão desempenham um papel crucial na vanguarda da pesquisa em séries temporais financeiras.

Com relação aos objetivos do trabalho, o estudo e coleta de séries temporais financeiras relevantes para o desenvolvimento do trabalho permitiu que explorasse dados e informações importantes, fornecendo uma base sólida para a criação dos modelos de predição. Ao estudar modelos lineares e não lineares de acoplamento de preditores de séries temporais, avaliou-se diferentes técnicas e avaliar pontos fortes e fracos de cada abordagem, ponderando sua representatividade em reduzir a magnitude do erro da previsão. Por fim, a experimentação da combinação de diferentes modelos de cópulas, mostrou-se uma técnica promissora e eficaz para a previsão de séries temporais, indicando também que quanto mais modelos de cópulas forem inseridos na ferramenta melhores podem ser os resultados.

Os resultados apresentados por meio da tese correspondem à literatura ao ilustrar que as estratégias de combinação, em geral, são superiores aos modelos individuais. Nesta tese, foram combinados três modelos individuais por meio de quatro abordagens. Olhando para os resultados, ficou claro que o formalismo de cópulas teve em geral os melhores resultados de acordo com as métricas aplicadas. Além disso, os modelos de cópulas refletem os pontos fortes e fracos de todos os modelos envolvidos. No entanto, em algumas séries como GU e PB, a função cópulas não obteve os melhores resultados em todas as métricas. Nessas séries, as funções de cópulas apresentaram bons resultados, mas não os melhores no conjunto de testes. De fato, algumas séries foram mais desafiadoras para os modelos de

cópuas. As razões para este resultado devem ser mais investigadas, mas um fato que pode ser destacado é que essas são as séries com mais observações entre as cinco estudadas.

Em termos de combinação, além dos modelos aqui utilizados, existem várias técnicas a serem ainda envolvidas, *e.g.* regressão baseada em vetores de suporte, floresta aleatória, *boosting*, entre outras. Em comparação com muitos trabalhos na literatura, o presente trabalho ilustrou o potencial do uso de funções de cópula como um combinador de previsões. Fica evidente também que a combinação dos modelos mais utilizados em suas áreas de previsão fortalece o conjunto combinado. Assim, a combinação de modelos por meio de cópulas parece ser um forte candidato para levar a resultados precisos no que diz respeito à previsão de séries temporais financeiras.

Durante o período do doutorado foi desenvolvido um software com todo o arcabouço para experimentação dos resultados do trabalho e registrado sob a licença de número do registro: BR5120210016157 com título: "Combinador de Preditores de Séries Temporais", Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial. Ademais, um trabalho com as considerações dos resultados dessa tese foi enviado e está sob avaliação da revista *Expert Systems with Applications*. Além disso, no mesmo período foram publicados trabalhos em: Congresso do Rbras (Região Brasileira da Sociedade Internacional de Biometria), RITA (Revista de Informática Teórica e Aplicada), ESWA (*Expert Systems with Applications*) e IJCA (*International Journal of Computer Applications*)

Como trabalhos futuros aos produzidos aqui podem ser abordadas duas linhas de pesquisa. Uma delas se estende pela concepção e criação de novos modelos de cópulas para verificar e validar a sua acurácia em relação a previsão de séries financeiras. Uma outra vertente dessa pesquisa é seguir a abordagem realizada aqui em uma investigação sobre o desempenho das funções de cópulas em comparação com abordagens de combinação como por exemplo o SVM (*support vector machine*).

A previsão de séries temporais é uma tarefa difícil. A série temporal financeira possui algumas propriedades que aumentam a complexidade de sua modelagem. No entanto, séries temporais financeiras podem ser um dos assuntos principais na previsão de séries temporais atualmente. Portanto, o uso do modelo mais robustos para previsão de séries temporais financeiras é indispensável. Finalmente, como mostrado nesta tese por seus resultados, combinar modelos de previsão usando funções de cópulas pode ser uma escolha atraente para previsão de séries temporais financeiras.

Referências bibliográficas

- ADHIKARI, R. A neural network based linear ensemble framework for time series forecasting. **Neurocomputing**, v. 157, p. 231–242, 2015. ISSN 0925-2312. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925231215000338>>.
- ADHIKARI, R.; AGRAWAL, R. K. A novel weighted ensemble technique for time series forecasting. In: TAN, P.-N. et al. (Ed.). **Advances in Knowledge Discovery and Data Mining**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 38–49. ISBN 978-3-642-30217-6.
- ALADAG, C. H.; EGRIOGLU, E.; YOLCU, U. Forecast combination by using artificial neural networks. **Neural Processing Letters**, Springer, v. 32, n. 3, p. 269–276, 2010.
- ALGHOUL, A. et al. Email classification using artificial neural network. **IJARW**, 2018.
- ANDRAWIS, R. R.; ATIYA, A. F.; EL-SHISHINY, H. Forecast combinations of computational intelligence and linear models for the nn5 time series forecasting competition. **International Journal of Forecasting**, v. 27, n. 3, p. 672–688, 2011. Disponível em: <<https://EconPapers.repec.org/RePEc:eee:intfor:v:27:y:i:3:p:672-688>>.
- ARMSTRONG, J. S. **Principles of forecasting: a handbook for researchers and practitioners**. [S.l.]: Springer, 2001. v. 30.
- AZZALINI, A. **The skew-normal and related families**. [S.l.]: Cambridge University Press, 2013. v. 3.
- BARZEKAR, H.; YU, Z. C-net: A reliable convolutional neural network for biomedical image classification. **Expert Systems with Applications**, Elsevier, v. 187, p. 116003, 2022.
- BATES, J. M.; GRANGER, C. W. The combination of forecasts. **Journal of the Operational Research Society**, Taylor & Francis, v. 20, n. 4, p. 451–468, 1969.
- BENGIO, Y. **Learning deep architectures for AI**. [S.l.]: Now Publishers Inc, 2009.
- BENGIO, Y.; GOODFELLOW, I.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press Cambridge, MA, USA, 2017. v. 1.
- BERGMEIR, C.; HYNDMAN, R. J.; KOO, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. **Computational Statistics & Data Analysis**, Elsevier, v. 120, p. 70–83, 2018.
- BISHOP, C. M.; NASRABADI, N. M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006. v. 4.

- BOX, G. E. et al. **Time series analysis: forecasting and control**. [S.l.]: John Wiley & Sons, 2015.
- BROWNLEE, J. Long short-term memory networks with python. **Machine learning mastery**, v. 1, p. 1–229, 2017.
- CAVALERI, R.; RIBEIRO, E. P. Combinação de previsões de volatilidade: um estudo. **Revista Economia**, v. 12, n. 2, p. 239–261, 2011.
- CHAGAS, C. S. et al. Utilização de redes neurais artificiais na classificação de níveis de degradação em pastagens. **Revista Brasileira de Engenharia Agrícola e Ambiental**, SciELO Brasil, v. 13, p. 319–327, 2009.
- CHATFIELD, C. **Time-series forecasting**. [S.l.]: Chapman and Hall/CRC, 2000.
- CHEN, D. et al. Ultrasonic signal classification and porosity testing for cfrp materials via artificial neural network. **Materials Today Communications**, Elsevier, v. 30, p. 103021, 2022.
- CHENG, D. et al. Financial time series forecasting with multi-modality graph neural network. **Pattern Recognition**, Elsevier, v. 121, p. 108218, 2022.
- CILIMKOVIC, M. Neural networks and back propagation algorithm. **Institute of Technology Blanchardstown, Blanchardstown Road North Dublin**, v. 15, n. 1, 2015.
- CLETTE, F. et al. Revisiting the sunspot number. **Space Science Reviews**, Springer, v. 186, n. 1, p. 35–103, 2014.
- CUVELIER, E.; NOIRHOMME-FRAITURE, M. Clayton copula and mixture decomposition. **ASMDA 2005**, p. 699–708, 2005.
- de Mattos Neto, P. S. et al. An adaptive hybrid system using deep learning for wind speed forecasting. **Information Sciences**, v. 581, p. 495–514, 2021. ISSN 0020-0255. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0020025521009919>>.
- DONALDSON, R. G.; KAMSTRA, M. Forecast combining with neural networks. **Journal of Forecasting**, Wiley Online Library, v. 15, n. 1, p. 49–61, 1996.
- DONATE, J. P. et al. Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm. **Neural Computing and Applications**, Springer, v. 22, n. 1, p. 11–20, 2013.
- DUNEA, D.; POHOATA, A.; IORDACHE, S. Using wavelet–feedforward neural networks to improve air pollution forecasting in urban environments. **Environmental monitoring and assessment**, Springer, v. 187, n. 7, p. 477, 2015.
- EHLERS, R. S. Análise de séries temporais. **Laboratório de Estatística e Geoinformação. Universidade Federal do Paraná**, v. 1, p. 1–118, 2007.
- EL-KHATIB, M. J.; ABU-NASSER, B. S.; ABU-NASER, S. S. Glass classification using artificial neural network. **IJARW**, 2019.

