UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

1.5em

Detecção de Outliers em Dados não Vistos de Séries Temporais por meio de Erros de Predição com SARIMA e Redes Neurais Recorrentes LSTM e GRU

Antonio Luiz Tonissi Migliato

Dissertação de Mestrado do Programa de Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria (MECAI)

ICMC USP
SÃO CARLOS

Antonio Luiz Tonissi Migliato

Detecção de Outliers em Dados não Vistos de Séries Temporais por meio de Erros de Predição com SARIMA e Redes Neurais Recorrentes LSTM e GRU

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria. *Versão revisada.*

Área de Concentração: Matemática, Estatística e Computação

Orientador: Prof. Dr. Moacir Antonelli Ponti

USP – São Carlos
Dezembro de 2021

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

M634d

Migliato, Antonio Luiz Tonissi
Detecção de Outliers em Dados não Vistos de SériesTemporais por meio de Erros de Predição com SARIMA eRedes Neurais Recorrentes LSTM e GRU / Antonio Luiz Tonissi Migliato; orientador Moacir Antonelli Ponti. -- São Carlos, 2021.
104 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2021.

1. Detecção de outliers. 2. Predição de séries temporais. 3. SARIMA. 4. LSTM. 5. GRU. I. Ponti, Moacir Antonelli, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Antonio Luiz Tonissi Migliato

Outlier Detection in Unseen Time Series Data via Prediction Errors with SARIMA and Recurrent Neural Networks LSTM and GRU

Dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – in accordance with the requirements of the Professional Master's Program in Mathematics Statistics and Computing Applied to Industry, for the degree of Master in Science. *Final version.*

Concentration Area: Mathematics, Statistics and Computing

Advisor: Prof. Dr. Moacir Antonelli Ponti

USP – São Carlos
December 2021

Dedico este trabalho à minha esposa Iara Regina e aos meus filhos Luiz Felipe e William que são a razão de minha existência.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me inspirado a iniciar essa jornada e por ter me dado saúde e determinação para desenvolver este trabalho. Agradeço à minha esposa Iara Regina e aos meus filhos Luiz Felipe e William pelo apoio, incentivo e compreensão ao longo de toda essa trajetória. Agradeço aos meus pais Sebastião e Lilia Márcia pelos ensinamentos e conselhos que sempre me ajudaram a fazer as escolhas corretas. Faço também um agradecimento especial ao Professor Moacir Antonelli Ponti por ter aceitado acompanhar-me nessa empreitada, pelas longas horas de dedicação, orientação, instrução e de transmissão de conhecimento sem os quais esse trabalho jamais seria possível. Agradeço aos colegas do grupo de pesquisa Visualization, Images, and Computer Graphics (VICG), principalmente à Leo Sampaio Ribeiro, pelo apoio e atenção na solução de questões técnicas. Agradeço à comissão examinadora, Prof. André Carvalho, Profa. Sarajane Peres e Prof. Mauro Masili, pelas sugestões e avaliação. Agradeço à Universidade de São Paulo e a todos os seus colaboradores que, em algum momento ao longo desse tempo, com seus conhecimentos e serviços, contribuíram para a execução e finalização deste trabalho. Entre eles, em especial, agradeço à Luzinete Silva (recepção), ao Bruno Chaaban (secretaria), à Monique Conceição (secretaria) e à Maria Lima (biblioteca).

"O homem que não se decide a cultivar o hábito de pensar,
perde o maior prazer da vida."
(Thomas Alva Edison)

RESUMO

MIGLIATO, A. L. T. **Detecção de Outliers em Dados não Vistos de Séries Temporais por meio de Erros de Predição com SARIMA e Redes Neurais Recorrentes LSTM e GRU**. 2021. 104 p. Dissertação (Mestrado – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

A atividade de identificar padrões nos dados que não estejam em conformidade com o comportamento esperado, ou detecção de outliers, como é conhecida, é um problema relevante em diversas áreas do conhecimento, como financeira, saúde, detecção de fraudes, entre outras. Em diversas dessas áreas, os dados apresentam-se em forma de séries temporais. Esse tipo de dado exige métodos que considerem a natureza sequencial das observações, visto que os valores em séries temporais são correlacionados e dependentes. Nesses casos, sistemas de detecção de outliers precisam lidar com situações nas quais os valores estão temporalmente associados. Visando encontrar respostas mais apropriadas para a detecção de outliers nessas situações, sistemas baseados em erros de predições realizadas com redes recorrentes LSTM tem sido propostos. Neste trabalho, foi estudado um modelo de detecção de outliers em dados não vistos baseado nas capacidades preditivas das redes neurais LSTM e GRU. A diferença entre os valores preditos e os valores observados foram calculados como erros de predição e utilizados para detectar outliers em três séries temporais univariadas de contexto econômico. Como linha de base para comparações, foi utilizado o modelo estatístico SARIMA. Primeiramente, utilizou-se um ”valor limite” específico para detecção de outliers, calculado a partir dos erros de predição do conjunto de treinamento. Num segundo momento, os modelos foram testados com todos os valores limites possíveis para detecção de outliers. Os resultados mostraram que o modelo SARIMA obteve melhor desempenho no geral, mas os desempenhos apresentados pelas redes neurais LSTM e GRU foram satisfatórios e merecem mais estudos.

Palavras-chave: Detecção de outliers, Predição de séries temporais, SARIMA, LSTM e GRU.

ABSTRACT

MIGLIATO, A. L. T. **Outlier Detection in Unseen Time Series Data via Prediction Errors with SARIMA and Recurrent Neural Networks LSTM and GRU**. 2021. 104 p. Dissertação (Mestrado – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

The activity of identifying patterns in data that do not comply with expected behavior, or detection of outliers, as it is known, is a relevant problem in several areas of knowledge, such as finance, health, fraud detection, among others. In several of these areas, data are presented in the form of time series. This type of data requires methods that consider the sequential nature of the observations, as the values in time series are correlated and dependent. In these cases, outlier detection systems need to deal with situations in which values are temporally associated. Aiming to find more appropriate answers for the detection of outliers in these situations, systems based on prediction errors with LSTM recurrent networks have been proposed. In this work, an outlier detection model in unseen data based on the predictive capabilities of LSTM and GRU neural networks was studied. The difference between predicted values and observed values were calculated as prediction errors and used to detect outliers in three univariate time series of economic context. As a baseline for comparisons, the SARIMA statistical model was used. First, a specific threshold was used to detect outliers, calculated from the training set prediction errors. Secondly, the models were tested with all possible thresholds for detecting outliers. The results showed that the SARIMA model had better overall performance, both in predicting and detecting outliers, but the performances achieved by the LSTM and GRU neural networks were satisfactory and deserve further studies.

Keywords: Outliers detection, Time series prediction, SARIMA, LSTM, GRU.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ALGORITMOS

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

Detecção de *outliers* é um problema relevante que tem sido estudado em diversas áreas tanto de pesquisa quanto de aplicação (CHALAPATHY; CHAWLA, 2019). O termo refere-se, de forma geral, à atividade de identificar comportamento nos dados que não estejam em conformidade com o padrão esperado (SINGH, 2017; TRAN; NGUYEN; THOMASSEY, 2019). Ainda, conforme definido por Hawkins (1980), outlier é uma observação que apresenta um desvio tão expressivo em relação a outras observações a ponto de levantar suspeitas de que foi gerada por um mecanismo diferente. Diversas áreas, tais como engenharia e economia, estão baseadas na suposição de que processos e comportamentos existentes na natureza obedecem princípios e regras, resultando em sistemas que se manifestam por meio de dados observáveis. A partir desses dados, é possível formular hipóteses sobre os processos subjacentes que descrevem seu comportamento "normal", implicitamente assumindo que os dados utilizados para gerar as hipóteses são típicos para esse processo (MEHROTRA; MOHAN; HUANG, 2017). Outlier é uma variação do que seria considerado um comportamento normal (DAVIS; RAINA; JAGANNATHAN, 2020; MEHROTRA; MOHAN; HUANG, 2017; TRAN; NGUYEN; THOMASSEY, 2019) e, embora existam várias técnicas para detecção de outliers, o desafio comum é identificar corretamente o comportamento normal e classificar o comportamento anômalo (DAVIS; RAINA; JAGANNATHAN, 2020; YAO *et al.*, 2017).

A análise de outliers tem inúmeras aplicações em diversas áreas, tais como, financeira, controle de qualidade, diagnósticos de erros, detecção de intrusões, e diagnósticos médicos (AGGARWAL, 2017; CHALAPATHY; CHAWLA, 2019; CHANDOLA; BANERJEE; KUMAR, 2009). A detecção de outliers pode ser particularmente útil nas áreas industriais e de manufatura, uma vez que pode auxiliar a identificar problemas graves com antecedência (DAVIS; RAINA; JAGANNATHAN, 2020; MEHROTRA; MOHAN; HUANG, 2017). A implementação de sistemas de detecção de outliers permite a análise de grande quantidade de dados para identificar padrões inesperados e tomar decisões mais efetivas (TRAN; NGUYEN; THOMASSEY, 2019). Por exemplo, na área de controle de qualidade, a detecção de outliers tem sido utilizada por

um longo tempo, indicando quando características de um determinado produto está abaixo dos requisitos mínimo de qualidade estabelecidos (AGGARWAL, 2017). Empresas de varejo constantemente monitoram seus ganhos e lucros, para planejarem suas atividades e evitarem problemas futuros. Isso envolve analisar os dados de venda e comparar as flutuações com dados passados. Detecção de outliers pode desempenhar um papel importante nesse contexto, ajudando a separar flutuações insignificantes (ruídos) de variações importantes com implicações significativas para o futuro da empresa. Empresas de diversos setores mantêm estoques de matéria prima e produtos acabados. Manter o controle do estoque de forma que o produto não falte e nem seja estocado em excesso é uma questão que tem alto impacto sobre a lucratividade de uma empresa. A detecção de outliers pode contribuir ao descrever o comportamento normal do estoque de um determinado produto, indicando quando a quantidade desse item alcança um valor não esperado (MEHROTRA; MOHAN; HUANG, 2017). Na área de Finanças, a análise de outliers tem permitido a detecção de fraudes em diversas aplicações como fraudes em cartões de crédito, fraudes em seguros, anomalias em preço de ações e anomalias em transações entre organizações, evitando, assim, prejuízos para instituições que atuam nessas áreas (AGGARWAL, 2017; BALA; SINGH *et al.*, 2019).

