UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Henri Makika

Previsão de séries temporais com aprendizagem profunda: *uma aplicação para taxa de câmbio*

Time series forecast with deep learning: an application for exchange rate

Campinas

2022

UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Henri Makika

Previsão de séries temporais com aprendizagem profunda: *uma aplicação para taxa de câmbio*

Time series forecast with deep learning: an application for exchange rate

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Telecomunicações e Telemática.

Orientador: Prof. Dr. João Marcos Travassos Romano

Coorientadora: Profa. Dra. Rosângela Ballini

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Henri Makika, e orientada pelo Prof. Dr. João Marcos Travassos Romano

Campinas

2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Makika, Henri, 1986-
M289p Previsão de séries temporais com aprendizagem profunda : uma aplicação para taxa de câmbio / Henri Makika. – Campinas, SP : [s.n.], 2022.

Orientador: João Marcos Travassos Romano.
Coorientador: Rosângela Ballini.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes neurais recorrentes. 2. Processamento de sinais. 3. Teoria da estimativa. 4. Aprendizado profundo. 5. Memória de longo e curto prazo. I. Romano, João Marcos Travassos, 1960-. II. Ballini, Rosângela, 1969-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações Complementares

Título em outro idioma: Time series forecast with deep learning : an application for exchange rate
Palavras-chave em inglês:
Recurrent neural networks
Signal processing
Estimation theory
Deep learning
Long short-term memory
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
João Marcos Travassos Romano [Orientador]
Cristiano Torezzan
Romis Ribeiro de Faissol Attux
Data de defesa: 15-12-2022
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-6666-8765
- Currículo Lattes do autor: http://lattes.cnpq.br/3834616701955251

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Henri Makika RA: 211042
Data de defesa: 15 de dezembro de 2022
Título da Tese: Previsão de séries temporais com aprendizagem profunda: *uma aplicação para taxa de câmbio*

Prof. Dr. João Marcos Travassos Romano (Presidente)
Prof. Dr. Cristiano Torezzan (Membro externo)
Prof. Dr. Romis Ribeiro de Faissol Attux (Membro interno)

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

in memoriam de
Raymond Makika Kwasa & Rose Tundula Musesa.

Agradecimentos

Esta dissertação é o culminar de três anos completos de estudo. Várias pessoas foram de grande ajuda durante esses três anos, seja por seu apoio ou por sua disponibilidade. Gostaria de agradecer a algumas delas.

Antes de mais, gostaria aqui agradecer imensamente ao professor João Marcos Travassos Romano, que não poupou esforços para concordar em dirigir a redação desta dissertação; suas observações, comentários e paciências me permitiram melhorar o conteúdo. Agradeço também respeitosamente à professora Rosângela Ballini. Sua habilidade analítica rigorosa influenciou bastante o conteúdo deste trabalho. Que ela encontre aqui minha grande admiração e estima por sua pessoa.

Aos professores Romis Ribeiro de Faissol Attux e Cristiano Torezzan, que compuseram a banca de qualificação desta dissertação e cujos comentários foram ricos para a elaboração desta versão final.

Agradeço aos meus queridíssimos pais, Raymond Makika e Rose Tundula, por ensinarem-me o valor do trabalho e a quem devo tudo o que sou. Com eles aprendi as coisas mais valiosas que hoje tenho dentro de mim e que me empurram no caminho da vida acadêmica: o amor pela Ciência. Lá no Céu, inspirem-me muito mais. Também agradeço às minhas irmãs e meus irmãos, pelo amor verdadeiro.

Agradeço à Inès Kwamiso, grande amiga. Seus mil e um sorrisos me fazem esquecer todos os aspectos menos agradáveis da vida. Esse sucesso é nosso.

Agradeço também aos meus amigos de tempo todo, com os quais aprendi a ver o Estudo de um modo diferente, ainda que é no Estudo, naquele trabalho bem feito de dia a dia, que encontramos o Grande Amigo. Em especial, agradeço à Sergio Consolmagno. Acredito que esta dissertação nunca teria visto a luz do dia se não fosse daquele encontro e conversa. Agradeço também à Guilherme Giacopini e outros, pelo apoio financeiro que me permitiu focar nos meus estudos e ainda mais.

Agradeço a todos que ensinaram-me algo até o presente momento. Agradeço especialmente aos professores Levy e Romis, com os quais aprendi a compreender melhor o campo de aprendizado de máquina. Agradeço também à Marcelo Rainho, um dos mais brilhantes colegas com quem já tive a oportunidade de conversar e que tem imensa participação nesta dissertação. Nas ferramentas do *LaTeX* e nas conversas sobre a econometria, sua ajuda foi essencial. O trabalho coletivo, a parceria em *TeamViewer* e as longas horas de discussão foram - e tenho certeza que continuarão sendo - extremamente

enriquecedoras para minhas pesquisas e para minha formação acadêmica.

Também agradeço aos familiares, especialmente à Guyslaine, Platini e Victorine, amigos de família, por tudo o que passamos juntos. A Augusto, Gabriel e João-Paulo, colegas de mestrado do IE onde tudo tinha começado, por tudo o que passamos naquele momento que cursamos disciplinas juntos. Agradeço ainda aos amigos Carlos Eduardo, Franciscarlos, Miguel Argollo e tantos outros que, de maneira direta ou indireta me fizeram lembrar da existência de uma vida "extra-acadêmica". Agradeço ao amigo Bonifácio Andrada, Bony, muito obrigado pelo português que o senhor me ensinou. A Jonathan Biangala, Pathy Ahundu e Alpha Olondo, companheiros de teto e de risadas na reta final da redação desta dissertação.

Agradeço à todos os funcionários da Faculdade de Engenharia Elétrica e Engenharia de Computação por todo apoio institucional. O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

O estudo de previsão de séries temporais tem tido grande interesse na área econômica e financeira. Vários problemas da análise econômica são baseados na tarefa de previsão de séries temporais, ou seja, identificar as características do processo no ponto futuro. Nos últimos anos, modelos de aprendizado de máquinas desempenharam um papel importante na abordagem e solução de problemas complexos, entre outros, a previsão de séries temporais. Modelos de redes neurais recorrente (com realimentação) e não-recorrente (*feedforward* ou sem realimentação), têm-se mostrado uma alternativa valiosa em relação aos modelos lineares tradicionais. Algumas séries temporais exibem características ou fatos estilizados complexos, como tendência, não-linearidade, longa memória, em que modelos de redes neurais são capazes de incorporar na modelagem.

A presente dissertação de mestrado centra-se na possibilidade de prever as taxas de câmbio. É realizada uma investigação empírica sobre até que ponto os modelos de redes neurais podem melhorar a previsibilidade das taxas de câmbio em comparação com os modelos lineares tradicionais. Duas famílias de modelos são empregadas: modelos lineares (ARIMA e ARFIMA) e modelos não-lineares (MLP, LSTM e GRU). Mostramos suas fundamentações teóricas, as arquiteturas montadas como os algoritmos de treinamento e implementação computacional. Os resultados das previsões são comparados com base em métricas da magnitude do erro (MAE, MSE, RMSE) e da acurácia que nada mais é a probabilidade de detectar a direção correta do valor previsto.

Utilizamos as séries das taxas de câmbio diárias de Real/Dólar e Euro/Dólar de dezembro de 2003 a maio de 2021. As previsões são realizadas para um horizonte temporal de 1 e 7 passos à frente. Os resultados indicaram que os modelos lineares têm mostrados o desempenho relativamente inferior aos modelos não-lineares tanto para previsão 1 passo à frente como para previsão 7 passos à frente. Além disso, quando comparamos os modelos de redes neurais, a estrutura do modelo GRU fornece o melhor desempenho de previsão tanto para medida da magnitude como para acurácia.

Palavras-chaves: Redes neurais recorrentes; Processamento de sinais; Teoria da estimativa; Aprendizado profundo; Memória de longo e curto prazo; Taxa de Câmbio.

Abstract

The study of time series forecasting has been a great interest in the economic and financial area. Several problems of economic analysis are based on the task of time series, forecasting that is, to identify the characteristics of the process in the future point. In recent years, machine learning models have played an important role in approaching and solving complex problems, among others, time series forecasting. Recurrent and non-recurrent neural network models have proved to be a valuable alternative to traditional linear models. Some time series exhibit complex stylized features or facts, such as trend, nonlinearity, long memory, which recurrent neural networks are able to incorporate into modeling.

This master's thesis focuses on the possibility of predicting exchange rates. An empirical investigation is carried out on the extent to which neural network models can improve the predictability of exchange rates compared to traditional linear models. Two families of models are used: linear models (ARIMA and ARFIMA) and non-linear models (MLP, LSTM and GRU). We show its theoretical foundations, the architectures assembled as the training algorithms and computational implementation. The results of the predictions are compared based on metrics of the magnitude of the error (MAE, MSE, RMSE) and the accuracy, which is nothing more than the probability of detecting the correct direction of the predicted value.

We use the Real/Dollar and Euro/Dollar daily exchange rate series, from December 2003 to May 2021. Forecasts are performed for a time horizon of 1 and 7 steps ahead. The results indicated that linear models have shown relatively inferior performance to non-linear models for both 1-step-ahead and 7-step-ahead forecasts. In addition, among the neural networks structures, the stucture of the GRU model provides the best prediction performance for both magnitude and accuracy measurements.

Keywords: Recurrent neural networks; Signal processing; Estimation theory; Deep learning; Long short-term memory; Exchange rate.

Lista de ilustrações

Lista de tabelas

Lista de Acrônimos e Abreviações

AdaGrad - *Adaptive Gradient Algorithm*

ADAM - *Adaptive Moment Estimation*

AIC - *Akaike Information Criterion*

ANNs - *Artificial Neural Networks*

ARCH - *Autoregressive Conditional Heteroskedasticity*

ARCH-LM - *Autoregressive Conditional Heteroskedasticity - Lagrange Multiplier*

ARFIMA - *Auto Regressive Fractionally Integrated Moving Average*

ARIMA - *Autoregressive Integrated Moving Average*

ARMA - *Autoregressive Moving Average*

BIC - *Bayesian Information Criterion*

BN - *Batch Normalization*

BPTT - *BackPropagation-Through-Time*

CNN - *Convolutional Neural Network*

CV - *Cross-Validation*

EDA - *Exploratory Data Analysis*

ELU - *Exponential Linear Unit*

FAC - Função de Autocorrelação

FACP - Função de Autocorrelação Parcial

FNNs - *Feedforward Neural Networks*

GRU - *Gated Recurrent Unit*

k-fold CV - *k-fold Cross-Validation*

Leaky ReLU - *Leaky Rectified Linear Unit*

LSTM - *Long Short-Term Memory*

MAE - *Mean Absolute Error*

ML - *Machine Learning*

MLP - *Multilayer Perceptron*

MQO - Mínimos Quadrados Ordinários

MSE - *Mean Squared Error*

NAG - *Nesterov Accelerated Gradient*

PReLU - *Parametric Rectified Linear Unit*

R$/US$ - Real Brasil

ReLU - *Rectified Linear Unit*

RMSE - *Root Mean Squered Error*

RMSProp - *Root Mean Square Propagation*

RNNs - *Recurrent Neural Networks*

SELU - *Scaled Exponential Linear Unit*

SVM - *Support Vector Machine*

Tanh - Tangente hiperbólica

US$ - Dólar Norte Americano

Sumário

1 Introdução

A previsão de variáveis macroeconômicas e financeiras, como taxas de câmbio, desempenha um papel importante nas decisões de política econômica e na avaliação do estado futuro da economia. A taxa de câmbio, por definição, reflete o preço de uma moeda em relação à outra. Também pode ser definida como o número de unidades de moeda nacional necessário para comprar uma unidade de moeda estrangeira.

Como um dos principais preços da economia, a literatura aponta que a taxa de câmbio sempre foi reconhecida dentro da teoria econômica como objeto de estudo desde o princípio da ciência econômica (ABEL *et al.*, 2013). Como observamos também na atual conjuntura econômica mundial, o câmbio ganha importância ainda mais relevante no que diz respeito ao grau de integração econômica entre nações. A crise cambial de um país pode afetar outros países de forma semelhante dada a interconexão entre os países (SOON; BAHARUMSHAH, 2021).

A taxa de câmbio é, geralmente, definida no mercado de câmbio, em que esta última é o mercado no qual todos os participantes vendem ou compram divisas (KRUGMAN; OBSTFELD, 2009). Os participantes no mercado de câmbio são: bancos centrais, bancos comerciais, empresas participantes do comércio internacional, instituições financeiras não bancárias, investidores, etc. Existe mercado de câmbio à vista e mercado à termo. No mercado à vista, as divisas são trocadas no momento, sendo que o preço já está fixado. O preço no mercado a termo é fixado para entrega futura.

No mercado a termo pode haver riscos associados a volatilidade na taxa de câmbio. Nesse mercado o objetivo principal dos especuladores é obter lucro com operação cambial mediante determinado risco. A volatilidade da taxa de câmbio pode ou não ser observável e expressa de maneira simples o grau de incerteza que o investidor possui (KRUGMAN; OBSTFELD, 2009).

Além disso, vários outros efeitos negativos podem afetar a economia. No caso do Brasil, uma desvalorização da moeda doméstica (R$) em relação ao dólar americano (US$), por exemplo, pode aumentar o risco soberano, o qual reflete a capacidade e a vontade de um país de cumprir suas obrigações econômica e financeira. Em contraste, uma depreciação da taxa de câmbio efetiva acaba tendo impacto significativo sobre o risco soberano para um país com grande exposição líquida em moeda estrangeira do setor privado. Nesse caso, uma desvalorização da taxa de câmbio nominal induz um aumento do risco soberano[1]. No mercado internacional, por outro lado, uma crise financeira induzirá

[1] Os termos valorização e desvalorização são usados quando um dado país opera no regime de taxas

a um aumento do risco soberano (KRUGMAN; OBSTFELD, 2009).

Podemos ter ainda dois tipos de regime de câmbio: o fixo e o flutuante. No Brasil é adotado o regime de câmbio flutuante desde 1999. Nesse regime a taxa de câmbio é determinada por fatores internos e externos, tais como a inflação, a taxa de juros, o saldo da conta de capital, o papel dos especuladores, a dívida do país, a estabilidade política e desempenho econômico, a força relativa de outras moedas fortes, os eventos macroeconômicos e geopolíticos (PATEL *et al.*, 2014).

Como os preços da moeda dependem de fatores internos e externos, a natureza estocástica e intermitente da taxa de câmbio faz com que sua previsão seja uma tarefa desafiadora. Dessa forma, o objetivo principal desta dissertação é investigar o desempenho de modelos empregados para previsão da taxa de câmbio. Duas famílias de modelos são empregadas: (i) modelos paramétricos, a saber, *Auto Regressive Integrated Moving Average* (ARIMA) e *Auto Regressive Fractionally Integrated Moving Average* (ARFIMA ou FARIMA); e (ii) modelos não-paramétricos, *Multilayer Perceptron* (MLP), *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU). O desempenho dessas abordagens é comparado por meio de medidas da magnitude do erro (MAE, MSE, RMSE), e do grau de precisão da direção (acurácia). Um melhor desempenho de previsão para taxa de câmbio pode levar também a melhores decisões econômicas e financeiras.

Em vários estudos, modelos de redes neurais e técnicas tradicionais de séries temporais vêm sendo comparados (COELHO *et al.*, 2008), (HENRIQUE *et al.*, 2019).

As comparações, em termos de desempenho, das redes neurais com abordagens tradicionais são ainda contraditórias. Alguns trabalhos concluem que redes neurais têm desempenho superior às técnicas tradicionais, enquanto que outras pesquisas concluem o contrário. Essas contradições podem ser explicadas por diferentes fatores, tais como: estrutura da rede neural como o número de camadas intermediárias, número de neurônios por cada camada e realimentação, ou diferentes técnicas de otimização (escolha dos hiperparâmetros, técnicas adaptativas, etc.), além do tipo de série temporal a ser prevista (pré-processamento, estacionariedade ou não-estacionariedade, e outros fatos estilizados).

(TANG *et al.*, 1991) compararam o desempenho dos modelos MLP e Box-Jenkins, usando séries temporais de tráfego de passageiros de vôo internacional, vendas de carros domésticos e vendas de carros importados nos Estados Unidos. Os resultados das três séries analisadas mostraram que o desempenho do modelo de Box-Jenkins superou ao modelo MLP para previsão a curto-prazo. Para previsão do longo-prazo, a rede MLP

de câmbio fixo e os termos apreciação e depreciação são utilizados quando o país está no regime de câmbio flutuante (BLANCHARD, 2019). Há apreciação de uma moeda nacional quando há aumento de seu preço em relação à moeda estrangeira, o contrario é a depreciação, isto é, quando ocorre uma diminuição do preço da moeda nacional em relação à moeda estrangeira. Assim, uma apreciação de uma moeda significa a diminuição na taxa de câmbio e a depreciação representa a relação inversa.

apresentou melhor desempenho.

(ZHANG, 2003) aplicou redes neurais RNAs para previsão de manchas solares, caça de linces no Canadá, e taxa de câmbio GBP/US e comparou com modelos ARIMA. O autor concluiu que as RNAs têm melhor desempenho ao tornar as séries estacionárias, ou seja, após a remoção dos componentes de tendência e sazonalidade.

(COELHO *et al.*, 2008) aplicaram redes neurais MLP, RBF (*Radial Basis Function Neural Networks*) e sistema nebuloso Takagi-Sugeno para previsão um passo-à-frente da série de retorno da taxa de câmbio R$/US$ com frequência de 15 minutos, 60 minutos, 120 minutos, diária e semanal, e compararam com modelos ARIMA-GARCH. Os resultados indicaram que o desempenho dos modelos está diretamente relacionado à frequência observada da série. De modo geral, os modelos de redes neurais obtiveram melhor desempenho que técnicas tradicionais.

(SHEN *et al.*, 2015) aplicaram modelos DBN (*Deep Belief Networks*) e Máquinas de Boltzmann restrita para previsão de três séries de câmbio. Além disso, o método do gradiente conjugado foi aplicado para acelerar o aprendizado da rede DBN. A comparação com modelo ARMA mostrou que o modelo DBN pode tratar séries não-lineares, com bom desempenho de previsão.

(AZZOUNI; PUJOLLE, 2017) compararam os modelos MLP, LSTM e ARAR, ARMA e algoritmo Holt-Winters (HW), usando a série temporal de tráfego do INRIX Roadway Analytics para previsão de curto e longo-prazo. As estruturas de redes MLP e LSTM foram de tamanhos relativamente pequenas, isto é, somente uma camada intermediária com 5 neurônios. Os resultados mostraram que, tanto no curto como no longo-prazo, o desempenho das redes MLP e LSTM foi superior aos modelos ARMA, ARAR e HW. Além disso, a rede LSTM superou ainda a rede MLP em termo de desempenho.

(ZHANG; KABUKA, 2018) aplicaram modelos RNN (*Recurrent Neural Network*), LSTM e GRU profundos para previsão de tráfego para gerenciamento de transporte inteligente para o Japão e compararam com os modelos ARIMA, SVM (*Support Vector Regression*) e *Random Forest Regression*. Os resultados mostraram que a rede GRU obteve bom desempenho preditivo com baixo erro de previsão.

(NI *et al.*, 2019) compararam redes RNN profunda, CNN profunda (Rede Neural Convolucional) e LSTM profunda, usando dados de séries de câmbio de diversos países. Explorando diferentes características espaço-temporais das séries, o modelo LSTM teve melhor desempenho que as redes RNN profunda e CNN profunda.

(QU; ZHAO, 2019) aplicaram redes neurais RNN e LSTM para previsões da taxa de câmbio €/US$. O índice do preço de câmbio e outras informações financeiras foram tomadas e consideradas como variáveis de entrada. Os resultados mostraram que a

rede LSTM alcançou menores erro quadrático médio (MSE) e erro absoluto médio (MAE) do que a rede RNN.

(JI *et al.*, 2019) compararam o modelo ARIMA-CNN-LSTM para previsão do preço futuro do carbono. O modelo ARIMA-CNN-LSTM emprega o modelo ARIMA e a estrutura das redes neurais profundas que combina as camadas CNN e LSTM para capturar recursos dos dados lineares e não-lineares. Na estrutura conjunta ARIMA-CNN-LSTM, o modelo ARIMA captura as relações lineares; a rede CNN captura a estrutura dos dados hierárquicos, enquanto a rede LSTM captura as dependências de longo prazo nos dados. Os resultados mostram que a estrutura conjunta ARIMA-CNN-LSTM alcançou desempenho superior quando comparado com as três estruturas separadas.

Existem várias características nas séries temporais macroeconômicas e financeiras, tais como: as curtas e longas dependências temporais e as não-estacionariedades. Uma série temporal com forte dependência reflete em um decaimento lento das funções de autocorrelação. Modelos ARIMA são considerados modelos de curta memória, porque o grau de diferenciação é um valor inteiro, assim, a previsão é realizada a partir de um número de atrasos limitados e não lida com problema de longa dependência. Os modelos ARFIMA, com grau de diferenciação fracionária, é capaz de agregar tanto as baixas quanto as altas dependências temporais. Assim, é chamado de modelo de longa memória.

Por sua vez, séries temporais financeiras que apresentam tendência estocástica, a sua média e variância não são constantes ao longo do tempo, caracterizando assim séries não estacionárias. A literatura indica que modelos de redes neurais conseguem capturar as não-linearidades presentes nessas séries.

A rede MLP é um mapeamento estático da entrada até a saída, não tendo realimentação (chamada rede não-recorrente). As redes LSTM e GRU possuem uma estrutura dinâmica, com realimentação da informação passada, sendo estruturas recorrentes. Isso leva dizer que essas duas redes capturam as dependências de longo prazo nos dados.

Neste trabalho, o modelo ARIMA é utilizado para prever a série da taxa de câmbio. O modelo ARFIMA é escolhido para remover a dependência temporal e realizar previsão de longa memória. Por sua vez, o modelo MLP captura não-estacionariedade e quebra estrutural, assim, realiza previsão estática. Por fim, os modelos LSTM e GRU são treinados para filtrar seletivamente qualquer informação irrelevante, e mantendo o que é importante, assim, são credenciados como candidatos naturais para lidar com problema de memória longa.

Essa dissertação propõe atingir os seguintes objetivos: fazer uma descrição dos modelos ARIMA, ARFIMA e dos modelos MLP, LSTM e GRU, além de destacar as principais diferenças entre estes; aplicar esses modelos nas séries diárias das taxas de

câmbio R$/US$ e €/US$. Outro aspecto importante considerado neste trabalho é a análise dos modelos não somente avaliando o desempenho da previsão usando medidas baseadas na magnitude do erro de previsão (MAE, MSE, RMSE, R^2, etc.), mas também avaliar a probabilidade de detectar a direção correta do valor previsto ou o grau de precisão da direção. Destaca-se que na literatura, de um modo geral, a análise é feita comparando-se erros de previsão.

Salienta-se que, para um investidor, o grau de precisão da direção é mais relevante para lucratividade, sendo que se os lucros não são observáveis, a precisão da direção das previsões pode auxiliar como medida da avaliação do desempenho da previsão. A precisão da direção, no caso, está diretamente relacionada à capacidade de *timing* do mercado, o qual reflete a tomada de decisões de compra ou venda de ativos tendo em conta as expectativas de volatilidade do mercado. Acrescenta-se ainda que ao saber a melhor direção da previsão, o investidor pode adaptar sua decisão ao acontecimento futuro do mercado (LO, 2005). Dessa forma, a precisão da direção pode servir ainda como indicador viável para a hipótese de mercados adaptativos, no qual reflete o máximo de informação ditada pela combinação de condições ambientais e do eco-sistema ou ecologia (ambiente econômico, evolução tecnológica, características institucionais, guerras, bolhas, etc.) (LO, 2017). Consideramos essa medida para este trabalho como uma métrica da avaliação do desempenho.

Após essa introdução, esse trabalho está organizado da seguinte forma:

O Capítulo 2: Modelos de Previsão de Séries Temporais, apresentamos os principais modelos que dão suporte para os estudos desta dissertação.

O Capítulo 3: Aplicação para Previsão da Taxa de Câmbio, aplicamos os modelos apresentados no Capítulo 2 às séries das taxas de câmbio R$/US$ e €/US$. Inicialmente, apresentamos as bases de dados utilizadas neste trabalho. Em seguida, descrevemos as metodologias das duas famílias de modelos. Por fim, apresentamos os resultados obtidos e a comparação desses a partir de medidas de erros e acurácia.

Por fim, apresentamos as conclusões gerais desse trabalho.

Adicionalmente, a dissertação conta ainda com um apêndice, onde detalhamos os testes de raiz unitária utilizados em nossos estudos.

2 Modelos de Previsão de Séries Temporais

Neste capítulo, exploramos os principais métodos que dão suporte para os estudos desta dissertação. O capítulo está dividido em três seções. Na Seção 2.1 apresentamos, de forma sintética, dois modelos lineares amplamente estabelecidos na área de previsão de séries temporais: modelos ARIMA e ARFIMA, o qual é uma extensão do primeiro, com a incorporação de uma ordem de integração fracionária. Após a apresentação algébrica, vamos discutir, ainda, sobre o procedimento de estimativa desses modelos.

Na Seção 2.2, procuramos mitigar diferentes modelos de inteligência computacional aplicados em análise econômica e financeira desde seus princípios. Vamos apresentar as arquiteturas de redes neurais MLP, LSTM e GRU. Aqui, discutimos sobre o processo de otimização não-linear (treinamento ou aprendizagem) das três arquiteturas.

Por fim, na Seção 2.3 apresentamos as métricas que serão empregadas para comparar os desempenhos dos modelos.

2.1 Modelos Lineares

A previsão de uma série temporal consiste em encontrar a função $F(\cdot)$, de tal forma que $F(x(n)) = \hat{y}(n)$. Em outras palavras, emprega-se uma série de observações, $x(n) = (x(n-1), x(n-2), ..., x(n-K))^T$, que produza uma estimativa adequada de $\hat{y}(n)$ (ROMANO *et al.*, 2011). Em resumo, o objetivo dessa formulação consiste em ajustar os valores de parâmetros $\mathscr{W}$ que levem a melhor estimativa de entrada-saída, isto é, busca resolver o problema de otimização linear, tal que: $F(x(n)) = \hat{y}(n) = \sum_{i=1}^{k} \mathscr{W}_i x(n-i)$ (Figura (2.1)).

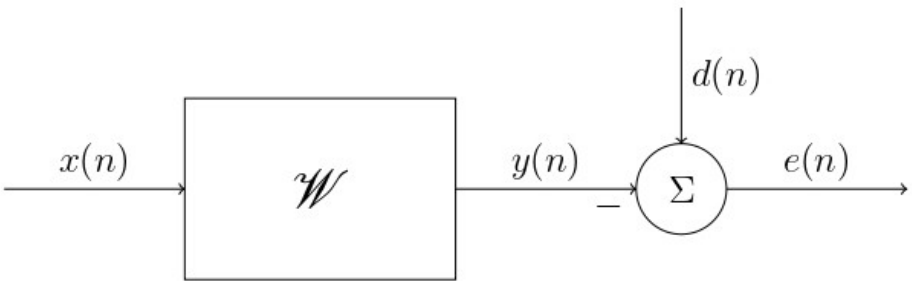


Figura 2.1 – Representação de um modelo linear.

O problema de otimização linear consiste em minimizar o erro $e(n)$ dado pela diferença entre a resposta esperada $d(n)$ e a resposta gerada pelo modelo $\hat{y}(n)$, ou seja,

$e(n) = d(n) - \hat{y}(n)$ (ROMANO *et al.*, 2011).

Usar as observações reais (série temporal) significa que existe uma dependência entre os valores, o que permite a realização da previsão dos valores futuros. De modo geral, uma série temporal x_t pode ser decomposta em quatro componentes, a saber:

• T_t um componente de tendência, que reflete a mudança de longo prazo no nível médio da série temporal. Esse componente pode ser causado, por exemplo, pelo crescimento demográfico, mudança gradual de hábitos de consumo, ou qualquer outro aspecto que afete a variável de interesse no longo prazo;

• C_t um componente cíclico, refletindo flutuações na certa periodicidade, que podem ser resultado de variações da economia como períodos de expansão ou recessão, ou fenômenos climáticos. Geralmente, são fenômenos que se repetem com periodicidade superior a um ano;

• S_t um componente sazonal, que ocorre a cada s período de tempo. Geralmente, tem duração inferior a um ano;

• ε_t um ruído branco, a parte não explicada, que espera-se ser puramente aleatória, com média zero e variância constante. Caso ε_t tenha uma distribuição normal é denominado *ruído branco gaussiano*.

Matematicamente, uma série temporal pode ser representada por um modelo aditivo:

$$x_t = T_t + C_t + S_t + \varepsilon_t \tag{2.1}$$

ou multiplicativo:

$$x_t = T_t \cdot C_t \cdot S_t \cdot \varepsilon_t \tag{2.2}$$

Os principais objetivos em se estudar séries temporais se resumem em analisar os componentes apresentados nas Equações (2.1) e (2.2), além de identificar os *outliers*, existência ou não de alterações estruturais (mudanças no padrão da tendência) e predizer valores futuros com base em valores passados (BUENO, 2012).

Neste trabalho, o objetivo principal é investigar o desempenho de modelos empregados na previsão da taxa de câmbio. Em termo ilustrativo, a Figura 2.2 apresenta a série temporal da taxa de câmbio R$/US$ entre 2004 a 2021. Visualmente, podemos inferir que essa série temporal não apresenta componentes sazonais e/ou cíclicos, resultando apenas nos componentes de tendência e aleatório.

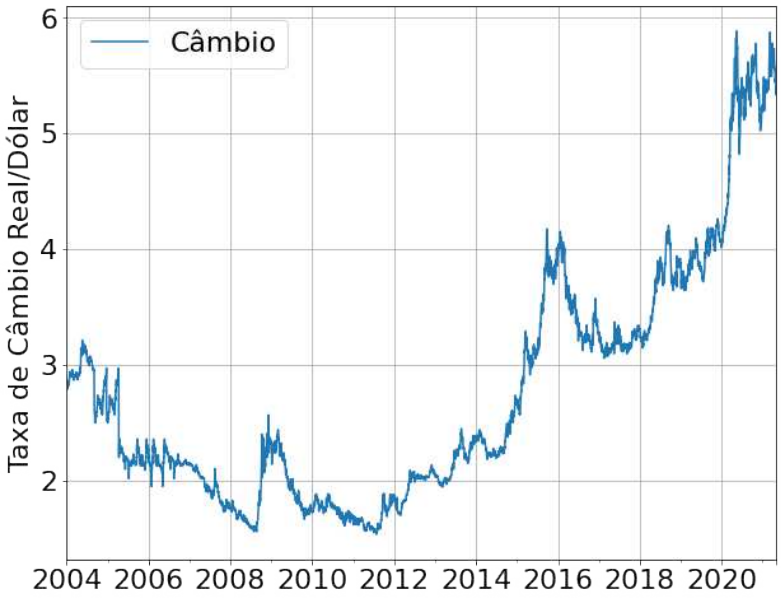


Figura 2.2 – Série Temporal da Taxa de Câmbio R$/US$ (2004-2021).