- ESHEL, G. The yule walker equations for the ar coefficients. **Internet resource**, v. 2, p. 68–73, 2003.
- FATHI, O. Time series forecasting using a hybrid arima and lstm model. **Velvet Consulting**, p. 1–7, 2019.
- FERREIRA, T. A.; VASCONCELOS, G. C.; ADEODATO, P. J. A new intelligent system methodology for time series forecasting with artificial neural networks. **Neural Process. Lett.**, Kluwer Academic Publishers, USA, v. 28, n. 2, p. 113–129, oct 2008. ISSN 1370-4621. Disponível em: <<https://doi.org/10.1007/s11063-008-9085-x>>.
- FIRMINO, P. R. A.; NETO, P. S. de M.; FERREIRA, T. A. Correcting and combining time series forecasters. **Neural Networks**, Elsevier, v. 50, p. 1–11, 2014.
- FRITSCH, S.; GUENTHER, F. **Training of Neural Networks**. [S.l.], 2019. R package version 1.4.2. Disponível em: <<https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf>>.
- FURGANG, K. **How the Stock Market Works**. [S.l.]: The Rosen Publishing Group, Inc, 2010.
- GENEST, C. Frank's family of bivariate distributions. **Biometrika**, Oxford University Press, v. 74, n. 3, p. 549–555, 1987.
- GHEYAS, I. A.; SMITH, L. S. A neural network approach to time series forecasting. In: **Proceedings of the World Congress on Engineering**. [S.l.: s.n.], 2009. v. 2, p. 1–3.
- GHIASSI, M.; SAIDANE, H.; ZIMBRA, D. A dynamic artificial neural network model for forecasting time series events. **International Journal of Forecasting**, Elsevier, v. 21, n. 2, p. 341–362, 2005.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.
- HAJIRAHIMI, Z.; KHASHEI, M. Hybrid structures in time series modeling and forecasting: A review. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 86, p. 83–106, 2019.
- HEBB, D. O. The first stage of perception: growth of the assembly. **The Organization of Behavior**, Wiley, New York, v. 4, p. 60–78, 1949.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.
- HOLLAUER, G.; ISSLER, J. V.; NOTINI, H. H. Prevendo o crescimento da produção industrial usando um número limitado de combinações de previsões. **Economia Aplicada**, SciELO Brasil, v. 12, n. 2, p. 177–198, 2008.
- HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982.

HYNDMAN, R. J.; ATHANASOPOULOS, G. Forecasting: Principles and practice. otexts; 2014. **Online at <http://otexts.org/fpp>**, 2014.

HYNDMAN, R. J.; ATHANASOPOULOS, G. **Forecasting: principles and practice**. [S.l.]: OTexts, 2018.

JAIN, A.; KUMAR, A. M. Hybrid neural network models for hydrologic time series forecasting. **Applied Soft Computing**, Elsevier, v. 7, n. 2, p. 585–592, 2007.

JOSHI, D. M.; RANA, N.; MISRA, V. Classification of brain cancer using artificial neural network. In: IEEE. **2010 2nd international conference on electronic computer technology**. [S.l.], 2010. p. 112–116.

JUNIOR, A. R. L.; FERREIRA, T. A. E. A hybrid method for tuning neural network for time series forecasting. In: **Proceedings of the 10th annual conference on Genetic and evolutionary computation**. [S.l.: s.n.], 2008. p. 531–532.

JUNIOR, A. R. L. et al. An experimental study of fitness function and time series forecasting using artificial neural networks. In: **Proceedings of the 12th annual conference companion on Genetic and evolutionary computation**. [S.l.: s.n.], 2010. p. 2015–2018.

KIRCHGÄSSNER, G.; WOLTERS, J.; HASSLER, U. **Introduction to modern time series analysis**. [S.l.]: Springer Science & Business Media, 2012.

KLOKE, J. D.; MCKEAN, J. W. Rfit: Rank-based estimation for linear models,. **The R Journal**, v. 4, n. 2, p. 57–64, 2012.

LEAL, D. M. B. **Aplicação de cópulas ao ramo vida: Risco de resgate e risco de taxa de juro**. Tese (Doutorado) — Instituto Superior de Economia e Gestão, 2010.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LIN, C. et al. Libd3c: ensemble classifiers with a clustering and dynamic selection strategy. **Neurocomputing**, Elsevier, v. 123, p. 424–435, 2014.

LIND, D. A. et al. **Basic statistics for business & economics**. [S.l.]: Boston: McGraw-Hill/Irwin,, 2006.

LJ, A. R.; NETO, P. S. de M.; FERREIRA, T. A. A prime step in the time series forecasting with hybrid methods: The fitness function choice. In: IEEE. **2009 International Joint Conference on Neural Networks**. [S.l.], 2009. p. 2703–2710.

LJUNG, G. M.; BOX, G. E. On a measure of lack of fit in time series models. **Biometrika**, Oxford University Press, v. 65, n. 2, p. 297–303, 1978.

MAKRIDAKIS, S.; HIBON, M. The m3-competition: results, conclusions and implications. **International journal of forecasting**, Elsevier, v. 16, n. 4, p. 451–476, 2000.

MAKRIDAKIS, S.; WINKLER, R. L. Averages of forecasts: Some empirical results. **Management science**, Informs, v. 29, n. 9, p. 987–996, 1983.

- MALETZKE, A. G. et al. Estudo do parâmetro tamanho de motif para a classificação de séries temporais de ecg. In: **Congresso da Sociedade Brasileira de Computação, Workshop de Informática Médica, Natal, Rio Grande do Norte**. [S.l.: s.n.], 2011. p. 1–10.
- MATSUMOTO, D. K. F. et al. Estudo em séries temporais financeiras utilizando redes neurais recorrentes. Universidade Federal de Alagoas, 2019.
- MAYER, M. J.; YANG, D. Pairing ensemble numerical weather prediction with ensemble physical model chain for probabilistic photovoltaic power forecasting. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 175, p. 113171, 2023.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.
- MCLEOD, A. I. Kendall rank correlation and mann-kendall trend test. **R Package Kendall**, Western Univ., 2005.
- MENEZES, L. M. D.; BUNN, D. W.; TAYLOR, J. W. Review of guidelines for the use of combined forecasts. **European Journal of Operational Research**, Elsevier, v. 120, n. 1, p. 190–204, 2000.
- MEYER, C. The bivariate normal copula. **Communications in Statistics-Theory and Methods**, Taylor & Francis, v. 42, n. 13, p. 2402–2422, 2013.
- MOODY, J.; DARKEN, C. J. Fast learning in networks of locally-tuned processing units. **Neural computation**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 1, n. 2, p. 281–294, 1989.
- MORETTIN, P. A. **Econometria financeira: um curso em séries temporais financeiras**. [S.l.]: Editora Blucher, 2017.
- (MSFT), M. C. **Profile, business summary**. **Yahoo! Finance**. 2020. Disponível em: <<https://finance.yahoo.com/>>.
- NAIK, S.; DESAI, P. Mango (*mangifera indica* l.) classification using convolutional neural network and linear classifiers. In: SPRINGER. **Proceedings of Third International Conference on Sustainable Computing**. [S.l.], 2022. p. 163–172.
- NELSEN, R. B. **An introduction to copulas**. [S.l.]: Springer Science & Business Media, 2007.
- NELSON, D. M. Q. Uso de redes neurais recorrentes para previsão de séries temporais financeiras. Universidade Federal de Minas Gerais, 2017.
- UFRPE E UFCA. A. B> Santos Neto, P. R. A. Firmino e T. A. E. Ferreira. **Combinador de Preditores de Séries Temporais**. 2021. BR512021001615-7, 10/02/2021.
- NETO, P. S. D. M. et al. Neural-based ensembles for particulate matter forecasting. **IEEE Access**, IEEE, v. 9, p. 14470–14490, 2021.