Em várias das áreas citadas acima, como por exemplo, a detecção de fraudes, detecção de falhas e na análise de lucratividade em empresas de varejo, os dados apresentam-se, em geral, em forma de séries temporais. Esse tipo de dado exige métodos que considerem a natureza sequencial das observações, visto que os valores em séries temporais são correlacionados e dependentes (CHATFIELD; XING, 2019). Nesses casos, passa a ser necessário lidar com os desafios associados aos problemas de detecção de outliers em situações nas quais os valores são temporalmente associados (BONTEMPS *et al.*, 2016; TRAN; NGUYEN; THOMASSEY, 2019). Além disso, modelos tradicionais de detecção de outliers, tais como os modelos estatísticos, frequentemente falham em capturar completamente a estrutura de dados complexos (DAVIS; RAINA; JAGANNATHAN, 2020; CHALAPATHY; CHAWLA, 2019), inclusive séries temporais (CHALAPATHY; CHAWLA, 2019). Visando encontrar respostas mais apropriadas para essas situações, diversos pesquisadores têm utilizado modelos de detecção de outliers baseados em erros de predição. Entre esses pesquisadores encontram-se: Bontemps *et al.* (2016), Chalapathy e Chawla (2019), Davis, Raina e Jagannathan (2020), Mehrotra, Mohan e Huang (2017), Singh (2017), Tran, Nguyen e Thomassey (2019). Nessas pesquisas, as predições foram realizadas com a rede neural recorrente Long Short-Term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) devido à sua habilidade em manter memórias de longo prazo (TRAN; NGUYEN; THOMASSEY, 2019). Essa rede é conhecida por possuir capacidade para representar os relacionamentos entre eventos atuais e eventos prévios, e lidar com o problema exposto acima (MALHOTRA *et al.*, 2015). Inclusive, segundo Chalapathy e Chawla (2019), a detecção de outliers é uma área de aplicação que tem se beneficiado significativamente com a evolução do *Deep Learning*.

Neste trabalho, foi proposto e avaliado um método para detecção de outliers em séries

temporais univariadas de contexto econômico baseado na capacidade preditiva das redes neurais recorrentes LSTM e *Gated Recurrent Unit* (GRU) (CHUNG *et al.*, 2014). A diferença entre os valores preditos e os valores observados foram calculados como erros de predição e utilizados para detectar outliers em dados não vistos de três séries temporais univariadas de contexto econômico. Como linha de base para comparações, foi utilizado o modelo estatístico SARIMA (BOX; JENKINS, 1970). Primeiramente, utilizou-se um "valor limite" específico para detecção de outliers, calculado a partir dos erros de predição do conjunto de treinamento. Num segundo momento, os modelos foram testados com todos os valores limites possíveis para detecção de outliers.

Dessa forma, buscou-se: (i) investigar a capacidade preditiva das redes neurais recorrentes LSTM e GRU em comparação com o modelo estatístico SARIMA, utilizando-se dois métodos de predição; (ii) estudar uma metodologia para detectar outliers em dados não vistos a partir de erros de predição das redes neurais LSTM e GRU e do método SARIMA; e (iii) avaliar a eficácia dos modelos de detecção de outliers gerados a partir dos erros de predição em diferentes cenários, bem como uma análise dos mesmos utilizando curvas *precision-recall*. Convém salientar que, embora o modelo de detecção de outliers desenvolvido neste estudo tenha sido empregado em dados previamente coletados, essa abordagem pode ser adaptável para detecção de outliers em tempo real.

Partindo-se da premissa de que é possível realizar predições em séries temporais por meio de redes neurais e modelos estatísticos, a hipótese adotada é a de que erros de predição, calculados a partir da diferença entre valores preditos e valores observados de séries temporais, fornecem informação suficiente para detecção de outliers em dados futuros.

O restante desta dissertação está organizado da seguinte forma: o Capítulo 2 apresenta uma revisão teórica dos principais conceitos abordados neste trabalho; o Capítulo 3 apresenta o método utilizado no procedimento experimental e as descrições das séries temporais abordadas neste estudo; o Capítulo 4 apresenta os resultados e discussão; e o Capítulo 5 apresenta a conclusão do trabalho. Os códigos desenvolvidos nesse trabalho encontram-se em: https://github.com/ToniMigliato

CAPÍTULO

2

FUNDAMENTAÇÃO TEÓRICA

Esse capítulo trata dos conceitos fundamentais envolvidos no presente estudo. Primeiramente, faz-se uma abordagem sobre séries temporais e suas principais características. Em seguida, descreve-se a utilização do modelo estatístico SARIMA para modelagem e predições de séries temporais, abordando também tópicos sobre predição de séries temporais com *Deep Learning*. E, por último, são abordados os fundamentos da detecção de outliers em séries temporais.

2.1 Séries Temporais

Este trabalho aborda séries temporais discretas e uniformemente amostradas ao longo do tempo. Para tal, serão adotadas as definições conforme Brockwell e Davis (2016).

Definição 2.1. Série temporal é uma sequência ordenada de observações X de tamanho m, na forma $X = x_1, x_2, ..., X_m$, onde $x_t \in \mathbb{R}$ é uma observação amostrada num instante de tempo $t \in T$.

Definição 2.2. Série temporal discreta é aquela em que o conjunto de tempo T, dos tempos nos quais as observações foram coletadas, é um conjunto discreto.

O objetivo primário da análise de séries temporais é desenvolver modelos matemáticos que ofereçam descrições plausíveis de dados amostrais. Com o intuito de se oferecer um conjunto estatístico que descreva características de séries temporais, assume-se que uma série temporal pode ser representada como uma coleção de variáveis aleatórias indexadas de acordo com a ordem com que foram obtidas no tempo. Por exemplo, pode-se considerar uma série temporal como uma sequência de variáveis aleatórias, $x_1, x_2, x_3, ...$, onde a variável aleatória x_1 denota o valor tomado pela série no primeiro ponto do tempo, i.e. $t = 1$, a variável x_2 denota o valor do segundo período do tempo, i.e. $t = 2$, e assim por diante (SHUMWAY; STOFFER, 2017).

A coleção de variáveis aleatórias X_t indexada por t pode ser modelada por um *processo estocástico*. Os valores assim observados são chamados de realizações desse processo esto-

cástico (SHUMWAY; STOFFER, 2017; CHATFIELD; XING, 2019). O relacionamento entre uma realização do processo estocástico e o processo em si é análogo, na estatística clássica, ao relacionamento entre uma amostra e a população da qual ela foi retirada (MILLS, 2019).

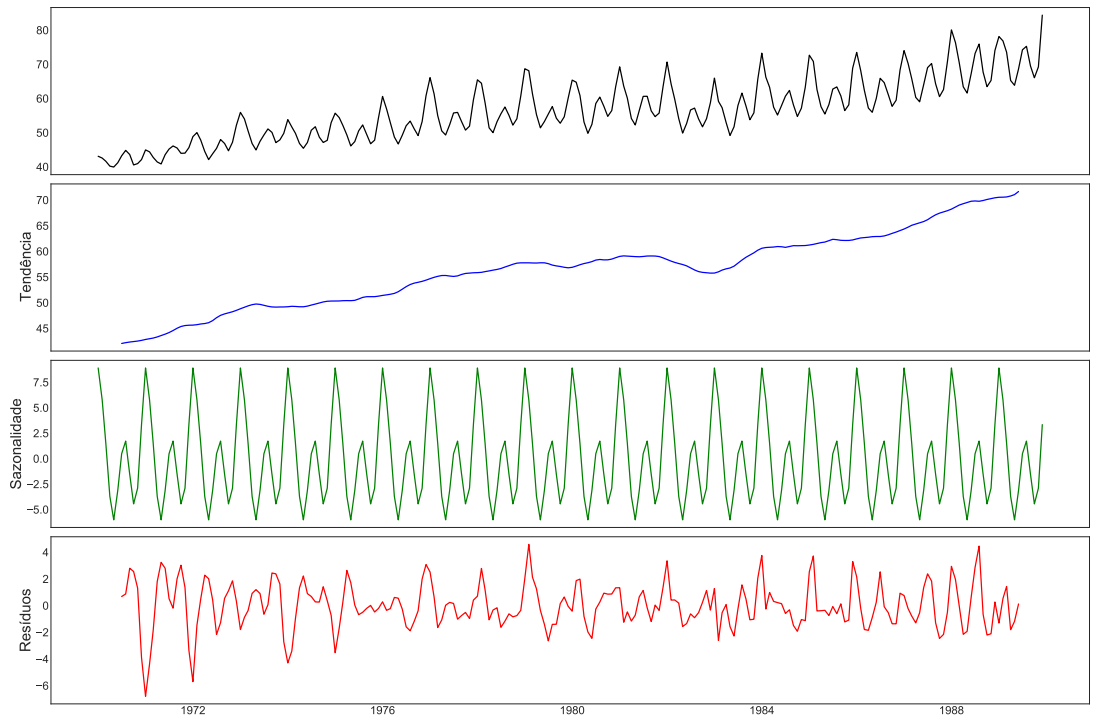
2.1.1 Componentes das Séries Temporais

Os componentes que formam uma série temporal, de acordo com Hyndman *et al.* (2008), são:

- Tendências (M_t): a direção de longo-prazo da série;
- Sazonalidade (S_t): um padrão que se repete com periodicidade conhecida;
- Ciclos: um padrão que se repete com alguma regularidade, mas com periodicidade desconhecida; e
- Resíduos (R_t): o componente imprevisível da série.

A Figura 1 mostra os principais componentes de uma série temporal.

Figura 1 – Componentes de Uma Séries Temporais - Produção de Energia



Nessa revisão, as séries temporais consideradas serão decompostas na forma

$$X_t = M_t + S_t + R_t, \tag{2.1}$$

Em vários casos, a presença de ciclos não é modelada explicitamente, mas incorporada na tendência.

2.1.2 Autocorrelação

Em séries temporais as observações próximas estão comumente correlacionadas (METCALFE; COWPERTWAIT, 2009). A autocorrelação mede a relação linear entre valores defasados de uma série temporal ou, em outras palavras, o quão similares são dois pontos separados por um atraso temporal k. Na prática, existem diversos coeficientes de correlação, correspondendo a diferentes defasagens k. Valores próximos a 0 indicam ausência de correlação, e valores próximos a 1 ou -1 correlação máxima, indicando, nesse caso, que uma observação pode ser explicada pela outra de forma perfeita.