Uma série temporal pode ter uma tendência determinística ou estocástica. Uma série com tendência determinística, ou seja, a mudança de longo prazo no nível médio da série temporal pode ser escrita da seguinte forma (MORETTIN; TOLOI, 2004):

$$x_t = \alpha_0 + \alpha_1 \mathbf{t} + \varepsilon_t \tag{2.3}$$

em que α_0 e α_1 são constantes a serem estimadas, $\mathbf{t}$ o valor amostral no tempo e ε_t representa o componente aleatório. Para tornar esta série estacionária é necessário remover o componente de tendência. O componente de tendência como apresentado na Equação (2.3) pode ser estimado usando método dos mínimos quadrados ordinário (MORETTIN; TOLOI, 2004). Após estimação, os resíduos podem ser obtidos da seguinte forma:

$$\hat{\varepsilon}_t = x_t - \hat{x}_t \tag{2.4}$$

em que $\hat{x}_t = \hat{\alpha}_0 + \hat{\alpha}_1 \mathbf{t}$.

Caso a série temporal seja representada por:

$$x_t = \delta_0 + x_{t-1} + \varepsilon_t \tag{2.5}$$

em que δ_0 é uma constante denominada *drift* e ε_t é um componente aleatório independente e identicamente distribuído, ou seja, $\varepsilon_t \sim i.i.d(0, \sigma_\varepsilon^2)$.

Dado o valor inicial x_0, a solução de (2.5) é dada por:

$$\begin{aligned} x_1 &= \delta_0 + x_0 + \varepsilon_1 \\ x_2 &= \delta_0 + x_1 + \varepsilon_2 = \delta_0 + \delta_0 + x_0 + \varepsilon_1 + \varepsilon_2 \\ &\vdots \\ x_t &= x_0 + \delta_0 \mathbf{t} + \sum_{i=1}^{t} \varepsilon_i \end{aligned} \tag{2.6}$$

Diz-se que a série tem tendência estocástica. De (2.5):

$$x_t - x_{t-1} = \Delta x_t = \delta_0 + \varepsilon_0 \tag{2.7}$$

que resulta em uma série com média δ_0 e variância σ_ε^2. Dessa forma, diz que neste caso, a Δx_t é estacionário.

Em resumo, a análise e a previsão de séries temporais precisam que os dados sejam estacionários, isto é, as principais propriedades (função densidade de probabilidade conjunta) não se alteram ao longo do tempo. A partir da identificação de média, variância, coeficientes de auto-covariância e autocorrelação e/ou construção dos gráficos das funções de auto-covariância e autocorrelação pode-se verificar as características da série temporal em relação à condição de estacionariedade. Feito isso, uma vez que a série é estacionária pode-se realizar previsão futura. Maiores informações sobre as propriedades assintóticas de uma série temporal podem ser obtidas em (BUENO, 2012) e (MORETTIN; TOLOI, 2004).

Prever uma série temporal consiste em encontrar um modelo adequado que descreva o comportamento desta série (ROMANO *et al.*, 2011). Na prática, algumas séries temporais, tal como taxas de câmbio, têm memória longa, isto é, existe uma forte dependência entre os valores da série temporal. Essas séries são, usualmente, caracterizadas por um decaimento lento das funções de autocorrelações.

A modelagem desse tipo de séries temporais é complexa na escolha de modelos lineares. Assim, (GRANGER; JOYEUX, 1980) desenvolveram um modelo conhecido como ARFIMA (*Auto Regressive Fractionally Integrated Moving Average*), onde uma ordem de integração fracionária é adicionado no modelo ARIMA.

A seguir, fornecemos uma breve apresentação dos modelos ARIMA e ARFIMA, assim com os principais métodos de estimação.

2.1.1 Modelo ARIMA

Na modelagem paramétrica de séries temporais, a metodologia proposta por (BOX; JENKINS, 1970) é bastante usada na literatura. O uso desses métodos permite estimar modelos auto-regressivos (AR), modelos médias móveis (MA), modelos auto-regressivos médias móveis (ARMA) e modelos integrados auto-regressivos médias móveis (ARIMA). Esses modelos são classificados como estacionários na média, e não na variância (COELHO *et al.*, 2008), (BROCKWELL; DAVIS, 2016).

A metodologia de Box & Jenkins consiste de quatro etapas realizadas em um ciclo iterativo, e baseadas nos dados apresentados (MORETTIN; TOLOI, 2004). As etapas são: (i) identificação do modelo, com base na análise de funções de autocorrelação (FAC), autocorrelação parcial (FACP) e/ou critérios de informação (AIC, BIC, etc.); (ii) estimação do modelo; esta etapa consiste em estimação dos parâmetros do modelo identificado; (iii) verificação do modelo estimado, a partir de análise dos resíduos; esta etapa consiste na verificação se o modelo identificado e estimado é adequado para realizar previsões. Se não é o caso, o processo retorna à primeira etapa, ou seja, para identificação, caso contrário, seguimos para a etapa (iv) de previsão.

Na prática, a primeira etapa é baseada no comportamento das FAC e FACP empíricas, chamada ainda análise da significância das autocorrelações (BUENO, 2012). É nessa etapa que se verifica a propriedade de estacionariedade da série analisada. Uma série temporal estocástica é estacionária em diferença, sendo a ordem d de integração o número de raízes unitárias obtido nessa série (BROCKWELL; DAVIS, 2016). Usualmente, é comum ajustar diferentes modelos e utilizar os critérios de informação para selecionar o modelo adequado. O modelo mais parcimonioso é escolhido.

2.1.1.1 Modelo AR

Um modelo auto-regressivo de ordem p, AR(p) estacionário, pode ser definido como sendo um modelo em que os valores correntes do processo x_t são conhecidos como sendo uma combinação linear dos p valores anteriores $x_{t-1}, x_{t-2}, x_{t-3}, ..., x_{t-p}$, mais um ruído branco de média zero e variância constante. O modelo AR(p), representado na Equação (2.8), em que $\alpha_1, ..., \alpha_p$ representam os coeficientes do modelo linear a serem ajustados e p a ordem do modelo. Como se trata de um modelo linear, os coeficientes podem ser ajustados a partir do método de minimização dos erros quadráticos médios (MQO).

$$x_t = c + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \alpha_3 x_{t-3} + ... + \alpha_p x_{t-p} + \varepsilon_t \tag{2.8}$$

Uma versão mais simples de um modelo auto-regressivo AR(p) é aquela em

que o processo x_t depende unicamente do seu primeiro atraso, ou seja, x_{t-1} e de ε_t. Tal modelo é definido como um modelo auto-regressivo de ordem 1, ou seja, AR(1):

$$x_t = c + \alpha_1 x_{t-1} + \varepsilon_t \tag{2.9}$$

A Equação (2.9) reflete uma equação em diferença não homogênea de primeira ordem, em que ε_t é um ruído branco $i.i.d(0, \sigma^2)$, c uma constante.

A identificação de um modelo auto-regressivo AR(p) é realizada por meio da análise das funções de autocorrelação (FAC) - a função de autocorrelação decai com o aumento das defasagens - e autocorrelação parcial (FACP), que define a defasagem, ou seja, o valor de p do modelo AR (BALLINI, 2000).

2.1.1.2 Modelo MA

Um modelo de médias móveis MA(q), onde q representa a defasagem mais elevada, é um processo em que existe uma dependência entre o valor x_t e os componentes aleatórios, ou seja (BROCKWELL; DAVIS, 2016):

$$x_t = \mu + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \beta_2 \varepsilon_{t-2} - \beta_3 \varepsilon_{t-3} - ... - \beta_q \varepsilon_{t-q} \tag{2.10}$$

Os coeficientes $\beta_1, ..., \beta_q$ podem ser estimados por meio do método MQO (BROCKWELL; DAVIS, 2016), ou método de máxima verossimilhança (BOX; JENKINS, 1970).

Usualmente, um processo MA(q) pode ser identificado a partir de suas primeiras q autocorrelações significativas e um padrão de decaimento lento ou alternado de suas autocorrelações parciais - com a condição de invertibilidade estabelecido em que $|\beta| < 1$. A FACP do modelo de médias móveis MA(q) decresce em medida que as defasagens aumentam.

2.1.1.3 Modelo ARMA

A junção dos modelos AR(p) e MA(q) gera um processo auto-regressivo e de médias móveis chamado modelo ARMA(p,q). Para uma série temporal estacionária x_t, o modelo misto é definido por:

$$x_t = c + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + ... + \alpha_p x_{t-p} - \beta_1 \varepsilon_{t-1} - \beta_2 \varepsilon_{t-2} - ... - \beta_q \varepsilon_{t-q} + \varepsilon_t \tag{2.11}$$

Para ajustar os coeficientes do modelo ARMA(p,q), os métodos baseados em maximização de Verossimilhança e de mínimos quadrados são utilizados (BUENO, 2012).

O Método dos Mínimos Quadrados consiste em minimizar a soma de quadrados dos resíduos. O Método de Máxima Verossimilhança maximiza a probabilidade de ocorrência da variável dependente. Vale ressaltar que a condição de estacionariedade depende somente dos parâmetros do modelo AR e não dos componentes relacionados à parte MA.

A Tabela 2.1 resume a etapa de identificação dos modelos apresentados até aqui, ou seja, modelos AR(p), MA(q) e ARMA(p,q).

Tabela 2.1 – Identificação de modelos AR(p), MA(q) e ARMA(p,q).

Modelo	Função FAC	Função FACP
AR(p)	declinante	truncada em k = p
MA(q)	truncada em k = q	declinante
ARMA(p,q)	declinante a partir de q	declinante a partir de p

Como descritos acima, usualmente, para identificar a ordem dos modelos a partir das funções FAC e FACP precisamos de uma análise da significância estatística das autocorrelações. Uma outra maneira de determinarmos a ordem de um modelo ARMA(p,q) é usar os critérios de informação AIC e/ou BIC. Neste caso, a escolha das ordens p e q são determinadas a partir do menor valor do AIC e/ou BIC. Após a identificação de p e q, estimamos os parâmetros do modelo usando o método de máxima verossimilhança. Em seguida, é necessário verificar a adequação do modelo identificado. Para tanto, devemos analisar a significância dos parâmetros estimados e analisar a autocorrelação dos resíduos. Ainda, para análise dos resíduos, vários testes estatísticas são empregados tais como: teste de normalidade dos resíduos, e teste ARCH-LM-heterocedasticidade condicional. Após a verificação dos resíduos, é realizada a etapa de previsão (MORETTIN; TOLOI, 2004).

2.1.1.4 Modelo ARIMA

Os processos estocásticos ARMA(p,q) são estacionários, isso quer dizer que as propriedades assintóticas não se alteram ao longo do tempo. Entretanto, as séries temporais econômicas e financeiras não exibem necessariamente estacionariedade (BROCKWELL; DAVIS, 2016), (SIMS, 1980). Em geral, as séries econômicas e financeiras apresentam tendência estocástica, o que resulta na hipótese de que sua média aumenta ou diminua em certa medida ao longo do tempo (GHYSELS; MARCELLINO, 2018). Para dar conta desse fato, um termo de integração (integrado) é incorporado ao modelo ARMA(p,q), gerando assim o modelo ARIMA(p,d,q). Basicamente, é uma transformação aplicada à série para estabilizar sua média. No modelo ARIMA, o termo de integração d indica a inclusão da diferença entre os valores correntes e os valores passados. Esta diferenciação pode ser realizada mais de uma vez.

O modelo ARIMA(p,d,q) para uma série temporal x_t pode ser representado por:

$$\alpha(\Psi)(1-\Psi)^d x_t = \beta(\Psi)\epsilon_t \tag{2.12}$$

em que α e β são os coeficientes associados às partes AR e MA, respectivamente, $\Delta x_t = x_t - x_{t-1} = (1-\Psi)x_t$ o operador de diferença; d indica a ordem de integração; $\Psi^j x_t = x_{t-j}$ representa o operador de atraso. Note que, é um modelo ARMA(p,q) que sofre uma transformação do tipo $\alpha(\Psi)(1-\Psi)^d x_t$.

Um modelo ARIMA(p,0,q) é simplesmente um modelo ARMA(p,q). Da mesma forma, um modelo ARIMA(p,0,0) indica um processo AR(p). Por sua vez, um modelo ARIMA(0,0,q) representa um processo MA(q). Vale ressaltar que o termo de integração d é um inteiro positivo: se $d = 1$, a série é estacionária com tendência linear, e se $d = 2$, a série torna estacionária no caso que a tendência for quadrática (BROCKWELL; DAVIS, 2016).

Como nos modelos vistos acima, para a identificação dos modelos ARIMA(p,d,q) temos as seguintes etapas: (i) a determinação das ordens p, d e q; (ii) a estimação dos parâmetros α_i, $i = 1, ..., p$, e β_j, $j = 1, ..., q$; (iv) a estimação da variância do ruído. A estimação de parâmetros dos modelos ARIMA(p,d,q) pode ser obtida por meio do método máxima verossimilhança (MORETTIN; TOLOI, 2004). A função verossimilhança, $\prod_{i=1}^{n} f(\mathbf{x}_i, \boldsymbol{\Theta})$ (função densidade de probabilidade conjunta, com $\boldsymbol{\Theta} = \alpha, \beta$), consiste em encontrar $\boldsymbol{\Theta}$ que fornece a máxima probabilidade de se obter a amostra x observada, ou seja, $f(x_1, \theta)f(x_2, \theta)\cdots f(x_n, \theta)$.

Os modelos AR(p), MA(q), ARMA(p,q) e ARIMA(p,d,q) possuem diversas variações, tais como SARIMA(p,d,q)×(P,D,Q) que combinam duas partes: sazonal e não-sazonal (MORETTIN; TOLOI, 2004), modelos ARCH(p,q) (modelos *autorregressivos condicionais heterocedásticos*) ou GARCH(p,q), que incorporam as volatilidades estocásticas e dependências temporais (BROCKWELL; DAVIS, 2016) e modelos ARFIMA, que assume que a ordem de integração d pode ser um valor fracionário (HOSKING, 1981), (GRANGER; JOYEUX, 1980).

2.1.2 Modelo ARFIMA

O modelo linear ARIMA(p,d,q) é definido como um modelo de memória curta, no qual a previsão é realizada a partir de um número de atrasos limitados. Além disso, o parâmetro d de integração desse modelo pode assumir um valor inteiro que estabelece o nível de diferenciações necessárias para tornar uma série temporal estacionária. Por sua vez, o modelo ARFIMA(p,d,q) é classificado na família de modelos de memória

longa, pois é responsável por capturar e modelar processos de longa dependência serial, onde $d \geqslant 0$ é um valor não-inteiro (GRANGER; JOYEUX, 1980), (HOSKING, 1981). O modelo ARFIMA(p,d,q) é capaz de agregar tanto as baixas quanto as altas dependências temporais. Além de assumir valores inteiros, d pode representar também os graus de diferenciação fracionários, isto é, assumir valores não-inteiros. É muito comum usar os critérios de informação AIC e BIC para a escolha do modelo adequado. Por isso, o modelo ARFIMA(p,d,q) é chamado de modelo de memória longa. As variáveis econômicas, como por exemplo, taxas de câmbio, costumam-se ter reputação de apresentar séries temporais com longa dependência como podemos verificar na Figura 2.3, da função de autocorrelação.

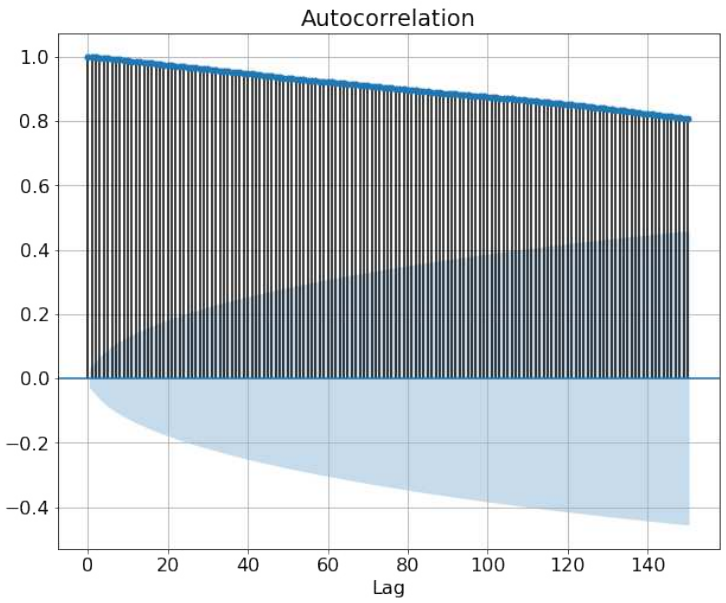


Figura 2.3 – Ilustração da autocorrelação simples de um processo de memória longa.

A Figura 2.3 mostra o comportamento da função de autocorrelação com decaimento lento. O modelo ARFIMA(p,d,q) é capaz de lidar com esse tipo de problema, de tal forma que a ordem do parâmetro d que corresponde às diferenciações deve explicar a estrutura de correlação de ordens mais altas para tornar a série estacionária. Assim, o modelo é capaz de descrever as dinâmicas de um modelo fracionário de curta e longa memória.

Existe uma parte auto-regressiva do modelo e outra parte média móvel. Isto é, se a ordem de integração é igual a um, o modelo é ARIMA(p,1,q). Caso a ordem de integração for um valor fracionário, por exemplo, $d = \frac{1}{2}$, é ARFIMA(p,0.5,q).

Duas características que fazem a escolha da família de modelos de memória longa (MORETTIN; TOLOI, 2004). A primeira diz respeito ao efeito da ordem de integração d. As observações anteriores decai de forma hiperbólica quando os atrasos aumentam. Por sua vez, os coeficientes associados às partes autoregressiva (α) e média móvel (β)

decrescem de forma exponencial. Segundo os autores supracitados, a ordem de integração d pode ser escolhida com intuito de explicar a estrutura de correlação das ordens seriais e os coeficientes α e β devem explicar a estrutura de correlação de baixas ordens. A segunda característica diz respeito à dependência prolongada (persistência no tempo) das autocorrelações seriais. A persistência prolongada entre observações pode fazer com que a função densidade espectral não seja limitada na frequência zero, ou seja, a função de autocorrelação não será inteiramente sintetizável. Tomando em consideração essas características, a família de modelos ARFIMA(p,d,q) permite prever adequadamente séries com longas dependências seriais.

Seja uma série temporal denotada por x_t. O modelo ARFIMA(p,d,q) pode ser definido por (GRANGER; JOYEUX, 1980):

$$\alpha(\Psi)(1-\Psi)^d x_t = \beta(\Psi)\varepsilon_t \tag{2.13}$$

em que $(1-\Psi)^d x_t = \mu_t$; $|d| \in [-0.5, 0.5]$ a ordem de integração fracionária (quando $d \geqslant 0.5$, a série não é estacionária, quando $d \in [-0.5, 0.0]$ a série tem uma pequena dependência, e quando $d \in [0.0, 0.5]$ a série tem uma longa dependência); $\alpha(\Psi)$ e $\beta(\Psi)$ representam os polinômios, tais que $\alpha(\Psi) = 1 - \alpha_1 x_{t-1} - \alpha_2 x_{t-2} - ... - \alpha_p x_{t-p}$ e $\beta(\Psi) = 1 - \beta_1 \varepsilon_{t-1} - \beta_2 \varepsilon_{t-2} - ... - \beta_q \varepsilon_{t-q}$. O operador de integração, por sua vez, toma uma forma tal que:

$$(1-\Psi)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k \Psi^k \tag{2.14}$$

em que k é o número de parâmetros a serem ajustados. Vale ressaltar que se os polinômios $\alpha(\Psi)$ e $\beta(\Psi)$ têm suas raízes fora do círculo unitário e que não possuem raízes comuns, o modelo $(1-\Psi)^d x_t$ é estacionário de segunda ordem e representa então um processo invertível (com $d = 0.5\ ou\ -0.5$) (HOSKING, 1981).

A identificação representa uma das etapas mais complexas no que se trata da modelagem de séries temporais. Porém, é comum que vários modelos candidatos sejam empregados para que possamos realizar uma escolha do modelo mais parcimonioso, isto é, escolher o modelo com menor número de parâmetros para estimação (MORETTIN; TOLOI, 2004). Neste trabalho, usamos as funções de autocorrelação (FAC) e autocorrelação parcial (FACP) amostrais para identificar os modelos ARIMA(p,d,q) e ARFIMA(p,d,q). Além disso, usamos também os critérios de penalizações utilizados na literatura, o chamado critério de informação AIC (*Akaike Information Criteria*) e o BIC (*Bayesian Information Criteria*).

AIC é a medida relativa da qualidade de ajuste do modelo estatístico estimado. BIC é uma medida de avaliação de modelos definida em termos da probabilidade *a posteriori*. Tanto o AIC quanto o BIC fundamentam-se na verossimilhança, impondo, entretanto, diferentes penalizações. Esses critérios são definidos como:

$$AIC(p,q) = ln\ \hat{\sigma}_{\varepsilon}^2 + n\ \frac{2}{N} \tag{2.15a}$$

$$BIC(p,q) = ln\ \hat{\sigma}_{\varepsilon}^2 + n\ \frac{ln\ N}{N} \tag{2.15b}$$

em que $n = p + q$; N o número de observação e $\hat{\sigma}_{\varepsilon}^2$ a variância dos erros estimada. Note que o critério BIC tende a escolher um modelo mais parcimonioso do que o critério AIC. O risco é que o critério AIC pode indicar modelos sobre-parametrizados ou super especificados. É por isso que o uso das funções de autocorrelação (FAC) e autocorrelação parcial (FACP) é bastante importante.

Uma vez os parâmetros p, d e q são determinados, passa-se ao próximo passo, ou seja, para a etapa de estimação dos parâmetros α, β e da variância σ_{ε}^2. Dois métodos são basicamente utilizados nesta etapa: - mínimos quadrados ordinários (MQO), - e máxima verossimilhança (MORETTIN; TOLOI, 2004). Para modelos da família ARIMA é recomendável o método de máxima verossimilhança. Após verificação do modelo estimado, ou seja, a etapa de análise dos resíduos, a partir da estatística Q de Box-Pierce, análise da hipótese de normalidade e teste ARCH-LM, assim como análise da significância dos parâmetros estimados, passa-se para a etapa de previsões.

A previsão é *estática* quando é realizada um passo à frente. Enquanto, a previsão é *dinâmica* quando é feita vários passos à frente. A qualidade da previsão pode ser medida a partir do coeficiente de determinação, sua magnitude por meio de medidas de erros e a acurácia determina a precisão da direção dos valores previstos.

2.2 Modelos de Redes Neurais Artificiais

As redes neurais artificiais são modelos não-paramétricos. São ferramentas bastante poderosas e adequadas para tratar uma grande diversidade de problemas tais como de classificação e reconhecimento de padrões, processamento de sinais, aproximação ou otimização de sistemas com desempenhos complexos e previsão de séries temporais.

Nesta seção apresentamos os aspectos conceituais e técnicos de modelos não-paramétricos, ou seja, redes neurais perceptron múltiplas camadas (*Multilayer Perceptron* - MLP), redes neurais recorrentes *Long Short-Term Memory* (LSTM) e *Gated Recurrent*

Unit (GRU), justificando assim suas aplicações como modelos adequados na previsão de séries temporais.

2.2.1 Um Breve Histórico da Área e sua Aplicação em Séries Temporais

O primeiro modelo algébrico de um neurônio artificial foi desenvolvido por McCulloch e Pitts em 1943 (HAYKIN, 2009). Neste modelo, as entradas eram binárias e a função de ativação do tipo degrau. Além disso, o modelo admitia a presença de sinapses inibitórias, simulando assim o comportamento e as funções do neurônio biológico.

Baseado no sistema nervoso biológico, o modelo proposto por McCulloch e Pitts foi composto de elementos computacionais, chamados de neurônios ordenados em padrões, no qual o modelo recebe informações de entrada para processamento, simulando os pesos sinápticos das diferentes conexões entre neurônios, e gera uma saída (BALLINI, 2000). No modelo de McCulloch e Pitts, a saída é calculada com base nos sinais de entrada (Figura 2.4). Ainda, a saída é determinada calculando-se a soma ponderada das entradas, com base nos *ganhos sinápticos*, isto é, os pesos de ponderação. Neste modelo não está presente as características adaptativas para ser um modelo de otimização inteligente.

Após o desenvolvimento do trabalho de McCulloch e Pitts, houve muitos outros avanços significativos na área de redes neurais. Em 1949, Hebb desenvolve um modelo de redes neurais artificiais, em que as conexões ativas se fortalecem.

Em 1958, baseado nas ideias de McCulloch e Pitts, Rosenblatt introduz as redes neurais artificiais chamadas *perceptrons*, nas quais existe uma única camada intermediária. O neurônio artificial calcula uma soma ponderada dos números $\sum_j w_j x_j$ que representam as atividades dos neurônios na camada intermediária (LECUN, 2019). O funcionamento do perceptron segue a lógica de que os neurônios que recebem os sinais de entrada pertencem a camada de entrada, enquanto os neurônios que recebem os sinais de saída via camada intermediária estão nesta camada. A última camada é a camada de saída (HAYKIN, 2009). No total, existe uma camada de entrada, camada intermediária e camada de saída. O modelo de Rosenblatt lida com valores contínuos e não apenas com valores binários.

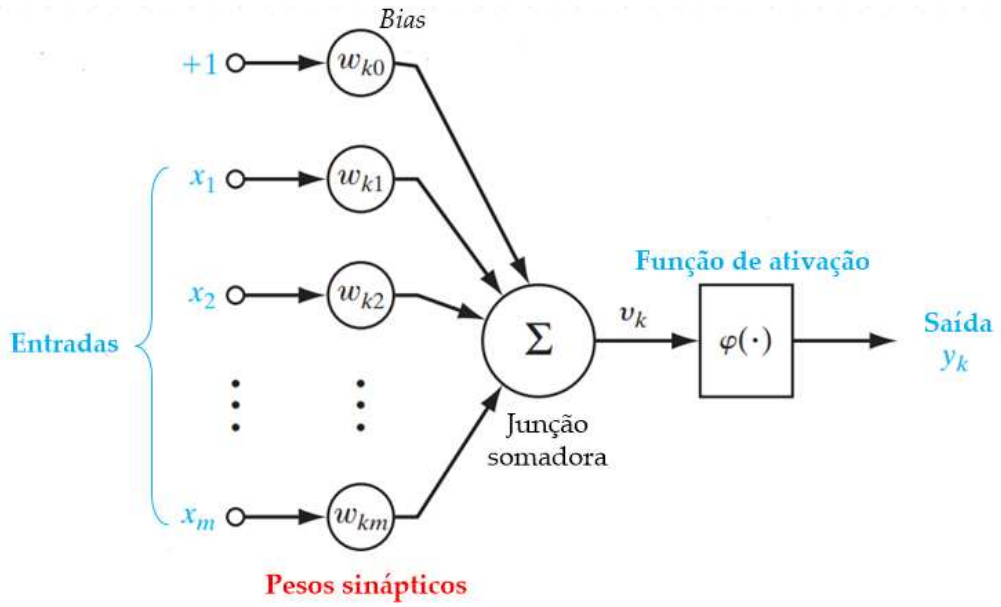


Figura 2.4 – Modelo de Redes neurais artificiais.
Adaptada de (BOCCATO; ATTUX, 2020).

A saída de um neurônio artificial pode ser expressa, matematicamente, como:

$$\hat{y}_k = \varphi\left(v_k\right) = \varphi\left(\sum_{j=1}^{m} w_{kj}x_j + w_{k0}\right) = \varphi\left(\sum_{j=0}^{m} w_{kj}x_j\right) \tag{2.16a}$$

$$\hat{y}_k = \varphi\left(\mathbf{w}_k^T\mathbf{x}\right) \tag{2.16b}$$

em que $\mathbf{w}_k = (w_{k0}, w_{k1}, ..., w_{km})$ e $\mathbf{x} = (x_1, x_2, ..., x_m)$ são vetores de pesos sinápticos e de sinais de entradas, respectivamente; $\varphi(\cdot)$ função de ativação; w_{k0} o termo de polarização. No total, existem sete elementos, a saber: sinais de entrada, pesos sinápticos, operador de agregação, termo de polarização ou limiar de ativação, potencial de ativação, função de ativação e sinal de saída.

A ideia do modelo de Rosenblatt foi de ajustar o modelo de rede neural artificial de forma que ela aprende com base no dado desejado, de forma que a rede precisa encontrar o valor ideal dos pesos sinápticos a partir de um algoritmo de otimização adequado. Apesar de pouco sucesso, Minsky e Papert, em 1969, provaram algumas limitações do perceptron com uma única camada intermediária frente aos problemas de não linearidade separáveis. Essas críticas apontaram restrição para aplicação dessa metodologia. Para mais informações sobre os avanços da área de redes neurais artificiais, ver por exemplo (LECUN, 2019) e (HAYKIN, 2009).

O ressurgimento da teoria de redes neurais artificiais foi observado a partir da década de 1980. Primeiro, o trabalho do Hopfield em 1982 incorporou o princípio físico de armazenamento de informação na configuração da estabilidade dinâmica. Este foi um

dos primeiros modelos a introduzir a dinâmica no perceptron. Romelhart et al., em 1986, propôs um método de treinamento de redes neurais perceptron de múltiplas camadas, o chamado *algoritmo de retro-propagação* (LECUN, 2019). O algoritmo proporcionou a estimação de pesos sinápticos. Desde esse momento, vários estudos foram realizados para fim de melhorar o algoritmo de retro-propagação: evitar sobre-ajuste no treinamento dos modelos (o problema de viés-variância ou *Underfitting* e *Overfitting*); evitar a lentidão de aprendizado; evitar mínimos locais; etc.

Muitos estudos foram conduzidos para abordar uma perspectiva de modelagem estatística. Os objetivos foram realizar previsão de séries temporais, entre outros, (WEIGEND *et al.*, 1990), (WEIGEND *et al.*, 1992), (ZHANG *et al.*, 1998). Nas últimas décadas, vários estudos foram investigados com ideias de comparar a metodologia baseada em redes neurais perceptrons de múltiplas camadas com a metodologia clássica proposta por Box & Jenkins 1970, para previsão de séries temporais: (BALLINI, 2000), (ZHANG, 2003), (MORELLI *et al.*, 2004), (COELHO *et al.*, 2008), (AZZOUNI; PUJOLLE, 2017), entre outros.