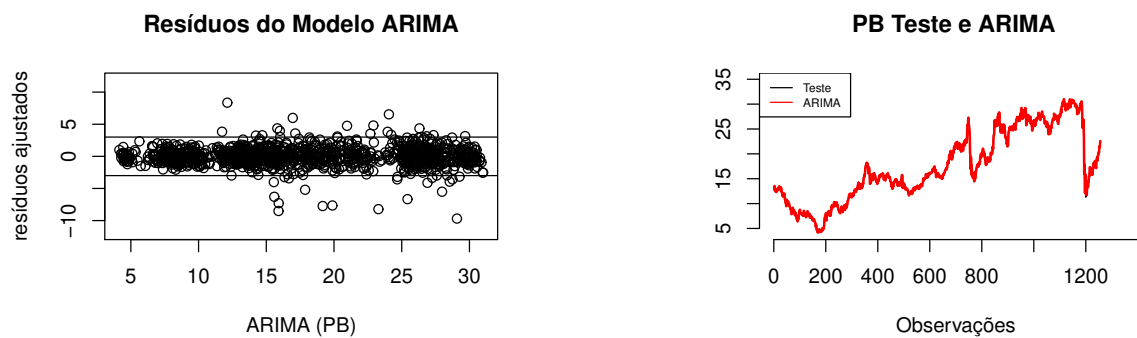
- NEUMAN, S. P. Maximum likelihood bayesian averaging of uncertain model predictions. **Stochastic Environmental Research and Risk Assessment**, Springer, v. 17, n. 5, p. 291–305, 2003.
- NÓBREGA, J. P.; OLIVEIRA, A. L. A combination forecasting model using machine learning and kalman filter for statistical arbitrage. In: IEEE. **2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.], 2014. p. 1294–1299.
- OLIVEIRA, J. F. de; LUDERMIR, T. B. A hybrid evolutionary decomposition system for time series forecasting. **Neurocomputing**, Elsevier, v. 180, p. 27–34, 2016.
- OLIVEIRA, M.; TORGO, L. Ensembles for time series forecasting. In: PMLR. **Asian Conference on Machine Learning**. [S.l.], 2015. p. 360–370.
- OLIVEIRA, R. T. de et al. Copulas-based time series combined forecasters. **Information Sciences**, Elsevier, v. 376, p. 110–124, 2017.
- OLIVEIRA, R. T. de et al. Combining time series forecasting models via gumbel-hougaard copulas. In: IEEE. **2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence**. [S.l.], 2013. p. 568–573.
- O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. **arXiv preprint arXiv:1511.08458**, 2015.
- QIU, H. et al. Linear and non-linear combination forecasting model of varicella incidence in chongqing. **Epidemiology & Infection**, Cambridge University Press, v. 149, 2021.
- R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2020. Disponível em: <<https://www.R-project.org/>>.
- REID, D. J. **A comparative study of time series prediction techniques on economic data**. [S.l.]: University of Nottingham, Library Photographic Unit, 1969.
- ROJAS, R. **Neural networks: a systematic introduction**. [S.l.]: Springer Science & Business Media, 2013.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- ROSS, S. M. **Stochastic processes**. [S.l.]: John Wiley & Sons, 1995.
- RUMELHART, D. E. et al. A general framework for parallel distributed processing. **Parallel distributed processing: Explorations in the microstructure of cognition**, Cambridge, MA: MIT Press, v. 1, n. 45-76, p. 26, 1986.
- SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, Elsevier, v. 61, p. 85–117, 2015.
- SHAH, V. et al. Stock market prediction using neural networks. **Int. J. Soft Comput. Eng**, v. 6, n. 1, 2016.

- SHENSHI, G.; ZHIQIAN, W.; JITAI, C. The fractal research and predicating on the times series of sunspot relative number. **Applied Mathematics and Mechanics**, Springer, v. 20, n. 1, p. 84–89, 1999.
- SILVA, I. N. D. et al. Artificial neural networks. **Cham: Springer International Publishing**, Springer, v. 39, 2017.
- SKLAR, A. Distribution functions of n dimensions and margins. **Publications of the Institute of Statistics of the University of Paris**, v. 8, p. 229–231, 1959.
- SORJAMAA, A. et al. Methodology for long-term prediction of time series. **Neurocomputing**, Elsevier, v. 70, n. 16-18, p. 2861–2869, 2007.
- TEALAB, A. Time series forecasting using artificial neural networks methodologies: A systematic review. **Future Computing and Informatics Journal**, Elsevier, v. 3, n. 2, p. 334–340, 2018.
- TEALAB, A.; HEFNY, H.; BADR, A. Forecasting of nonlinear time series using ann. **Future Computing and Informatics Journal**, Elsevier, v. 2, n. 1, p. 39–47, 2017.
- TISSOT, H. C.; CAMARGO, L. C.; POZO, A. T. Treinamento de redes neurais feedforward: comparativo dos algoritmos backpropagation e differential evolution. In: **Brazilian Conference on Intelligent Systems**. [S.l.: s.n.], 2012.
- TORRES, J. F. et al. Deep learning for time series forecasting: a survey. **Big Data**, Mary Ann Liebert, Inc., publishers 140 Huguenot Street, 3rd Floor New . . . , v. 9, n. 1, p. 3–21, 2021.
- TSAY, R. S. **Analysis of Financial Time Series**. [S.l.]: John Wiley & Sons, 2010. v. 762.
- VOKÓ, Z.; PITTER, J. G. The effect of social distance measures on covid-19 epidemics in europe: an interrupted time series analysis. **GeroScience**, Springer, v. 42, n. 4, p. 1075–1082, 2020.
- WALCZAK, S. Artificial neural networks. In: **Encyclopedia of Information Science and Technology, Fourth Edition**. [S.l.]: IGI Global, 2018. p. 120–131.
- WANG, J.; ZHANG, N.; LU, H. A novel system based on neural networks with linear combination framework for wind speed forecasting. **Energy conversion and management**, Elsevier, v. 181, p. 425–442, 2019.
- WEISS, C. E.; RAVIV, E.; ROETZER, G. Forecast combinations in r using the forecastcomb package. **R Journal**, v. 10, n. 2, 2018.
- WELCH, I.; GOYAL, A. A comprehensive look at the empirical performance of equity premium prediction. **The Review of Financial Studies**, Society for Financial Studies, v. 21, n. 4, p. 1455–1508, 2008.
- WICHARD, J. D.; OGORZALEK, M. Time series prediction with ensemble models. In: **IEEE. 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)**. [S.l.], 2004. v. 2, p. 1625–1630.