Seja r_k o valor que mede a autocorrelação com defasagem k, então estamos medindo a correlação entre x_t e x_{t-k}. Por exemplo, r_1 mensura a relação entre x_t e x_{t-1}. Formalmente, um coeficiente de autocorrelação r_k é dado por:

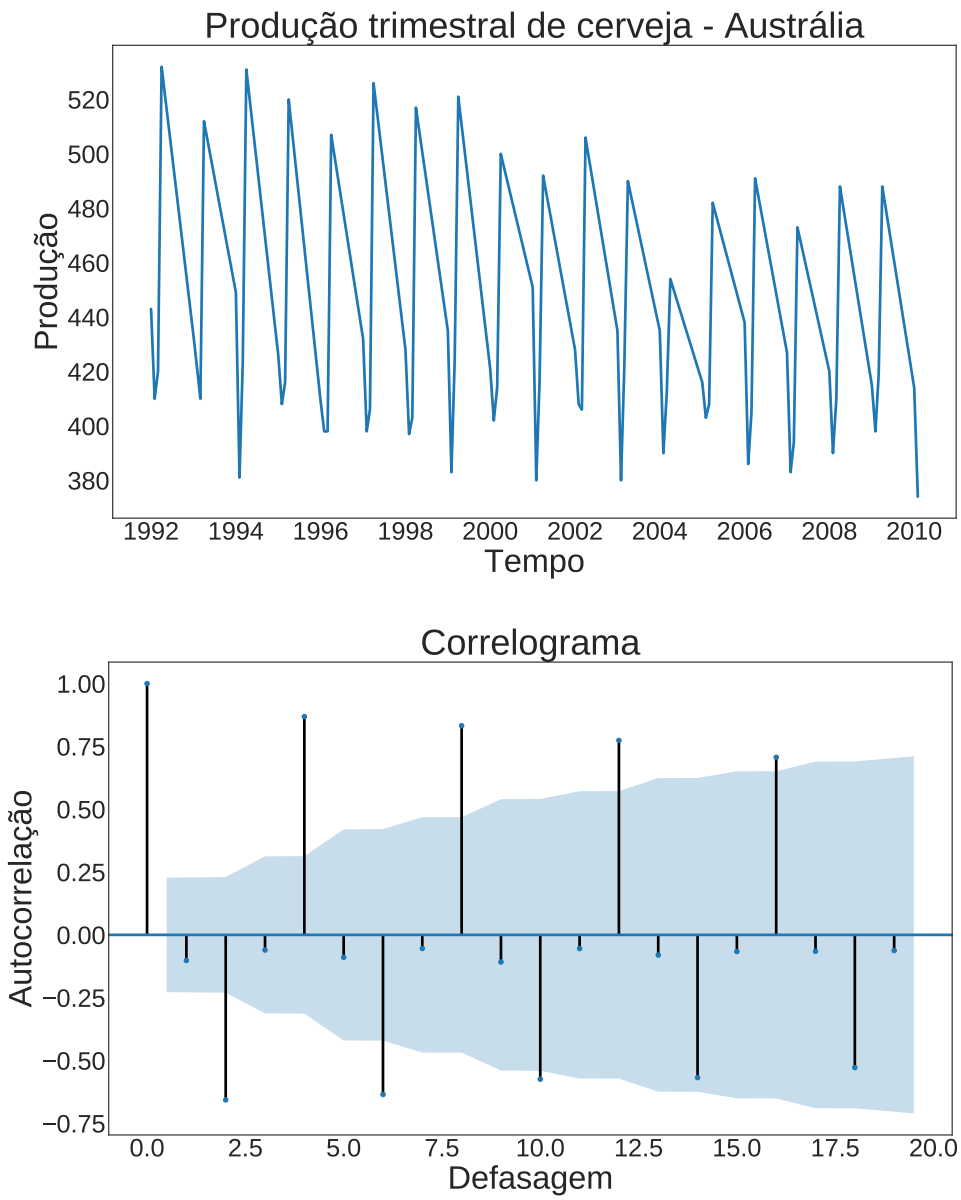
$$r_k = \frac{\sum_{t=k+1}^{T}(x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^{T}(x_t - \bar{x})^2} \tag{2.2}$$

onde T é o comprimento da série temporal e $\bar{x}$ é a média dos valores. A equação 2.2 é conhecida como Função de Autocorrelação, ou ACF (do inglês *Autocorrelation Function*). Os coeficientes de autocorrelação, r_k, de uma função de autocorrelação podem ser apresentados graficamente. Esse gráfico é chamado de correlograma (HYNDMAN; ATHANASOPOULOS, 2018).

A Figura 2 mostra o gráfico trimestral da produção de cerveja na Austrália de 1992 a 2010 e seu correlograma. O correlograma é um auxílio visual para interpretação de um conjunto de coeficientes de autocorrelação r_k. No correlograma, os coeficientes são plotados contra as defasagens de autocorrelação k para $k = 0, 1, 2, ...N$. A região colorida em azul no correlograma indica se as correlações são significativamente diferentes de zero. Isso porque quando uma série é puramente aleatória (esse tipo de série será descrito mais abaixo), r_k, para $k \geq 1$, será aproximadamente $\mathcal{N}(0, 1/m)$, onde m é o número de observações em uma série temporal. Assim, se uma série temporal é puramente aleatória, pode-se esperar que os seus coeficientes de autocorrelação r_k, para $k \geq 1$, estejam situados entre $\pm 1.96/\sqrt{m}$. Convencionou-se mostrar no correlograma a região demarcada por esses valores, e quando os coeficientes de autocorrelação situam-se fora dessa região, são tidos como estatisticamente significativos. Importante ressaltar que o correlograma também auxilia na modelagem de séries temporais (CHATFIELD; XING, 2019), conforme será mostrado mais adiante.

No correlograma da série temporal da figura 2, é possível observar que r_0 possui valor máximo pois indica a correlação sem defasagem, ou seja, compara cada ponto com ele mesmo. Podemos analisar os coeficientes de maior valor (positivo ou negativo) para interpretar a série: (i) r_4 é o maior coeficiente para $k > 0$. Isso se deve ao padrão sazonal nos dados. Os picos e fundos desse gráfico tendem a ocorrer a cada quatro trimestres. (ii) r_2 apresenta o maior valor negativo porque os vales da série tendem a estar dois trimestres atrás dos picos.

Figura 2 – Produção trimestral de cerveja na Austrália (acima) e seu correlograma (abaixo).



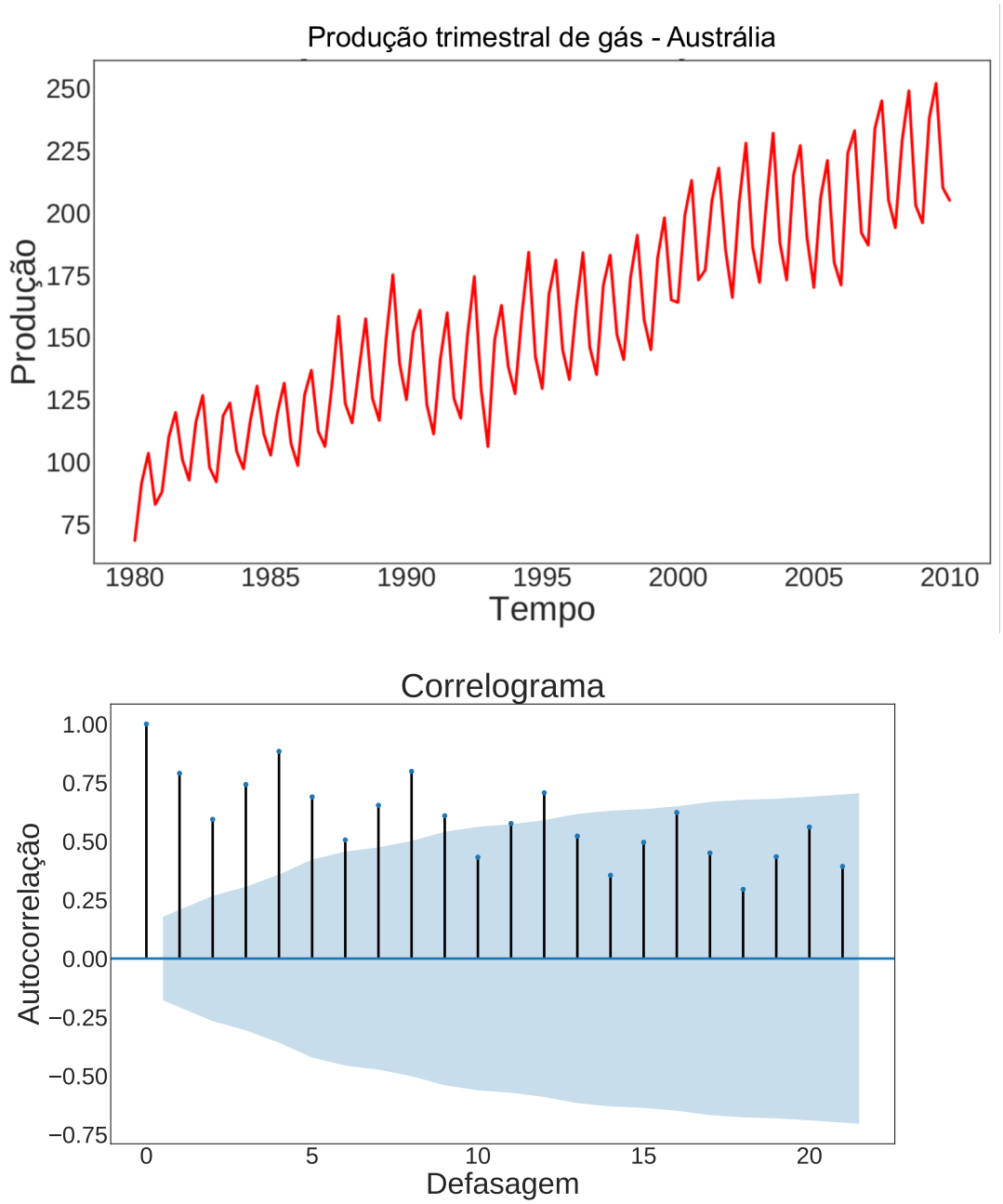
2.1.3 Analisando a Tendência e Sazonalidade das Séries Temporais

Quando a série temporal possui tendência, a autocorrelação para defasagens menores tende a ser alta e positiva porque as obervações próximas no tempo também são próximas em magnitude. Portanto, a ACF de séries com tendências tende a ter valores positivos que decrescem lentamente à medida que a defasagem aumenta. Quando a série temporal é sazonal, as autocorrelações tendem a ser mais altas para as defasagens sazonais (com valores múltiplos ao da frequência da sazonalidade) do que para as outras defasagens. Quando a série temporal possui tendência e sazonalidade, poderão ser vistos ambos os efeitos.

A quantidade de gás produzido na Austrália e seu correlograma estão representados na Figura 3. O decaimento lento nos coeficientes de autocorrelação à medida que a defasagem

aumenta é devido à tendência, enquanto que a forma ondulada é devida à sazonalidade.

Figura 3 – Produção trimestral de gás na Austrália (acima) e seu correlograma (abaixo).