Em resumo, as arquiteturas de redes neurais perceptrons se definem como a forma no qual diversos neurônios artificiais estão arranjados uns em relação aos outros. Desta forma, é possível dividir em dois grupos com diferentes arquiteturas: (i) redes neurais *feedforward* ou progressivas, e (ii) recorrentes.

Existem vários tipos de aprendizagem em redes neurais: aprendizagens supervisionada, não-supervisionada, semi-supervisionada e por reforço, entre outras (HAYKIN, 2009). Aprendizagem supervisionada é utilizada quando há um valor desejado associado com cada entrada do conjunto de dados de treinamento. Dessa forma, é definido uma função custo que expresse uma medida de erro entre a resposta que o modelo gerou e a resposta esperada para cada padrão do conjunto de treinamento. Nesta dissertação utilizamos apenas aprendizagem supervisionada para tarefa de previsão. A próxima seção trata-se de arquiteturas de redes neurais perceptrons de múltiplas camadas.

2.2.2 Arquiteturas de Redes MLP

A arquitetura de redes neurais *multilayer perceptron* (MLP) é a generalização das redes neurais perceptrons de Rosenblatt. Ela se caracteriza em uma ou mais camadas intermediárias (ou escondidas ou ocultas) de neurônios processadores. Essa arquitetura é direta, isto é, as informações de entrada são processadas diretamente até a saída. O número de neurônios e das camadas ocultas dependem diretamente do tipo e da complexidade do problema que desejamos tratar.

Os sinais ou observações de entrada passam informações na primeira camada

oculta sem nenhuma modificação ocorrida. Por sua vez, as camadas ocultas transmitem informações através das conexões entre camadas de entrada e saída. Por fim, a camada de saída fornece a resposta final (BALLINI, 2000). A conexão dos neurônios de uma camada oculta é realizada unicamente com os neurônios da próxima camada. Portanto, os neurônios não fazem conexão com outros neurônios da mesma camada. Além disso, cada neurônio de uma camada é conectado a todos os neurônios da próxima camada. As ligações entre sinais de entrada e camadas ocultas têm em cada uma, um peso sináptico correspondente w_{ij}. Esse peso é ponderado a cada observação de entrada x_i. Uma rede *feedforward* com apenas uma camada oculta, e função de ativação do tipo linear, é nada mais que uma regressão linear simples.

Uma arquitetura de redes MLP é chamada de aprendizagem profunda quando esta apresenta as seguintes características:

1. Número arbitrário de camadas ocultas e neurônios entre entrada e saída da rede;
2. Neurônios de uma camada l estão conectados a todos os neurônios da camada seguinte, ou seja, $l+1$, ..., L. Assim, é chamada de estrutura densa ou totalmente conectada;
3. Função de ativação escolhida.

Na Figura 2.5, a camada de entrada recebe as entradas e propaga para a próxima camada; as camadas ocultas ou intermediárias realizam mapeamentos temporais não-lineares. Os neurônios da camada de saída combinam a informação recebida da última camada oculta, e produz as respostas da rede para o padrão de entrada.

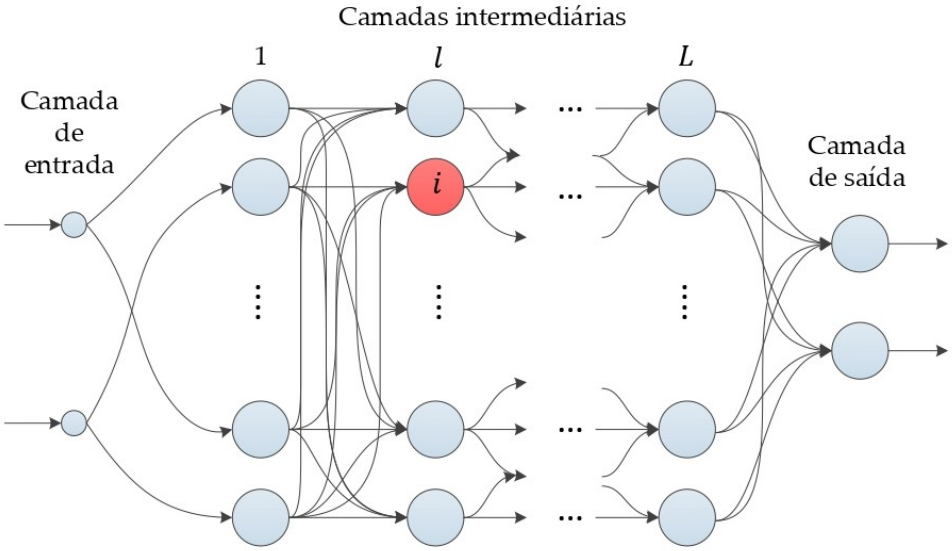


Figura 2.5 – Arquitetura de redes neurais MLP.
Extraída de (BOCCATO; ATTUX, 2020).

Algebricamente, a saída do modelo MLP pode ser expressa como:

$$\hat{y}_i^l = \varphi^l \left(\sum_{j=1}^{n_{l-1}} w_{ij}^l y_j^{l-1} + w_{i0}^l \right) \tag{2.17}$$

em que $i = 1, ..., n_l$ representa o i-ésimo neurônio da l-ésima camada oculta $l = 1, ..., L$; w_{ij}^l é o peso sináptico da conexão do j-ésimo neurônio da camada $(l-1)$ ao i-ésimo neurônio da camada l; $y_j^0 = x_j^0$, $j = 1, ..., m$, são os sinais temporais de entrada da primeira camada oculta.

A função de ativação $\varphi(\cdot)$ determina o caráter do mapeamento temporal realizado pelo neurônio. Por isso, a função a ser usada na camada de saída de redes MLP depende do problema tratado. Em geral, em previsão de séries temporais, o neurônio de saída tem função de ativação igual à identidade, de modo que o valor previsto é obtido por meio de uma combinação linear das ativações dos neurônios da última camada oculta (GOODFELLOW *et al.*, 2016). Existem várias funções de ativações, na qual pode ser uma função linear, retificadora, logística, ou tangente hiperbólica.

2.2.2.1 Funções de Ativação

A escolha da função de ativação dos neurônios tem influência sobre alguns aspectos do modelo. No caso da previsão de séries temporais, uma melhor escolha realizada da função de ativação pode levar a uma convergência fácil e rápida na otimização dos parâmetros. As funções de ativação mais conhecidas são:

1. Função de ativação linear (CYBENKO, 1992):

é a função de ativação mais simples, em que não altera a saída de um neurônio. Geralmente é utilizada na camada de saída quando se trata de redes neurais para regressão. Algebricamente, a função linear é expressa como:

$$\varphi(v_k) = v_k \tag{2.18}$$

2. Função de ativação sigmóide (CYBENKO, 1992):

é frequentemente utilizada nas camadas ocultas de redes neurais *feedforward*, que precisam ter como saída apenas números positivos. A função sigmóide assume valores apenas entre 0 (não ativação) e 1 (ativação):

$$\varphi(v_k) = \frac{1}{1 + e^{-pv_k}} \tag{2.19a}$$

$$\varphi^{'}(v_k) = p\ \varphi(v_k)(1 - \varphi(v_k)) \tag{2.19b}$$

em que p é o parâmetro de inclinação da função sigmóide, variando este parâmetro, obtemos funções sigmóides com diferentes inclinações.

O uso da função sigmóide no problema de previsão de séries temporais pode levar a algumas dificuldades durante a etapa de treinamento, pois quando a derivada da função sigmóide tende a zero (Figura 2.6(b)), a propagação do gradiente nesta iteração dissipa-se nessas regiões causando dificuldades no treinamento. Dessa forma, o treinamento pode parar enquanto que não atingiu o mínimo erro (GOODFELLOW *et al.*, 2016).

3. Função de ativação retificadora (GÉRON, 2019):

com o advento das técnicas de *deep learning*, funções lineares por partes ganharam terreno por apresentarem uma melhor relação custo-benefício entre eficiência (convergência rápida) no ajuste de pesos sinápticos e desempenho de otimização. A função de ativação retificadora (ReLU, do inglês *Rectified Linear Unit*) é bastante usada nos modelos de aprendizado profundo para previsão de séries temporais, e são dadas por:

$$\varphi(v_k) = max(0, v_k) \tag{2.20a}$$

$$\varphi^{'}(v_k) = \begin{cases} 1, & se\ v_k \geqslant 0 \\ 0, & c.c. \end{cases} \tag{2.20b}$$

As redes neurais profundas com função de ativação ReLU são simples de serem aplicadas, pois essa função é parecida com a função identidade. A diferença é que a função ReLU produz zero em metade do seu domínio (Figura 2.6(c)). As derivadas da função de ativação ReLU são estáveis, sendo 1, quando $v_k \geqslant 0$ e 0 quando $v_k < 0$. A segunda derivada da função ReLU tem valor zero em todo o domínio. A ativação ReLU é muito mais eficiente do que a função logística e é uma das descobertas que contribuiu de forma significante para a recente popularidade de *Deep Learning* (GÉRON, 2019). Ainda, essa função é bastante utilizada nos modelos recorrentes para previsão de séries temporais financeiras (MOSCATELLI *et al.*, 2020).

4. Função de ativação tangente hiperbólica (GÉRON, 2019):

a função de ativação tangente hiperbólica (tanh) também possui uso comum em redes neurais cujas saídas variam de -1 a 1, em vez de 0 a 1 como no caso da função sigmóide. Esse intervalo tende a tornar a saída de cada camada mais ou menos centrada em torno de 0 no início do treinamento, o que geralmente ajuda a acelerar a convergência. Também, é comum usar essa função no caso de modelos de redes neurais recorrentes para previsão de séries temporais, sendo dada por:

$$\varphi(v_k) = tanh(pv_k) \tag{2.21a}$$

$$\varphi^{'}(v_k) = p(1 - tanh^2(pv_k)) \tag{2.21b}$$

em que p é uma constante positiva.

Segundo Géron (2019), a função tangente hiperbólica, assim como a função sigmóide, tem um formato de *S*. Isso significa que, as saturações podem estar presentes no momento de treinamento. Mas, o valor da derivada é maior, chegando a 1 quando $v_k = 0$.

Existe, na literatura, outras funções de ativação tais como *Leaky* ReLU, PReLU, ELU e SELU (PEDAMONTI, 2018). Essas alternativas são usadas no caso em que os neurônios apresentam valores negativos como respostas a vários estímulos de entrada. Nesse caso, esses neurônios praticamente deixam de participar do processamento da rede e não sofrem ajustes em seus parâmetros, pois a derivada da função de ativação é nula.

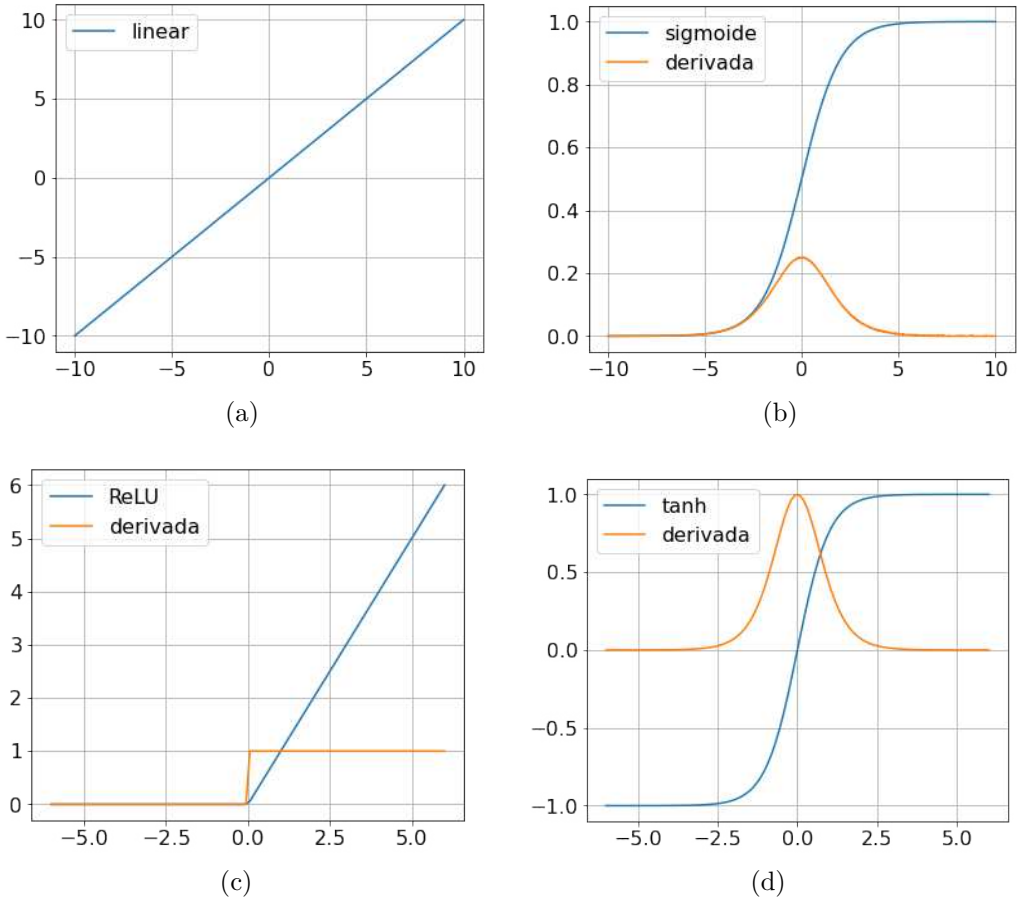


Figura 2.6 – Funções de ativações e suas derivadas.

2.2.2.2 Processo de Treinamento: algoritmo de retro-propagação

Treinar uma rede neural MLP consiste no processo de ajuste dos pesos sinápticos w_{ij}^{l} de todas as camadas, em que buscamos os valores que levem ao melhor mapeamento entrada-saída possível (HAYKIN, 2009). Em outras palavras, busca-se resolver o problema de otimização não-linear irrestrito (GÉRON, 2019). Isto é, minimizar a função custo ($\min_{\mathbf{w}} J(\mathbf{w})$) que representa uma medida de erro entre saída fornecida pelo modelo e saída desejada, ou simplesmente os valores previstos e valores reais.

O processo de treinamento da rede MLP supervisionado pode ser realizado por meio do algoritmo de retro-propagação do erro (BP, do inglês *backpropagation*), desenvolvido por Rumelhart e co-autores em 1986. O algoritmo de retro-propagação é baseado em uma regra de correção do erro. O BP pode ser visto como uma generalização do algoritmo de filtro adaptativo ou, ainda, um caso especial do algoritmo de mínimos quadrados (MOSCATELLI *et al.*, 2020). Uma alternativa para o cálculo do BP é o uso do método iterativo baseado no gradiente que busca minimizar o erro quadrático médio (MSE) entre

a saída observada e a saída desejada:

$$\min_{\mathbf{w}} J(\cdot) = \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{r} \left(d_j(n) - y_j^{L+1}(n) \right)^2 \tag{2.22}$$

Para minimizar a função de perda, o algoritmo BP utiliza uma técnica de busca baseada no gradiente descendente (LUENBERGER; YE, 2016). A forma mais usual é descrita ainda em (GÉRON, 2019).

$$\nabla J(\mathbf{w}) = \left(\frac{\partial J(\mathbf{w})}{\partial w_1}, \frac{\partial J(\mathbf{w})}{\partial w_2}, ..., \frac{\partial J(\mathbf{w})}{\partial w_n} \right)^T \tag{2.23}$$

A Equação (2.23) é chamada de vetor gradiente, em que os pesos sinápticos são atualizados conforme uma regra de atualização da seguinte forma (GÉRON, 2019):

$$\mathbf{w}(k) - \eta \ \nabla J(\mathbf{w}(k)) \rightarrow \mathbf{w}(k+1) \tag{2.24}$$

em que $\mathbf{w}$ é o vetor com todos os parâmetros (pesos) do modelo, η é a taxa de aprendizagem (*learning rate*), e $\nabla(\cdot)$ o operador gradiente. Os métodos de gradiente descendente são métodos de busca local, isso significa que a convergência do modelo é realizada a partir de mínimos locais. Como aponta Géron (2019), a busca do mínimo local apresenta uma solução ótima em relação a seus vizinhos; o mínimo global também é uma solução ótima, mas, não apenas em relação a seus vizinhos, mas a todo o domínio do modelo (Figura 2.7).

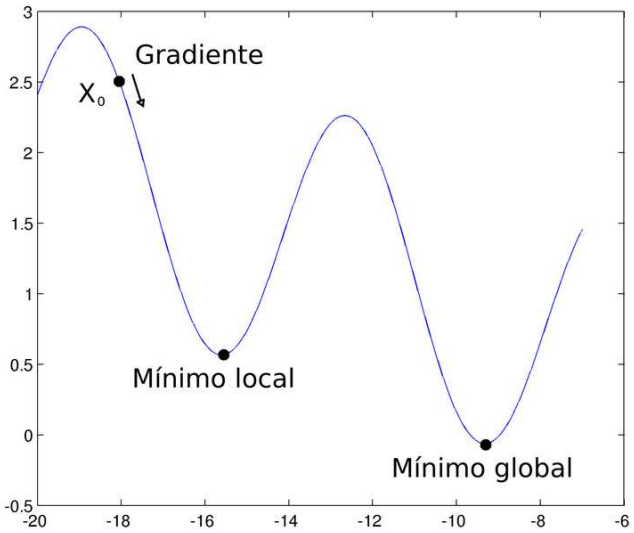


Figura 2.7 – Gradiente descendente com mínimo local e global.

Vale ressaltar que na Equação (2.22), os pesos sinápticos do modelo não aparecem de forma explícita. Para que a dependência de $J(\cdot)$ em relação aos pesos seja obtida é necessário aplicar a regra de retro-propagação, que corresponde a um processo no qual os erros são propagados de trás para frente, ou seja, da saída no qual se observa o erro para a entrada. Portanto, a regra resultou no conhecido algoritmo de retro-propagação do erro. Para detalhes das derivações dos pesos ver (HAYKIN, 2009).

Desde seu princípio, o algoritmo de retro-propagação é considerado como a *pedra angular* no treinamento das redes neurais. O algoritmo BP consiste em duas fases (HAYKIN, 2009): fase *forward* e fase *backward*. A fase *forward* ou passo para frente consiste na propagação das entradas pela rede, resultando na saída da rede. A fase *backward* ou passo para trás consiste no calculo do gradiente da função custo na camada final e usa o mesmo gradiente para aplicar recursivamente a regra da cadeia para atualizar os pesos sinápticos (retro-propagação).

Na prática, existem várias formas para implementação do algoritmo de retro-propagação. De acordo com Haykin (2009) o ajuste dos parâmetros pode ser dada por: (i) estimação padrão-a-padrão; (ii) estimação em batelada (*batch*); e (iii) estimação com pequenos lotes de dados chamada de *mini-batch*. Neste trabalho, usamos a estimação com pequenos lotes de dados. A estimação a partir do meio-termo de observações é bastante eficiência, o que traduz uma redução significativa do tempo de aprendizagem (GÉRON, 2019). O processo de estimação (aprendizagem) usando *mini-batch* ocorre a partir de um dado número de épocas (*epoch*), no qual após atingir o máximo de épocas, e varrer o conjunto de dados de treinamento, o processo se encerra.

Algoritmo 1: Algoritmo baseado em *mini-batch*

Entrada: $(\mathbf{x}_i, \mathbf{y}_i)_{i \in (N_T)} \Leftarrow$ Conjunto de dados de treinamento;
Defina uma condição inicial para o vetor de pesos $\mathbf{w}$ e o valor do passo η;
Faça $k = 0$ e calcule $J(\mathbf{w}(k))$;
enquanto *o critério de parada não for atendido* **faça**
  para cada *l variando de 1 até* N_T **faça**
    Apresente o padrão l de entrada, que compões um *mini-batch* à rede
    Calcule $J_l(\mathbf{w}(k))$ e $\nabla J_l(\mathbf{w}(k))$
    $\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\eta}{m} \sum_{l=1}^{m} \nabla J_l(\mathbf{w}(k))$
    $k = k + 1$
    Calcule $J(\mathbf{w}(k))$
  fim
fim
Saída: $\mathbf{w}(k) - \eta \nabla J(\mathbf{w}(k)) \rightarrow \mathbf{w}(k+1)$

Além da técnica baseada em *mini-batch*, que reduz o tempo de aprendizagem, existe outras abordagens no qual aumenta a velocidade de aprendizagem sem, portanto, aumentar a taxa de aprendizagem (η). Essas ferramentas são mais usadas para aprendi-

zagem em redes neurais recorrentes. Problemas tais como não-convexidade da superfície de erro e convergência com menor número de iterações, levaram à proposição de técnicas de otimização bastante eficientes: (i) Gradiente com momento; (ii) *Nesterov Accelerated Gradient* (NAG); (iii) Algoritmo com passo adaptativo; (iv) Algoritmo adaptativo *Ada-Grad*; (v) RMSProp; (vi) *Adaptive Moment Estimation* (ADAM). A técnica baseada em ADAM é utilizada neste trabalho, e sua formalização é apresentada mais adiante.

2.2.3 Arquiteturas de Redes LSTM

A arquitetura da célula LSTM (*Long Short-Term Memory*) é similar a rede *feedforward*. A diferença principal reside no que diz respeito às realimentações (HOCHREITER; SCHMIDHUBER, 1997). Assim, a célula LSTM tem potencialidade de modelar os dados temporais (JASON, 2018).

Existem três principais tipos de redes neurais recorrentes, a saber:

1. Rede neural recorrente simples ou de memória curta (RNN, do inglês *Recurrent Neural Networks*).

2. Rede neural recorrente de memória de longo-prazo e curto-prazo (LSTM, do inglês *Long Short-Term Memory*) (HOCHREITER; SCHMIDHUBER, 1997).

3. Rede neural recorrente *Gated Recurrent Unit* (GRU) (CHO *et al.*, 2014).

As redes neurais *feedforward* são capazes de resolver tarefas de alta complexidade, tais como realizar previsões de séries temporais e/ou reconhecimento de padrões e, ainda, elas constroem mapeamentos estáticos da entrada até a saída. Diante de uma amostra de entrada x_t, a rede não tem a capacidade de reutilizar informações processadas de amostras passadas $x_{t-1}, x_{t-2}, ..., x_{t-n}$. Esse fato impede às redes neurais *feedforward* a serem redes dinâmicas.

Por sua vez, redes neurais recorrentes RNNs usam sequências dinâmicas de eventos anteriores. Isto é, há uma persistência de informações ao longo do tempo. Em outras palavras, as RNNs possuem ciclos de propagação das ativações de neurônios de períodos passados como entradas que influenciam as previsões no período futuro. As ativações de neurônios são armazenadas nos *estados internos* da rede, com objetivos de guardar informações no tempo. A rede RNN é um tipo básico de redes recorrentes, capazes de aprender padrões em curto-prazo (curta-memória).

Por fim, as células LSTM e GRU têm as mesmas características das redes RNN, ou seja, são redes recorrentes que retêm a dependência de longo e curto-prazos da

sequência de eventos anteriores. A principal diferença entre as estruturas LSTM e GRU reside no fato de que a célula LSTM tem três portas e a GRU tem duas portas. Nesta dissertação, utilizamos as redes neurais LSTM e GRU (Figura 2.8). A seguir, apresentamos a arquitetura da rede LSTM.

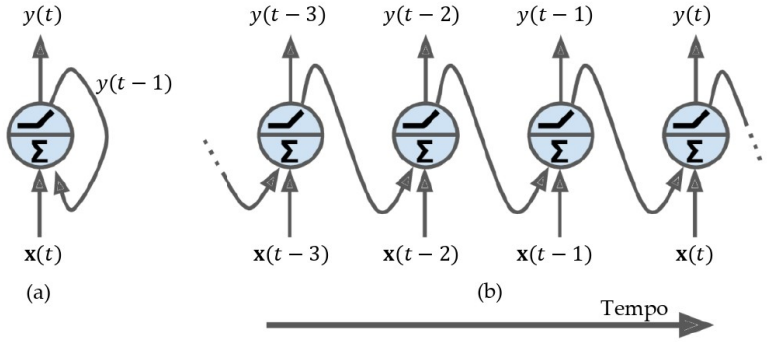


Figura 2.8 – Arquitetura com recorrência. Adaptada de (GÉRON, 2019)

.

Nestas arquiteturas existem retroalimentação, isto é, a cada período do tempo t, a rede recebe o vetor de entrada $\mathbf{x}(t)$ e o vetor de saída do período anterior, i.e. $y(t-1)$ (Figura 2.8(a)), e assim por diante (Figura 2.8(b)). Uma rede neural recorrente possui dois vetores de pesos sinápticos: $\mathbf{w}_x$ que pondera as entradas $\mathbf{x}(t)$ e $\mathbf{w}_y$ que pondera as realimentações $y(t-1)$. A célula LSTM tem um funcionamento diferente da rede *feedforward*. A Figura (2.9) apresenta uma célula LSTM.

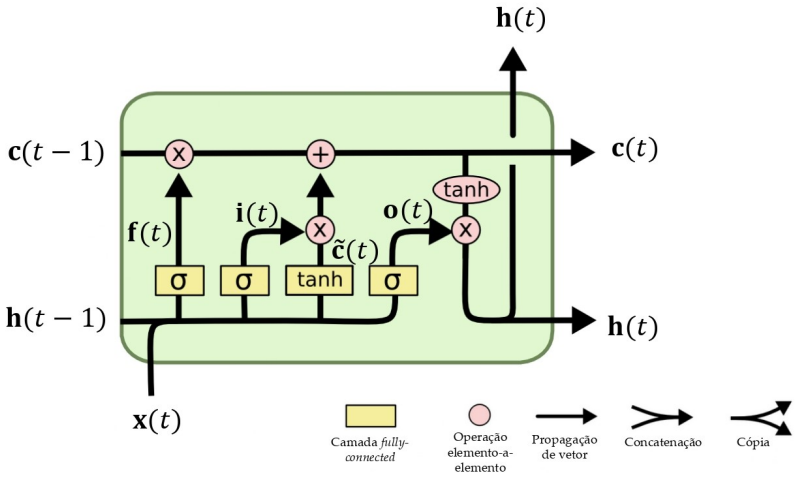


Figura 2.9 – Estrutura interna da célula LSTM.
Adaptada de (OLAH, 2015).

Na Figura 2.9, $\mathbf{c}(t)$ é o vetor de estado que armazena informações de longo prazo (ou simplesmente vetor de longo prazo). Uma vez que a rede aprende, as informações úteis são armazenadas no vetor de estado; caso contrário as informações são descartadas; $\mathbf{h}(t)$ é o vetor de curto prazo, isto é, a saída de informações úteis da célula (GRAVES, 2012).

As operações sobre o vetor de longo prazo são controladas por três portas, a saber: (i) porta de esquecimento (*forget gate*); (ii) porta de entrada (*input gate*); e (iii) porta de saída (*output gate*). Dependendo da função de ativação escolhida, cada elemento do vetor de saída assume os valores no intervalo entre 0 e 1. Isto é, quando o valor for 0, a informação é removida, enquanto que para o valor 1, a informação é mantida. A seguir, apresentamos passo a passo a célula LSTM.

1. **Porta de esquecimento (*forget gate*):** Sua importância é selecionar os elementos que devem ser guardados no instante t dentro do vetor de longo prazo ou vetor de estado, $\mathbf{c}(t-1)$ (Figura 2.9), em que $\mathbf{f}(t)$ é o vetor que controla o desempenho da porta; $\mathbf{x}(t)$ é a entrada atual; $\mathbf{h}(t-1)$ é a saída passada; ou seja (GRAVES, 2012):

$$\mathbf{f}(t) = \sigma([\mathbf{w}_{hf}\mathbf{h}(t-1), \mathbf{w}_{xf}\mathbf{x}(t)] + \mathbf{b}_f). \tag{2.25}$$

2. **Porta de entrada (*input gate*):** O objetivo desta porta é de tomar uma decisão sobre a intensidade dos elementos do vetor de estado a serem modificados. Nessa porta, $\mathbf{i}(t)$ (Figura 2.9) representa o sinal de controle desta porta, com uma função de ativação do tipo sigmóide; por sua vez, $\widetilde{\mathbf{c}}(t)$ é uma camada densa com função de ativação do tipo tangente hiperbólica. Na camada densa todas as informações novas devam ser armazenadas no vetor de estados (GRAVES, 2012):

$$\mathbf{i}(t) = \sigma([\mathbf{w}_{hi}\mathbf{h}(t-1), \mathbf{w}_{xi}\mathbf{x}(t)] + \mathbf{b}_i) \tag{2.26a}$$
$$\widetilde{\mathbf{c}}(t) = \tanh([\mathbf{w}_{hc}\mathbf{h}(t-1), \mathbf{w}_{xc}\mathbf{x}(t)] + \mathbf{b}_c) \tag{2.26b}$$

   O novo vetor de longo prazo $\mathbf{c}(t)$ é composto pela adição de valores armazenados pela porta de esquecimento e das novas informações da porta de entrada, ou seja:

$$\mathbf{c}(t) = \mathbf{f}(t) \bigotimes \mathbf{c}(t-1) + \mathbf{i}(t) \bigotimes \widetilde{\mathbf{c}}(t). \tag{2.27}$$

   em que $\bigotimes$ é a multiplicação elemento a elemento.

3. **Porta de saída (*output gate*):** Esta porta tem como objetivo escolher quais partes do novo vetor de longo prazo $\mathbf{c}(t)$ devem servir na saída da célula LSTM. Como ilustrada na Figura 2.9, antes de servir como a saída, o novo vetor de longo prazo $\mathbf{c}(t)$ passa pela função de ativação tanh.

$$\mathbf{o}(t) = \sigma([\mathbf{w}_{ho}\mathbf{h}(t-1), \mathbf{w}_{xo}\mathbf{x}(t)] + \mathbf{b}_o) \tag{2.28a}$$
$$\hat{\mathbf{y}}(t) = \mathbf{h}(t) = \mathbf{o}(t) \bigotimes \tanh(\mathbf{c}(t)) \tag{2.28b}$$

Um neurônio da célula LSTM pode ser realimentado por sinal de entrada e por sua própria saída. Ao contrário da rede *feedforward* simples ou profunda, que usa parâmetros diferentes em cada camada, a célula LSTM compartilha os mesmos parâmetros em todas as etapas. Isso reflete o fato de que estamos executando a mesma tarefa em cada etapa, apenas com entradas diferentes. Tratando de séries temporais, as realimentações são acompanhadas de defasagens de uma ou mais unidades do tempo (Figura 2.8). Esse tipo de abordagem amplifica-se todas as possibilidades de modelar as estruturas de auto-dependência de dados temporais.