- XIE, Q. et al. Multi-scale analysis of influencing factors for soybean futures price risk: Adaptive fourier decomposition mathematical model applied for the case of china. **International Journal of Wavelets, Multiresolution and Information Processing**, World Scientific, v. 19, n. 05, p. 2150017, 2021.
- YOLCU, U.; EGRIOGLU, E.; ALADAG, C. H. A new linear & nonlinear artificial neural network model for time series forecasting. **Decision support systems**, Elsevier, v. 54, n. 3, p. 1340–1347, 2013.
- YOU, Y. et al. Large-batch training for lstm and beyond. In: **Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis**. [S.l.: s.n.], 2019. p. 1–16.
- YULE, G. U. Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers. **Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character**, The Royal Society London, v. 226, n. 636-646, p. 267–298, 1927.
- ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with artificial neural networks:: The state of the art. **International journal of forecasting**, Elsevier, v. 14, n. 1, p. 35–62, 1998.
- ZHANG, G. P. Time series forecasting using a hybrid arima and neural network model. **Neurocomputing**, Elsevier, v. 50, p. 159–175, 2003.
- ZHANG, G. P.; QI, M. Neural network forecasting for seasonal and trend time series. **European journal of operational research**, Elsevier, v. 160, n. 2, p. 501–514, 2005.
- ZIVOT, E.; WANG, J. **Modeling Financial Time Series with S-PLUS**. [S.l.]: Springer Science & Business Media, 2013. v. 191.

Apêndices

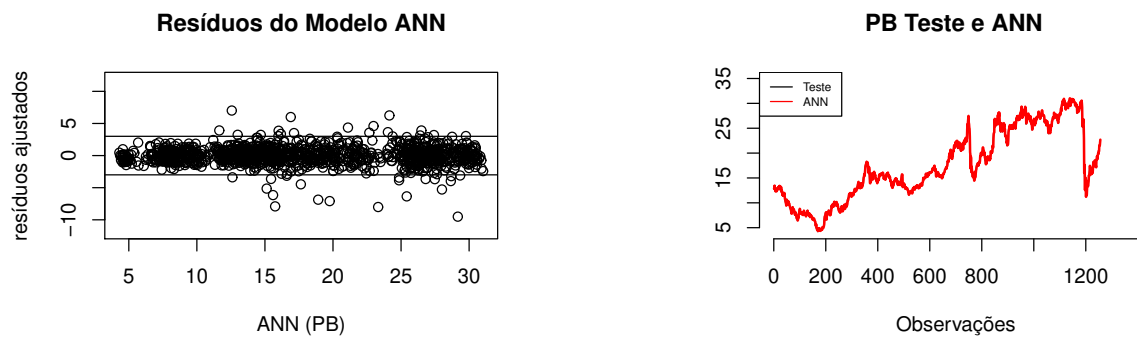
APÊNDICE A – Petrobras (PB)



(a) Resíduos Estudenterizados ARIMA

(b) Previsões do modelo ARIMA

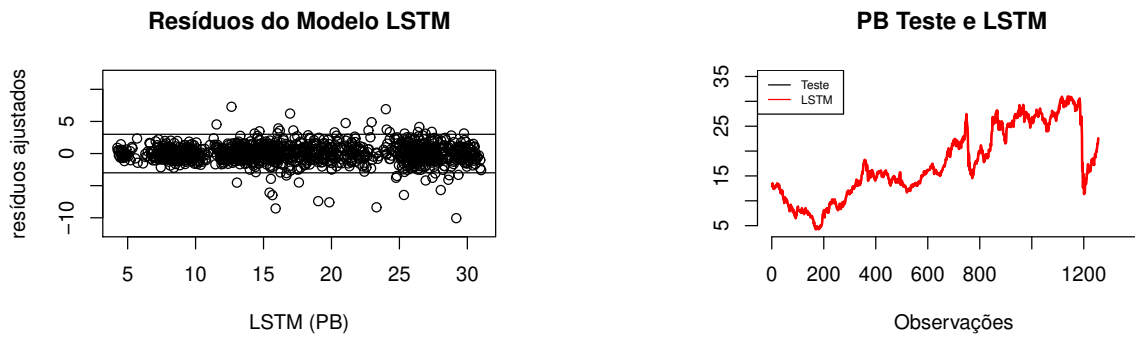
Figura 31 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo ARIMA



(a) Resíduos Estudenterizados ANN

(b) Previsões do modelo ANN

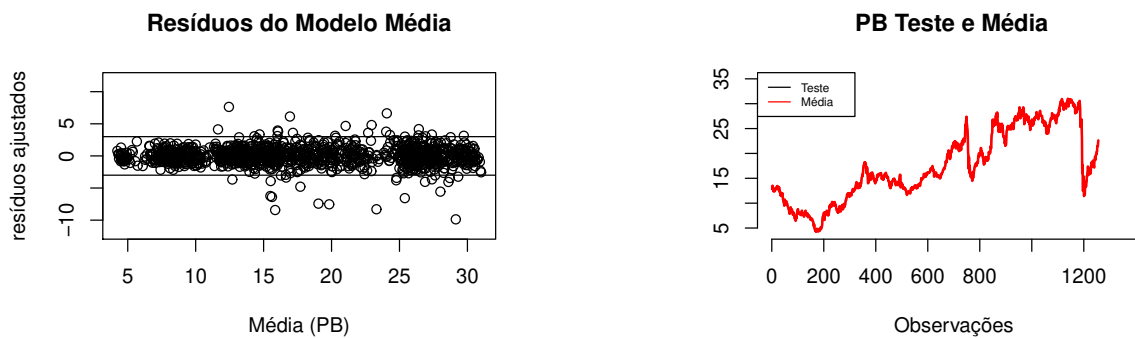
Figura 32 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo ANN



(a) Resíduos Estudenterizados LSTM

(b) Previsões do modelo LSTM

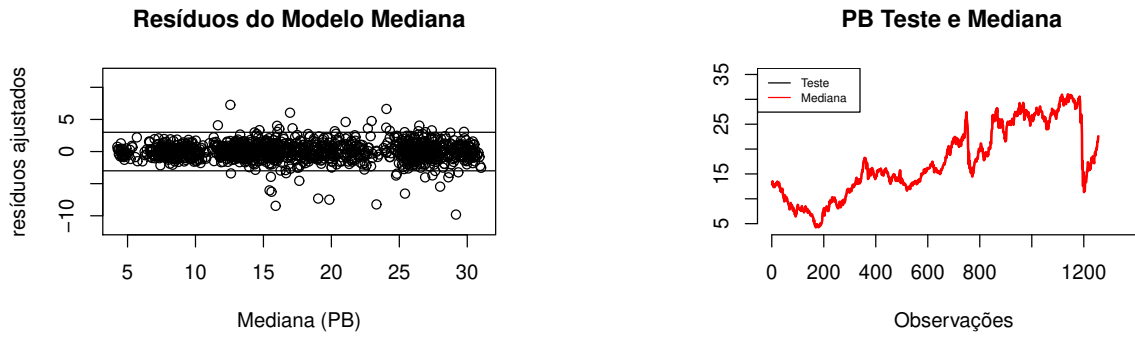
Figura 33 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo LSTM



(a) Resíduos Estudenterizados Média

(b) Previsões do modelo Média

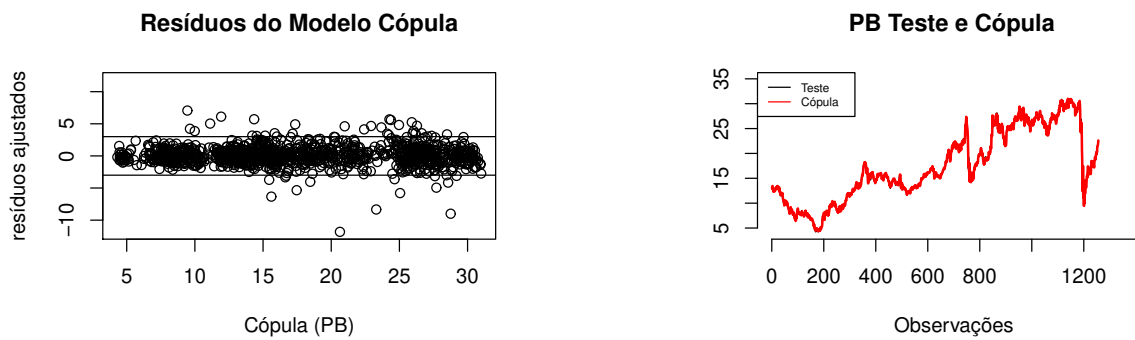
Figura 34 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo Média