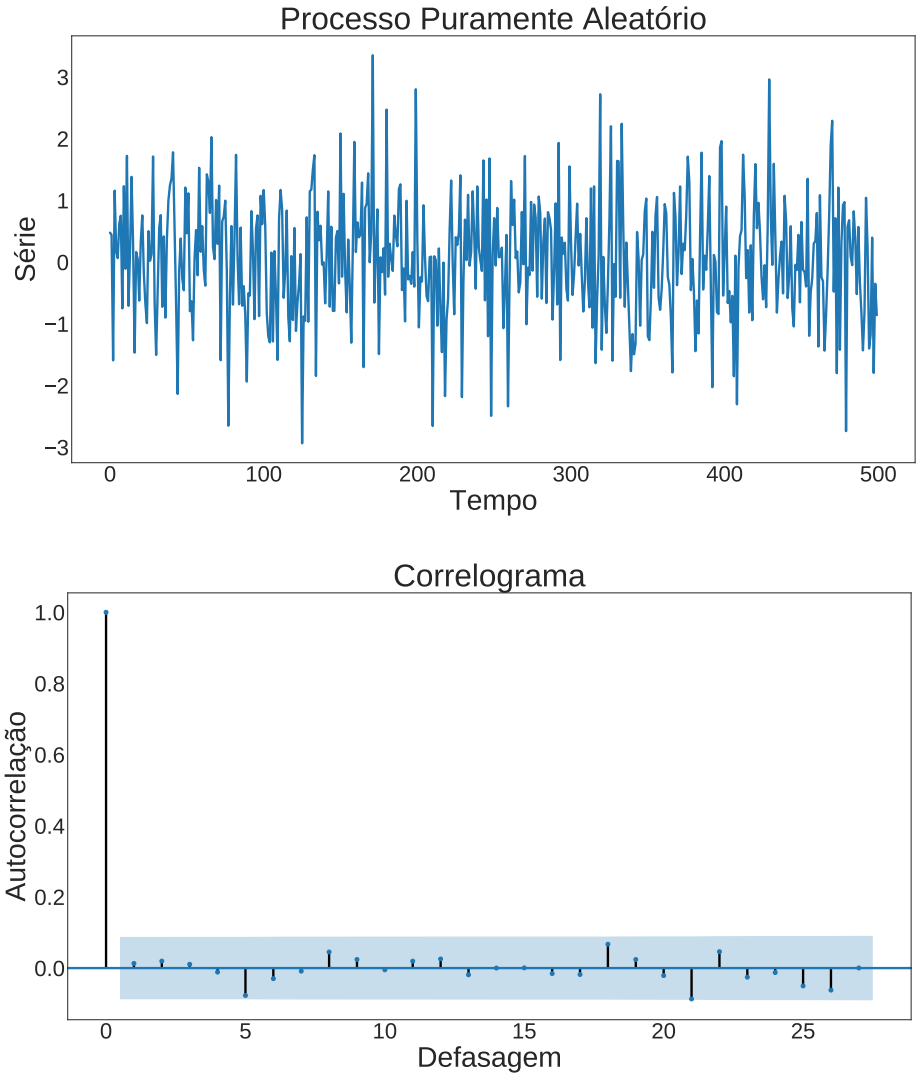


2.1.4 Processos Puramente Aleatórios

Um tipo simples de série é quando se tem uma coleção de valores (variáveis aleatórias) não correlacionadas, z_t, com média 0 e variância σ^2. Esse processo pode ser denotado como $w_t \sim \text{zn}(0, \sigma^2)$ (MILLS, 2019; SHUMWAY; STOFFER, 2017). Assim, as séries temporais classificadas como processos puramente aleatórios possuem autocorrelações aproximadamente nula para $k > 0$ (CHATFIELD; XING, 2019). A Figura 4 mostra um exemplo de processo

puramente aleatório e seu correlograma. É possível notar que, como a série é aleatória, as autocorrelações entre x_t e x_{t-k} para qualquer defasagem k é próxima de zero (HANKE, 2014).

Figura 4 – Série puramente aleatória com variância $\sigma_Z^2 = 1$ (acima) e seu correlograma (abaixo).



2.1.5 Estacionariedade e Diferenciação

Estacionariedade

Um aspecto importante na análise de séries temporais envolve avaliar processos cujas propriedades, pelo menos algumas delas, não variam com o tempo (BROCKWELL; DAVIS, 2016). Dessa forma, no desenvolvimento de modelos para séries temporais, é importante fazer suposições de que haja alguma forma de equilíbrio estatístico. Uma suposição especialmente útil é que a série temporal seja estacionária (BOX *et al.*, 2015). Uma série temporal estacionária é aquela cujas propriedades não dependem do tempo no qual ela é observada. Assim, séries temporais com tendências ou com sazonalidades não são estacionárias (HYNDMAN; ATHANASOPOULOS, 2018). De acordo com Chatfield e Xing (2019), um processo é considerado

estacionário de segunda ordem, ou fracamente estacionário, se sua média for constante e se sua função de autocovariância depender somente da defasagem, de forma que

$$E[x(t)] = \mu \tag{2.3}$$

e

$$Cov[(x(t)), x(t+\tau)] = \gamma(\tau) \tag{2.4}$$

Assim, estacionariedade fraca requer regularidade na média e nas funções de autocorrelações, de forma que essas quantias possam ser estimadas por seus valores médios (SHUMWAY; STOFFER, 2017). Na definição de estacionariedade fraca não se faz nenhuma exigência para momentos acima da segunda ordem. Convém notar que existe também a estacionariedade estrita. Na prática, porém, a mais utilizada é a estacionariedade fraca, ou de segunda ordem (CHATFIELD; XING, 2019).

Diferenciação

Uma maneira de se converter uma série temporal não estacionária para estacionária é computando diferenças entre observações consecutivas (SHUMWAY; STOFFER, 2017). Isso é conhecido como diferenciação. Diferenciação pode ajudar a estabilizar a média da série temporal removendo alterações em seu nível e, portanto, eliminando ou reduzindo tendências e sazonalidades (HYNDMAN; ATHANASOPOULOS, 2018). A diferenciação é um tipo especial de filtragem da série temporal, útil para remoção de sua tendência, até que se torne estacionária. Primeira diferenciação é largamente utilizada e frequentemente funciona bem (CHATFIELD; XING, 2019). Segundo Shumway e Stoffer (2017), a diferenciação desenvolve um papel central na análise de séries temporais e, por isso, recebe sua própria notação. A diferenciação de primeira ordem é denotada por

$$\nabla x_t = x_t - x_{t-1} \tag{2.5}$$

O gráfico da função de autocorrelação também é útil para identificar séries temporais não estacionárias. Para uma série estacionária, a ACF irá decair para zero com relativa rapidez, enquanto que a ACF de uma série não estacionária irá decrescer lentamente. Também, para uma série não estacionária, o valor de r_1 é frequentemente alto e positivo (HYNDMAN; ATHANASOPOULOS, 2018; SHUMWAY; STOFFER, 2017).

Algumas vezes, a série diferenciada ainda não será estacionária e pode ser necessário diferenciar a série por uma segunda ou até mais vezes para se chegar a uma série estacionária.

Teste de Raiz Unitária

Uma forma objetiva de se determinar a estacionariedade de uma série é usar um teste de raiz unitária. É um teste de hipótese de estacionariedade desenvolvido para se determinar a necessidade de diferenciação. Existem diversos testes de raiz unitária, baseados em pressupostos diferentes e que podem levar a resultados conflitantes. Entre esses testes estão o Augmented

Dickey Fuller Test (ADF) e o teste de Kwiatkowski-Phillips-Schmidt-Shin (KPSS). Nesses testes, a hipótese nula é que os dados são estacionários e procura-se evidências de que essa hipótese seja falsa. Consequentemente, pequenos valores de p (menores do que 0.05) sugerem que a diferenciação da série temporal seja necessária (HYNDMAN; ATHANASOPOULOS, 2018).

2.2 Predição em Séries Temporais com Modelos Autoregressivos Integrados de Médias Móveis (ARIMA)

Essa seção introduz uma classe de modelos que podem oferecer predições precisas de séries temporais baseada em padrões históricos implícitos nos dados. Os modelos autoregressivos integrados de médias móveis (ARIMA - do inglês *autoregressive integrated moving average*) constituem-se em uma das abordagens mais utilizadas para predição de séries temporais (BOX *et al.*, 2015). Os modelos ARIMA são uma classe de modelos lineares capazes de representarem séries temporais tanto estacionárias como não estacionárias. Esse modelos dependem pesadamente dos padrões de autocorrelações existentes nas séries temporais e buscam, assim, descrevê-los. A metodologia para identificar, ajustar e avaliar a adequação de modelos ARIMA teve grandes avanços com os trabalhos de G. E. P. Box e G. M. Jenkins (HANKE, 2014). Os modelos ARIMA são compostos por três partes: a parte autoregressiva (AR), a parte de integração (I) e a parte de médias móveis (MA) (HYNDMAN; ATHANASOPOULOS, 2018).

2.2.1 Modelos Autoregressivos (AR)

Modelos autoregressivos são baseados na ideia de que o valor atual de uma série temporal, x_t, pode ser explicado em função de valores passados, $x_{t-1}, x_{t-2}, \ldots, x_{t-m}$ (SHUMWAY; STOFFER, 2017). Convém lembrar que autocorrelação implica que valores da variável dependente em um período de tempo estão linearmente relacionadas a valores dessa variável em um outro período de tempo. Em uma abordagem de regressão, a variável dependente defasada em um ou mais períodos de tempo pode ser usada como variável independente ou preditora (HANKE, 2014). Assim, em modelos autoregressivos, a variável de interesse é prevista utilizando-se uma combinação linear dos valores passados da variável (BOX *et al.*, 2015). Segundo Hyndman e Athanasopoulos (2018), o termo *auto*regressão significa regressão de uma variável contra ela mesmo. Assim, um modelo autoregressivo de ordem p pode ser descrito como:

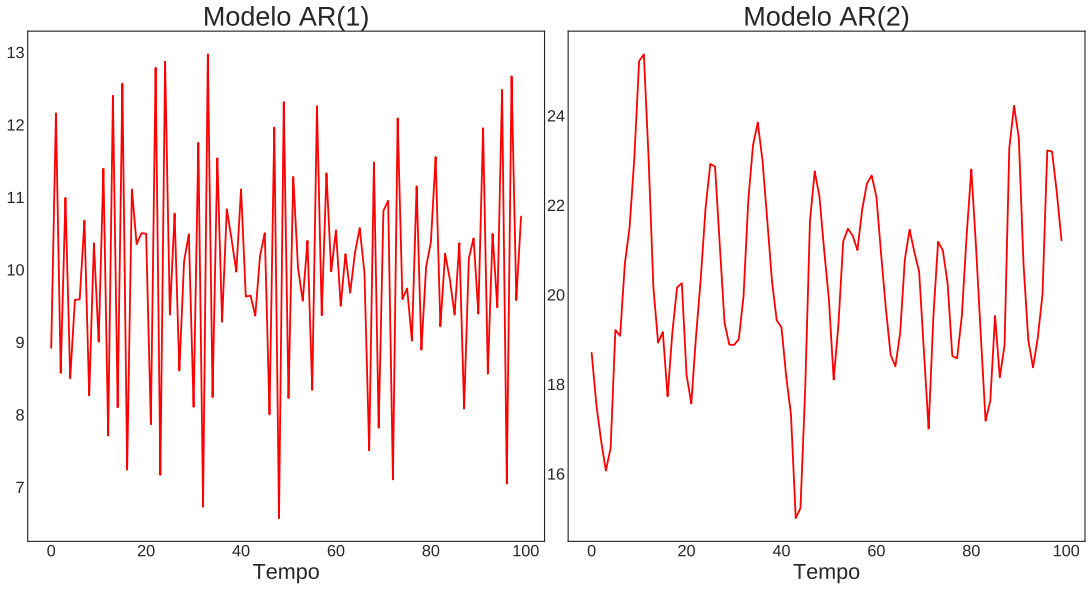
$$x_t = c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t, \tag{2.6}$$

onde ε_t é o ruído. Presume-se que o erro possui as mesmas propriedades de um modelo usual de regressão (HANKE, 2014).