2.2.3.1 Retro-Propagação através do Tempo

O treinamento do modelo LSTM supervisionado é realizado por meio do algoritmo de retro-propagação do erro através do tempo (BPTT, do inglês *backpropagation-through-time*) (WERBOS, 1990). Existem também outros algoritmos de treinamento, como, por exemplo, algoritmo de aprendizagem recorrente em tempo real e o filtro de Kalman estendido. O algoritmo BPTT é uma extensão do algoritmo de retro-propagação, visto acima. O tempo, no entanto, é simplesmente expresso por uma série ordenada e bem definida de operações, ligando um passo de tempo ao seguinte.

Sabemos que a saída do neurônio LSTM é dada por:

$$\mathbf{y}(t) = \varphi\left(\mathbf{x}(t)\mathbf{w}_x + \hat{\mathbf{y}}(t-1)\mathbf{w}_y + \mathbf{b}\right) \tag{2.29}$$

$\varphi(\cdot)$ função de ativação; $x(t)$ a entrada atual; $\mathbf{y}(t-1)$ a realimentação; w_x e w_y pesos sinápticos; $\mathbf{b}$ termo de polarização. Suponha a função custo medida da seguinte forma:

$$\min_{\mathbf{w}} \; J(\mathbf{w}) = \min_{\mathbf{w}} \; \frac{1}{2}\sum_{t=1}^{N} \left\|(\mathbf{x}(t), \hat{\mathbf{y}}(t-1), \mathbf{w}) - \mathbf{y}(t)\right\|^2 \tag{2.30}$$

A solução de (2.30) é obtida via processos iterativos, usando a técnica de treinamento do tipo *on-line* ou *mini-batches*.

O algoritmo BPTT calcula os gradientes relativos a todos os parâmetros da rede, soma e usa somente a média desses para atualizar o próximo parâmetro. Como no treinamento da rede MLP, o objetivo aqui também é de calcular a derivada da função custo em relação ao vetor $\mathbf{w}$, ou seja,

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial J(\mathbf{w})}{\partial w_x} + \frac{\partial J(\mathbf{w})}{\partial \hat{\mathbf{y}}(t-1)}\frac{\partial \hat{\mathbf{y}}(t-1)}{\partial w_y} \tag{2.31}$$

Os pesos são atualizados da seguinte forma:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \, \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \tag{2.32}$$

em que $\eta \in [0, 1]$ representa a taxa de aprendizagem. A escolha do valor de η é muito importante durante o treinamento, pois uma taxa próxima de um pode fazer com que o gradiente tenda a infinito e uma taxa próxima de zero leva o gradiente a dissipar-se. Para atenuar esses problemas, algumas técnicas alternativas são usadas, como por exemplo, a técnica baseada em *batch normalization* (BN) (IOFFE; SZEGEDY, 2015).

O uso da técnica BN permite remover a média dos gradientes e normaliza a variância dos valores somados (Equação (2.32)) para atualização afim de reescalar e deslocar o resultado. Algebricamente, a camada do tipo BN funciona da seguinte forma:

$$\boldsymbol{\mu}_B = \frac{1}{N_B} \sum_{i=1}^{N_B} \mathbf{x}(i); \quad \boldsymbol{\sigma}_B^2 = \frac{1}{N_B} \sum_{i=1}^{N_B} (\mathbf{x}(i) - \boldsymbol{\mu}_B)^2 \tag{2.33}$$

$$\hat{\mathbf{x}}(i) = \frac{\mathbf{x}(i) - \boldsymbol{\mu}_B}{\sqrt{\boldsymbol{\sigma}_B^2 + \boldsymbol{\varepsilon}}}; \quad \mathbf{z}(i) = \boldsymbol{\rho} \bigotimes \hat{\mathbf{x}}(i) + \boldsymbol{\beta} \tag{2.34}$$

em que $\boldsymbol{\mu}_B$ e $\boldsymbol{\sigma}_B^2$ são vetores de média e desvio padrão, respetivamente; $\hat{\mathbf{x}}(i)$ é o vetor gaussiano de média nula e variância unitária; $\mathbf{z}(i)$ é o vetor de observações gerado pelo BN (com saída reescalada e deslocada), onde cada observação possui média e variância definidas pelos vetores $\boldsymbol{\rho}$ e $\boldsymbol{\beta}$; $\boldsymbol{\rho}$ é o vetor que pondera cada atributo normalizado em $\hat{\mathbf{x}}(i)$; $\boldsymbol{\beta}$ contém os parâmetros ajustáveis de BN, ou seja, o *offset* relativo a cada atributo normalizado; $\boldsymbol{\varepsilon}$ representa um número pequeno que impede a divisão por zero; $\bigotimes$ é o produto elemento a elemento.

A técnica baseada em BN é apenas utilizada na etapa de aprendizagem (treinamento) da rede, ainda que na etapa de teste, os vetores $\boldsymbol{\mu}_B$ e $\boldsymbol{\sigma}_B^2$ são substituídos pelos resultados de médias móveis com decaimento exponencial, que foram calculados durante a etapa de treinamento.

Uma outra técnica baseada em ADAM (*Adaptive Moment Estimation*) é utilizada para desafiar o problema de lentidão na convergência do gradiente descendente (KINGMA; BA, 2015). O otimizador ADAM ajusta adaptativamente a taxa de aprendizagem η. A regra de atualização é descrita da seguinte forma:

$$\mathbf{m}(n) = \beta \mathbf{m}(n-1) - (1-\beta) \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \tag{2.35a}$$

$$\mathbf{s}(n) = \alpha \mathbf{s}(n-1) + (1-\alpha) \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \bigotimes \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \tag{2.35b}$$

em que $\mathbf{m}(n)$ e $\mathbf{s}(n)$ são normalmente inicializados como vetores nulos, isto é, $\mathbf{m}(0)$ e $\mathbf{s}(0)$. Ainda, esses momentos têm um viés, sendo necessário aplicar uma correção:

$$\hat{\mathbf{m}}(n) = \frac{\mathbf{m}(n)}{1-\beta^n}; \quad \hat{\mathbf{s}}(n) = \frac{\mathbf{s}(n)}{1-\alpha^n}, \tag{2.36}$$

assim, a regra de atualização dos pesos sinápticos assume a seguinte forma:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta\ \hat{\mathbf{m}}(n) \oslash \sqrt{\hat{\mathbf{s}}(n) + \varepsilon}. \tag{2.37}$$

em que $\oslash$ denota uma divisão elemento a elemento; os hiperparâmetros sejam iguais a $\beta = 0.9$; $\alpha = 0.999$ e $\varepsilon = 10^{-8}$. A rede neural LSTM é treinada a partir de uma dada época.

2.2.3.2 Técnicas da Validação dos Modelos

O processo de aprendizagem consiste em uma seleção adequada dos parâmetros e hiperparâmetros de modelos para um determinado conjunto de dados para treinar. Como vimos na família de modelos lineares, aqui, também, a escolha desses parâmetros e hiperparâmetros pode ser feita em um conjunto de arquiteturas de redes (*feedforward* ou recorrentes) candidatas a ser escolhido como modelo adequado. Para isso existem diferentes técnicas que buscam otimizá-los que resulta em uma melhor acurácia do modelo. Aqui vejamos algumas delas.

No pré-processamento o conjunto de dados disponível é usualmente dividido em conjunto de treinamento ($p\%$) e teste ($1-p\%$). O conjunto de treinamento é utilizado para indução e ajuste dos parâmetros internos dos modelos, enquanto, o conjunto de teste é usado para avaliar a sua capacidade de generalização nos dados ainda desconhecidos. Isto é, desejamos que, ao mesmo tempo, o modelo absorva as informações relevantes contidas nos dados de treinamento e apresente a melhor capacidade de generalização possível em conjunto de dados de teste. Repare que essa divisão de dados de treino e teste pode ser feita aleatoriamente ou dependendo dos objetivos a serem atingidos.

Se a escolha dessa divisão é mal feita, esse mecanismo pode também conduzir ao problema conhecido como *viés-variância* (do inglês, *bias-variance trade-off*) (GEMAN *et al.*, 1992), (BISHOP, 2006). O *efeito viés* ou *underfitting* significa que os erros de treinamento e teste ao longo das iterações tendem a ser elevados; por sua vez, a *variância* ou *overfitting* estipula que embora o erro de treinamento seja baixo, o erro de teste é elevado. Porém, existem várias estratégias que auxilia na obtenção de um modelo com melhor capacidade de generalização (ALPAYDIN, 2014):

1. Validação cruzada (CV, do inglês *cross-validation*);

2. Validação cruzada com k pastas (*k-fold cross-validation*);

3. Parada antecipada (*Early stopping*).

A técnica CV é uma estratégia que consiste em dividir o conjunto de dados para treinamento em dois subgrupos, a saber: (i) conjunto de treinamento, isto é, as observações que são efetivamente empregadas na indução e ajuste dos parâmetros internos; (ii) conjunto de validação, que representam as observações que são utilizadas para desempenhar a capacidade de generalização de modelos. Assim, o conjunto de dados de teste é utilizado unicamente após o modelo já ter sido ajustado e avaliado.

No total, o uso da CV permite na obtenção de um modelo com melhor capacidade de generalização e sugere que os conjuntos de dados (treino, validação e teste) sejam suficientemente representativos para efetuar a previsão. Existem algumas desvantagens dessa estratégia. Por exemplo, alguns dados disponíveis sempre serão usados no ajuste dos parâmetros, enquanto outros nunca serão usados para este fim, pois compõem o conjunto de teste. Além disso, não sabemos de qualquer indicativo sobre como o modelo varia com diferentes dados de treinamento. Uma forma de contornar esses pontos é usar a *k-fold CV* (BENGIO; GRANDVALET, 2005).

A técnica *k-fold CV* consiste em dividir o conjunto de dados para treinamento em k conjuntos. No treinamento são usados $k-1$ conjuntos de dados para indução e ajuste de parâmetros e um outro para a validação. A *k-fold CV* é uma técnica bastante usada para uma melhor capacidade de generalização. O erro é então calculado sobre os k conjuntos e representa a média dos desempenhos dos k conjuntos de validação, chamada de erro de validação cruzada ($\frac{\varepsilon_1+\varepsilon_2+...+\varepsilon_n}{k} = \frac{1}{k}\sum_{i=1}^{k}\varepsilon_n$) (ilustração na Figura 2.10). O conjunto de dados temporais disponível aparece $k-1$ vezes no conjunto de treinamento e uma vez no conjunto de validação em todos os k subconjuntos.

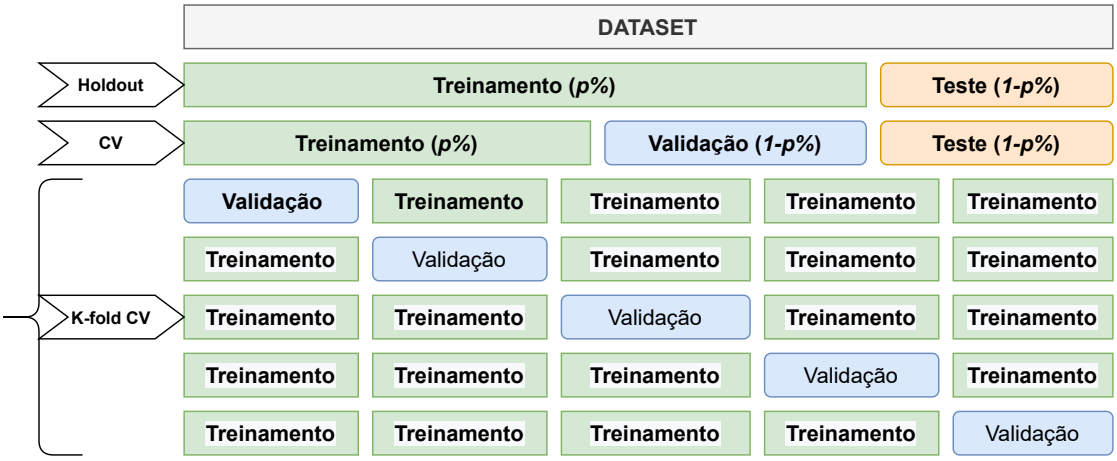


Figura 2.10 – Diferentes técnicas para validação de modelos.

Pela técnica *k-fold CV*, o procedimento consiste em:

a) particionar o conjunto de dados em k subconjuntos, sem sobreposição, de mesmo tamanho.

b) para cada $n = 1, ..., k$, faça o ajuste dos parâmetros do modelo considerando todos os dados, menos aqueles que pertencem ao n-ésimo subconjunto; compute as saídas estimadas do modelo para os dados do n-ésimo subgrupo, assim, calcule o erro de validação.

c) no final, o desempenho do modelo é dado pela média dos erros de validação calculados para cada subconjunto.

Por fim, a *early stopping* é uma técnica de regularização que consiste em interromper o treinamento da rede quando o valor do erro de validação começa a aumentar por p iterações sucessivas. De forma implícita, a técnica *early stopping* controla também a norma do vetor de parâmetros internos, ou seja, $\mathbf{w} = \mathbf{w}^*$, em que $\mathbf{w}^*$ é o vetor peso mínimo, que pode ser obtido a partir do mínimo local ou global (HAYKIN, 2009).

2.2.4 Arquiteturas de Redes GRU

Uma célula GRU é a versão simplificada da célula LSTM padrão e funciona de forma análoga (GREFF *et al.*, 2017). Introduzida por (CHO *et al.*, 2014), a célula GRU combina a porta de esquecimento e a porta de entrada em uma única porta chamada de *porta de atualização*. A arquitetura da GRU permite capturar adaptativamente dependências temporais de sequências de dados sem descartar informações de partes anteriores da sequência. Isso é alcançado através de suas unidades de portas (Figura 2.11).

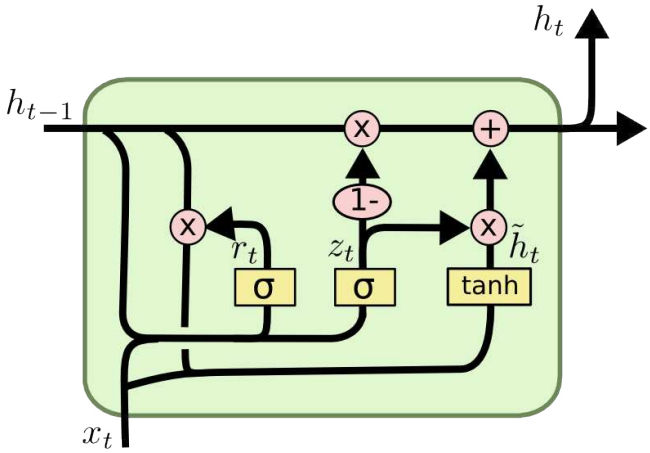


Figura 2.11 – Estrutura interna da célula GRU.
Adaptada de (OLAH, 2015).

As principais simplificações da célula GRU são: (i) há somente um vetor de estado $\mathbf{h}_t$; há único vetor $\mathbf{z}_t$ que controla a porta de esquecimento (*forget gate*) e a porta de entrada (*input gate*), respectivamente; (ii) não há mais a porta de saída (*output gate*). Portanto, há agora o vetor $\mathbf{r}_t$ que toma decisão sobre a parte do vetor de estado $\mathbf{h}_{t-1}$ que deve ser mostrado na camada de saída $\widetilde{\mathbf{h}}_t$.

Com isso, podemos dizer que a célula GRU contém apenas duas portas: (i) a porta de atualização z_t; e (ii) a porta de redefinição (ou reinicialização) r_t. Assim como a célula LSTM, a célula GRU é treinada para filtrar seletivamente qualquer informação irrelevante, e mantendo o que é importante. Matematicamente, a célula GRU funciona da seguinte forma:

$$\mathbf{z}_t = \sigma([\mathbf{w}_{hz}\mathbf{h}_{t-1} + \mathbf{w}_{xz}\mathbf{x}_t] + \mathbf{b}_z) \tag{2.38a}$$

$$\mathbf{r}_t = \sigma([\mathbf{w}_{hr}\mathbf{h}_{t-1} + \mathbf{w}_{xr}\mathbf{x}_t] + \mathbf{b}_r) \tag{2.38b}$$

$$\widetilde{\mathbf{h}}_t = \tanh([\mathbf{w}_{x\widetilde{h}}\mathbf{x}_t + \mathbf{w}_{h\widetilde{h}}(\mathbf{r}_t \bigotimes \mathbf{h}_{t-1})] + \mathbf{b}_{\widetilde{h}}) \tag{2.38c}$$

$$\mathbf{h}_t = \mathbf{z}_t \bigotimes \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \bigotimes \widetilde{\mathbf{h}}_t \tag{2.38d}$$

A célula GRU expõe todo o estado da célula para outras unidades da rede. Ao contrário, a célula LSTM controla a exposição do conteúdo da sua memória. A unidade LSTM tem uma porta de atualização e esquecimento separada, enquanto, a GRU executa ambas as operações juntas por meio da sua porta de reinicialização. No mais, a célula GRU é treinada seguindo o mesmo procedimento que a célula LSTM. A Tabela 2.2 sumariza as diferenças entre os modelos MLP, LSTM e GRU.

Tabela 2.2 – Diferenças entre os modelos MLP, LSTM e GRU.

Modelo	**Tipo**	**Comentários**
MLP	Estático	• Modelo sem recorrência ou direta; • Constrói mapeamento não-recorrente da entrada até a saída; • Treinamento realizado por meio do algoritmo de retro-propagação do erro.
LSTM	Dinâmico	• Modelo com recorrência • Há estado interno para armazenamento de informações; • Existem três portas: porta de esquecimento, porta de entrada, e porta de saída; • LSTM controla a exposição do conteúdo da sua memória; • Treinamento realizado por meio do algoritmo de retro-propagação do erro através do tempo.
GRU	Dinâmico	• Modelo com recorrência; • Há estado interno para armazenamento de informações; • Existem duas portas: porta de atualização e porta de reinicialização; • GRU expõe todo o estado da célula para outras unidades da rede; • Treinamento realizado por meio do algoritmo de retro-propagação do erro através do tempo.

2.3 Métricas da Avaliação do Desempenho

Como já vimos na etapa anterior, a aprendizagem dos modelos apresentados até agora consiste em determinar os valores que levem ao melhor mapeamento entrada-saída possível. Isto dá origem a um problema de otimização linear e não-linear irrestrito, no qual, sem perda de generalização, desejamos minimizar uma **função custo ou função de perda** que expressa uma medida de erro entre as saídas fornecidas pelos modelos e as saídas desejadas.

Na área de previsão de séries temporais é muito comum usar métricas relacionadas à regressão. As métricas comumente usadas para avaliar a magnitude das previsões são: erro quadrático médio (MSE, do inglês *Mean Squared Error*); raiz do erro quadrático médio (RMSE, do inglês Root Mean Squared Error); e erro absoluto médio (MAE, do inglês *Mean Absolute Error*) (HYNDMAN; ATHANASOPOULOS, 2018). A seguir, apresentamos cada uma dessas métricas.

O MAE representa o valor médio da soma das diferenças absolutas entre saídas desejadas e saídas fornecidas pelo modelo:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{\mathbf{y}}(x_i)| \tag{2.39}$$

O MSE determina a média dos quadrados dos erros, ou seja, a diferença quadrática média entre os valores previstos pelo modelo e os valores reais:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{\mathbf{y}}(x_i))^2 \tag{2.40}$$

A RMSE penaliza os termos de erro maiores e tende a se tornar cada vez maior que o MAE para *outliers*, os quais são obtidos a partir do erro quadrático médio:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{\mathbf{y}}(x_i))^2} \tag{2.41}$$

em que $\hat{y}$ é o previsto, y_i é o desejado e N representa o número total de observações. A métrica MAE dá o mesmo peso a todos os erros, enquanto que RMSE tende a penalizar os modelos que cometem erros maiores. Essas métricas possuem a mesma unidade de medida da variável de interesse. Para que as métricas de avaliação de desempenho sejam bem-sucedida, algumas técnicas de pré-processamento podem ser utilizadas.

Embora as métricas MAE, MSE e $RMSE$ são comumente usadas, elas não são adequadas quando for conhecer a correta direção das previsões realizadas (MOOSA; VAZ, 2015). Para um investidor, por exemplo, a direção da previsão é um dos critérios de avaliação do desempenho mais relevantes, em que ele se preocupe mais com a mudança de direção da taxa de câmbio no futuro do que seus valores previstos. Enquanto que os lucros não são observáveis, a precisão da direção das previsões é sugerida como critério de avaliação do desempenho da previsão. A precisão da direção, no caso, está diretamente relacionada à capacidade de *timing* do mercado.

Como métrica da avaliação do desempenho, adicionamos a precisão da direção (*direction accuracy*) que denominamos simplesmente *acurácia* expressa por:

$$Acurácia = \frac{1}{N}\sum_{i=1}^{N} 1_{sign(y_i - y_{i-1}) == sign(\hat{\mathbf{y}}(x_i) - y_{i-1})} \tag{2.42}$$

em que $\hat{y}_i$ representa o valor previsto no tempo i, y_i o valor desejado (valor real) e N representa o número total de observações, $sign(\cdot)$ a função sinal e $\mathbf{1}$ é a função de indicador. A acurácia pode ser usada ainda como indicador para que o investidor adapta-se a sua decisão ao acontecimento futuro do mercado.

3 Previsão da Taxa de Câmbio

Neste capítulo aplicamos os modelos de séries temporais apresentados no Capítulo 2, para as taxas de câmbio R$/US$ e €/US$. Inicialmente, na Seção 3.1, fazemos uma breve exposição da importância de prever a taxa de câmbio. Na Seção 3.2, realizamos uma análise exploratória de dados. Particularmente, apresentamos a fonte de extração de dados e suas trajetórias ao longo do período de tempo, assim como suas respectivas características.

Na Seção 3.3 aplicamos os modelos lineares na previsão da taxa de câmbio. Na Seção 3.4, por sua vez, aplicamos os modelos não-lineares na previsão da taxa de câmbio. Em cada passo, apresentamos a metodologia desenvolvida, o que representa também uma das contribuições desta dissertação. Na Seção 3.5 discutimos os resultados e fazemos a comparação dos resultados para os modelos empregados.

3.1 Importância de Prever a Taxa de Câmbio

Definimos a taxa de câmbio como o número de unidades de moeda nacional necessário para comprar uma unidade de moeda estrangeira. A taxa de câmbio é definida no mercado de câmbio, no qual todos os participantes vendem ou compram divisas. O mercado de câmbio é um mercado líquido, onde os ativos (ações, títulos e outros) são trocados por meio de dinheiro.

No mercado de câmbio comercial a moeda é investida e pode ser vendida ou comprada de acordo com sua variabilidade. A tomada de decisão é feita pelos investidores, e esses podem analisar o momento oportuno para vender ou comprar moeda, avaliando os diferentes fatores que podem causar mudanças no mercado (PATEL *et al.*, 2014). Os investidores investem na moeda que acreditam que irá aumentar o preço em um determinado intervalo temporal. Assim, os investidores ou tomadores de decisão precisam ter uma previsão do preço da moeda. A previsão desse preço é uma tarefa desafiadora, porque as mudanças são causadas pelos fatores internos e externos à economia. A volatilidade da taxa de câmbio causa dificuldade para realizar uma previsão adequada (SOON; BAHARUMSHAH, 2021), (EVANS; LYONS, 2005).

As mudanças que acontecem no mercado de câmbio são, geralmente, rápidas e repetitivas, a menos que acontece uma crise econômica e/ou financeira como foi o caso da crise *subprime* de 2007-2008, ou ainda um evento destruidor raro que pode quebrar

a economia[1]. Na era dos algoritmos de inteligência computacional, as mudanças nos dados ou não-estacionariedades podem ser consideradas nas previsões de dados históricos, pois os modelos de redes neurais usam funções não-lineares para realizar o mapeamento. Muitos desses modelos foram testados para previsão de séries temporais macroeconômicas e financeiras com resultados satisfatórios em termos de desempenho, na maioria dos estudos.

Além disso, tendo em conta a descentralização e a desregulamentação do mercado cambial, as volatilidades no preço da moeda acontecem muito rapidamente, o que pode prejudicar tomadores de decisão como investidores, bancos e os outros participantes do mercado, embora a economia dos países seja altamente dependente de seu valor de moeda no mercado internacional por conta da globalização dos mercados (SOON; BAHARUMSHAH, 2021). Uma boa previsão da taxa de câmbio permite a um tomador de decisão diminuir o volume de risco e evitar grandes perdas e ajudar um país a melhorar sua política econômica, dado que a taxa de câmbio pode afetar o desenvolvimento econômico de um país.

Em resumo, as transações do mercado de câmbio dão uma visão clara do tamanho da importância do mercado monetário, qual o montante de dinheiro está sendo trocado diariamente, afetando o país doméstico e o resto do mundo. Dessa forma, a previsão para taxa de câmbio pode alertar os investidores ou tomadores de decisão sobre o mercado.

Acrescenta-se que, como o mercado de câmbio é fundamentalmente líquido, as informações são públicas e acessíveis a todos os participantes que compartilham as mesmas expectativas. Ainda, a taxa de câmbio torna-se mais dependente dos fatores macroeconômicos e geopolíticos (PATEL *et al.*, 2014).

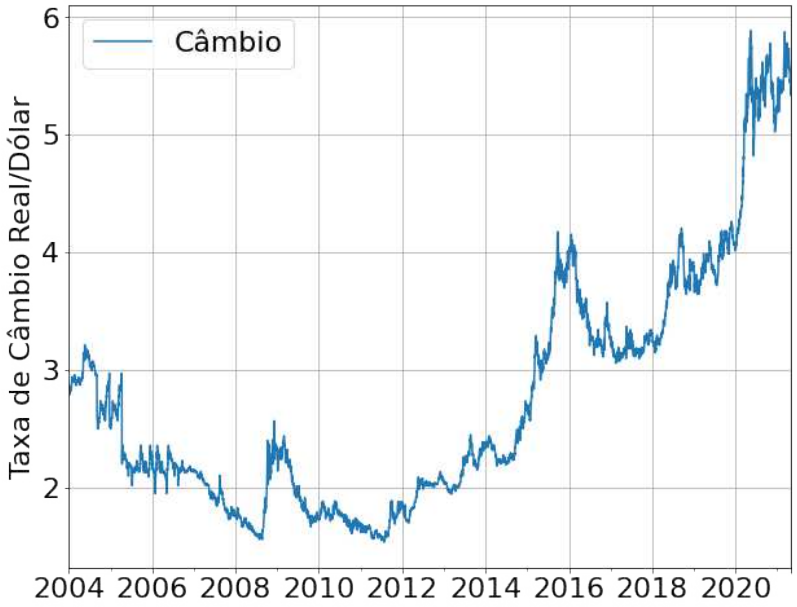
3.2 Dados

Os dados usados neste trabalho foram extraídos do *yahoo finance*, que estão disponíveis publicamente como Taxas de Câmbio BRL=X (R$/US$) e EURUSD=X (€/US$). A ferramenta *yahoo finance* é uma rede destinada ao público que fornece notícias financeiras, mas, também, dados históricos financeiros. Ambas as séries são médias diárias das taxas de câmbio. Os períodos considerados foram de 31 de Dezembro de 2003 a 04 de Maio de 2021 para os dados R$/US$ e de 31 de Dezembro de 2003 a 05 de Maio

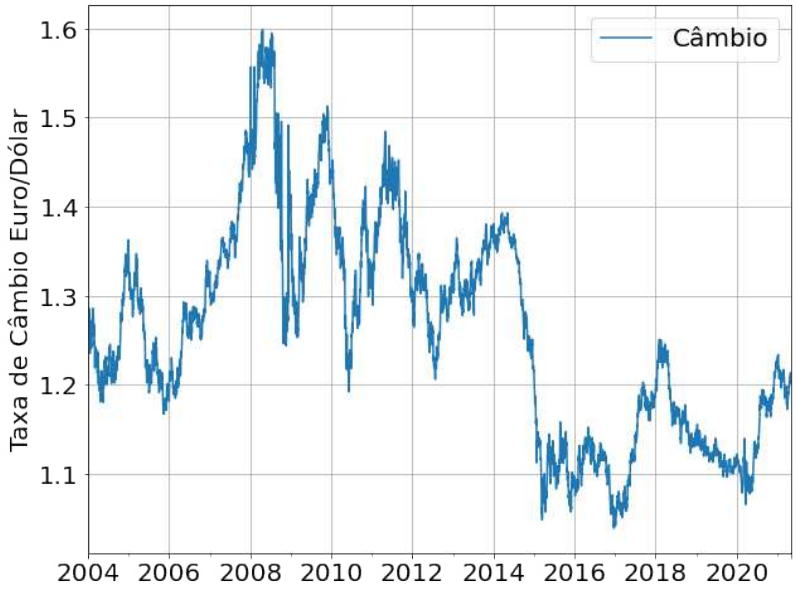
[1] Os inventos imprevistos (ou acontecimento improvável) não são levados em consideração nas previsões econômicas e financeiras. A consideração desses eventos em um modelo de inteligência computacional pode estar sujeita a restrições que podem causar discriminação, cujos resultados finais podem ser catastróficos no mundo social (O'NEIL, 2016). Eventos improváveis, é o que Nassim Nicholas Taleb chama de lógica do Cisne Negro: o impacto do altamente improvável (TALEB, 2007).

de 2021 para os dados €/US$.

As Figuras 3.1(a) e 3.1(b) ilustram a evolução das taxas de câmbio R$/US$ e €/US$, respectivamente. Vale ressaltar que a taxa de câmbio efetiva é definida como uma média das taxas de câmbio e de índices de preços de países com os quais um país doméstico tem relações econômicas, ponderada pelos respectivos pesos nas relações externas (KRUGMAN; OBSTFELD, 2009).



(a)



(b)

Figura 3.1 – Evolução das Séries: (a) Taxa de Câmbio R$/US$ e (b) Taxa de Câmbio €/US$.

Da Figura 3.1(a), observamos picos na evolução da taxa de câmbio R$/US$ no período entre Dezembro de 2003 a Maio de 2021. No início do período nota-se quedas

sucessivas da taxa de câmbio para um patamar em torno de 2,00 R$/US$ em 2004-2005. Em 2008, provocada pela crise financeira de 2007-2008, a um aumento da taxa de câmbio, seguido por outro aumento provocado pela crise política no Brasil no final de 2015 até início de 2016, encerrando o período com uma tendência de aumento sucessivo na taxa de câmbio. A oscilação da taxa de câmbio R$/US$ no período de análise atingiu o valor máximo de R$ 5.88 = US$ 1, em 13 e 14 de Maio de 2020, e um valor mínimo de R$ 1.53 = US$ 1, em 27 de Julho de 2011.