(a) Resíduos Estudenterizados Mediana

(b) Previsões do modelo Mediana

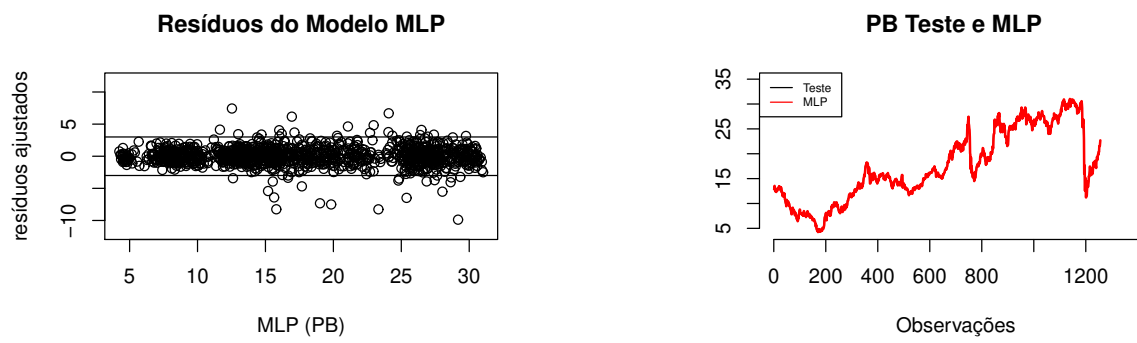
Figura 35 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo Mediana



(a) Resíduos Estudenterizados Cópula

(b) Previsões do modelo Cópula

Figura 36 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo Cópula

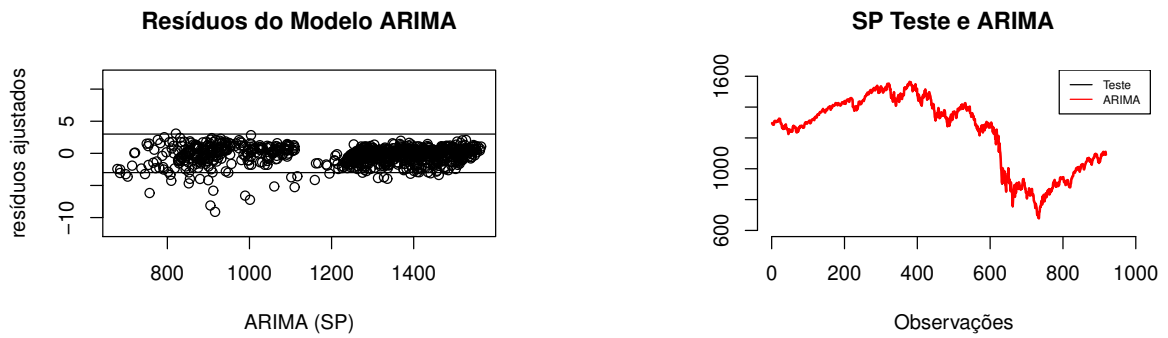


(a) Resíduos Estudenterizados MLP

(b) Previsões do modelo MLP

Figura 37 – Análise residual estudenterizada para a série PB a respeito das previsões do modelo MLP

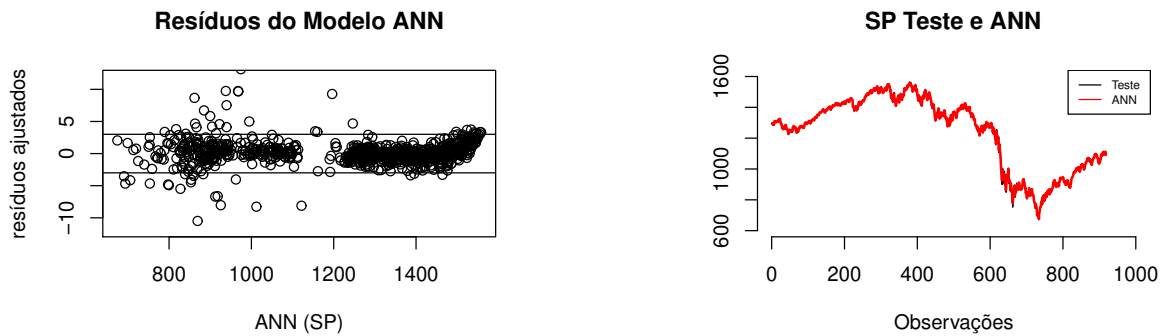
APÊNDICE B – S&P500 (SP)



(a) Resíduos Estudenterizados ARIMA

(b) Previsões do modelo ARIMA

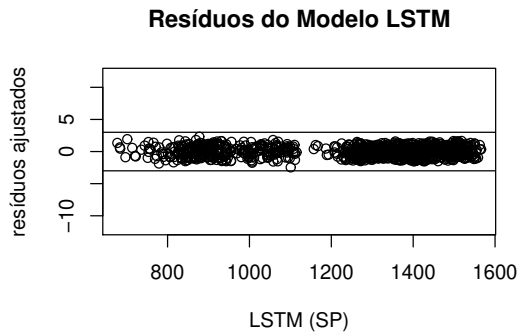
Figura 38 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo ARIMA



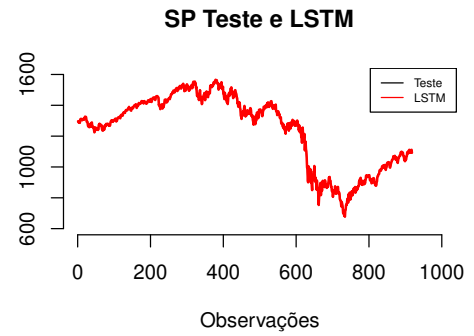
(a) Resíduos Estudenterizados ANN

(b) Previsões do modelo ANN

Figura 39 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo ANN

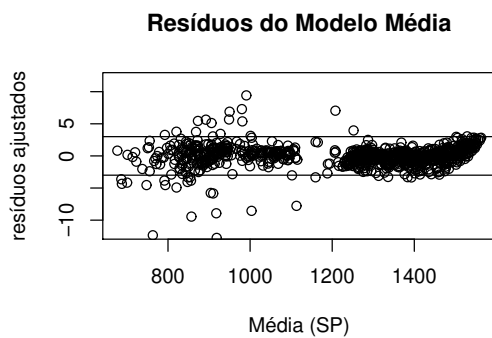


(a) Resíduos Estudenterizados LSTM

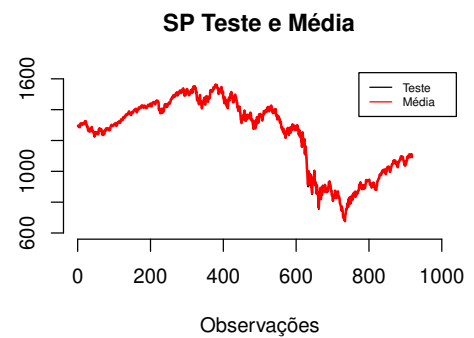


(b) Previsões do modelo LSTM

Figura 40 – Análise residual estudenterizada para a série sp a respeito das previsões do modelo LSTM