Em um modelo de autoregressão, as predições são expressadas em função de valores passados da série temporal (HANKE, 2014). Isso seria como uma multiregressão mas com valores defasados de x_t como preditores. Esse modelo é conhecido como modelo autoregressivo

de ordem p, ou **modelo AR(p)**. Os modelos autoregressivos são bastante flexíveis podendo lidar com diversos padrões das séries temporais (HYNDMAN; ATHANASOPOULOS, 2018). Além disso, formam um subconjunto dos modelos ARIMA (HANKE, 2014), detalhados mais adiante. A Figura 5 mostra dois exemplos de modelos AR(p), sendo um do modelo AR(1) e outro do modelo AR(2).

Figura 5 – Dois exemplos de dados originados a partir de modelos autoregressivos com diferentes parâmetros. Esquerda: AR(1) com $x_t = 18 - 0.8x_{t-1} + \varepsilon_t$. Direita: AR(2) com $x_t = 8 + 1.3x_{t-1} - 0.7x_{t-2} + \varepsilon_t$. Em ambos os casos, ε_t é um ruído branco normalmente distribuído com média igual a zero e variância igual a 1.



2.2.2 Modelos de Médias Móveis (MA)

Os modelos de médias móveis (MA) fazem predições de séries temporais baseados na combinação linear entre erros passados, enquanto que modelos autoregressivos (AR) fazem predições por meio de funções lineares de valores passados (HANKE, 2014). Assim, segundo Hyndman e Athanasopoulos (2018), ao invés de utilizar valores passados da variável a ser prevista na regressão, o modelo de médias móveis utiliza os erros passados em um modelo de regressão:

$$x_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q}, \tag{2.7}$$

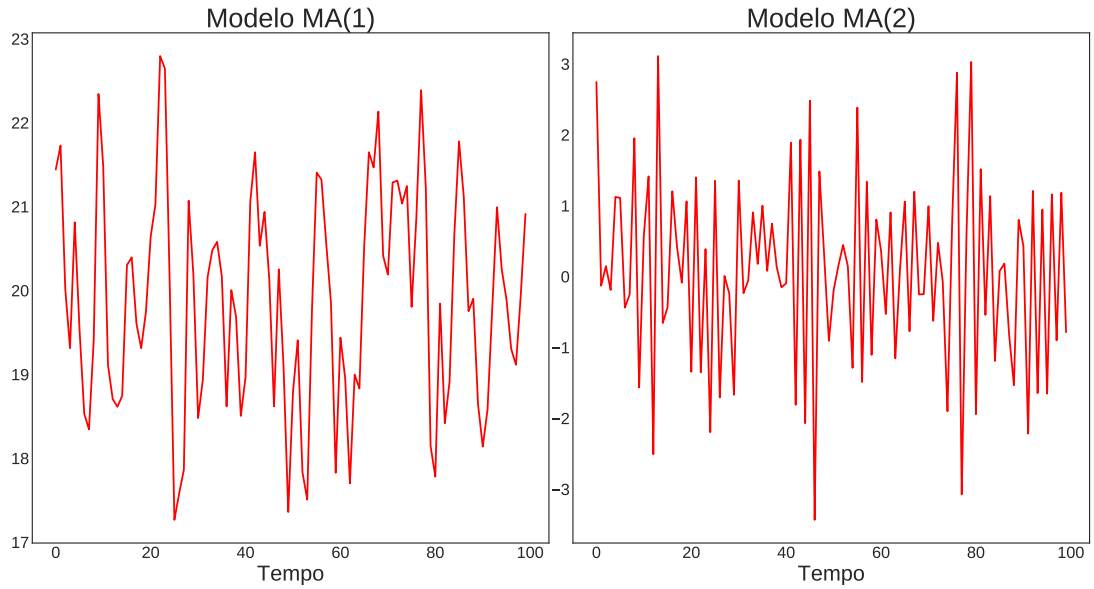
onde ε_t é um ruído. Esse modelo é referido como **modelo MA(q)**, ou modelo de médias móveis de ordem q (MILLS, 2019).

O termo *médias móveis* utilizados aqui não pode ser confundido com o simples procedimento de médias móveis. No presente caso, o termo médias móveis referem-se ao fato de que o desvio da resposta em relação à sua média, $x_t - \mu$, é uma combinação linear entre erros presentes e passados e que, à medida que o tempo avança, os erros envolvidos nessa combinação linear avançam também (HANKE, 2014).

Assim, é possível observar que cada valor x_t pode ser visto como resultado de médias móveis ponderadas de alguns erros passados. Por isso, modelos de médias móveis não devem ser confundidos com, por exemplo, procedimentos de suavização por médias móveis. Modelos de médias móveis são utilizados para previsões de valores futuros, enquanto que a suavização por média móveis é utilizada para estimar valores passados da tendência (HYNDMAN; ATHANASOPOULOS, 2018).

A Figura 6 mostra dados de modelos MA(1) e MA(2). Alterando-se os parâmetros $\theta_1, \ldots, \theta_q$ resultará em séries temporais com padrões diferentes. Assim como os modelos autoregressivos, a variância do termo de erro ε_t irá somente alterar a escala da série, mas não seus padrões.

Figura 6 – Dois exemplos de dados originados a partir de modelos de médias móveis com diferentes parâmetros. Esquerda: MA(1) com $x_t = 20 + \varepsilon_t + 0.8\varepsilon_{t-1}$. Direita: MA(2) com $x_t = \varepsilon_t - \varepsilon_{t-1} + 0.8\varepsilon_{t-2}$. Em ambos os casos, ε_t é um ruído branco normalmente distribuído com média igual a zero e variância igual a 1.



2.2.3 Modelos ARIMA não Sazonais

Com a combinação entre diferenciação, autoregressão e médias móveis, obtém-se o modelo ARIMA não sazonal (HYNDMAN; ATHANASOPOULOS, 2018). Nesse contexto, integração é a operação reversa da diferenciação. O modelo é descrito por:

$$x'_t = c + \phi_1 x'_{t-1} + \cdots + \phi_p x'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \tag{2.8}$$

onde x'_t é a série diferenciada (pode ser diferenciada mais de uma vez). Os "preditores"incluem tanto os valores defasados de x_t e os erros defasados. Esse modelo é conhecido como **modelo ARIMA**(p,d,q), onde p é a ordem do modelo AR, d é o grau de diferenciação envolvida e q é a ordem do modelo MA.

As mesmas condições de estacionariedade que são empregadas nos modelos autoregressivos e de médias móveis também se aplicam aos modelos ARIMA. Inclusive, muitos dos modelos discutidos anteriormente são casos especiais do modelo ARIMA, conforme mostra o Quadro 1.

Quadro 1 – Casos especiais do modelo ARIMA.

Casos	Parâmetros
Processos puramente aleatórios	ARIMA(0,0,0)
Passeio aleatório	ARIMA(0,1,0)
Autorregressão	ARIMA(p,0,0)
Médias móveis	ARIMA(0,0,q)

2.2.4 Modelos ARIMA Sazonais

Os modelos ARIMA são capazes também de modelar dados sazonais. Um modelo ARIMA sazonal inclui termos sazonais ao modelo ARIMA visto anteriormente. É dado da seguinte forma:

$$\begin{array}{lcc} \text{ARIMA} & (p,d,q) & (P,D,Q)_m \\ & \uparrow & \uparrow \\ & \text{Parte não sazonal do modelo} & \text{Parte sazonal do modelo} \end{array} \tag{2.9}$$

onde m é o número de observações por ano. A parte sazonal do modelo consiste de termos que são similares aos termos não sazonais, mas envolve deslocamento do período sazonal. Por exemplo, um modelo ARIMA$(1,1,1)(1,1,1)_4$ é para dados trimestrais ($m = 4$)

2.2.5 Seleção da Ordem do Modelo: Critérios de Informação

O critério de informação Akaike (AIC) auxilia a determinar a ordem de um modelo ARIMA. Esse critério pode ser escrito como:

$$\text{AIC} = -2\log(L) + 2(p+q+k+1), \tag{2.10}$$

onde L é a verossimilhança dos dados, $k = 1$ se $c \neq 0$ e $k = 0$ se $c = 0$. O último termo do parênteses é o número de parâmetros do modelo (incluindo σ^2, a variância dos resíduos). Para os modelos ARIMA, o AIC corrigido (AIC_c) é dado por:

$$\text{AICc} = \text{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}, \tag{2.11}$$

e o critério de informação Bayesiano é dado por:

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p+q+k+1). \tag{2.12}$$

Bons modelos são obtidos ao se minimizar AIC, AIC_c ou BIC. Importante notar que esses critérios de informação são apropriados somente para indicar a ordem dos parâmetros p e q,

mas tendem a não ser apropriados como guias da ordem de diferenciação (d) de um modelo. Isso acontece porque a diferenciação altera os dados nos quais a verossimilhança é calculada, fazendo com que valores do AIC entre modelos com diferentes ordens de diferenciação sejam incomparáveis. Portanto, é necessária uma outra abordagem para se chegar em d.

2.3 Predições de Séries Temporais com *Deep Learning*

Aprendizado de máquina (*Machine Learning* e mais especificamente abordagens baseadas em apredizado profundo (*deep learning*) são técnicas emergentes em análise de dados situadas na área de inteligência artificial. Essas abordagens de aprendizado levam o processo de análise de dados a um outro patamar, no qual os modelos construídos são direcionados pelos dados e não pelos modelos. O aprendizado de máquina e as abordagens baseadas em aprendizado profundo abriram novas possibilidades para a análise de séries temporais (SIAMI-NAMINI; TAVAKOLI; NAMIN, 2019). No entanto, grande parte dos métodos de aprendizado de máquina possuem fundamentação teórica que assume apenas com distribuição de probabilidades fixas, não modelando a dinâmica dos dados como em fluxos de dados e séries temporais (MELLO; PONTI, 2018).

Redes neurais permitem o projeto de técnicas de aprendizagem profunda que alcançam excelentes resultados em uma ampla variedade de tarefas na área de aprendizado de máquina, tanto supervisionado como não supervisionado (PONTI *et al.*, 2017). Porém, a despeito dessa capacidade de aprendizado, as redes neurais tradicionais possuem limitações. Uma limitação importante é a suposição de independência nos dados. Após o processamento de cada exemplo, o estado atual da rede é inteiramente perdido. Se os dados de entrada foram gerados de forma independente, isso não traz nenhum problema. Porém, se os pontos de dados estão relacionados no tempo ou no espaço, a suposição de independência passa a ser inaceitável. Além disso, redes neurais tradicionais geralmente requerem que os exemplos sejam vetores de comprimento fixo (LIPTON; BERKOWITZ; ELKAN, 2015). De acordo com Murugan (2018), quando se pretende criar sistemas inteligentes para analisar e prever séries temporais, as redes neurais tradicionais ”feed forward” irão falhar. Nesse sentido é necessário adequar os modelos aos dados e problema em questão, evitando ser enganado pelo erro de treinamento já que redes neurais profundas tem alta capacidade de se ajustar aos dados de treinamento mas podem falhar ao generalizar para dados futuros, o que é ainda mais crítico quando se trata de séries temporais (PONTI *et al.*, 2021). Por exemplo, o uso de mecanismos de recorrência e auto-atenção permitem modelar dados sequenciais Ribeiro *et al.* (2020). Assim, tornou-se desejável ampliar o potencial de aprendizado das redes neurais para que pudessem modelar dados com estruturas temporal e sequencial, e que permitissem variações nos comprimentos de entrada e saídas de dados o que é possível alcançar em particular com redes neurais recorrentes (LIPTON; BERKOWITZ; ELKAN, 2015).