A Figura 3.1(b) indica que desde 31 de Dezembro de 2003 até 05 de Maio de 2021 a taxa de câmbio €/US$ oscilou entre 1.60 e 1.03. Em termos de comparação entre as duas moedas, podemos dizer que a variabilidade da taxa de câmbio R$/US$ é maior que €/US$. Vale ressaltar que a moeda Euro tem variação histórica relativamente mais estável comparativamente ao Real. Esse fato pode fazer com que o desempenho da previsão alcançado pela série R$/US$ seja ligeiramente inferior, ou seja, menos adequado do que o desempenho alcançado pela série €/US$.

Explorando ainda as estatísticas descritivas das duas séries temporais, a Tabela 3.1 traz informações sobre as principais medidas exploratórias (média, mediana, desvio padrão, assimetria, curtose e quartis) dos dois conjuntos de dados utilizados nesta pesquisa.

Tabela 3.1 – Estatísticas para as Séries das Taxas de Câmbio R$/US$ e €/US$

Estatística	**R$/US$**	**€/US$**
Observações	4525	4526
Média	2.7467	1.2599
Mediana	2.2938	1.2582
Desvio padrão	1.0314	0.1211
Assimetria	1.1006	0.3571
Curtose	0.5551	-0.5076
Mínimo	1.5337	1.0390
Máximo	5.8864	1.6018
Primeiro quartil (25%)	1.9695	1.1563
Amplitude (50%)	2.2968	1.2586
Terceiro quartil (75%)	3.2886	1.3454

A assimetria (*skewness*) mede o grau de simetria da curva em relação a distribuição normal, ou seja: $\hat{A}(X) = \frac{1}{(N-1)\hat{\sigma}^3}\sum_{t=1}^{N}(X_t - \hat{\mu})^3$. Para uma série normalmente distribuída, $\hat{A}(X)$ é próxima de zero. Uma assimetria positiva (com $\hat{A}(X) > 0$), significa que a cauda da distribuição está mais para a direita. Uma assimetria negativa (com $\hat{A}(X) < 0$), significa que a cauda da distribuição está mais para a esquerda. Notamos que ambas taxas de câmbio são assimétricas, sendo que a taxa R$/US$ tem uma cauda mais acentuada à direita que a taxa €/US$.

Por sua vez, a medida de curtose (*kurtosis*) avalia o grau de achatamento da curva em relação a distribuição normal, ou seja: $\hat{K}(X) = \frac{1}{(N-1)\hat{\sigma}^4}\sum_{t=1}^{N}(X_t - \hat{\mu})^4$. Se o valor da curtose é igual a zero, temos uma distribuição mesocúrtica. Se o valor for maior que zero, leptocúrtica, correspondendo a uma curva mais pontuda, ou seja, apresenta cauda longa. Se for menor que zero, platicúrtica, ou seja, uma curva mais achatada, com maior variabilidade dos dados. Da Tabela 3.1, notamos que a taxa de câmbio R$/US$ é leptocúrtica enquanto a taxa €/US$ é platicúrtica. Essas medidas são geralmente utilizadas para testar se a amostra de dados provêm de uma distribuição normal. Vejam (MORETTIN, 2008) e (MORETTIN; TOLOI, 2004) para uma discussão mais elaborada. A partir dessas duas medidas podemos concluir que ambas as séries não seguem uma distribuição normal.

Ainda, podemos observar que há uma diferença entre as duas taxas de câmbio em termos de variabilidade, em que a taxa de câmbio R$/US$ é mais volátil que a €/US$. A dinâmica da taxa de câmbio na condução de política econômica tem uso diferente dependendo de características de cada país. O Euro representa característica de uma moeda forte comparado ao Real. Na hierarquia de moedas, por exemplo, o Real representa as características de uma moeda periférica que têm especificidades diferentes quando na condução de política econômica. Nota-se que uma moeda periférica é aquela que não exerce nenhuma de suas funções (meio de troca, reserva de valor e unidade de conta) em âmbito internacional (CONTI *et al.*, 2014).

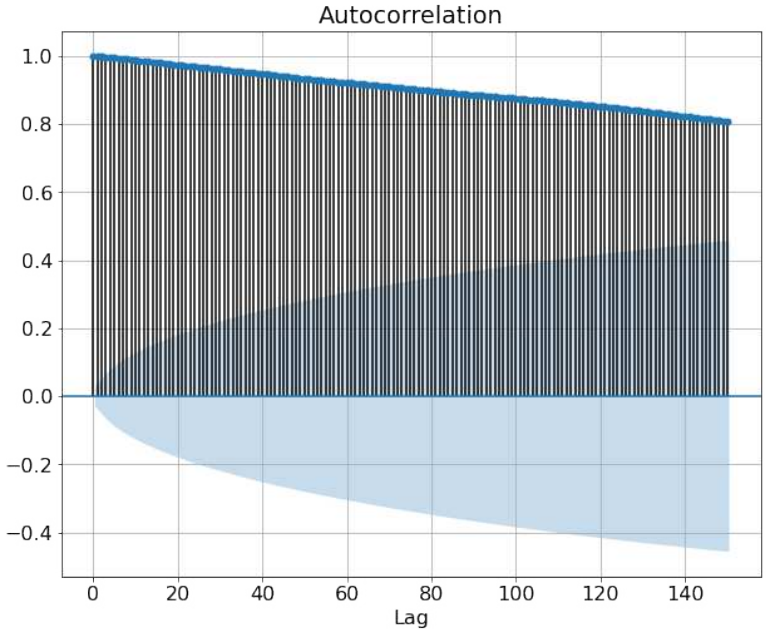
Como já destacamos, a curtose da taxa de câmbio R$/US$ é alta, mostrando a não-normalidade da série, com uma cauda da distribuição mais para direita. A curtose da taxa de câmbio €/US$ é negativa, o que indica uma distribuição platicúrtica, com a cauda da distribuição centralizada.

Neste trabalho, após analisar as propriedades estatísticas e fatos estilizados (tendência, sazonalidade, não-linearidade, heteroscedasticidade) nos dados, o objetivo é prever os preços diários das moedas, sendo que estes são não-lineares, com existência de longa memória. Destaca-se que a incorporação das dinâmicas não-lineares no ajuste da taxa de câmbio pode levar a melhor desempenho.

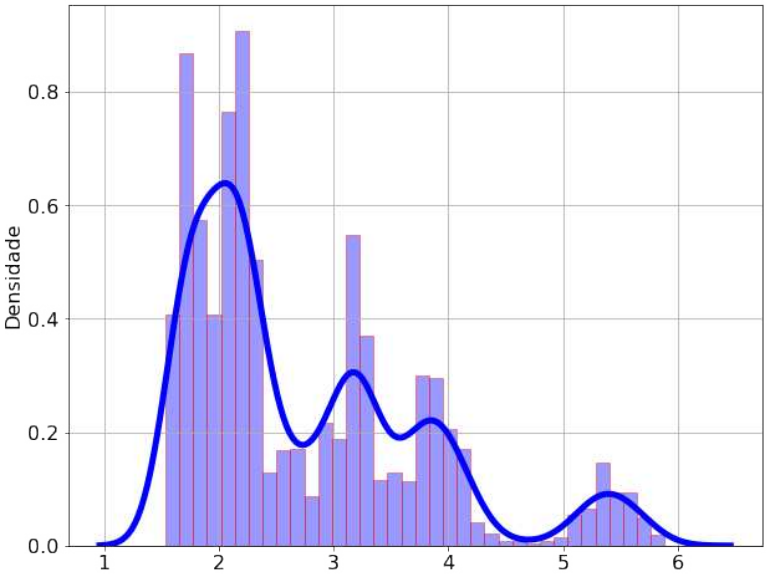
A previsão dos preços diários da moeda foi obtida para 1 e 7 passos à frente. Para estes horizontes, calculamos os erros obtidos para fim de comparação. Com o objetivo de compararmos o desempenho dos modelos ajustados, foram retirados os últimos 150 dias, ou seja, os modelos foram ajustados em um período distinto do período de análise do desempenho que corresponderam aos últimos 5 meses das amostras. A comparação dos modelos é feita a partir das métricas de avaliação do desempenho apresentadas no Capítulo 2.

3.3 Aplicação dos Modelos Lineares

Nesta seção apresentamos a metodologia e os principais resultados obtidos a partir dos modelos lineares, ajustados para as séries das taxas de câmbio R$/US$ e €/US$. As Figuras 3.2(a) e 3.2(b) mostram a função de autocorrelação (FAC) e o histograma dos preços de câmbio R$/US$, respectivamente. Essa série é não estacionária em sua média. A função de autocorrelação indica que há persistência no tempo, levando à conclusão de que a série de câmbio R$/US$ tem comportamento de memória longa.



(a)

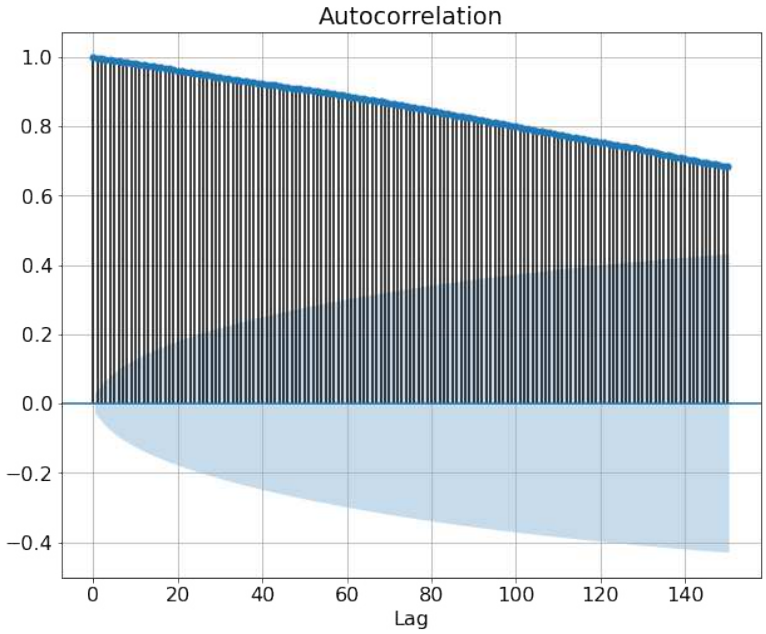


(b)

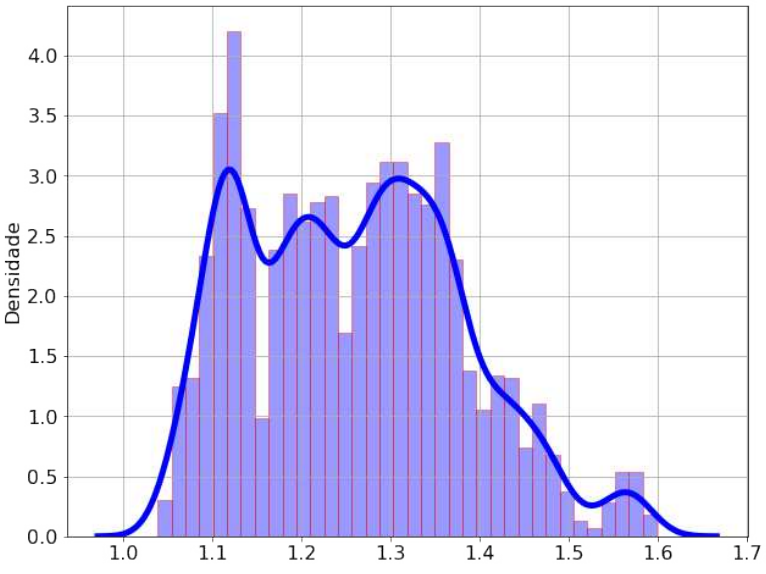
Figura 3.2 – (a) Função de autocorrelação e (b) Histograma dos preços diários de R$/US$.

As Figuras 3.3(a) e 3.3(b) ilustram a função de autocorrelação (FAC) e o

histograma dos preços de câmbio €/US$, respectivamente. Essa série também é não estacionária em sua média. A função de autocorrelação indica que há persistência no tempo, levando à conclusão de que a série de câmbio €/US$ tem, também, a característica de memória longa.



(a)



(b)

Figura 3.3 – (a) Função de autocorrelação e (b) Histograma dos preços diários de €/US$.

3.3.1 ARIMA

A Figura 3.4 mostra o procedimento empregado para a previsão das taxas de câmbio R$/US$ e €/US$. Foi utilizada a linguagem de programação *Python* para estimar os modelos e para a previsão.

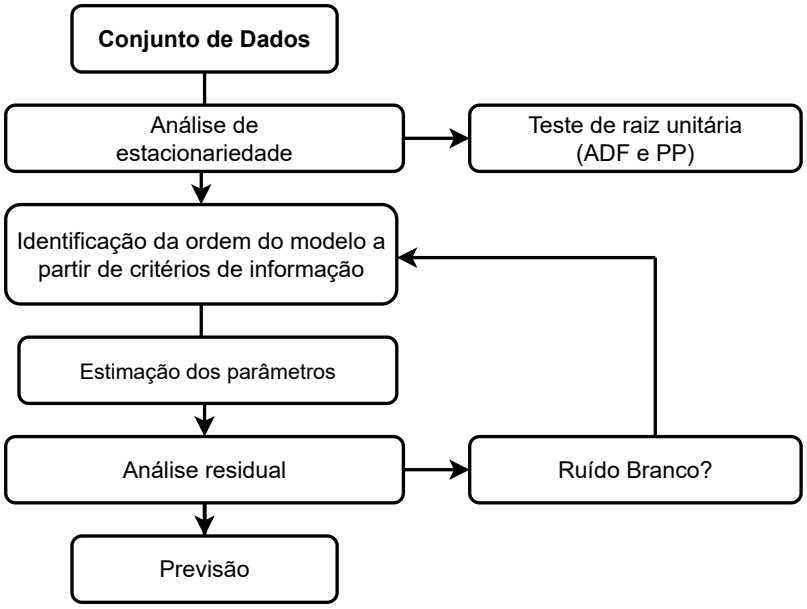


Figura 3.4 – Procedimento para previsão dos modelos lineares.

Para a análise de estacionariedade das séries, neste trabalho empregamos dois testes de raiz unitária: teste de Dickey Fuller Aumentado (ADF) e teste de Phillips-Perron (PP) (MORETTIN, 2008), (MORETTIN; TOLOI, 2004). A hipótese nula de ambos testes é que a série tem uma raiz unitária (ou seja, não estacionária), e a alternativa é que a série não tem uma raiz unitária, portanto, é estacionária. O Apêndice A descreve os testes de raiz unitária. Os resultados dos testes são apresentados na Tabela 3.2.

O software RStudio é utilizado para verificar a presença da raiz unitária. O pacote *urca* que tem implementada a função *ur.adf* e *ur.pp* permite obter resultados de ambos testes. Os parâmetros importantes da função são: (i) conjunto de dados; (ii) escolha do modelo (constante, tendência e sem termos determinísticos); (iii) número de defasagens, a partir de critérios de informação; (iv) teste sobre os coeficientes do modelo; (v) teste sobre as estatísticas do modelo.

Tabela 3.2 – Testes de raiz unitária para identificação da ordem de integração

Teste de Dickey-Fuller Aumentado (ADF)							
Série	Termo determinístico	Defasagem	Valor do teste	Valor crítico			p-value
				1%	5%	10%	
Câmbio (R$/US$)	Constante	2	0.5621	-3.4317	-2.8621	-2.5671	0.9866
	Constante e Tendência	2	-1.5550	-3.9607	-3.4114	-3.1276	0.8095
	Nenhuma	2	1.3090	-2.5662	-1.9410	-1.6167	0.9514
△ Câmbio (R$/US$)	Constante	1	-53.9001	-3.4317	-2.8621	-2.5671	0.000
	Constante e Tendência	1	-53.9581	-3.9607	-3.4114	-3.1276	0.000
	Nenhuma	1	-53.8943	-2.5662	-1.9410	-1.6167	0.000
Câmbio (€/US$)	Constante	2	-2.0076	-3.4317	-2.8621	-2.5671	0.2832
	Constante e Tendência	2	-2.5485	-3.9607	-3.4114	-3.1276	0.3041
	Nenhuma	2	-0.3198	-2.5662	-1.9410	-1.6167	0.5683
△ Câmbio (€/US$)	Constante	1	-56.2577	-3.4317	-2.8621	-2.5671	0.000
	Constante e Tendência	1	-56.2520	-3.9607	-3.4114	-3.1276	0.000
	Nenhuma	1	-56.2638	-2.5662	-1.9410	-1.6167	0.000
Teste de Phillips-Perron (PP)							
Câmbio (R$/US$)	Constante	3	0.476	-3.43	-2.86	-2.57	0.984
	Constante e Tendência	3	-1.635	-3.96	-3.41	-3.13	0.779
	Nenhuma	3	1.231	-2.57	-1.94	-1.62	0.944
△ Câmbio (R$/US$)	Constante	2	-78.695	-3.43	-2.86	-2.57	0.000
	Constante e Tendência	2	-78.776	-3.96	-3.41	-3.13	0.000
	Nenhuma	1	-78.687	-2.57	-1.94	-1.62	0.000
Câmbio (€/US$)	Constante	2	-2.15	-3.43	-2.86	-2.57	0.225
	Constante e Tendência	2	-2.741	-3.96	-3.41	-3.13	0.219
	Nenhuma	2	-0.311	-2.57	-1.94	-1.62	0.572
△ Câmbio (€/US$)	Constante	1	-84.413	-3.43	-2.86	-2.57	0.000
	Constante e Tendência	1	-84.404	-3.96	-3.41	-3.13	0.000
	Nenhuma	1	-84.423	-2.57	-1.94	-1.62	0.000

Da Tabela 3.2, notamos que as duas séries são estacionárias tomando a primeira diferença, o que significa que a ordem de integração do modelo ARIMA(p,d,q) é unitária ($d = 1$).

A identificação do modelo adequado é a fase mais crítica do processo. Para identificar a ordem do modelo ARIMA utilizamos a biblioteca *pmdarima* que tem implementada a função *auto-arima* a qual permite selecionar um conjunto de modelos, usando um número máximo de defasagens. Consideramos o critério de informação *Akaike* (AIC) para escolher o modelo adequado: que é baseado na variância estimada dos erros e no número de parâmetros a serem ajustados. O objetivo neste critério é considerar o modelo que apresentar o menor AIC. Os parâmetros importantes da função são: (i) conjunto de dados de ajuste; (ii) valores iniciais das ordens p e q; (iii) a ordem de integração; (iv) os valores máximos para p e q. Neste trabalho, tomamos p e q de 0 a 10, respectivamente, para a escolha do modelo, de acordo com o AIC.

Após identificar o modelo adequado, passamos a estimar seus parâmetros. O método usado para estimação dos parâmetros de forma consistente foi o método de máxima verossimilhança. A Tabela 3.3 apresenta diversos modelos candidatos ajustados, juntos com o critério de informação AIC e os parâmetros do modelo escolhido.

Tabela 3.3 – Seleção dos modelos ARIMA

Taxa de câmbio R$/US$					**Taxa de câmbio €/US$**				
Modelo	Parâmetro	Desvio-padrao	Prob.	AIC	Modelo	Parâmetro	Desvio-padrão	Prob.	AIC
ARIMA(0,1,0)				-12726.858	ARIMA(0,1,0)				-27643.173
ARIMA(1,1,0)				-14332.062	ARIMA(1,1,0)				-27892.138
ARIMA(0,1,1)				-16005.376	ARIMA(0,1,1)				-27645.157
ARIMA(2,1,0)				-15062.128	ARIMA(2,1,0)				-27923.175
ARIMA(0,1,2)				-16363.636	ARIMA(0,1,2)				-27922.61
ARIMA(1,1,1)				-16243.533	ARIMA(1,1,2)				-27923.175
ARIMA(2,1,1)				-16351.102	ARIMA(2,1,1)				-27817.827
ARIMA(2,1,2)				-15062.128	ARIMA(2,1,2)				-27542.278
ARIMA(1,1,1)				**-16395.613**	**ARIMA(1,1,1)**				**-27925.526**
	$\hat{\alpha_1}$= 0.144	0.058	0.013			$\hat{\alpha_1}$= 0.086	0.029	0.003	
	$\hat{\beta_1}$= -0.279	0.057	0.000			$\hat{\beta_1}$=-0.344	0.03	0.000	

Observando a Tabela 3.3, os modelos adequados segundo o critério de informação AIC são: ARIMA(1,1,1) da série de câmbio R$/US$ e ARIMA(1,1,1) da série de câmbio €/US$. Nota-se que o ajuste é efetuado no conjunto de $N-150$ amostras, ou seja, 4.375 amostras para a taxa de câmbio R$/US$ e 4.376 para €/US$.

Utilizamos a biblioteca *statsmodels* que fornece a capacidade de ajustar o modelo ARIMA da seguinte forma: definimos o modelo chamando a função *ARIMA()*, com os argumentos para os parâmetros p = 1, d = 1 e q = 1 e o método de máxima verossimilhança como método de estimação. O modelo é ajustado no conjunto de dados do treinamento a partir da função *fit*. A previsão é realizada com a função *predict()*, indicando o horizonte temporal definido.

Logo, os modelos ARIMA(1,1,1) para as séries das taxas de câmbio R$/US$ e €/US$ são escritos, respectivamente, como:

$$\hat{y}_t = 0,144x_{t-1} - 0,279\hat{\varepsilon}_{t-1} \tag{3.1}$$

$$\hat{y}_t = 0,086x_{t-1} - 0,344\hat{\varepsilon}_{t-1} \tag{3.2}$$

Para verificar a adequação dos modelos, o resíduo $\hat{\varepsilon}_t$ é calculado como:

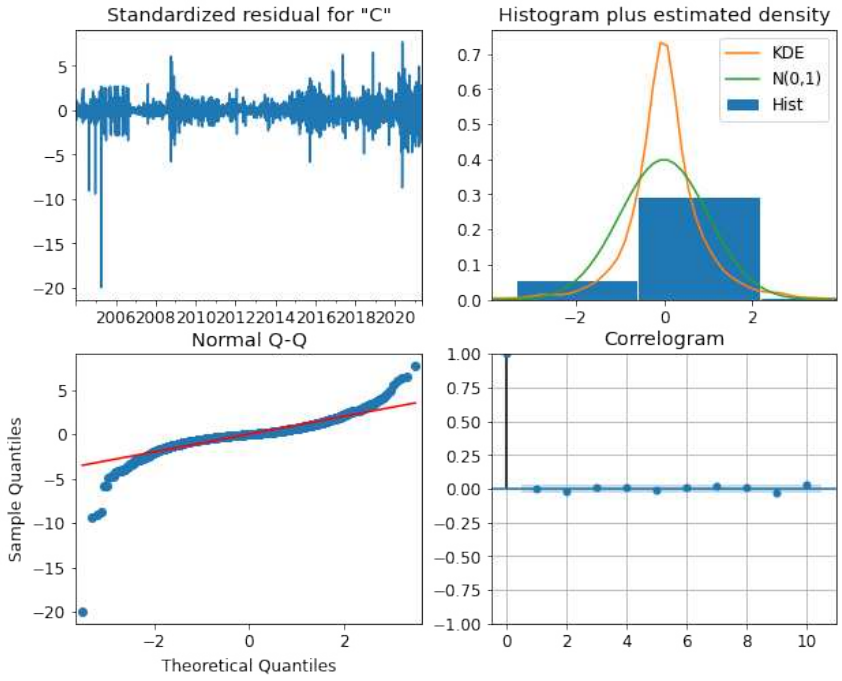
$$\hat{\varepsilon}_t = y_t - \hat{y}_t \tag{3.3}$$

em que $t = 1, \cdots, N$. O próximo passo é a verificação de $\hat{\varepsilon}_t$, ou seja, verificar se $\hat{\varepsilon}_t$ segue um ruído branco com média zero e variância constante.

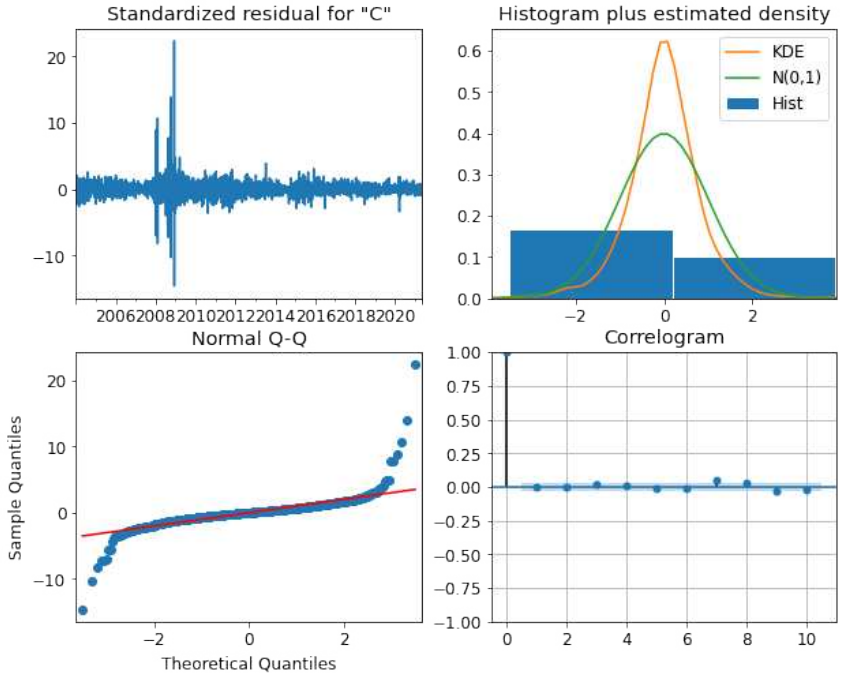
Com base nos resíduos estimados foram obtidos os histogramas das distribuições com a função densidade ajustada, gráfico de normal-plot e a função de autocorrelação (Figura 3.5).

Como podemos observar, nas duas séries das taxas de câmbio R$/US$ e €/US$ os histogramas de densidade mostram resultados de resíduos próximos de uma distribuição normal com média zero e variância unitária. Da mesma forma, o correlograma indica que os resíduos são não correlacionados. O modelo ARIMA(1,1,1) ajustado a partir da série de câmbio R$/US$ conseguiu capturar de melhor forma as informações contidas nesse conjunto de dados.

O gráfico de normal-plot também é utilizado para avaliar a normalidade dos resíduos. Os eixos são construídos contrastando os quantis teóricos de uma distribuição normal, com os quantis observados a partir do conjunto de dados de treinamento (ajuste).



(a)



(b)

Figura 3.5 – (a) Diagnósticos do modelo ARIMA(1,1,1) ajustado na série da taxa de câmbio R$/US$ e (b) diagnósticos do modelo ARIMA(1,1,1) ajustado na série da taxa de câmbio €/US$.

Após diagnósticos dos modelos, realizamos as previsões um passo à frente. Isto é, realizamos previsões estáticas fora da amostra de ajuste para as duas séries.

As Figuras 3.6 e 3.7 ilustram as previsões, um passo à frente, para as taxas de câmbio R$/US$ e €/US$, respectivamente.

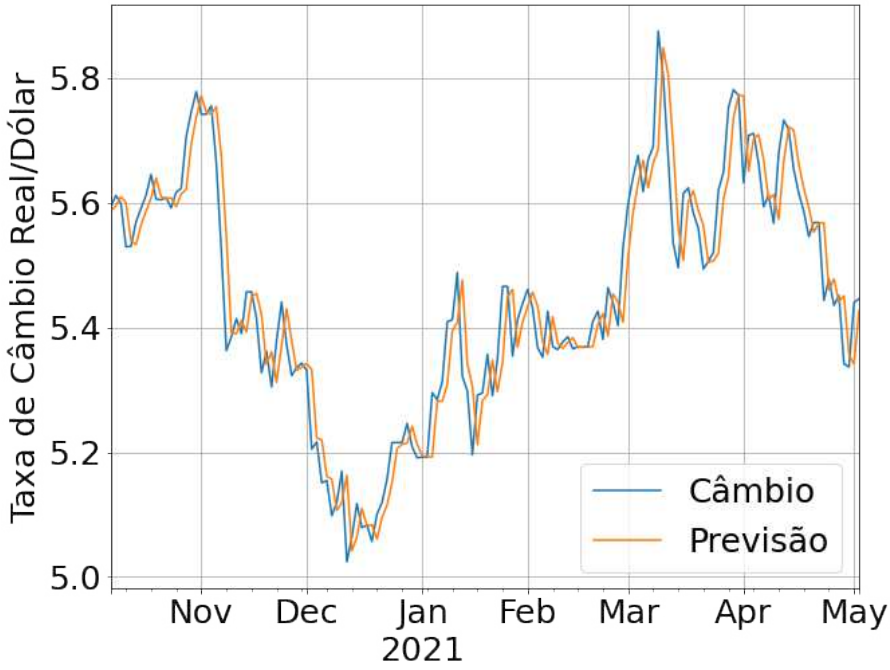


Figura 3.6 – Previsão gerada pelo modelo ARIMA(1,1,1): taxa de câmbio R$/US$. São os últimos 150 dias (07/10/2020 a 04/05/2021) que ilustramos juntos aos valores previstos.

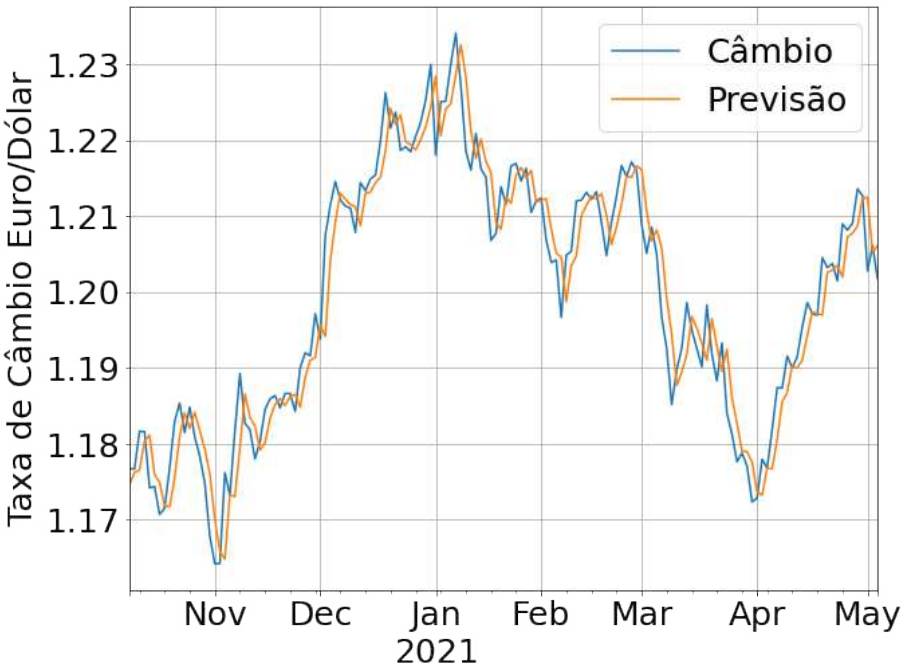


Figura 3.7 – Previsão gerada pelo modelo ARIMA(1,1,1): taxa de câmbio €/US$. São os últimos 150 dias (08/10/2020 a 05/05/2021) que ilustramos juntos aos valores previstos.