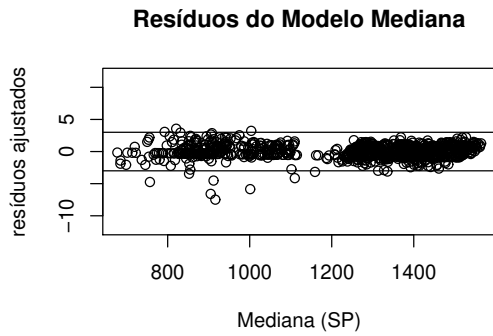


(a) Resíduos Estudenterizados Média

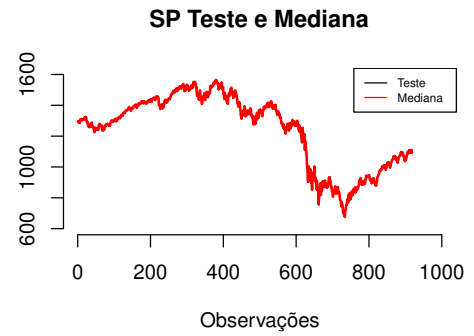


(b) Previsões do modelo Média

Figura 41 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo Média

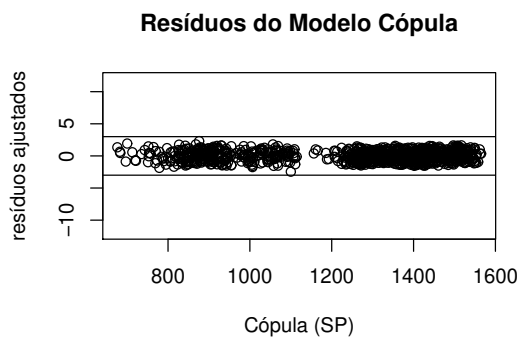


(a) Resíduos Estudentizados Mediana

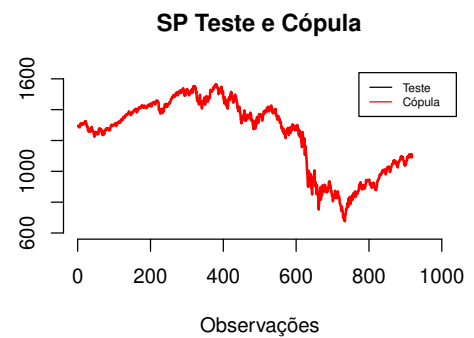


(b) Previsões do modelo Mediana

Figura 42 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo Mediana

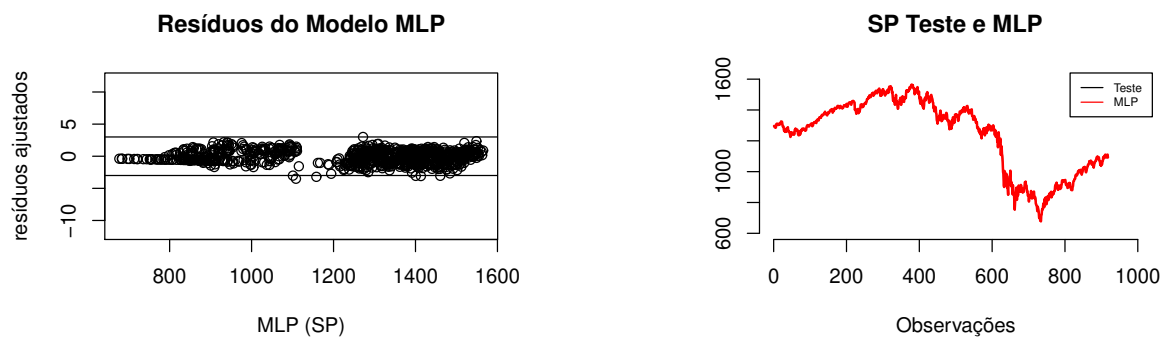


(a) Resíduos Estudentizados Cópula



(b) Previsões do modelo Cópula

Figura 43 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo Cópula

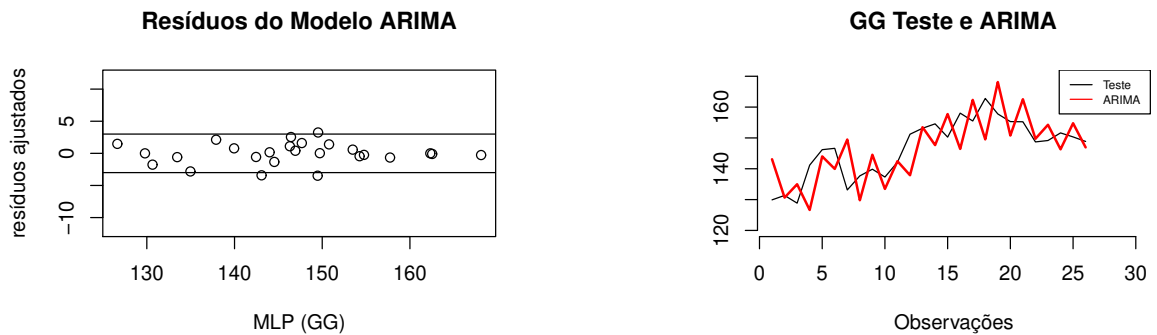


(a) Resíduos Estudenterizados MLP

(b) Previsões do modelo MLP

Figura 44 – Análise residual estudenterizada para a série SP a respeito das previsões do modelo MLP

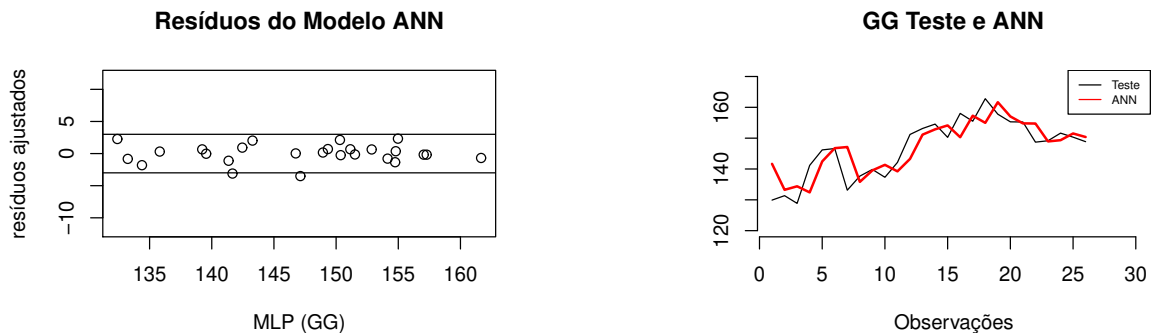
APÊNDICE C – Google (GG)



(a) Resíduos Estudenterizados ARIMA

(b) Previsões do modelo ARIMA

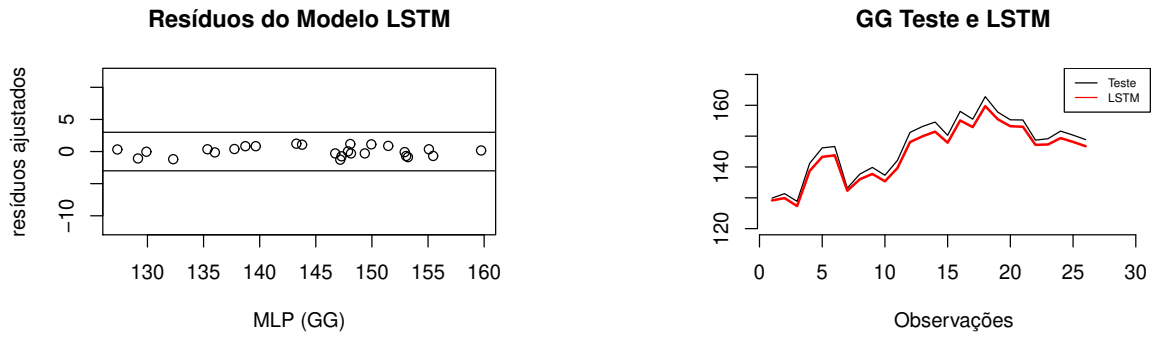
Figura 45 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo ARIMA



(a) Resíduos Estudenterizados ANN

(b) Previsões do modelo ANN

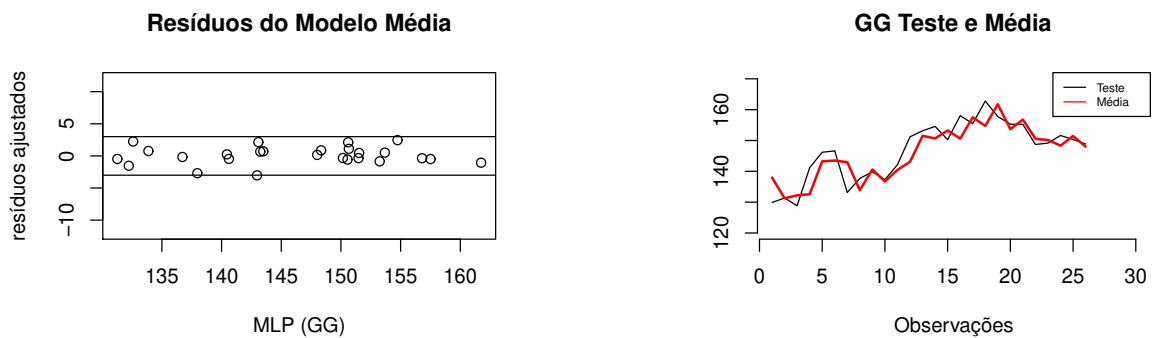
Figura 46 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo ANN