2.3.1 Redes Neurais Recorrentes (RNN)

As redes neurais recorrentes (RNN - do inglês *Recurrent Neural Network*) são uma extensão das redes neurais convencionais (*Feed Forward*), com a capacidade de lidar com sequências de comprimentos variados. Diferentemente das redes neurais convencionais, as quais geralmente não são capazes de lidar com dados sequenciais e suas entradas necessitam ser independentes, as RNNs oferecem *gates* que armazenam informações prévias. Essa memória especial da RNN é chamada de estados escondidos recorrentes (*recurrent hidden states*) e dão à RNN a capacidade para predizer o valor da próxima entrada da sequência de dados. Em teoria, uma RNN é capaz de aproveitar as informações sequenciais anteriores mesmo para sequências longas. Na prática, porém, devido às limitações de memória da RNN, o comprimento da informação sequencial é limitada a somente alguns passos para trás.

Para uma definição formal de redes neurais recorrentes, assume-se que $x = (x_1, x_2, x_3, ..., x_m)$ representa um sequência de comprimento m, h_t representa a memória da RNN no tempo t e o modelo de RNN atualiza suas informações de memória usando:

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b_t) \tag{2.13}$$

onde σ é uma função não linear (por exemplo, função logística ou sigmoidal, função hiperbólica tangente ou função unidade linear retificada), W_x e W_h são matrizes de pesos, e b_t é uma constante (viés) (SIAMI-NAMINI; TAVAKOLI; NAMIN, 2019).

Em geral, existem diversos tipos de RNNs: uma entrada para muitas saídas, muitas entradas para muitas saídas, e muitas entradas para uma saída. No entanto, o aprendizado com redes recorrentes pode ser especialmente desafiador devido às dificuldades de se aprender as dependências entre os dados no longo prazo. Dois problemas especialmente comuns são a explosão ou desvanecimento de gradientes. Esses problemas ocorrem quando os erros são propagados ao longo de diversas camadas. No caso do desvanecimento, a informação, ou o valor dos gradientes, tende a desvanecer ou desaparecer ao longo das camadas. Nesse caso, o algoritmo da RNN aloca valores cada vez menores (< 1) para a matriz de pesos. No problema de explosão de gradientes, a informação resultará em gradientes com valores extremamente altos. Isso ocorre quando o algoritmo da RNN aloca valores cada vez mais altos para a matriz de pesos (LIPTON; BERKOWITZ; ELKAN, 2015).

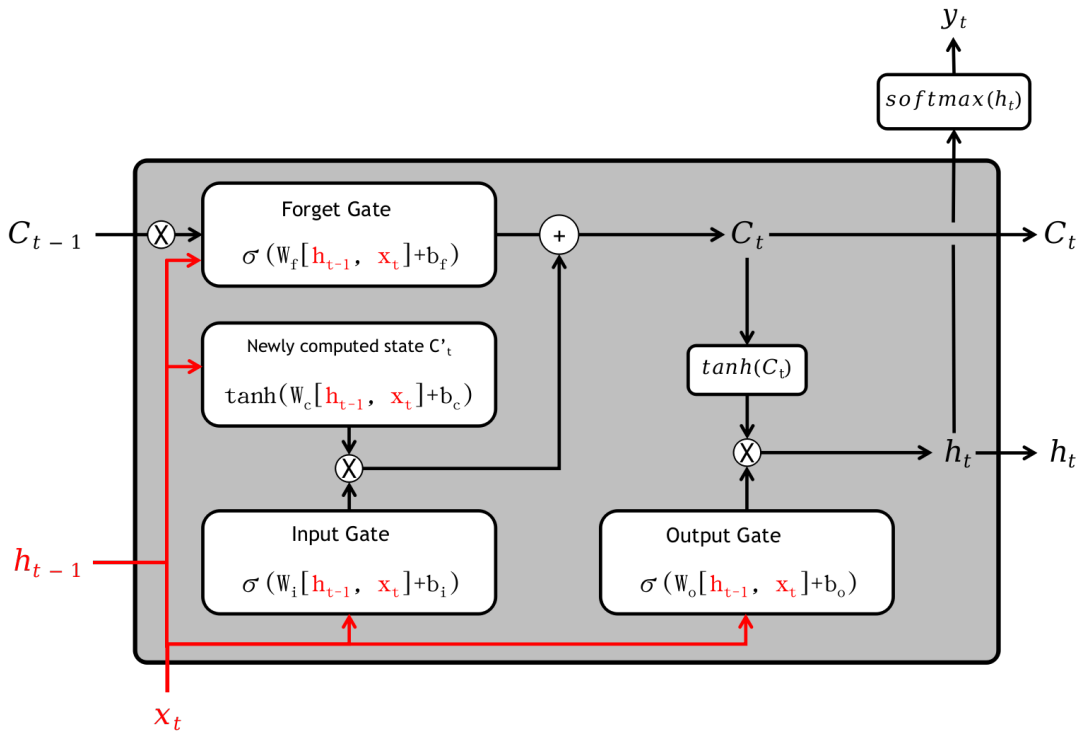
As RNN tem um longo histórico de aplicações em várias tarefas de aprendizado com dados sequenciais. Mas a despeito de seu sucesso inicial, a dificuldade de treinamento, até mesmo de simples RNN's, encorajou várias propostas de aprimoramento de sua arquitetura básica. Entre as variações mais bem sucedidas estão as redes LSTM (do inglês Long Short-Term Memory) e GRU (Gated Recurrent Unit), as quais podem, a princípio, armazenar e recuperar informações sobre longos períodos de tempo (KARPATHY; JOHNSON; FEI-FEI, 2015).

2.3.2 Redes LSTM (Long Short-Term Memory)

Redes recorrentes com memória de longo e curto prazos (LSTM) emergiram como um modelo efetivo e escalável para diversos problemas de aprendizado relacionados a dados sequenciais (HOCHREITER; SCHMIDHUBER, 1997). As redes LSTM são genéricas e efetivas em capturar dependências temporais de longo prazo. Elas não possuem as dificuldades de otimização que possuem as RNN e tem sido utilizadas para avançar o estado da arte para muitos problemas de elevada complexidade Greff *et al.* (2016), incluindo problemas de predição de séries temporais (como exemplo, veja (PARMEZAN; SOUZA; BATISTA, 2019; BALA; SINGH *et al.*, 2019; SAGHEER; KOTB, 2019), entre outros).

A ideia central por trás da arquitetura da LSTM é uma célula de memória que pode manter seu estado ao longo do tempo, e unidades não lineares chamadas de *gates* (portões) que regulam o fluxo de informações que entram e saem da célula. Um portão é uma unidade sigmoidal que gera uma ativação a partir da entrada atual x_t e da camada escondida anterior h_{t-1} (LIPTON; BERKOWITZ; ELKAN, 2015). Na LSTM, tem-se três portões: o portão do esquecimento, o portão de entrada e o portão de saída. Um bloco esquemático da LSTM pode ser visto na Figura 7.

Figura 7 – Esquema detalhado de uma rede neural recorrente LSTM.



A descrição abaixo acerca do funcionamento da LSTM foi baseada em Greff *et al.* (2016), Lipton, Berkowitz e Elkan (2015), Murugan (2018), Sagheer e Kotb (2019), Sezer, Gudelek e Ozbayoglu (2020), Siami-Namini, Tavakoli e Namin (2019). Deixe x_t ser o vetor de entrada no tempo t, e h_{t-1} a camada escondida anterior, no tempo $t-1$, e as seguintes matrizes de pesos

para uma camada LSTM:

- Pesos de entrada: $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_{C'}, \mathbf{W}_o \in \mathbb{R}^{N \times M}$
- Pesos dos vieses: $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_{c'}, \mathbf{b}_o \in \mathbb{R}^N$

A primeira etapa da LSTM é o portão do esquecimento, responsável por descartar informações que não são mais necessárias ao estado da célula. Esse portão recebe a entrada x_t no tempo t e h_{t-1}, multiplicados por matrizes de peso W_f e somados a um viés b_f, que são processados por uma função logística, produzindo como saída valores entre 0 e 1. Cada valor do estado da célula anterior C_{t-1} é, então, multiplicado pelo resultado do portão de esquecimento. Isso determina o quanto do estado da célula anterior irá ser utilizado no estado da célula atual. O valor 1 implica que a informação será completamente preservada, e o valor 0 significa que a informação será totalmente descartada. A equação do portão de esquecimento é dada por

$$f_t = \sigma(\mathbf{W}_f[h_{t-1}, x_t] + \mathbf{b}_f), \tag{2.14}$$

onde σ é a função logística de ativação

$$\sigma(x) = \frac{1}{1 + \exp -x}. \tag{2.15}$$

A segunda etapa no processamento de uma LSTM implica em decidir quanto da nova informação será mantida no estado da célula. Esse portão possui duas partes. Na primeira parte, tem-se o portão de entrada. Esse portão recebe a entrada x_t no tempo t e h_{t-1}, multiplicados por matrizes de peso W_i e somados a um viés b_i, que também são processados por uma função logística, produzindo como saída valores entre 0 e 1. Esse portão é dado pela seguinte equação:

$$i_t = \sigma(\mathbf{W}_i[h_{t-1}, x_t] + \mathbf{b}_i). \tag{2.16}$$

Na segunda parte, um novo estado C'_t é computado por meio de uma função tangente hiperbólica e multiplicado pelo resultado do portão de entrada. Esse novo estado é dado por:

$$C'_t = \tanh(\mathbf{W}_{c'}[h_{t-1}, x_t] + \mathbf{b}_c), \tag{2.17}$$

onde tanh é a função tangente hiperbólica de ativação

$$\tanh(x) = \frac{\exp^{2x} - 1}{\exp 2x + 1}. \tag{2.18}$$

Essas duas fases geram uma atualização para o estado da célula. Ou seja, o estado da célula anterior C_{t-1} é atualizado para um novo estado C_t, adicionando-se $i_t \odot C'_t$ a $f_t * C_{t-1}$, calculado previamente na primeira etapa. O novo estado C_t é, então, dado por

$$C_t = f_t * C_{t-1} + i_t \odot C'_t, \tag{2.19}$$

onde $\odot$ denota a multiplicação entre dois vetores.