3.3.2 ARFIMA

Nesta seção iremos realizar previsões nas mesmas bases de dados usando o modelo ARFIMA. O uso dessa família de modelo considera que a série temporal analisada

tem um comportamento de memória longa.

O procedimento para atingir a previsão é o mesmo apresentado na Figura 3.4. As linguagens de programação *Python* e *RStudio* foram utilizadas para mitigar a tarefa da previsão. Uma classe de transformador *FracDiff* que realiza a diferenciação fracionária foi utilizada para a escolha dos parâmetros d, p e q, relacionados à ordem do modelo.

O software RStudio foi utilizado para verificar a presença da raiz unitária. O pacote *fracdiff* que tem implementada a função *fracdiff.adf* e *fracdiff.pp* permite obter resultado dos testes ADF e PP. Os parâmetros importantes da função são: (i) conjunto de dados; (ii) escolha do modelo (constante, tendência e sem termos determinísticos); (iii) número de defasagens; (iv) teste sobre os coeficientes do modelo; (v) teste sobre as estatísticas do modelo.

Após encontrar a ordem da integração fracionária nas duas séries em estudo, verificamos ainda as estatísticas dos testes de raiz unitária. A Tabela 3.4 apresenta os resultados do teste de Dikey Fuller Aumentado (ADF) e teste de Phillips-Perron (PP) (MORETTIN; TOLOI, 2004). A hipótese nula dos testes foi rejeitada, pois, as duas séries de câmbio são fraccionadamente estacionárias.

Tabela 3.4 – Testes de raiz unitária para identificação da ordem de integração fracionária

Teste de Dickey-Fuller Aumentado (ADF)							
Série	Termo determinístico	Defasagem	Valor do teste	Valor crítico			p-value
				1%	5%	10%	
Câmbio (R$/US$)	Constante	2	0.5621	-3.4317	-2.8621	-2.5671	0.9866
	Constante e Tendência	2	-1.5550	-3.9607	-3.4114	-3.1276	0.8095
	Nenhuma	2	1.3090	-2.5662	-1.9410	-1.6167	0.9514
△ fracdif Câmbio (R$/US$)	Constante	1	-13.69377	-3.4322	-2.86236	-2.56720	0.000
	Constante e Tendência	1	-17.33861	-3.96135	-3.41174	-3.12778	0.000
	Nenhuma	1	-8.43646	-2.56637	-1.94107	-1.61674	0.000
Câmbio (€/US$)	Constante	2	-2.0076	-3.4317	-2.8621	-2.5671	0.2832
	Constante e Tendência	2	-2.5485	-3.9607	-3.4114	-3.1276	0.30414
	Nenhuma	2	-0.3198	-2.5662	-1.9410	-1.6167	0.5683
△ fracdif Câmbio (€/US$)	Constante	1	-18.57416	-3.43226	-2.86238	-2.56722	0.000
	Constante e Tendência	1	-18.73509	-3.96142	-3.41177	-3.1278	0.000
	Nenhuma	1	-7.32213	-2.56639	-1.94108	-1.61674	0.000
Teste de Phillips-Perron (PP)							
Câmbio (R$/US$)	Constante	2	0.476	-3.43	-2.86	-2.57	0.984
	Constante e Tendência	2	-1.635	-3.96	-3.41	-3.13	0.779
	Nenhuma	2	1.231	-2.57	-1.94	-1.62	0.944
△ fracdif Câmbio (R$/US$)	Constante	1	-18.488	-3.43	-2.86	-2.57	0.000
	Constante e Tendência	1	-24.044	-3.96	-3.41	-3.13	0.000
	Nenhuma	1	-10.777	-2.57	-1.94	-1.62	0.000
Câmbio (€/US$)	Constante	2	-2.15	-3.43	-2.86	-2.57	0.225
	Constante e Tendência	2	-2.741	-3.96	-3.41	-3.13	0.219
	Nenhuma	2	-0.311	-2.57	-1.94	-1.62	0.572
△ fracdif Câmbio (€/US$)	Constante	1	-26.692	-3.43	-2.86	-2.57	0.000
	Constante e Tendência	1	-26.963	-3.96	-3.41	-3.13	0.000
	Nenhuma	1	-9.298	-2.57	-1.94	-1.62	0.000

Após a identificação da ordem d fracionária do modelo, passamos para a próxima etapa que consiste em ajustar diferentes modelos concorrentes a partir da escolha dos parâmetros p e q dos modelos, indicado pelo critério de informação Akaike (AIC).

Utilizamos o método de máxima verossimilhança para ajuste dos parâmetros. A Tabela 3.5 apresenta os modelos ajustados e o modelo selecionado para as duas séries temporais.

Tabela 3.5 – Seleção dos modelos ARFIMA

Taxa de câmbio R$/US$					**Taxa de câmbio €/US$**				
Modelo	Parâmetro	Desvio-padrao	Prob.	AIC	Modelo	Parâmetro	Desvio-padrão	Prob.	AIC
ARFIMA(0,d,0)	$\hat{d}$=0.1			-3.6133586	ARFIMA(0,d,0)	$\hat{d}$=0.1			-5.5647930
ARFIMA(1,d,0)	$\hat{d}$=0.2			-3.6129028	ARFIMA(1,d,0)	$\hat{d}$=0.2			-5.5190449
ARFIMA(0,d,1)	$\hat{d}$=0.25			-3.2680173	ARFIMA(0,d,1)	$\hat{d}$=0.25			-5.9059252
ARFIMA(2,d,0)	$\hat{d}$=0.28			-3.6104911	ARFIMA(2,d,0)	$\hat{d}$=0.28			-6.3943010
ARFIMA(0,d,2)	$\hat{d}$=0.3			-3.6103383	ARFIMA(0,d,2)	$\hat{d}$=0.3			-6.3938777
ARFIMA(1,d,2)	$\hat{d}$=0.35			-3.5740352	ARFIMA(1,d,2)	$\hat{d}$=0.35			-6.3943010
ARFIMA(2,d,1)	$\hat{d}$=0.4			-3.609099	ARFIMA(2,d,1)	$\hat{d}$=0.4			-6.3964284
ARFIMA(2,d,2)	$\hat{d}$=0.45			-3.2587163	ARFIMA(1,d,1)	$\hat{d}$=0.45			-5.9227986
ARFIMA(1,d,2)	$\hat{\mathbf{d}} = \mathbf{0.481}$			**-3.6134865**	**ARFIMA(2,d,2)**	$\hat{\mathbf{d}} = \mathbf{0.46}$			**-6.397386**
	$\hat{\alpha_1}$= 0.1262	0.00252	0.000			$\hat{\alpha_1}$= 0.0181	0.004965	0.00024	
	$\hat{\beta_1}$= -0.0502	0.0377	0.000			$\hat{\alpha_2}$= 0.9622	0.005394	0.000	
	$\hat{\beta_2}$= 0.0556	0.0251	0.000			$\hat{\beta_1}$=0.74887	0.000	0.000	
						$\hat{\beta_2}$=-0.23471	0.000292	0.000	

Da Tabela 3.5, o modelo ajustado para a taxa de câmbio R$/US$ é o ARFIMA(1,0.481,2) e para série da taxa de câmbio €/US$, o modelo adequado é o ARFIMA(2,0.46,2). A previsão é realizada a partir desses dois modelos. Nota-se que o ajuste é efetuado no conjunto de $N - 150$ amostras, ou seja, 4.375 amostras para a taxa de câmbio R$/US$ e 4.376 para €/US$.

Utilizamos o pacote *rugarch* que fornece a capacidade de ajustar o modelo ARFIMA da seguinte forma: definimos o modelo a partir da função *ARFIMA()*, e os argumentos dos parâmetros p, d e q escolhidos (Tabela 3.5) e informando o método de máxima verossimilhança como método de estimação. O modelo é ajustado no conjunto de dados do treinamento a partir da função *fit*. A previsão é realizada com a função *predict()*, indicando o horizonte temporal definido.

Logo, os modelos ARFIMA(1,0.481,2) ajustado para a série da taxa de câmbio R$/US$ e ARFIMA(2,0.46,2) ajustado para série €/US$ são representados, respectivamente, como:

$$\hat{y}_t = 0,1262(1-\Psi)^{0.481}x_{t-1} - 0,0502\hat{\varepsilon}_{t-1} + 0,0556\hat{\varepsilon}_{t-2} \tag{3.4}$$

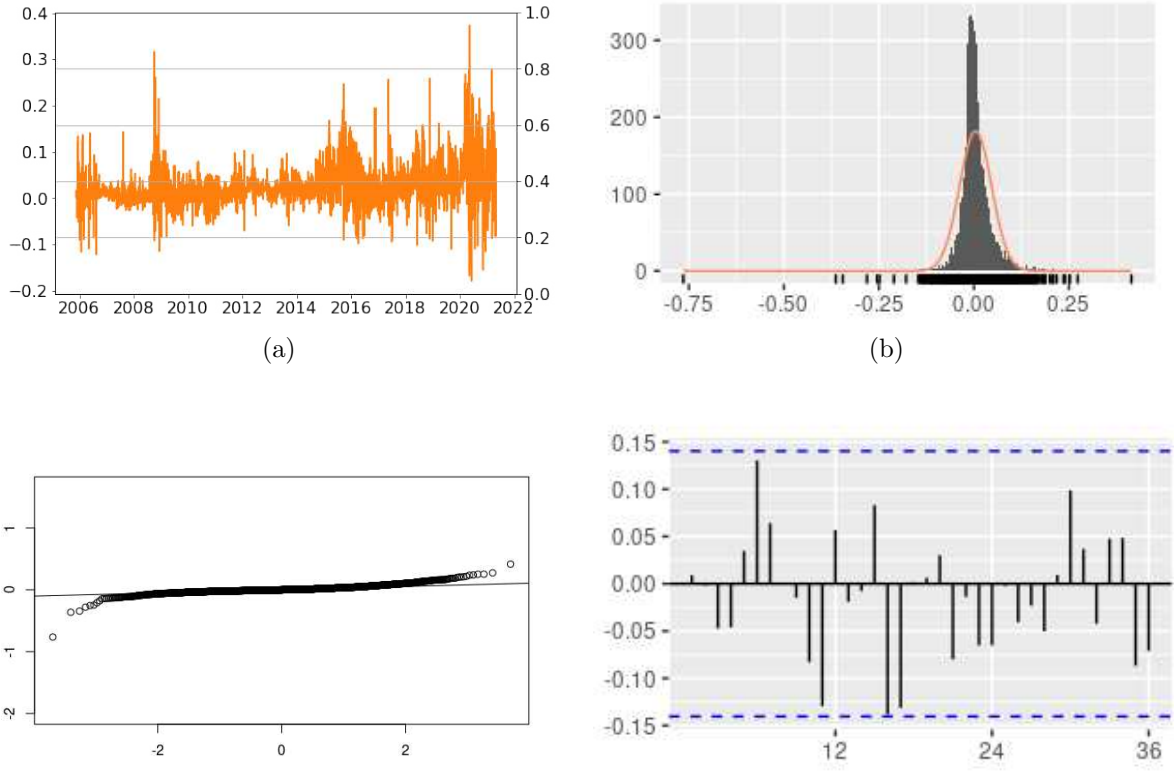
$$\hat{y}_t = 0,0181(1-\Psi)^{0.46}x_{t-1} + 0,9622(1-\Psi)^{0.46}x_{t-2} + 0,74887\hat{\varepsilon}_{t-1} - 0,23471\hat{\varepsilon}_{t-2} \tag{3.5}$$

A adequação dos modelos é feita a partir da análise do resíduo. O modelo adequado é aquele em que os resíduos são ruído branco, ou seja, $\hat{\varepsilon}_t = (y_t - \hat{y}_t) \sim i.id(0, \sigma_\varepsilon^2)$.

Os diagnósticos das estimativas são apresentados nas Figuras 3.8 e 3.9. Com base nos resíduos estimados foram obtidos o histograma da distribuição com a função densidade ajustada, gráfico de normal-plot e a função de autocorrelação.

Um modelo é bem ajustado quando os resíduos seguem características de ruído branco. Como podemos observar, nas duas séries das taxas de câmbio R$/US$ e €/US$ os histogramas de densidade mostram resultados de resíduos próximos de uma distribuição normal com média zero e variância unitária. Da mesma forma, os correlogramas indicam que os resíduos são não correlacionados.

Os gráficos de normal-plot também são utilizados para avaliar a normalidade dos resíduos. Os eixos são construídos contrastando os quantis teóricos de uma distribuição normal, com os quantis observados a partir do conjunto de dados de treinamento. Quanto mais os pontos se comportam em cima da reta, mais próxima é a distribuição conjunta de dados a uma distribuição normal.



(a) (b)

(c) (d)

Figura 3.8 – Diagnóstico do modelo ARFIMA(1,0.481,2) ajustado na série: (a) taxa de câmbio R$/US$, (b) histograma da distribuição com a função densidade ajustada, (c) gráfico de normal-plot, e (d) função de autocorrelação.

Observamos que os resíduos obtidos dos modelos ARFIMA, apresentam um comportamento mais adequado que os resíduos obtidos dos modelos ARIMA.

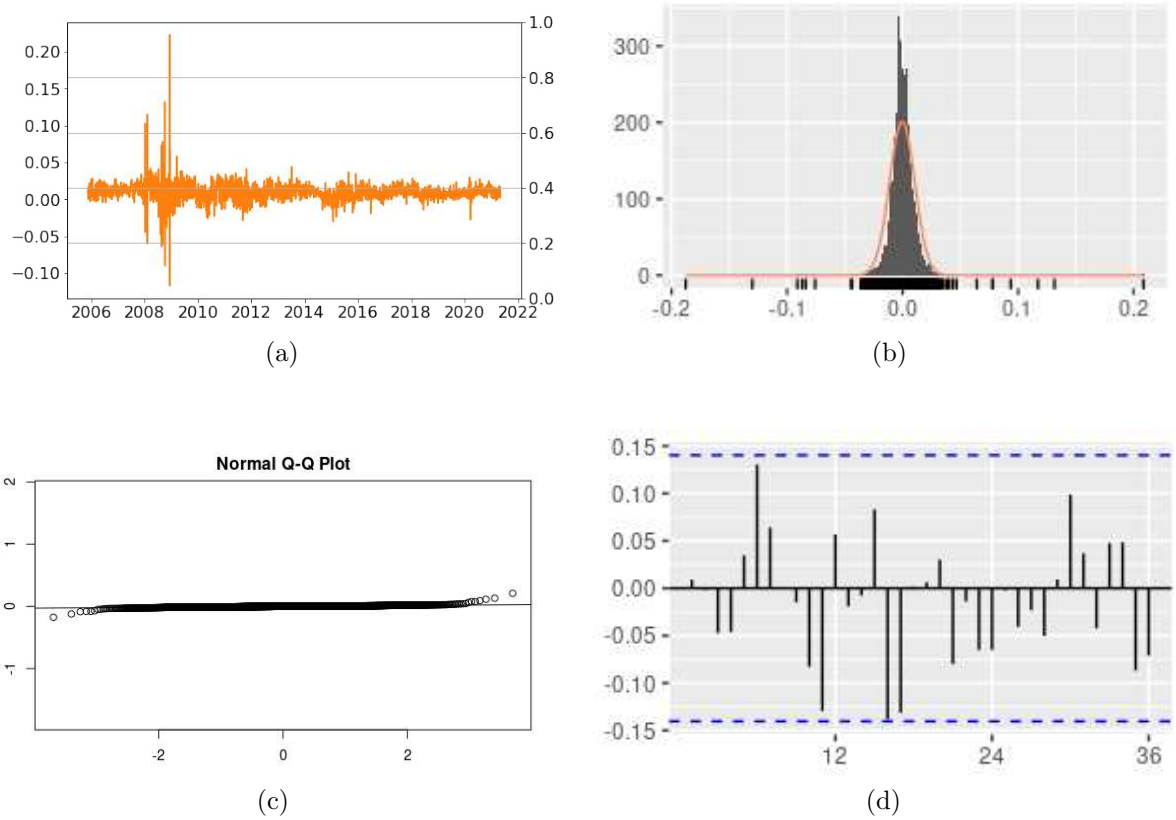


Figura 3.9 – Diagnóstico do modelo ARFIMA(2,0.46,2) ajustado na série: (a) taxa de câmbio €/US$, (b) histograma da distribuição com a função densidade ajustada, (c) gráfico de normal-plot, e (d) função de autocorrelação.

Da Figura 3.9, além do histograma da distribuição e a função de autocorrelação, o gráfico de normal-plot dos resíduos indica que os pontos estão bem comportados em cima da reta, o que significa que os resíduos aproximam-se de uma distribuição normal.

A próxima etapa após análise dos modelos, é a realização das previsões. A Figura 3.10 apresenta os resultados das previsões geradas um passo à frente para o modelo ARFIMA(1,0.481,2) na série da taxa de câmbio R$/US$.

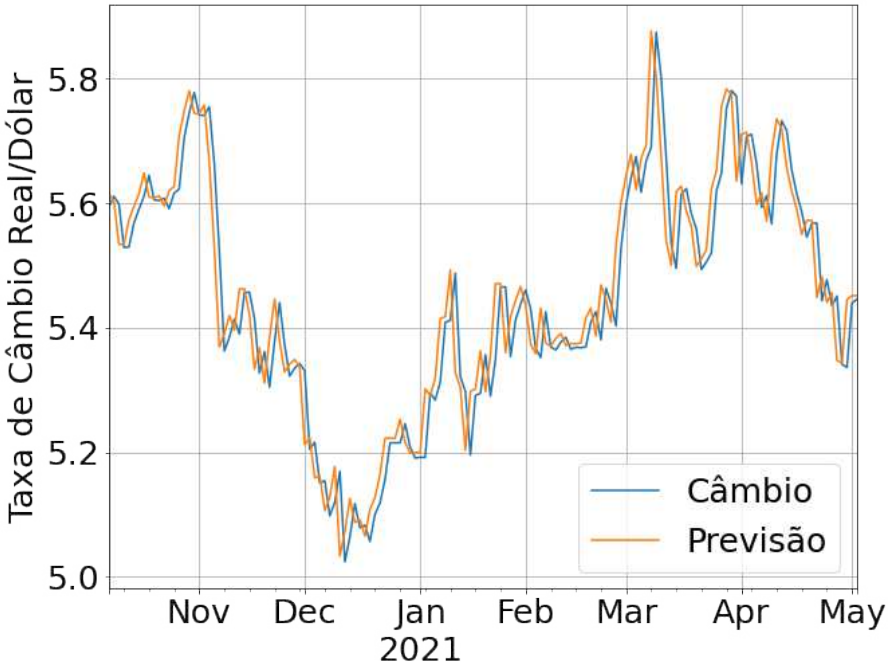


Figura 3.10 – Previsão gerada pelo modelo ARFIMA(1,0.481,2): taxa de câmbio R$/US$. São os últimos 150 dias (07/10/2020 a 04/05/2021) que ilustramos juntos aos valores previstos.

Por sua vez, a Figura 3.11 apresenta os resultados das previsões um passo à frente geradas pelo modelo ARFIMA(2,0.46,2) da série de câmbio €/US$.

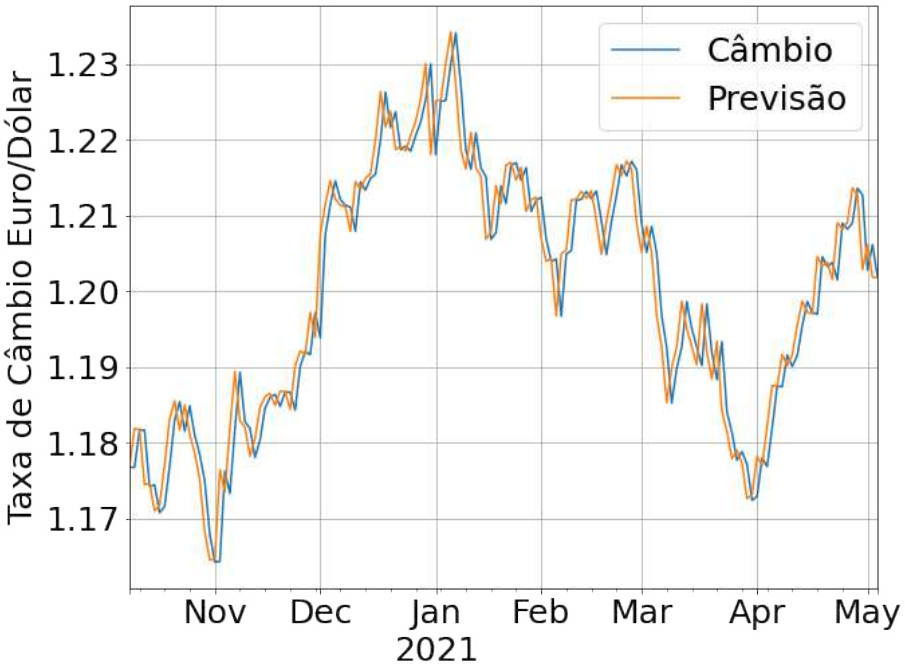


Figura 3.11 – Previsão gerada pelo modelo ARFIMA(2,0.46,2): taxa de câmbio €/US$. São os últimos 150 dias (08/10/2020 a 05/05/2021) que ilustramos juntos aos valores previstos.

3.4 Aplicação dos Modelos Não-Lineares

Nesta seção aplicamos os modelos não-lineares nas duas séries das taxas de câmbio. Inicialmente, um pré-processamento de dados é realizado, o qual é muito utilizado no treinamento das redes neurais para avaliação da capacidade de generalização desses modelos (HAYKIN, 2009). O pré-processamento é um conjunto de etapas que envolvem preparação, organização e estruturação dos dados. Trata-se de uma etapa fundamental que precede a realização de análises e previsões. Essa etapa é de grande importância, pois é determinante para a qualidade dos resultados.

Após, aplicamos a técnica de validação cruzada tal que uma parte de dados é usada para treinamento (amostras para ajuste dos coeficientes internos ou pesos sinápticos). Uma outra parte para validação que são amostras consideradas para minimização dos erros, enquanto que o modelo aprende com dados de treinamento. E a última parte é de teste em que são amostras não apresentadas ao modelo, e que são utilizadas para previsão.

Para a aplicação dos modelos não lineares, inicialmente é feita a normalização dos dados. O método *standard scaler* (*central limit*) foi aplicado.

Dessa forma, a normalização dos dados é dada por:

$$\mathbf{x}_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)} \tag{3.6}$$

em que $\mathbf{x}_i \in [0, 1]$.

A Figura 3.12 ilustra o procedimento empregado para atingir a tarefa da previsão de séries das taxas de câmbio R$/US$ e €/US$. Foi utilizada a linguagem de programação *Python* para realizar as previsões. Vale ressaltar que, utilizamos as bibliotecas: *keras*, *skit-learn*, *Tensorflow* e *matplotlib* para plotar os gráficos.

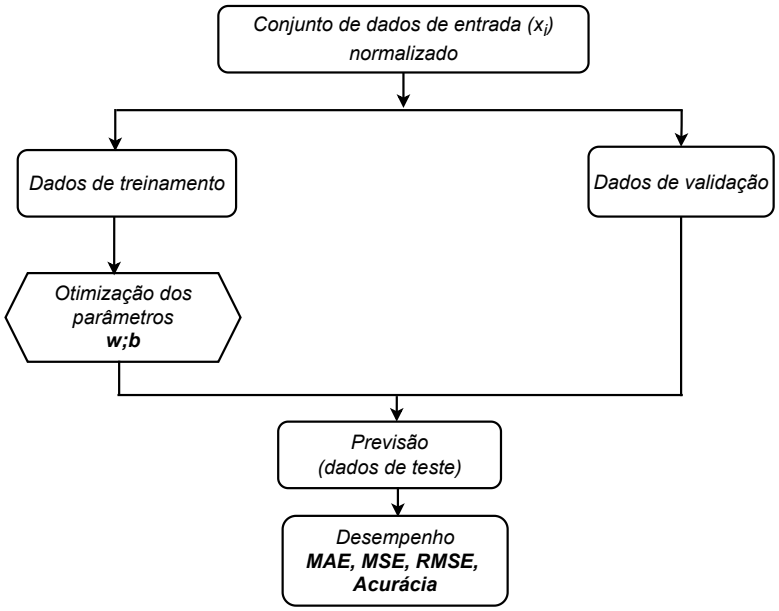


Figura 3.12 – Procedimento para previsão dos modelos não-lineares.

Os três modelos não-lineares empregados nesta dissertação, a saber, MLP, LSTM e GRU usam as mesmas estruturas. Para uma comparação equânime das métricas da avaliação do desempenho, utilizamos as mesmas configurações de parâmetros e hiperparâmetros. Um parâmetro do modelo é uma variável da configuração interna ao modelo, em que o valor pode ser estimado a partir dos dados. Por sua vez, um hiperparâmetro do modelo se refere a uma configuração externa ao modelo, em que o valor não pode ser estimado a partir dos dados, mas sim, escolhido de forma externa. A Tabela 3.6 apresenta um resumo dos parâmetros e hiperparâmetros empregados na nossa análise.

Os modelos usam função de ativação linear na camada de saída, duas camadas intermediárias com número de neurônios determinado no intervalo entre 5 a 50, ajustado de acordo com os conjuntos de dados. Utilizamos função de ativação ReLU nas camadas intermediárias. Empregamos o algoritmo ADAM para ajuste dos pesos sinápticos. Inicializamos os pesos utilizando a técnica *LeCun initialization*, de tal forma: $\mathbf{w}_{ij}^{l} \sim N(0, \frac{1}{L})$, L o número total dos dados de entrada (LECUN, 2019).

Tabela 3.6 – Parâmetros e hiperparâmetros

Modelo	**Símbolo**	**Descrição**	**Valor** (*série R$/US$*)	**Valor** (*série €/US$*)
	N	observações	4525	4526
	-	treinamento	3500	3501
	-	validação	875	875
	-	teste	150	150
	η	*learning rate*	0.001	0.001
MLP	-	número de camadas ocultas	2	2
	-	número de neurônios ocultos	50,10,1	50,10,1
	-	número máximo de épocas	150	150
	-	tamanho de lote	16	16
	-	*verbose*	1	1
	p	*patience*	2	2
	N	observações	4525	4526
	-	treinamento	3500	3501
	-	validação	875	875
	-	teste	150	150
	η	*learning rate*	0.001	0.001
LSTM	-	número de camadas ocultas	2	2
	-	número de neurônios ocultos	20,10,1	20,10,1
	-	número máximo de épocas	150	150
	-	tamanho de lote	16	16
	-	*verbose*	1	1
	p	*patience*	2	2
	N	observações	4525	4526
	-	treinamento	3500	3501
	-	validação	875	875
	-	teste	150	150
	η	*learning rate*	0.001	0.001
GRU	-	número de camadas ocultas	2	2
	-	número de neurônios ocultos	15,7,1	15,7,1
	-	número máximo de épocas	150	150
	-	tamanho de lote	16	16
	-	*verbose*	1	1
	p	*patience*	2	2

3.4.1 MLP

O Modelo MLP construído neste estudo tem uma camada de entrada, duas camadas ocultas e uma camada de saída, totalmente conectada ou densa. A camada de entrada representa o vetor $\mathbf{x}$ da série da taxa de câmbio. Na primeira camada oculta, o número de neurônios é 50, a função de ativação do tipo ReLU. Na segunda camada oculta, o número de neurônios é 10, com a função de ativação do tipo ReLU. A camada de saída é composta por um neurônio com função de ativação do tipo linear.

No total, 621 parâmetros foram ajustados. Da camada de entrada para primeira camada oculta, existem 1*50 = 50 pesos. A primeira camada oculta têm 50 neurônios e isso leva a 50 + 50 = 100 parâmetros. Da primeira camada oculta a segunda camada oculta há 510 parâmetros, ou seja, 50 * 10 + 10 = 510 parâmetros, pois há 10 parâmetros relacionados ao viés. Da segunda camada oculta para camada de saída temos 10*1 = 10

parâmetros, e um único viés, levando a $100 + 510 + 10 + 1 = 621$ parâmetros a serem ajustados.

Como se trata de um problema de previsão de séries temporais, o modelo MLP é compilado usando a métrica para função de perda do tipo *Mean Squared Error*, a função optimizador do tipo *ADAM* e a métrica da avaliação do desempenho do tipo *rmse*.

Como vimos no Capítulo 2, a raiz quadrada do erro médio é basicamente o desvio padrão dos resíduos. Utilizamos a técnica *Early Stopping*. A parada antecipada é utilizada para interromper o processo de ajuste do modelo, caso não haja melhoria no desempenho em termos de perda. Para especificar qual hiperparâmetro queremos monitorar durante o treinamento do modelo usamos o chamado parâmetro de monitor, ou seja, utilizamos a função de perda (loss) com $patience = 2$ e $verbose = 1$.

Ajustamos o modelo utilizando o conjunto de dados do treinamento composto por 3500 observação para a taxa de câmbio R$/US$, e por 3501 para a taxa de câmbio €/US$. O conjunto de validação é composto por 875 observações para ambas séries analisadas. O tamanho de lote é de 16 com 150 épocas. O processo de aprendizado foi interrompido após realizar 126 épocas para a série da taxa de câmbio R$/US$ e 135 épocas para série da taxa de câmbio €/US$.

Ajustando o modelo MLP, obtemos as previsões nos dados de teste que corresponde aos últimos 150 dias, ou seja, 07/10/2020 a 04/05/2021 para a taxa de câmbio R$/US$, e de 08/10/2020 a 05/05/2021 para a taxa de câmbio €/US$. A Figura 3.13 mostra as previsões um passo à frente geradas pelo modelo MLP a partir da série da taxa de câmbio R$/US$.