(a) Resíduos Estudenterizados LSTM

(b) Previsões do modelo LSTM

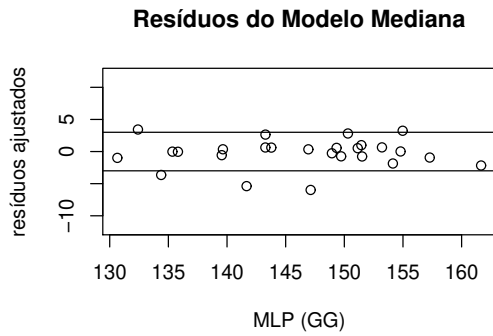
Figura 47 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo LSTM



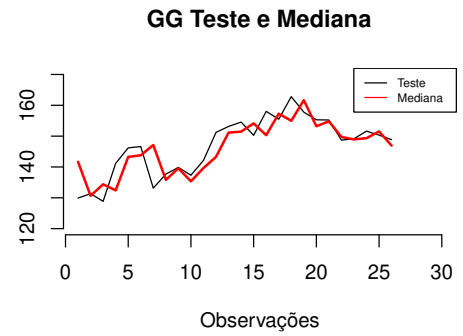
(a) Resíduos Estudenterizados Média

(b) Previsões do modelo Média

Figura 48 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo Média

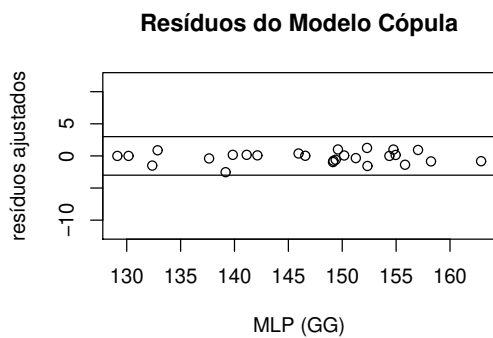


(a) Resíduos Estudentizados Mediana

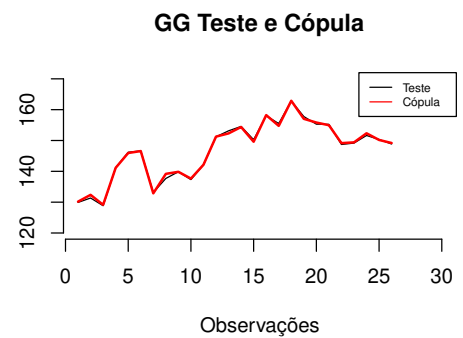


(b) Previsões do modelo Mediana

Figura 49 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo Mediana

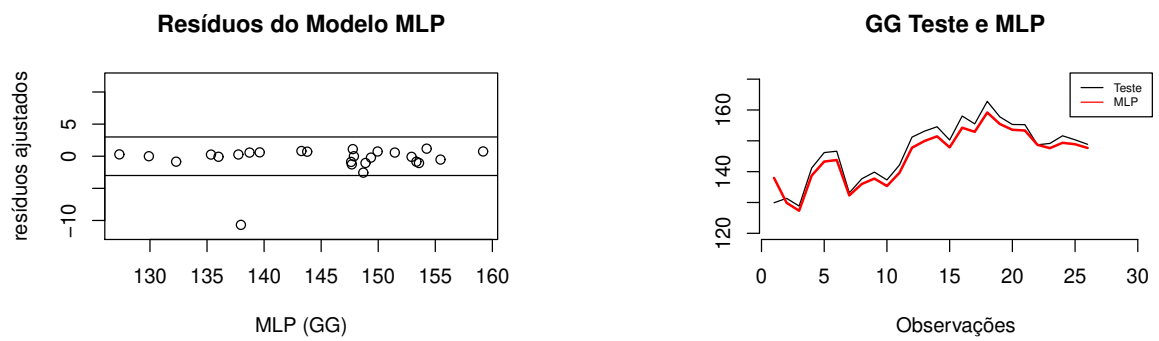


(a) Resíduos Estudentizados Cópula



(b) Previsões do modelo Cópula

Figura 50 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo Cópula

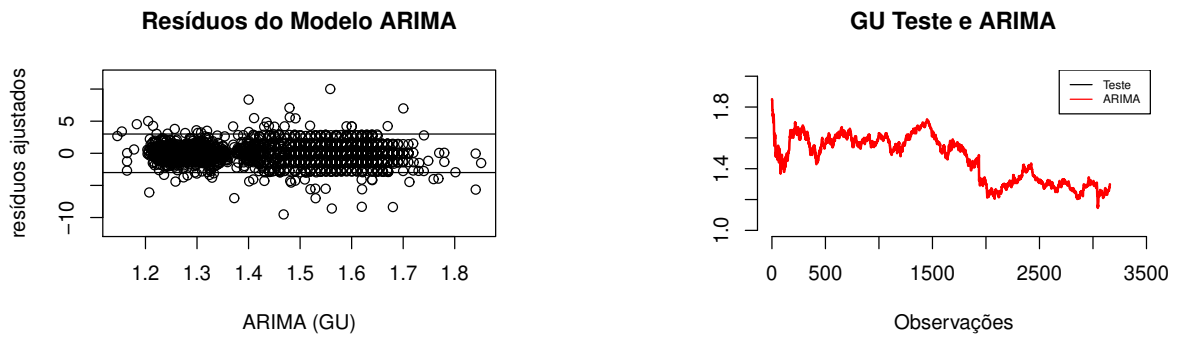


(a) Resíduos Estudenterizados MLP

(b) Previsões do modelo MLP

Figura 51 – Análise residual estudenterizada para a série GG a respeito das previsões do modelo MLP

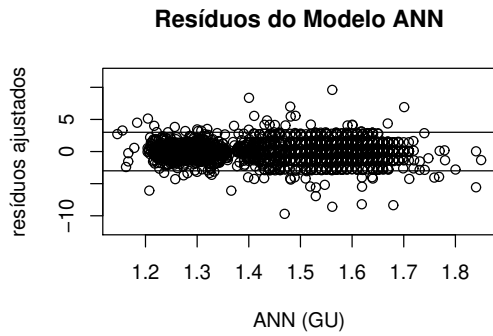
APÊNDICE D – Taxa de câmbio GBP para USD (GU)