E finalmente calcula-se a saída da LSTM. Primeiro, passa-se o estado da célula por uma função tangente hiperbólica para achatar os valores entre -1 e 1. Em seguida, esse valor é multiplicado pelo resultado do portão de saída O_t, responsável por decidir quanto do estado atual estará presente na saída da LSTM. O portão de saída O_t é dado por

$$o_t = \sigma(\mathbf{W}_o[h_{t-1}, x_t] + \mathbf{b}_o), \tag{2.20}$$

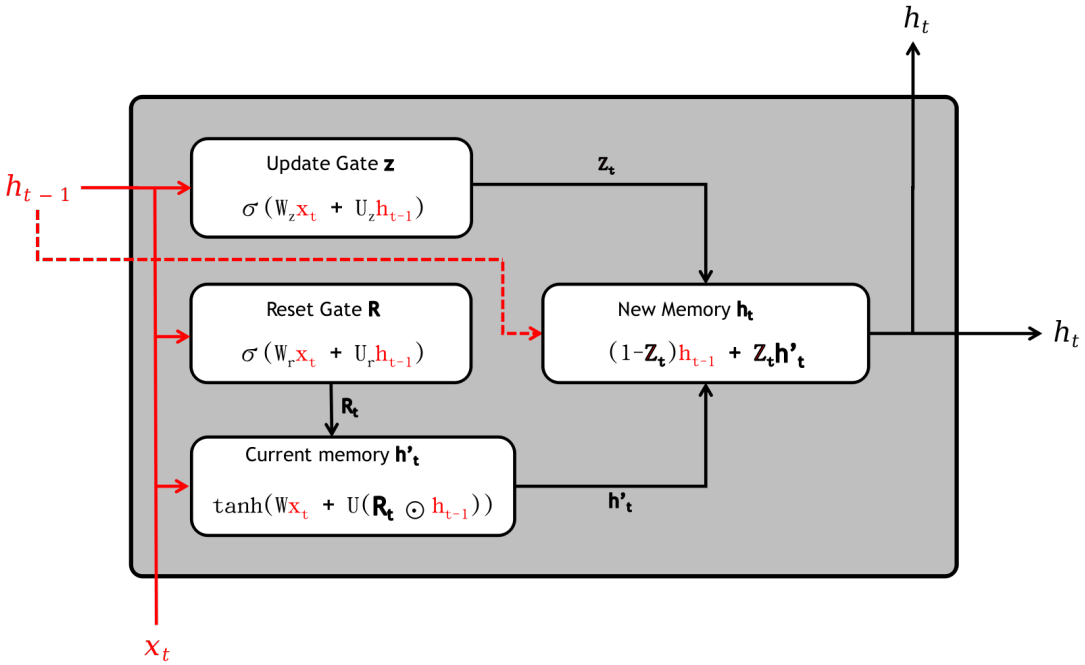
e a saída h_t é dada por

$$h_t = o_t \odot \tanh(C_t). \tag{2.21}$$

2.3.3 Redes GRU (Gated Recurrent Unit)

A GRU é uma rede recorrente que apresenta características semelhantes às da LSTM, porém com uma estrutura mais simples (GAO *et al.*, 2020). Ela foi introduzidas por Cho *et al.* (2014) e vem apresentando resultados satisfatórios. Uma representação do funcionamento geral da GRU é apresentado na figura 8. Nesta seção, será apresentado o funcionamento dessa rede.

Figura 8 – Esquema detalhado de uma rede neural recorrente GRU.



Assim como a LSTM, a GRU também possui unidades com portões que modulam o fluxo de informação. Porém, a GRU não faz separação entre as células de memórias (CHUNG *et al.*, 2014). Conforme mostra a figura 8, a ativação h_t da GRU no tempo t é uma interpolação linear entre a ativação prévia h_{t-1} e o candidato à ativação h'_t:

$$h_t = (1 - z_t)h_{t-1} + z_t h'_t \tag{2.22}$$

onde o portão de atualização (*update gate*) z_t decide o quanto a unidade irá atualizar sua ativação, ou conteúdo. O portão de atualização é computado da seguinte forma:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{2.23}$$

Esse procedimento de calcular a soma linear entre o estado existente e o novo estado computado é similar ao que ocorre na LSTM. No entanto, a GRU não possui um mecanismo que controla o grau com que o estado é exposto e expõe o estado inteiro a cada ativação. O candidato a ativação h'_t é calculado de forma similar ao da unidade recorrente tradicional:

$$h'_t = \tanh(W x_t + U(r_t \odot h_{t-1})) \tag{2.24}$$

onde r_t é o portão de *reset* e $\odot$ é uma multiplicação elemento a elemento. Quando desligado (r_t próximo a zero), este portão efetivamente faz a unidade agir como se estivesse lendo o primeiro símbolo de uma sequência de entrada, permitindo à unidade esquecer o estado computado anteriormente. O portão de *reset* r_t é calculado de forma similar ao portão de esquecimento:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{2.25}$$

2.3.4 Predição de Séries Temporais com Deep Learning

Na literatura, é possível encontrar diversos trabalhos nas mais variadas áreas que utilizaram modelos de deep learning e, mais especificamente, redes neurais LSTM e GRU, para predição de séries temporais. Por exemplo, Gao *et al.* (2020) utilizaram redes GRU e LSTM para predizer enchentes a partir de séries temporais com dados sobre escoamento de chuvas. Em outro estudo, conduzido por Zhang *et al.* (2019), foram pesquisados processos químicos em contexto temporal, importantes para o monitoramento da qualidade e segurança durante o processo de produção na indústria de refinamento de óleo e processos metalúrgicos. Para esse estudo, foram comparadas algumas variações de LSTM, que se mostraram eficientes na extração de informações importantes para os processoas acima mencionados.

Além disso, as redes LSTM também têm sido aplicadas em contextos financeiros. Em um estudo de Bala, Singh *et al.* (2019), foram utilizadas redes LSTM para predição de séries temporais financeiras não-estacionárias, tanto para prazos curtos quanto para prazos longos. O estudo comparou um modelo de LSTM desenvolvido pelos autores com outros modelos, tais como *Extreme Learning Machine*, Redes Neurais Artificiais convencionais (MLP), *Online Suport Vector Regression*, entre outros, e demonstrou que o modelo de LSTM desenvolvido pelos autores obteve performance superior. Outro estudo nessa área é o de Cao, Li e Li (2019), que também utilizaram a LSTM para predição de séries temporais financeiras. Nesse estudo, os autores tiveram como objetivo aprimorar a acurácia de predições para o mercado de ações por meio do desenvolvimento de um modelo de LSTM. A dificuldade, nesse caso, é que séries temporais com preços de ações, além de não-estacionárias, são aleatórias. Os autores concluíram

que, quando comparado com outros modelos, como LSTM tradicional, *Suport Vector Machine* e *Multi-Layer Perceptron*, o modelo desenvolvido no estudo demonstrou resultados mais eficazes para prazos mais curtos. Diversos outros estudo também foram realizados com séries temporais de preços de ações, entre eles: Althelaya, El-Alfy e Mohammed (2018), Choi (2018), Fischer e Krauss (2018), Yoshihara, Seki e Uehara (2016).

O uso de LSTM para predição de demanda também é bastante vasto. Entre esses estudos destacam-se dois realizados pela empresa Uber. Em um desses estudo, realizado por Laptev *et al.* (2017), são utilizadas redes LSTM para predição de eventos extremos. No caso da Uber, os eventos extremos são, por exemplo, feriados e o objetivo é predizer a demanda nesses períodos pode levar a uma alocação mais eficiente de motoristas resultando em tempo de espera menor pelos usuários. No outro estudo dessa empresa, Zhu e Laptev (2017) propuseram um modelo bayesiano de LSTM para fazer predições em séries temporais com o mesmo objetivo do estudo descrito anteriormente. Outro estudo que visa realizar previsões de demanda com modelo otimizado de LSTM é aquele desenvolvido por Abbasimehr, Shabani e Yousefi (2020). Nesse estudo, os autores propuseram uma metodologia para encontrar automaticamente a melhor arquitetura de LSTM para um problema de predição de demanda. O modelo de LSTM encontrado pela metodologia foi comparado com diversos outros modelos, tanto estatísticos (ARIMA e Suavização Exponencial), quanto de aprendizado de máquina (MLP, KNN, RNN e LSTM convencional). O modelo aplicado com a arquitetura encontrada pela metodologia utilizada no estudo apresentou melhor desempenho.

Além desses exemplos, as redes neurais LSTM e GRU têm sido utilizadas também em diversas outras situações de predições de séries temporais, como por exemplo:

- predição de séries temporais para produção de petróleo: Sagheer e Kotb (2019).
- Predição de demanda no turismo: Law *et al.* (2019)
- reconhecimento de atividades humanas (*wearable*): Ordóñez e Roggen (2016)
- Outros estudos mais genéricos: Bianchi *et al.* (2017), Sezer, Gudelek e Ozbayoglu (2020), Yunpeng *et al.* (2017)

Também é válido mencionar o trabalho de Tealab (2018), que realizou uma revisão sistemática da literatura sobre a utilização de redes neurais para predição de séries temporais, analisando trabalhos publicados entre 2006 e 2016.

Todos esses estudos demonstram a importância que as redes neurais recorrentes LSTM e GRU vêm adquirindo atualmente em diversos contextos, inclusive econômicos e financeiros.

2.3.5 Comparação entre Modelos Estatísticos e Deep Learning na Predição de Séries Temporais

Diversos são os trabalhos que visam determinar quais os melhores métodos para predição de séries temporais, tecendo comparações entre métodos estatísticos e métodos de deep learning. Um desses estudos é o realizado por Song *et al.* (2020), que propôs o uso de um modelo baseado na LSTM para predizer a produção poços horizontais fraturados em uma reserva vulcânica. O estudo comparou o modelo baseado em LSTM com diversos outros modelos, inclusive modelos estatístico tais como o modelo ARMA e o modelo ARIMA. Os resultados do experimento demonstraram que o modelo proposto de LSTM superam os resultados dos outros modelo, inclusive aqueles dos modelos estatísticos. Segundo os autores, o modelo de LSTM proposto pode capturar com mais precisão as complexas variações nos padrões das séries temporais utilizadas no estudo. Além disso, a capacidade de generalização do modelo proposto é maior ao fazer predições para dados nunca vistos.

Parmezan, Souza e Batista (2019) apresentam, segundo os próprios autores, uma das mais extensiva, imparcial e compreensível avaliação experimental sobre a área de predições de séries temporais. Os autores utilizaram 95 conjuntos de dados para avaliar onze preditores, sendo sete paramétricos (Médias Móveis, Suavização Exponencial Simples, Suavização Exponencial de Holt, Suavização Exponencial Sazonal de Holt-Winter Aditivo e Multiplicativo, ARIMA e SARIMA) e quatro não-paramétricos (MLP, SVM, KNN e LSTM). Os resultados mostraram que o modelo estatístico SARIMA é o único capaz de superar, mas sem diferença estatística significativa, os seguintes modelos de aprendizado de máquina: Redes Neurais Artificiais, SVM, e kNN-TSPI. No entanto, ressaltam os autores, tal desempenho é obtido á custa do uso de um grande número de parâmetros. A LSTM não teve desempenho de destaque entre os modelos utilizados no estudo.