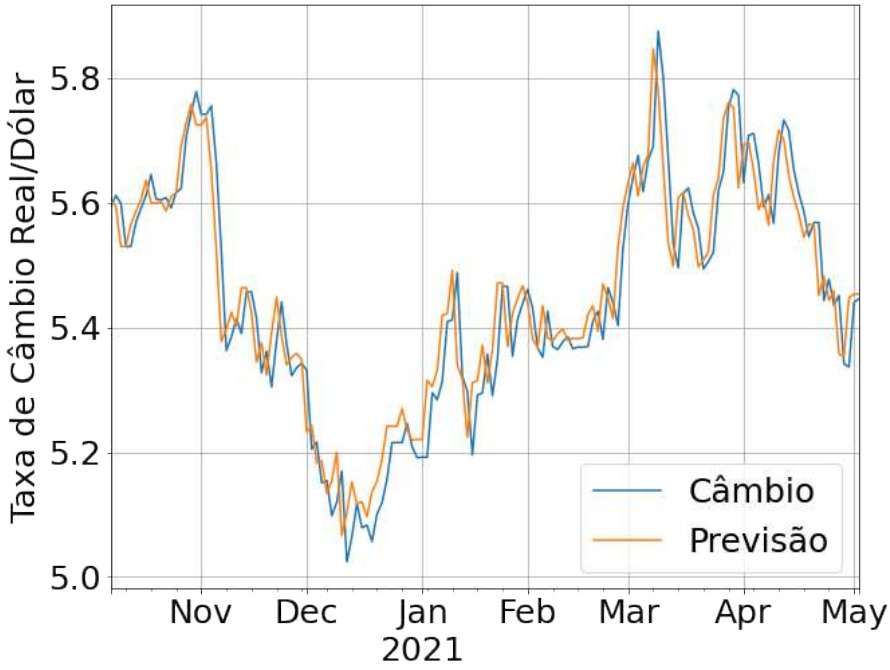


Figura 3.13 – Previsão 1 passo à frente gerada pelo modelo MLP: taxa de câmbio R$/US$.

Por sua vez, a Figura 3.14 apresenta as previsões um passo à frente geradas pelo modelo MLP a partir da série da taxa de câmbio €/US$.

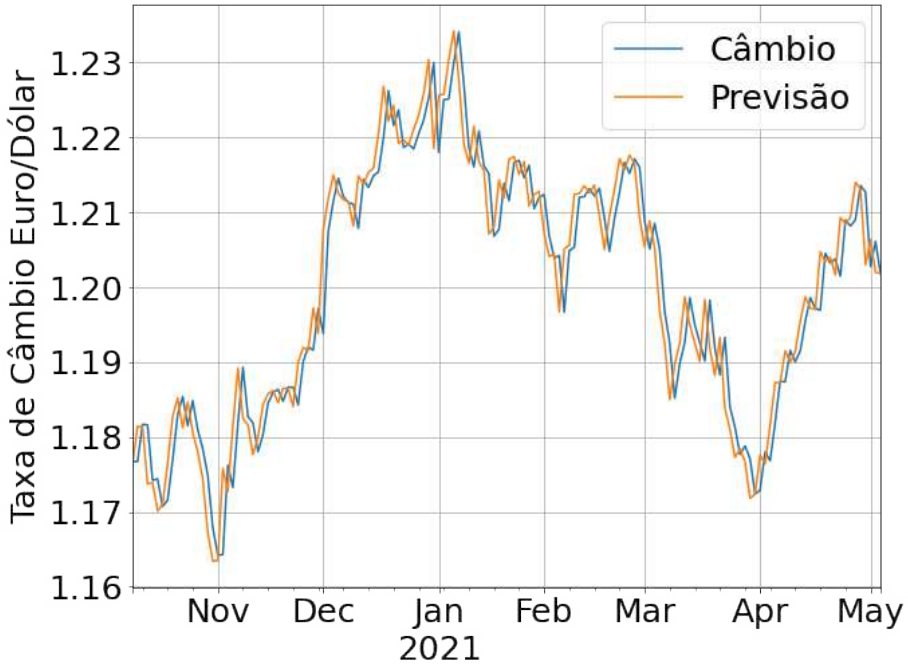


Figura 3.14 – Previsão 1 passo à frente gerada pelo modelo MLP: taxa de câmbio €/US$.

3.4.2 LSTM

O modelo LSTM construído neste trabalho contém uma camada de entrada, duas camadas ocultas e uma camada de saída, totalmente conectada e sequencial.

No total, 3011 parâmetros foram ajustados, constituídos da seguinte forma: da camada de entrada para a primeira camada oculta há 20 neurônios, ou seja, 1760 parâmetros; da primeira camada oculta para a segunda camada oculta há 10 neurônios, ou seja, 1240 parâmetros; da segunda camada oculta para camada de saída tem um neurônio, ou seja, $10 + 1 = 11$ parâmetros ajustáveis. Isso leva a um total de 3011 parâmetros para treinamento. Usamos a função de ativação do tipo ReLU nas duas camadas ocultas e a função linear na camada de saída.

O objetivo do algoritmo de ajuste dos parâmetros do modelo LSTM é minimizar o erro quadrático médio com otimizador ADAM. Os conjuntos de dados para treinamento e validação foram utilizados para ajuste dos coeficientes internos (pesos sinápticos). A Tabela 3.6 fornece maior informação sobre os parâmetros e hiperparâmetros usados. Para evitar o problema de *underfitting* ou *overfitting* utilizamos a técnica parada antecipada. O algoritmo foi encerrado após totalizar 132 épocas na série de câmbio R$/US$ e 135 na série de câmbio €/US$.

Os resultados das previsões um passo à frente geradas pelo modelo LSTM a partir da série da taxa de câmbio R$/US$ é apresentado na Figura 3.15 e para a taxa de câmbio €/US$ na Figura 3.16.

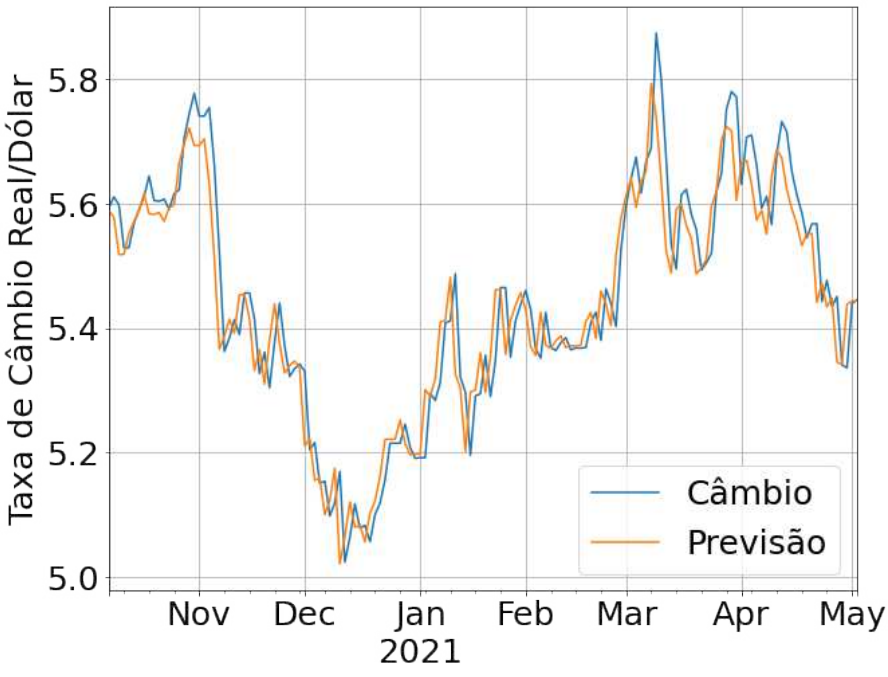


Figura 3.15 – Previsão 1 passo à frente gerada pelo modelo LSTM: taxa de câmbio R$/US$.

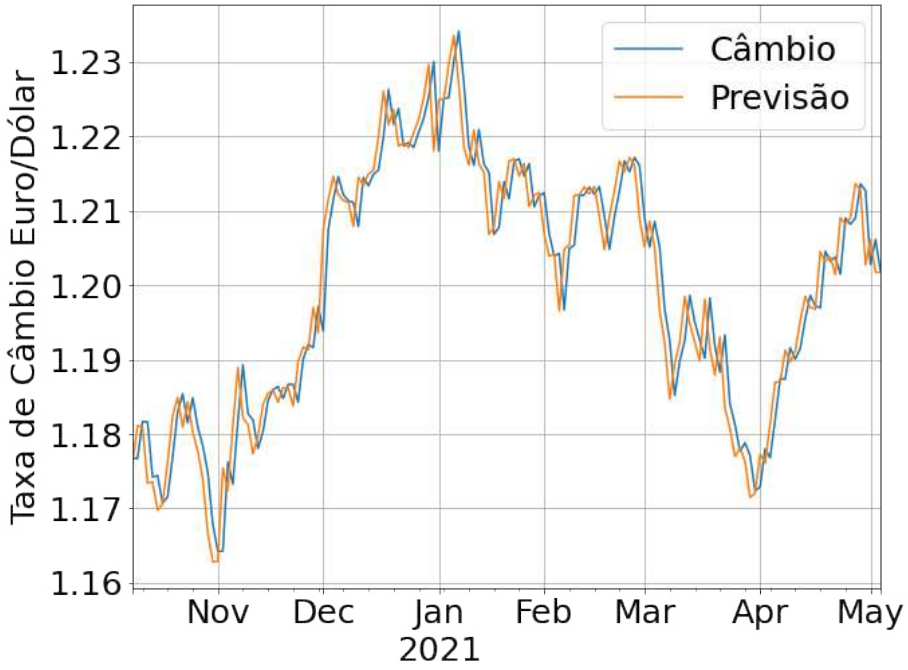


Figura 3.16 – Previsão 1 passo à frente gerada pelo modelo LSTM: taxa de câmbio €/US$.

3.4.3 GRU

O modelo GRU construído tem uma camada de entrada, duas camadas ocultas e uma camada de saída, totalmente conectadas. A sua estrutura é organizada da seguinte maneira: da camada de entrada para primeira camada oculta têm 15 neurônios, ou seja, 810 parâmetros ajustáveis; da primeira camada oculta para segunda camada oculta temos 7 neurônios, ou seja, 504 parâmetros ajustáveis; da segunda camada oculta para camada de saída tem um neurônio, ou seja, $7 + 1 = 8$ parâmetros ajustáveis. No total, temos $810 + 504 + 8 = 1322$ parâmetros ajustáveis.

A função de ativação ReLU é utilizada nas duas camadas ocultas e a função linear na camada de saída. O algoritmo consiste em minimizar o erro quadrático médio. O algoritmo otimizador usado é o ADAM. Ajustamos o modelo utilizando os conjuntos de dados do treinamento e validação para minimizar o erro no ajuste dos parâmetros internos. Também, utilizamos a *Early Stopping*, e o ajuste foi encerrado após atingir 128 épocas na série da taxa de câmbio R$/US$ e 132 épocas na série da taxa de câmbio €/US$. A Tabela 3.6 fornece outras informações sobre os hiperparâmetros utilizados.

Os resultados das previsões um passo à frente geradas pelo modelo GRU é ilustrado na Figura 3.17 para a série da taxa de câmbio R$/US$ e na Figura 3.18 para a taxa de câmbio €/US$.

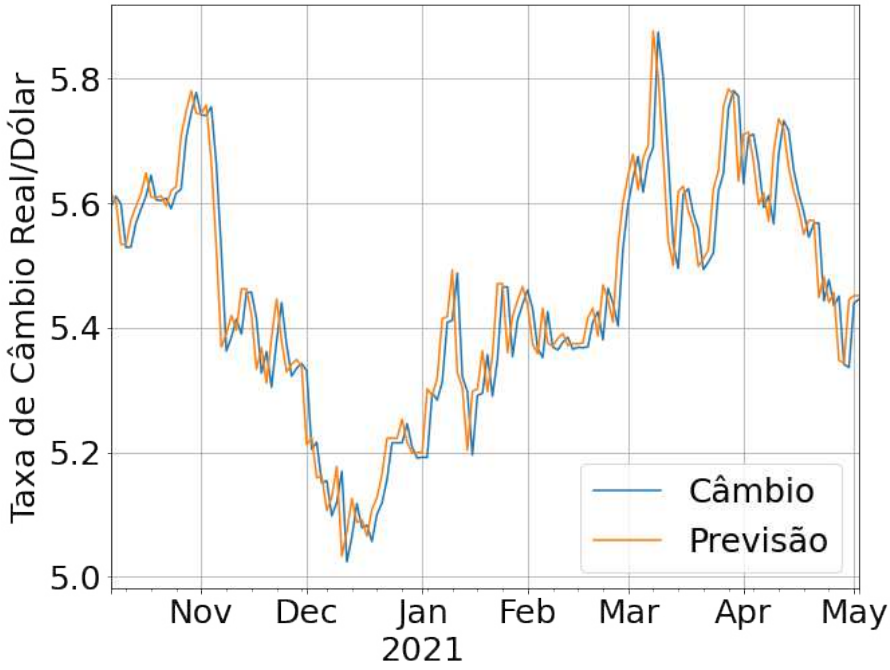


Figura 3.17 – Previsão 1 passo à frente gerada pelo modelo GRU: taxa de câmbio R$/US$.

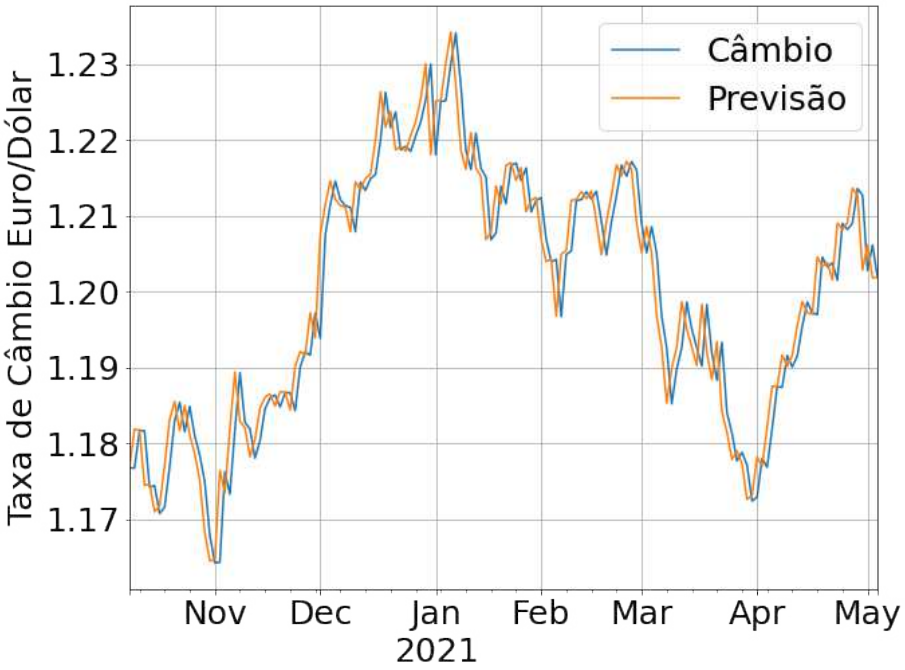


Figura 3.18 – Previsão 1 passo à frente gerada pelo modelo GRU: taxa de câmbio €/US$.

3.5 Comparação dos Resultados

De modo a comparar os modelos empregados nesta dissertação quanto ao desempenho de previsão alcançada, utilizamos as métricas de erro e da acurácia descritas na Seção 2.3. Os resultados apresentados em todas as tabelas refletem o desempenho dos modelos no conjunto de dados de teste.

Os modelos foram avaliados usando os últimos 150 valores para previsão 1 e 7 passos à frente. Ou seja, foram usados os dados de 07 de Outubro de 2020 a 04 de Maio de 2021 para a taxa de câmbio R$/US$ e 08 de Outubro de 2020 a 05 de Maio de 2021 para a taxa de câmbio €/US$.

A crença econômica se baseia na hipótese de que o mercado financeiro é eficiente, ou seja, os preços dos ativos atuais refletem todas as informações contidas em seu histórico passado (TAYLOR, 2005). Em outras palavras, o movimento passado ou a tendência de um preço de ativos financeiros não pode ser usado para prever seu movimento futuro. Isso levou à formulação da hipótese de passeio aleatório (RW, do inglês *Random Walk*). No caso, a taxa de câmbio prevista em $t+1$ é igual à taxa de câmbio no instante t, ou seja: $y_{t+1} = y_t + \varepsilon_t$. Para razão de verificação, utilizamos o modelo de passeio aleatório (RW) afim de provar esta hipótese.

Os resultados das métricas de avaliação do desempenho dos modelos lineares e não-lineares são apresentados na Tabela 3.7. Nela, é apresentado o erro absoluto médio (MAE), o erro quadrático médio (MSE), a raiz do erro quadrático médio (RMSE) e a precisão da direção (ACURÁCIA) de previsão 1 passo à frente que cobre o período de 07 de Outubro de 2020 a 04 de Maio de 2021 para série R$/US$, e de 08 de Outubro de 2020 a 05 de Maio de 2021 para série €/US$.

Observamos que todos os modelos de redes neurais recorrentes (LSTM e GRU) e não-recorrente (MLP) apresentaram MAE, MSE e RMSE menores que os modelos tradicionais, ARIMA, ARFIMA e Passeio Aleatório. Além disto, o modelo ARIMA apresentou maior RMSE quando comparados aos demais modelos, e a sua precisão da direção (ACURÁCIA) é ligeiramente menor comparado ao modelo ARFIMA.

Tabela 3.7 – Comparação dos Resultados: Previsão 1 Passo à Frente

Modelo/Métrica	**MAE**	**MSE**	**RMSE**	**ACURÁCIA**
ARIMA: 1 Passo à Frente				
ARIMA(1,1,1): Série R$/US$	0.1804	0.0469	0.2166	0.5146
ARIMA(1,1,1): Série €/US$	0.0844	0.0204	0.1429	0.5251
ARFIMA: 1 Passo à Frente				
ARFIMA(1,0.481,2): Série R$/US$	0.1461	0.0301	0.1735	0.5381
ARFIMA(2,0.46,2): Série €/US$	0.0634	0.0171	0.1208	0.5468
Passeio Aleatório-RW				
Passeio aleatório: Série R$/US$	0.1273	0.0262	0.1618	-
Passeio aleatório: Série €/US$	0.1132	0.0123	0.1109	-
MLP: 1 Passo à Frente				
MLP: Série R$/US$	0.0794	0.0119	0.1091	0.6442
MLP: Série €/US$	0.0659	0.0110	0.1049	0.6542
LSTM: 1 Passo à Frente				
LSTM: Série R$/US$	0.0562	0.0096	0.0979	0.6725
LSTM: Série €/US$	0.0433	0.0068	0.0825	0.6852
GRU: 1 Passo à Frente				
GRU: Série R$/US$	0.0387	0.0053	0.0728	0.6964
GRU: Série €/US$	0.0271	0.0037	0.0616	0.7073

As previsões um passo à frente do modelo ARIMA apresentou o resultado menos adequado quando comparamos ao modelo ARFIMA ou RW. Além disso, a acurácia, para as duas séries das taxas de câmbio, é ligeiramente inferior aos demais modelos, sejam lineares ou não-lineares.

O modelo ARFIMA apresentou a acurácia de 0.5381 para série da taxa de câmbio R$/US$, e 0.5468 para série €/US$. Como destacamos, em termos de magnitude, este modelo apresentou bons resultados, com RMSE menor tanto para série R$/US$ como para €/US$ quando comparamos com o modelo ARIMA. Isso leva dizer que o modelo ARFIMA consegue capturar de melhor forma as dependências temporais diferentemente do modelo ARIMA. O modelo ARFIMA acompanhou de melhor forma a tendência das séries analisadas.

O modelo passeio aleatório (RW) é melhor para previsão 1 passo à frente (Tabela 3.7). O RMSE do modelo passeio aleatório é ainda inferior ao modelo ARIMA, e mais próximo ao modelo ARFIMA nas séries analisadas.

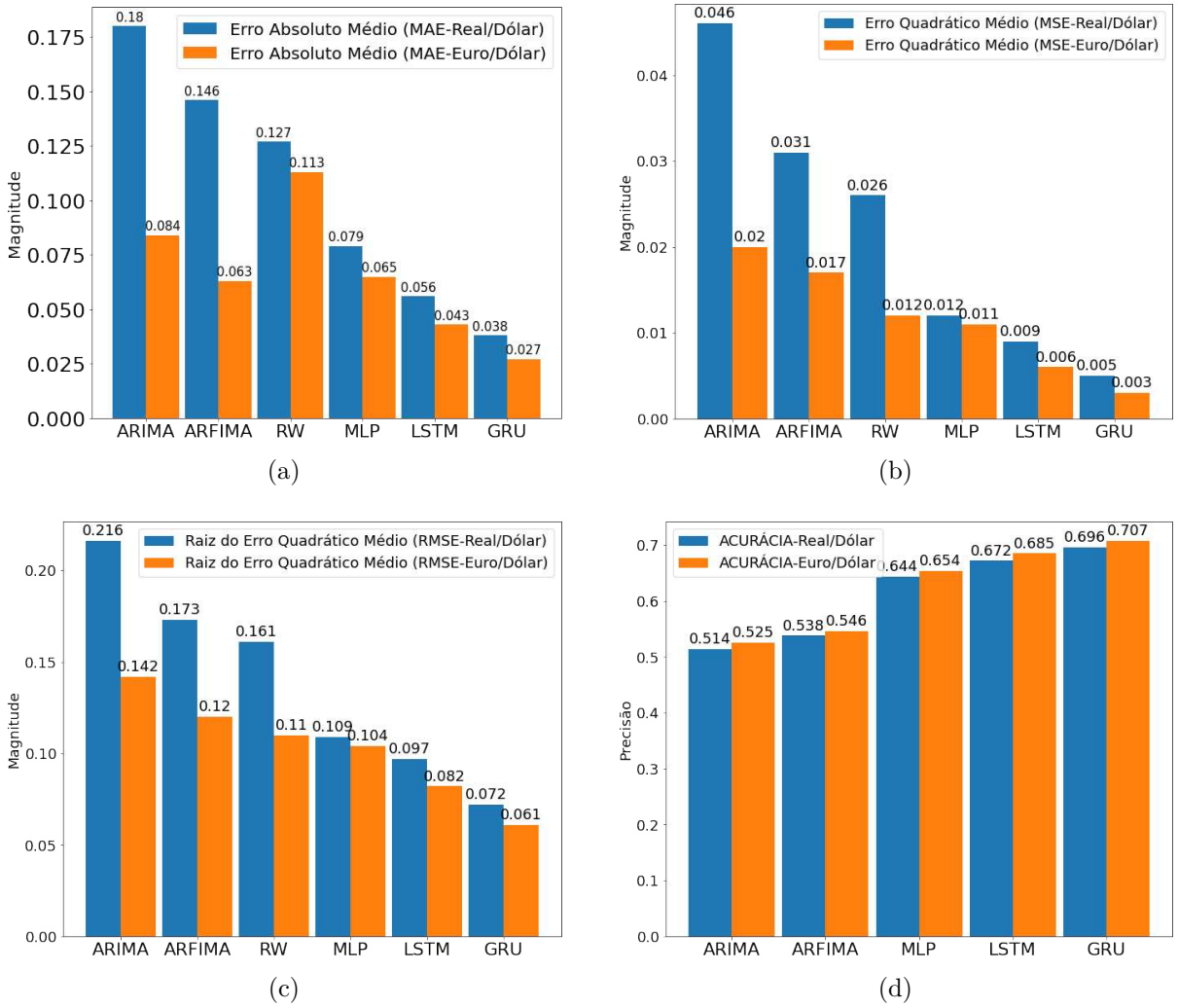


Figura 3.19 – Erros de Previsão das Taxas de Câmbio R$/US$ e €/US$ - Período de Outubro de 2020 a Maio de 2021 (Previsão 1 Passo à Frente).

Além disso, como as mudanças nas taxas de câmbio são não-linearmente dependentes, o resultado indica que estas não podem ser previstas de forma adequada com o uso dos modelos lineares. Geralmente, o modelo RW exibe um desempenho similar ao modelo ARIMA ou ARFIMA, sendo que os ARIMA e ARFIMA incorporam todo histórico da série para o ajuste de parâmetros. A hipótese de que o modelo Passeio Aleatório (TAYLOR, 2005) tem desempenho similar aos modelos lineares em termos de previsão de taxas de câmbio não se sustenta quando comparamos os resultados com os modelos não-lineares, como indica a Figura 3.19.

Ao contrário dos modelos lineares, os modelos não-lineares conseguem ter melhores desempenhos de previsão 1 passo à frente nas duas séries das taxas de câmbio tanto em magnitude (Figuras 3.19(a), (b) e (c)) como na precisão da direção (Figura 3.19(d)).

Uma comparação envolvendo somente as medidas de magnitude MAE, MSE e RMSE do modelo MLP indica que o modelo alcançou desempenho menos adequado

que os modelos LSTM e GRU nas duas séries das taxas de câmbio. O MAE do modelo ARFIMA para série €/US$ é ligeiramente inferior ao modelo MLP (Figura 3.19(a)). Outra comparação é feita desta vez considerando as duas séries de dados. Observamos o fato de que a série €/US$ alcançou um desempenho ligeiramente superior à série R$/US$. Entendemos que essa superioridade pode se explicar por fato da eficiência do mercado Euro.

Os resultados do modelo LSTM são melhores que os modelos ARIMA, ARFIMA, RW e MLP. Interessante notar que o modelo LSTM se adequa aos características das séries analisadas. Há predominância em termos da magnitude e da acurácia. No total, o modelo LSTM alcançou um desempenho superior para previsão 1 passo à frente. A acurácia do modelo é de 0.6725 para série R$/US$ e 0.6852 para série €/US$.

O modelo GRU superou em termos da magnitude e da acurácia comparando aos demais modelos empregados nesta dissertação. A acurácia é 0.6964 para série R$/US$ e 0.7073 para série €/US$. O RMSE é 0.0616 sendo o menor valor em ralação à todos os modelos empregados.

Em relação ao desempenho dos modelos não-lineares recorrentes (LSTM e GRU), observamos que uma arquitetura simples (uma ou duas camadas) é suficiente para capturar a dinâmica de todas as características das moedas analisadas nesta pesquisa, dado que os graus de precisão da direção são superiores aos demais modelos (ARIMA, ARFIMA e MLP).

Os modelos recorrentes LSTM e GRU apresentados nesta dissertação pertencem ao subconjunto dos modelos vistos como estatisticamente robustos. Estes modelos superam as abordagens tradicionais, ARIMA e ARFIMA, em termos da magnitude do erro e da acurácia, para todas as séries das taxas de câmbio analisadas. Esses resultados indicam que estes modelos são capazes de lidar com não-linearidades (ou não-estacionariedade) e memória longa, e podem ser considerados como modelos robustos para previsão de séries temporais.

A seguir, são analisados os erros de previsão 7 passos à frente dos modelos ARIMA, ARFIMA, MLP, LSTM e GRU para as séries das taxas de câmbio R$/US$ e €/US$. Os erros obtidos para o período de Outubro de 2020 a Maio de 2021 são mostrados na Tabela 3.8, para a previsão 7 passos à frente. Ainda, para maior visualização, a Figura 3.20 mostra os erros MAE, MSE, RMSE e a acurácia para a previsão 7 passos à frente.

Tabela 3.8 – Comparação dos Resultados: Previsão 7 Passos à Frente

Modelo/Métrica	**MAE**	**MSE**	**RMSE**	**ACURÁCIA**
ARIMA: 7 Passos à Frente				
ARIMA(1,1,1): Série R$/US$	0.5357	0.3442	0.5868	0.4531
ARIMA(1,1,1): Série €/US$	0.4426	0.3042	0.5515	0.4637
ARFIMA: 7 Passos à Frente				
ARFIMA(1,0.481,2): Série R$/US$	0.4134	0.3016	0.5492	0.4709
ARFIMA(2,0.46,2): Série €/US$	0.3412	0.2671	0.5168	0.4786
MLP: 7 Passos à Frente				
MLP: Série R$/US$	0.2943	0.2467	0.4967	0.5876
MLP: Série €/US$	0.2634	0.2200	0.4690	0.5964
LSTM: 7 Passos à Frente				
LSTM: Série R$/US$	0.1622	0.0898	0.2997	0.6578
LSTM: Série €/US$	0.1447	0.0780	0.2792	0.6605
GRU: 7 Passos à Frente				
GRU: Série R$/US$	0.0958	0.0531	0.2304	0.6824
GRU: Série €/US$	0.0735	0.0378	0.1944	0.6953

Analisando os erros de previsão 7 passos à frente (Tabela 3.8 e Figura 3.20), nota-se que os modelos empregados neste trabalho apresentaram pequena variação, comparado com a previsão 1 passo à frente, tanto nas medidas de magnitude como na precisão, sendo que o modelo GRU apresentou ainda melhor desempenho. No total, os modelos LSTM e GRU apresentaram melhor desempenho em termos de MAE e RMSE que forneceram erros menores que os modelos ARIMA e ARFIMA.

Para a previsão 7 passos à frente (Figura 3.20) os erros alcançados pelos modelos ARIMA e ARFIMA tanto para a série R$/US$ como para série €/US$ forneceram uma pequena diferença, onde o MAE e RMSE apresentaram erro maior do que a previsão 1 passo à frente (Figura 3.19).

Para a previsão 7 passos à frente os erros obtidos pelos modelos MLP, LSTM e GRU tanto para a série R$/US$ como para série €/US$ forneceram uma ligeira variação, em que o MAE e RMSE apresentaram erro ligeiramente maior do que a previsão 1 passo à frente.

Comparando todos os modelos empregados neste trabalho, pode-se verificar que tanto para a previsão 1 passo à frente como para a previsão 7 passos à frente o modelo GRU apresentou um bom desempenho que os modelos ARIMA e ARFIMA, isso para as duas séries analisadas. Vale ainda ressaltar que a diferença é relativamente pequena. O modelo LSTM apresentou melhor desempenho comparado aos modelos ARIMA, ARFIMA e MLP. Em relação aos modelos lineares, o modelo ARFIMA forneceu melhores resultados do que o modelo ARIMA tanto para a previsão 1 passo à frente como para a previsão 7

passos à frente, para as duas séries das taxas de câmbio.