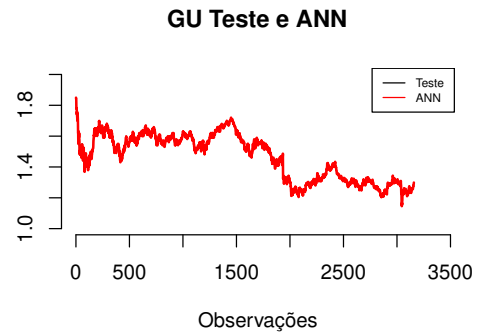
(a) Resíduos Estudentizados ARIMA

(b) Previsões do modelo ARIMA

Figura 52 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo ARIMA

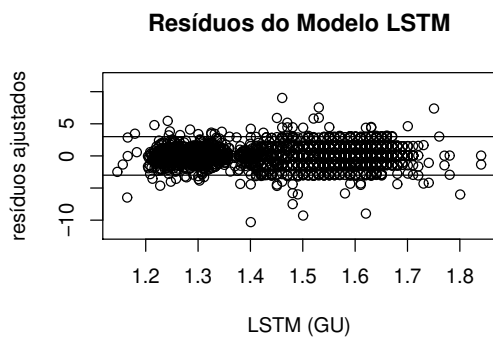


(a) Resíduos Estudenterizados ANN

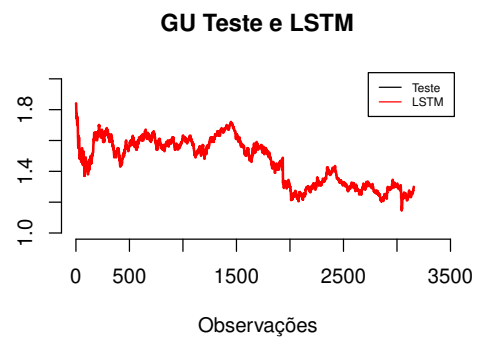


(b) Previsões do modelo ANN

Figura 53 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo ANN

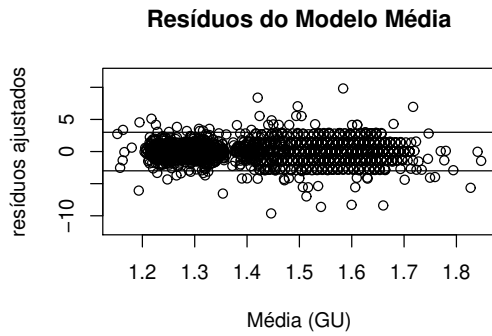


(a) Resíduos Estudenterizados LSTM

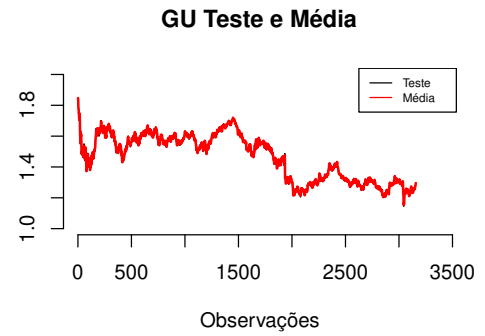


(b) Previsões do modelo LSTM

Figura 54 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo LSTM

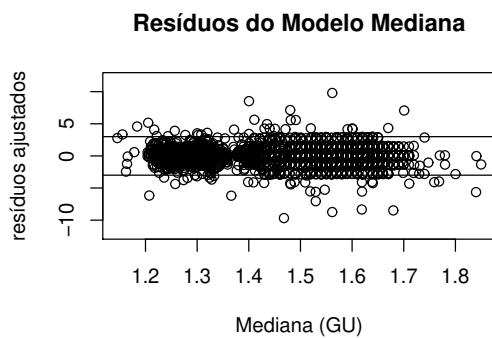


(a) Resíduos Estudenterizados Média

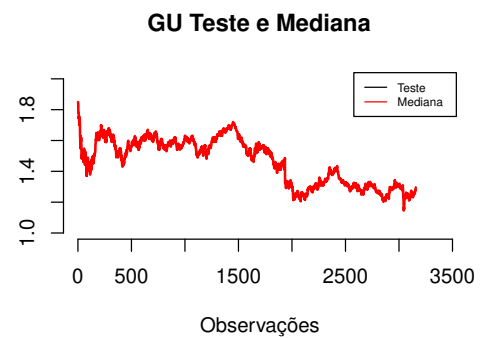


(b) Previsões do modelo Média

Figura 55 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo Média

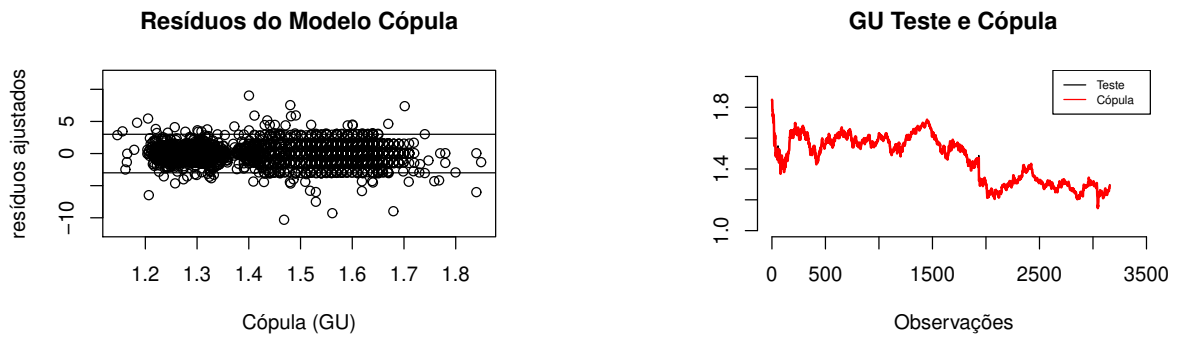


(a) Resíduos Estudenterizados Mediana



(b) Previsões do modelo Mediana

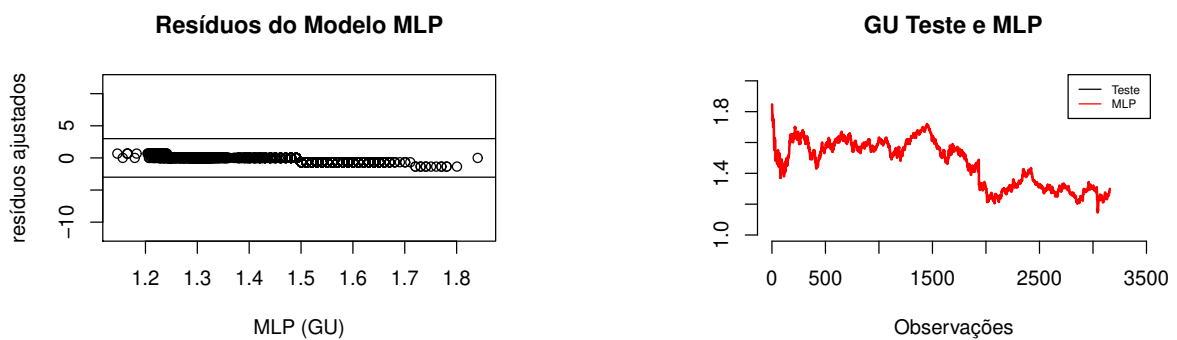
Figura 56 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo Mediana



(a) Resíduos Estudentizados Cópula

(b) Previsões do modelo Cópula

Figura 57 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo Cópula

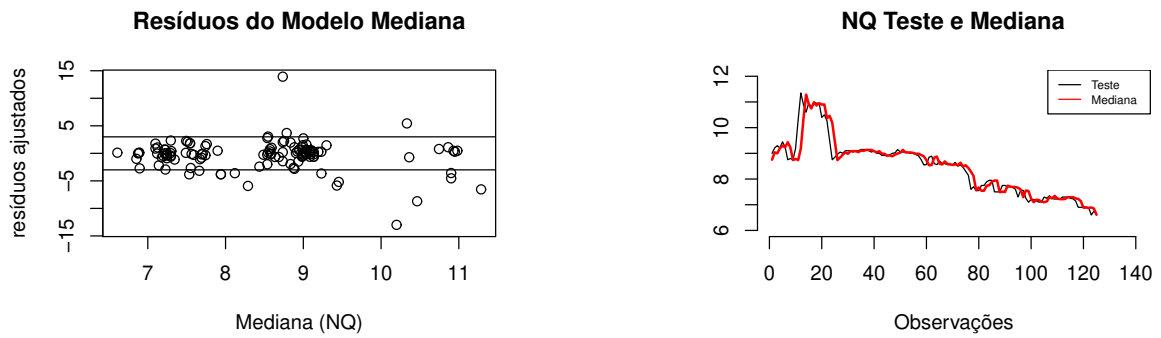


(a) Resíduos Estudentizados MLP

(b) Previsões do modelo MLP

Figura 58 – Análise residual estudenterizada para a série GU a respeito das previsões do modelo MLP

APÊNDICE E – Nasdaq (NQ)



(a) Resíduos Estudenterizados Mediana

(b) Previsões do modelo MLP

Figura 59 – Análise residual estudenterizada para a série NQ a respeito das previsões do modelo MLP