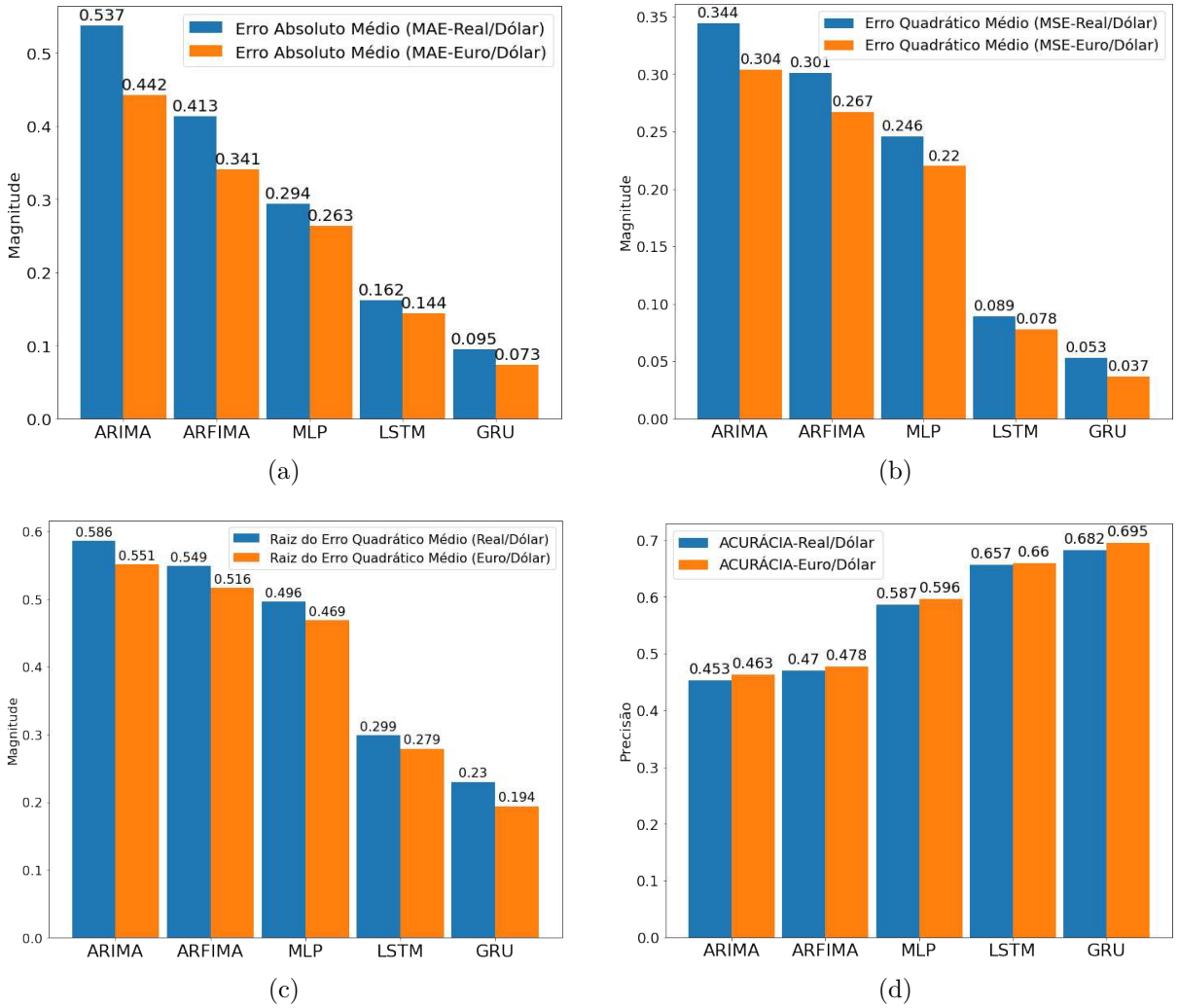


Figura 3.20 – Erros de Previsão das Taxas de Câmbio R$/US$ e €/US$ - Período de Outubro de 2020 a Maio de 2021 (Previsão 7 Passos à Frente).

Em relação a todos os modelos, o modelo GRU com duas camadas intermediárias, e 15 e 7 neurônios em cada camada apresentou os erros MAE, MSE e RMSE significativamente menores para as duas séries de câmbio R$/US$ e €/US$ tanto para a previsão 1 passo à frente como para a previsão 7 passos à frente.

A partir dos testes de raiz unitária (ADF e PP), mostramos a presença de não-estacionariedade nas duas séries das taxas de câmbio, além de longa dependência, isso leva a dizer que o uso de modelagens não-lineares são mais adequados. Nos modelos não-lineares, portanto, a incorporação de dinâmica pode ser um fator determinante na qualidade do desempenho obtido.

4 Conclusão

Nesta dissertação, investigamos o desempenho de alguns modelos lineares e não-lineares na previsão das Taxas de Câmbio Real/Dólar (R$/US$) e Euro/Dólar (€/US$). Em nossos estudos, empregamos duas famílias de modelos - modelos paramétricos: *Auto Regressive Integrated Moving Average* (ARIMA) e *Auto Regressive Fractionally Integrated Moving Average* (ARFIMA), e - modelos não-parmétricos: *Multilayer Perceptron* (MLP), *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU). O desempenho dessas abordagens foi comparado por meio de medidas da magnitude do erro (MAE, MSE, RMSE), e do grau de precisão da direção (Acurácia).

Os dados utilizados foram extraídos do *yahoo.finance*. O período considerado foi de 31 de Dezembro de 2003 a 04 de Maio de 2021 para os dados R$/US$ e de 31 de Dezembro de 2003 a 05 de Maio de 2021 para os dados €/US$, totalizando 4 525 e 4 526 observações, respectivamente. Ambas as séries são médias diárias.

Os ajustes dos modelos ARIMA e ARFIMA foram efetuados no conjunto de $N - 150$ observações, ou seja, 4 375 observações para a taxa de câmbio R$/US$, correspondendo ao período de 31 de Dezembro a 06 de Outubro de 2020, e 4 376 observações para a taxa de câmbio €/US$, correspondendo ao período de 31 de Dezembro de 2003 a 07 de Outubro de 2020. O conjunto de teste aborda o período de 07 de Outubro de 2020 a 04 de Maio de 2021, para a série R$/US$ e 08 de Outubro de 2020 a 05 de Maio de 2021, para a série €/US$.

Para o ajuste dos modelos MLP, LSTM e GRU foi aplicada a técnica de validação cruzada. Tal técnica divide o conjunto de dados em 3 partes: uma para treinamento ou ajuste dos pesos sinápticos, outra para validação usada para a minimização dos erros, e a terceira, chamada de conjunto de teste, utilizada na fase de previsão. Assim, neste trabalho foram usados os dados de 31 de Dezembro de 2003 a 30 de Maio de 2017, da série R$/US$, e 31 de Dezembro de 2003 a 31 de Maio de 2017, da série Euro/US$, para compor o conjunto de treinamento. O conjunto de validação compreende o período de 31 de Maio de 2017 a 06 de Outubro de 2020, da série R$/US$, e 01 de Junho de 2017 a 07 de Outubro de 2020, da série Euro/US$, e o conjunto de teste de 07 de Outubro de 2020 a 04 de Maio de 2021, da série R$/US$ e 08 de Outubro de 2020 a 05 de Maio de 2021, da série Euro/US$. O desempenho dos modelos lineares e não-lineares é analisado para previsão 1 e 7 passos à frente no período de teste.

A partir dos testes de raiz unitária (ADF e PP), mostramos a presença de não-estacionariedade (ou não-linearidade) nas duas séries das taxas de câmbio, além de

longa dependência detectada por meio do decaimento lento das funções de autocorrelação.

Os resultados indicaram que em todos os modelos de redes neurais recorrentes (LSTM e GRU) e não-recorrente (MLP) as medidas MAE, MSE e RMSE foram menores que os modelos tradicionais, ARIMA, ARFIMA e Passeio Aleatório. Além disso, o modelo ARIMA apresenta maior RMSE quando comparados aos demais modelos, e a sua precisão da direção (Acurácia) foi ligeiramente menor comparado ao modelo ARFIMA.

Uma comparação envolvendo apenas os modelos lineares mostra que o Passeio Aleatório (RW) alcançou o desempenho ligeiramente superior aos modelos ARIMA e ARFIMA. Além disso, como as mudanças nas taxas de câmbio são não-linearmente dependentes, o resultado indica que estas não podem ser previstas de forma adequada com uso do modelo ARIMA.

Uma comparação envolvendo somente os modelos não-lineares indica que o modelo MLP alcançou desempenho menos adequado que os modelos LSTM e GRU nas duas séries das taxas de câmbio. Entendemos que os modelos conseguiram capturar não-estacionariedade nos dados. Os resultados dos modelos LSTM e GRU são superiores quando comparado ao modelo MLP para previsão 1 passo à frente. Notamos que os modelos recorrentes se adaptam de melhor forma aos características das séries analisadas (não-linearidade e longa dependência). Há predominância em termos da magnitude e da acurácia.

Outra comparação foi feita considerando as duas séries de dados. Observamos o fato de que a série €/US$ alcançou um desempenho ligeiramente superior à série R$/US$. Entendemos que essa superioridade pode ser explicada pelo fato da eficiência de mercado Euro. A taxa de câmbio R$/US$ é volátil e apresenta variabilidade historicamente mais alta que a série €/US$. Esse fato dificulta as previsões da série, e como consequência o desempenho deteriora. Notamos que em vista das diferenças entre as duas moedas, a dinâmica da taxa de câmbio na condução de política econômica tem uso diferente dependendo de características do país.

Ainda, analisamos o desempenho de previsões 7 passos à frente. Notamos que todos os modelos empregados neste trabalho apresentaram pequena variação, comparado com a previsão 1 passo à frente, tanto nas medidas de magnitude como a acurácia, sendo que o modelo GRU apresentou ainda melhor desempenho. No total, os modelos LSTM e GRU apresentaram melhor desempenho em termos de MAE e RMSE que forneceram um erro ligeiramente menor aos demais modelos. Para a previsão 7 passos à frente os erros alcançados pelos modelos ARIMA e ARFIMA tanto para a série R$/US$ como para série €/US$ forneceram uma pequena diferença, onde o MAE e RMSE foram maiores do que para a previsão 1 passo à frente. O mesmo observamos para os modelos MLP, LSTM e

GRU tanto para a série R$/US$ como para série €/US$ em que o MAE e RMSE foram maiores do que a previsão 1 passo à frente.

Finalmente, destacamos que os modelos recorrentes LSTM e GRU apresentados nesta dissertação pertencem ao subconjunto dos modelos vistos como estatisticamente robustos. Estes modelos superaram as abordagens tradicionais, ARIMA e ARFIMA, em termos da magnitude do erro e da acurácia, para todas as séries das taxas de câmbio analisadas. Esses resultados indicam que estes modelos são capazes de lidar com não-linearidades (ou não-estacionariedade) e memória longa e podem ser considerados como modelos robustos para tratar séries temporais com estas características.

Referências

ABEL, A.; BERNANKE, B.; CROUSHORE, D. *Macroeconomics*. 8th. ed. [S.l.]: Pearson Education, 2013. 672 p. ISBN 9780133252170. Citado na página 16.

ALPAYDIN, E. *Introduction to Machine Learning*. 3rd. ed. [S.l.]: MIT Press, 2014. ISBN 9780262028189. Citado na página 47.

AZZOUNI, A.; PUJOLLE, G. *A Long Short-Term Memory Recurrent Neural Network Framework for Network Traffic Matrix Prediction*. arXiv, 2017. Disponível em: <https://arxiv.org/abs/1705.05690>. Citado 2 vezes nas páginas 18 e 34.

BALLINI, R. *Análise e Previsões de Vasões Utilizando Modelos de Séries Temporais, Redes Neurais e Redes Neurais Nebulosas*. Tese (Doutorado) — Universidade Estadual de Campinas, 2000. Citado 4 vezes nas páginas 26, 32, 34 e 35.

BENGIO, Y.; GRANDVALET, Y. Bias in Estimating the Variance of K-fold Cross-Validation. In: _____. *Statistical Modeling and Analysis for Complex Data Problems*. Boston, MA: Springer US, 2005. p. 75–95. ISBN 978-0-387-24555-3. Disponível em: <https://doi.org/10.1007/0-387-24555-3_5>. Citado na página 48.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. 1st. ed. Berlin, Heidelberg: Springer-Verlag, 2006. 738 p. ISBN 0387310738. Citado na página 47.

BLANCHARD, O. *Macroeconomics*. 8th. ed. [S.l.]: Pearson, 2019. 576 p. ISBN 9780134897899. Citado na página 17.

BOCCATO, L.; ATTUX, R. *Notas de aulas do curso "Aprendizado de Máquina". IA048-FEEC-Unicamp*. 2020. Disponível em: <http://www.dca.fee.unicamp.br/~lboccato/IA048_2s2020.html>. Citado 2 vezes nas páginas 33 e 36.

BOX, G.; JENKINS, G. *Time Series Analysis: Forecasting and Control*. 2nd. ed. [S.l.]: Holden-Day, 1970. 553 p. ISBN 9780816210947. Citado 3 vezes nas páginas 25, 26 e 34.

BROCKWELL, P. J.; DAVIS, R. A. *Introduction to Time Series and Forecasting*. 3rd. ed. [S.l.]: Springer International Publishing, 2016. XIV, 425 p. ISBN 9783319298528. Citado 4 vezes nas páginas 25, 26, 27 e 28.

BUENO, D. *Econometria de Séries Temporais*. 2nd. ed. [S.l.]: Cengage Learning, 2012. 341 p. ISBN 9788522128259. Citado 6 vezes nas páginas 22, 24, 25, 26, 93 e 94.

CHO, K.; MERRIENBOER, B. van; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. [S.l.]: Association for Computational Linguistics, 2014. p. 1724–1734. Citado 2 vezes nas páginas 42 e 49.

COELHO, L. d. S.; SANTOS, A. A. P.; JR., N. C. A. d. C. Podemos Prever a Taxa de Câmbio Brasileira? Evidência Empírica Utilizando Inteligência Computacional e Modelos Econométricos. *Gestão e Produção*, v. 15, n. 3, p. 635–647, Dez 2008. ISSN 0104-530X. Citado 4 vezes nas páginas 17, 18, 25 e 34.

CONTI, B. M.; PRATES, D. M.; PLIHON, D. A Hierarquia Monetária e suas Implicações para as Taxas de Câmbio e de Juros e a Política Econômica dos Países Periféricos. *Economia e Sociedade*, v. 23, n. 2, p. 341–372, Dez 2014. Citado na página 57.

CYBENKO, G. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, v. 5, n. 4, p. 455–455, Dez 1992. ISSN 0932-4194, 1435-568X. Citado na página 37.

EVANS, M. D.; LYONS, R. K. Meese-Rogoff Redux: Micro-Based Exchange-Rate Forecasting. *American Economic Review*, v. 95, n. 2, p. 405–414, 2005. Citado na página 53.

GEMAN, S.; BIENENSTOCK, E.; DOURSAT, R. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, v. 4, n. 1, p. 1–58, Jan 1992. ISSN 0899-7667, 1530-888X. Citado na página 47.

GHYSELS, E.; MARCELLINO, M. *Applied Economic Forecasting Using Time Series Methods*. 1st. ed. [S.l.]: Oxford University Press, 2018. 616 p. ISBN 9780190622015. Citado na página 27.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. 1st. ed. Cambridge, MA, USA: MIT Press, 2016. Citado 2 vezes nas páginas 36 e 37.

GRANGER, C.; JOYEUX, R. An Introduction to Long-Memory Time Series Models and Fractional Differencing. *Journal of Time Series Analysis*, Wiley-Blackwell, Wiley, v. 1, n. 1, p. 15–29, 1980. ISSN 0143-9782. Citado 4 vezes nas páginas 24, 28, 29 e 30.

GRAVES, A. *Supervised Sequence Labelling with Recurrent Neural Networks*. 1st. ed. [S.l.]: Springer, 2012. v. 385. 131 p. ISBN 978-3-642-24796-5. Citado 2 vezes nas páginas 43 e 44.

GREFF, K.; SRIVASTAVA, R. K.; KOUTNIK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, v. 28, n. 10, p. 2222–2232, Out 2017. ISSN 2162-237X, 2162-2388. Citado na página 49.

GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd. ed. [S.l.]: O'Reilly Media, Inc., 2019. ISBN 1492032646. Citado 6 vezes nas páginas 37, 38, 39, 40, 41 e 43.

HAYKIN, S. *Neural Networks and Learning Machines*. 3rd. ed. [S.l.]: Prentice Hall, 2009. ISBN 9780131471399. Citado 7 vezes nas páginas 32, 33, 34, 39, 41, 49 e 71.

HENRIQUE, B. M.; SOBREIRO, V. A.; KIMURA, H. Literature Review: Machine Learning Techniques Applied to Financial Market Prediction. *Expert Systems with Applications*, v. 124, p. 226–251, Jun 2019. ISSN 09574174. Citado na página 17.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, Nov 1997. Citado na página 42.

HOSKING, J. R. M. Fractional Differencing. *Biometrika*, v. 68, n. 1, p. 165–176, 1981. ISSN 0006-3444, 1464-3510. Citado 3 vezes nas páginas 28, 29 e 30.

HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: Principles and Practice*. 3rd. ed. [S.l.]: OTexts, 2018. ISBN 9780987507112. Citado na página 51.

IOFFE, S.; SZEGEDY, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. arXiv, 2015. Disponível em: <https://arxiv.org/abs/1502.03167>. Citado na página 46.

JASON, B. *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. 1st. ed. [S.l.]: Machine Learning Mastery, 2018. v. 1.4. ISBN 9781119682363. Citado na página 42.

JI, L.; ZOU, Y.; HE, K.; ZHU, B. Carbon Futures Price Forecasting Based with ARIMA-CNN-LSTM Model. *Procedia Computer Science*, v. 162, p. 33–38, 2019. ISSN 1877-0509. 7th International Conference on Information Technology and Quantitative Management (ITQM 2019): Information technology and quantitative management based on Artificial Intelligence. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050919319660>. Citado na página 19.

KINGMA, D. P.; BA, J. ADAM: A Method for Stochastic Optimization. In: BENGIO, Y.; LECUN, Y. (Ed.). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, Maio 7-9, 2015, Conference Track Proceedings*. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1412.6980>. Citado na página 46.

KRUGMAN, P.; OBSTFELD, M. *International Economics: Theory and Policy*. 8th. ed. [S.l.]: Pearson Addison-Wesley, 2009. ISBN 9780321553980. Citado 3 vezes nas páginas 16, 17 e 55.

LECUN, Y. *Quand la Machine Apprend: La Révolution des Neurones Artificiels et de L'Apprentissage Profond*. 1st. ed. [S.l.]: Odile Jacob, 2019. ISBN 9782738149329. Citado 4 vezes nas páginas 32, 33, 34 e 72.

LO, A. W. Reconciling Efficient Markets with Behavioral Finance: The Adaptive Markets Hypothesis. *Journal of investment consulting*, v. 7, n. 2, p. 21–44, 2005. Citado na página 20.

LO, A. W. *Adaptive Markets: Financial Evolution at the Speed of Thought*. 1st. ed. [S.l.]: Princeton University Press, 2017. ISBN 9780691135144. Citado na página 20.

LUENBERGER, D. G.; YE, Y. *Linear and Nonlinear Programming*. 4th. ed. [S.l.]: Springer International Publishing, 2016. v. 228. ISBN 9783319188416. Citado na página 40.

MOOSA, I.; VAZ, J. Directional Accuracy, Forecasting Error and the Profitability of Currency Trading: Model-Based Evidence. *Applied Economics*, Routledge, v. 47, n. 57, p. 6191–6199, 2015. Disponível em: <https://doi.org/10.1080/00036846.2015.1068917>. Citado na página 52.

MORELLI, M. J.; MONTAGNA, G.; NICROSINI, O.; TRECCANI, M.; FARINA, M.; AMATO, P. Pricing Financial Derivatives with Neural Networks. *Physica A: Statistical Mechanics and its Applications*, v. 338, n. 1–2, p. 160–165, Jul 2004. ISSN 03784371. Citado na página 34.

MORETTIN, P. A. *Econometria Financeira: Um Curso em Séries Temporais Financeiras*. 1st. ed. [S.l.]: São Paulo: Edgard Blucher, 2008. Citado 3 vezes nas páginas 57, 60 e 93.

MORETTIN, P. A.; TOLOI, C. M. d. C. *Análise de Séries Temporais*. 1st. ed. [S.l.]: São Paulo: Edgard Blucher, 2004. ISBN 9788521203896. Citado 11 vezes nas páginas 23, 24, 25, 27, 28, 29, 30, 31, 57, 60 e 66.

MOSCATELLI, M.; PARLAPIANO, F.; NARIZZANO, S.; VIGGIANO, G. Corporate Default Forecasting with Machine Learning. *Expert Systems with Applications*, Elsevier, v. 161, p. 113567, Dez 2020. ISSN 09574174. Citado 2 vezes nas páginas 38 e 39.

NI, L.; LI, Y.; WANG, X.; ZHANG, J.; YU, J.; QI, C. Forecasting of FOREX Time Series Data Based on Deep Learning. *Procedia Computer Science*, v. 147, p. 647–652, 2019. ISSN 1877-0509. 2018 International Conference on Identification, Information and Knowledge in the Internet of Things. Citado na página 18.

OLAH, C. *Understanding LSTM Networks*. 2015. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Citado 2 vezes nas páginas 43 e 50.

O'NEIL, C. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. 1st. ed. [S.l.]: Crown Publishing Group (NY), 2016. ISBN 9780553418811. Citado na página 54.

PATEL, P. J.; PATEL, N. J.; PATEL, A.; NORTH, H. Factors Affecting Currency Exchange Rate, Economical Formulas and Prediction Models. In: . [S.l.]: International Journal of Application or Innovation in Engineering & Management, 2014. v. 3, n. 3, p. 53–56. Citado 3 vezes nas páginas 17, 53 e 54.

PEDAMONTI, D. *Comparison of Non-Linear Activation Functions for Deep Neural Networks on MNIST Classification Task*. arXiv, 2018. Disponível em: <https://arxiv.org/abs/1804.02763>. Citado na página 38.

QU, Y.; ZHAO, X. Application of LSTM Neural Network in Forecasting Foreign Exchange Price. *Journal of Physics: Conference Series*, IOP Publishing, v. 1237, n. 4, p. 042036, Jun 2019. Disponível em: <https://doi.org/10.1088/1742-6596/1237/4/042036>. Citado na página 18.

ROMANO, J. M. T.; ATTUX, R.; CAVALCANTE, C. C.; SUYAMA, R. *Unsupervised Signal Processing: Channel Equalization and Source Separation*. [S.l.]: CRC Press, 2011. ISBN 9781420019469. Citado 3 vezes nas páginas 21, 22 e 24.

SHEN, F.; CHAO, J.; ZHAO, J. Forecasting Exchange Rate Using Deep Belief Networks and Conjugate Gradient Method. *Neurocomputing*, Elsevier, v. 167, p. 243–253, 2015. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231215005408>. Citado na página 18.

SIMS, C. A. Macroeconomics and Reality. *Econometrica: journal of the Econometric Society*, JSTOR, v. 48, n. 1, p. 1–48, Jan 1980. ISSN 00129682. Citado na página 27.

SOON, S.-V.; BAHARUMSHAH, A. Z. Exchange Rates and Fundamentals: Further Evidence Based on Asymmetric Causality Test. *International Economics*, v. 165, p. 67–84, Maio 2021. ISSN 21107017. Citado 3 vezes nas páginas 16, 53 e 54.

TALEB, N. N. *The Black Swan: The Impact of the Highly Improbable*. 1st. ed. [S.l.]: Random House, 2007. 400 p. ISBN 9781400063512. Citado na página 54.

TANG, Z.; ALMEIDA, C. de; FISHWICK, P. A. Time Series Forecasting Using Neural Networks vs. Box-Jenkins Methodology. *SIMULATION*, v. 57, n. 5, p. 303–310, Nov 1991. ISSN 0037-5497, 1741-3133. Citado na página 17.

TAYLOR, S. *Asset Price Dynamics, Volatility, and Prediction*. 3rd. ed. [S.l.]: Princeton University Press, 2005. 525 p. ISBN 9780691115375. Citado 2 vezes nas páginas 79 e 81.

WEIGEND, A. S.; HUBERMAN, B. A.; RUMELHART, D. E. Predicting the Future: A Connectionist Approach. *International Journal of Neural Systems*, v. 01, n. 03, p. 193–209, Jan 1990. ISSN 0129-0657, 1793-6462. Citado na página 34.

WEIGEND, A. S.; HUBERMAN, B. A.; RUMELHART, D. E. Predicting Sunspots and Exchange Rates with Connectionist Networks. *Advanced Research Workshop on Nonlinear Modeling and Forecasting*, p. 395–432, Set 1992. Disponível em: <https://cds.cern.ch/record/245460>. Citado na página 34.

WERBOS, P. Backpropagation Through Time: What It Does and How To Do It. *Proceedings of the IEEE*, v. 78, n. 10, p. 1550–1560, Out 1990. ISSN 00189219. Citado na página 45.

ZHANG, D.; KABUKA, M. R. Combining Weather Condition Data to Predict Traffic Flow: A GRU-Based Deep Learning Approach. *IET Intelligent Transport Systems*, Institution of Engineering and Technology, v. 12, p. 578–585(7), Set 2018. ISSN 1751-956X. Citado na página 18.

ZHANG, G. Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. *Neurocomputing*, v. 50, p. 159–175, Jan 2003. ISSN 09252312. Citado 2 vezes nas páginas 18 e 34.

ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with Artificial Neural Networks: The State of the Art. *International Journal of Forecasting*, v. 14, n. 1, p. 35–62, 1998. ISSN 0169-2070. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207097000447>. Citado na página 34.

APÊNDICE A – Capítulo 2: Testes de Raiz Unitária

Para a análise de estacionariedade das séries analisadas neste trabalho, foram empregados dois testes de raiz unitária: teste de Dickey Fuller Aumentado (ADF) e teste de Phillips-Perron (PP) (BUENO, 2012), (MORETTIN, 2008).

A.1 Teste de Dickey Fuller

Testar a existência de 1 raiz unitária (RU) em x_t quando a série for expressa por uma das expressões abaixo:

$$x_t = \alpha + \beta \mathbf{t} + \gamma x_{t-1} + \varepsilon_t \;\Rightarrow \Delta x_t = \alpha + \beta \mathbf{t} + \rho x_{t-1} + \varepsilon_t \tag{A.1a}$$

$$x_t = \alpha + \gamma x_{t-1} + \varepsilon_t \;\Rightarrow \Delta x_t = \alpha + \rho x_{t-1} + \varepsilon_t \tag{A.1b}$$

$$x_t = \gamma x_{t-1} + \varepsilon_t \;\Rightarrow \Delta x_t = \rho x_{t-1} + \varepsilon_t \tag{A.1c}$$

em que α e $\beta\mathbf{t}$ expressam componentes determinísticos, ou seja, constante e tendência, respectivamente; ε_t o ruído branco com média zero e variância constante.

As hipóteses nula e alternativa são dadas por:

$$H_0 : \gamma = 1 \Longleftrightarrow \rho = 0 \; (1 \; RU) \tag{A.2a}$$

$$H_1 : \gamma < 1 \Longleftrightarrow \rho < 0 \; (0 \; RU) \tag{A.2b}$$

As estatísticas dos testes (Equações A.1(a) a A.1(c)) são tratados como modelo com constante e tendência determinísticas (Equação A.1(a)); modelo com constante (Equação A.1(b)); modelo sem termos determinísticas (Equação A.1(c)).

A hipótese nula (H_0) é de que a série tenha raiz unitária e, portanto, não seja estacionária. A hipótese alternativa (H_1) é de que a série não tenha raiz unitária e, portanto, é estacionária.

A versão aumentada do teste de DF é usada por considerar a existência de alguma estrutura de autocorrelação para os erros nas equações do teste. Se essa estrutura não for considerada, há perda de eficiência do estimador de MQO para ρ, e as estatísticas

dos termos determinísticos ficam enviesadas (BUENO, 2012). O objetivo desse procedimento é eliminar uma possível existência de autocorrelação serial no termo de erro ε_t. Assim, ao invés de estimar as equações A.1a, A.1b e A.1c de cada uma das três opções do teste de Dickey Fuller, estima-se as seguintes equações a seguir:

$$\Delta x_t = \alpha + \beta \mathbf{t} + \rho x_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta x_{t-i} + \varepsilon_t \tag{A.3a}$$

$$\Delta x_t = \alpha + \rho x_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta x_{t-i} + \varepsilon_t \tag{A.3b}$$

$$\Delta x_t = \rho x_{t-1} + \sum_{i=1}^{p-1} \delta_i \Delta x_{t-i} + \varepsilon_t \tag{A.3c}$$

em que p a defasagem máxima de Δx_t. Nas três opções (Equações A.3a, A.3b e A.3c), aplica-se o mesmo procedimento de testar as hipóteses nula e alternativa. Assim, em cada uma delas, H_0 continua tendo as mesmas interpretações. A implementação do teste de ADF sugere os seguintes passos: (i) verificação da opção (A.3a, A.3b e A.3c) do teste, (ii) determinação de defasagem máxima das Equações A.3a, A.3b e A.3c, (iii) verificação da estatística de teste computada, (iv) decisão final.

A.2 Teste de Phillips-Perron

O teste PP generaliza o ADF para uma classe de modelos em que os erros ε_t são auto-correlacionados e heterogeneamente distribuídos. O teste baseia-se na mesma hipótese nula e estrutura do teste ADF. O teste PP é baseado nas respectivas estatísticas:

$$\Delta x_t = \alpha + \beta \mathbf{t} + \rho x_{t-1} + \varepsilon_t \longrightarrow \mathbf{Z}_{\tau,\beta} \tag{A.4a}$$

$$\Delta x_t = \alpha + \rho x_{t-1} + \varepsilon_t \longrightarrow \mathbf{Z}_{\tau,\alpha} \tag{A.4b}$$

$$\Delta x_t = \rho x_{t-1} + \varepsilon_t \longrightarrow \mathbf{Z}_{\tau} \tag{A.4c}$$

em que a obtenção das estatísticas $\mathbf{Z}_{\tau,\beta},\ \mathbf{Z}_{\tau,\alpha}\ \mathbf{Z}_{\tau}$ depende do cálculo da variância de longo prazo dos resíduos:

$$\hat{\nu}^2 = \hat{\sigma}^2 + \frac{2}{T} \sum_{j=1}^{M} \omega \left(\frac{j}{M+1} \right) \sum_{t=j+1}^{T} \hat{\varepsilon}_t \hat{\varepsilon}_{t-1} \tag{A.5}$$

em que $\hat{\sigma}^2$ variância estimada, M defasagem, T tamanho da amostra, $\omega \left(\frac{j}{M+1} \right)$ função de ponderação (BUENO, 2012).