Já o estudo realizado por Siami-Namini, Tavakoli e Namin (2019) teve por objetivo comparar duas redes neurais recorrentes, LSTM e BiLSTM. No entanto, utilizaram como *baseline* o modelo estatístico ARIMA, o que acabou permitindo comparações entre a rede neural LSTM e o modelo ARIMA. Os resultados do estudo mostraram que a rede BiLSTM superou o desempenho da LSTM no contexto abordado na pesquisa. No entanto, o estudo também mostra que a LSTM superou os resultados do modelo ARIMA. Para esse estudo, foram utilizadas séries temporais de preços de ações.

Em Makridakis, Spiliotis e Assimakopoulos (2018), os autores apresentam os resultados e conclusões da Competição M4. A Competição M4 é uma continuação de três competições anteriores que se iniciaram há mais de 45 anos atrás com o propósito de verificar possibilidades de aprimoramento da acurácia de modelos para predições de séries temporais. Essa edição da competição recebeu 50 trabalhos válidos que ao todo utilizaram cem mil séries temporais. Segundo os próprios autores, esse pode ter sido o elevado número de bases de dados pode ter desencorajado a aplicação de modelos de *Deep Learning* mais complexos devido ao alto

custo de processamento. Mas, a despeito disso, os cinco maiores resultados dessa competição são: 1. Dos 17 trabalhos com melhor desempenho, 12 foram combinações de modelos que tiveram, em sua maioria, abordagens estatísticas; 2. A maior surpresa foi uma abordagem híbrida que combinou modelos estatísticos com aprendizado de máquina; 3. A segunda abordagem com melhor desempenho foi uma combinação de 7 modelos estatísticos com um modelo de aprendizagem de máquina; 4. Os dois métodos com melhor desempenho também alcançaram grande sucesso ao especificarem corretamente 95% dos intervalos de predição; 5. Cinco modelos puros de aprendizado de máquina tiveram desempenho pobre, com nenhum deles sendo mais preciso do que a combinação utilizada como *benchmark* (uma combinação baseada na média simples aritmética dos modelos de suavização exponencial simples, de Holt e *damped*) e somente um teve melhor desempenho que o modelo estatístico *Naïve*.

2.4 Detecção de Anomalias em Séries Temporais

Detecção de outliers tem se tornado um campo de interesse para muitos pesquisadores e é agora uma das principais tarefas quando se trata de análises de séries temporais (BLÁZQUEZ-GARCÍA *et al.*, 2020). A detecção de outliers têm sido estudada e aplicada em diversas áreas (AGGARWAL, 2017; CHANDOLA; BANERJEE; KUMAR, 2009) , tais como detecção de fraudes, detecção de invasões em cybersegurança, em diagnóstico de falhas na área industrial, na área financeira e na área da saúde, entre outras (GUPTA *et al.*, 2014; MEHROTRA; MOHAN; HUANG, 2017). No caso de detecção de outliers em séries temporais, o objetivo é examinar comportamentos anômalos ao longo do tempo (GUPTA *et al.*, 2014). Muitas definições para o termo *outlier* e vários métodos para detecção têm sido propostos na literatura. Portanto, parece ainda não haver consenso nas definições e diversos termos têm sido utilizados para se referir a esse fenômeno, como por exemplo, anomalias, observações discordantes, discordantes, exceções, aberrações, surpresas, peculiaridades ou contaminações (CARREÑO; INZA; LOZANO, 2019). No presente trabalho, serão utilizados os termos *outlier* e anomalia, como sinônimos.

Uma definição clássica para *outlier* é aquela dada por Hawkins (1980), que diz que outliers, ou anomalias, são observações que se desviaram tanto de outras observações do conjunto de dados ao ponto de levantarem suspeitas de que foram geradas por algum outro mecanismo. Portanto, anomalias ou *outliers* são variações substanciais da norma (MEHROTRA; MOHAN; HUANG, 2017) e detectá-las implica no problema de encontrar padrões nos dados que não estejam em conformidade com o comportamento esperado (INOUE *et al.*, 2017; TRAN; NGUYEN; THOMASSEY, 2019). Assume-se, também, que os valores normais dos dados estejam localizados nas regiões de alta probabilidade do modelo estocástico, enquanto que as anomalias encontram-se nas regiões de baixa probabilidade (DAVIS; RAINA; JAGANNATHAN, 2020).

Colocado de outra forma, conforme Mehrotra, Mohan e Huang (2017), muitos campos da ciência estão baseados na suposição de que existem processos e comportamentos na natureza

que seguem certas regras e princípios amplos, resultando no estado de determinado sistema. Esse estado é observável por meio de dados, a partir dos quais são formuladas hipóteses sobre a natureza do processo subjacente. Essas hipóteses visam descrever o comportamento normal de um sistema, assumindo-se implicitamente que os dados utilizados para gerar as hipóteses sejam típicos daquele sistema sobre estudo. Assim, outliers seriam variações desse comportamento normal. Para esse autores, o objetivo da tarefa de detecção de outliers é descobrir tais variações do comportamento normal nos dados observados. A ideia de continuidade, de acordo com Aggarwal (2017), desempenha um papel importante do problema de detecção de outliers. Ou seja, espera-se que os padrões nos dados não se alterem abruptamente, a menos que algum processo anormal comece a operar. Na verdade, como bem colocam o autor, quanto mais complexo os dados, mais o analista deve realizar inferências preliminares sobre o que é considerado normal para as séries em estudo.

De acordo com Aggarwal (2017), no contexto de séries temporais, a detecção de outliers pode ter dois diferentes objetivos, dependendo da finalidade para a qual a detecção é realizada. Em alguns casos, os outliers são dados indesejados e busca-se conhecê-los com o objetivo de suprimi-los e promover uma limpeza no conjunto de dados. Em outros casos, principalmente na área de séries temporais, o objetivo pode ser detectar e analisar um fenômeno incomum. Detecção de fraudes é um exemplo recente disso. Nesse caso, o objetivo da detecção é o próprio outlier, visando conhecer melhor suas características.

De acordo com Blázquez-García *et al.* (2020), as técnicas de detecção de outliers em séries temporais dependem de três aspectos:

- O tipo dos dados de entrada: Esse primeiro aspecto descreve os tipos de dados de entrada que o método de detecção de outliers terá que lidar. Nesse caso, tem-se que a série temporal tida como dado de entrada pode ser univariada ou multivariada (BLÁZQUEZ-GARCÍA *et al.*, 2020). Segundo Gupta *et al.* (2014), o tipo de dados da série temporal terá impacto significativo sobre o método de detecção de outliers a ser escolhido.

- O tipo de outlier: O segundo aspecto descreve o tipo de outlier que o método de detecção visa encontrar. Os outliers podem ser pontos isolados em séries temporais, ou seja, quando um valor da série apresenta comportamento incomum em um instante específico no tempo, em comparação com outros valores da série temporal. Os outliers podem ser também uma sequência de pontos, ou seja, pontos consecutivos no tempo que apresentam, quando considerados em conjunto, comportamento anormal, mesmo que cada observação isolada não se constitua em um outlier pontual. Ou ainda uma série temporal inteira pode ser considerada outlier, que poderá ser detectada somente quando os dados de entrada forem séries temporais multivariadas (BLÁZQUEZ-GARCÍA *et al.*, 2020). De forma semelhante, em Aggarwal (2017), outliers são classificados como contextuais ou coletivos. Os outliers são contextuais quando valores mudam em momentos específicos do

tempo, e são coletivos quando séries temporais inteiras ou uma sequência longa de valores apresentam comportamento incomum.

- A natureza dos métodos de detecção de outliers: o terceiro aspecto analisa a natureza dos métodos de detecção de outliers empregados, ou seja, se o método de detecção empregado é univariado ou multivariado. Um método univariado irá considerar somente uma série temporal enquanto que métodos multivariados podem considerar, simultaneamente, mais de uma série temporal. Nota-se que o método pode ser univariado mesmo quando a série for multivariada. Nesse caso, o método considerará uma série por vez.

Gupta *et al.* (2014) descrevem os diferentes aspectos da análise de outliers em séries temporais de uma forma um pouco mais abrangente. Os autores denominam esses aspectos como facetas da análise de outliers e enfatizam que a área é tão rica que nenhuma categorização sozinha pode capturar totalmente a complexidade dos problemas que surgem nessa área, uma vez que tais facetas podem ser combinadas de formas abstratas. Segundos os autores, algumas dessas facetas são:

- Séries temporais versus dados multi-dimensionais: em séries temporais, a questão da continuidade é importante. Por outro lado, a análise de dados multi-dimensionais pode não depender de aspectos temporais;
- Ponto *versus* window: com a análise de outliers, busca-se detectar um ponto na série temporal ou um padrão incomum de mudanças?
- O tipo de dado: diferentes tipos de dados (séries contínuas, séries discretas, fluxos de dados multi-dimensionais, dados de redes sociais etc) requerem diferentes tipos de métodos de detecção de outliers;
- Supervision: Existem exemplos de outliers previamente detectados?

Quando métodos de detecção de outliers são utilizados, três resultados possíveis precisam ser considerados:

1. Detecção correta: anomalias detectadas nos dados correspondem exatamente com anomalias no processo;
2. Falsos positivos: O processo ocorre normalmente, mas dados são classificados como outliers, provavelmente devido a ruídos do sistema; e
3. Falsos negativos: O processo se torna anormal, mas isso não é captado pelos algoritmos de detecção de outliers, em parte devido ao fato da anormalidade não ser consideravelmente forte quando comparada ao ruído do sistema.

A maioria dos sistemas de detecção de outliers, em situações reais, não conseguirão detectar corretamente 100% dos outliers. A tarefa do analista de dados envolve também reconhecer esse fato, e desenvolver mecanismos para minimizar os impactos dos falsos negativos e dos falsos positivos, permitindo, dependendo da situação, que um ocorra mais do que o outro (MEHROTRA; MOHAN; HUANG, 2017).

Quando comparada com outras áreas de análise de dados, como classificação e clusterização, a detecção de outliers apresenta seus próprios desafios. Alguns desse desafios foram mapeados por Gupta *et al.* (2014), conforme listados abaixo:

- Classificação faz uso de dados rotulados para aprender um classificador que classifique os dados. Detecção de outliers é uma tarefa não supervisionada. Técnicas de detecção de outliers aprendem similaridades entre os dados sem usar qualquer tipo de rótulo. Devido à sua natureza não supervisionada, a detecção de outliers em séries temporais pode ser um desafio.

- Clusterização está bastante relacionada com detecção de outliers. Clusterização é também uma tarefa não supervisionada que visa agrupar dados similares. Dados que não podem ser alocados em nenhum grupo tendem a ser ignorados pelo processo de clusterização. Esses dados podem ser avaliados como outliers. Clusterização temporal implica em manter a informação dos grupos ao longo do tempo. Detecção de outlier pode ser desafiadora por ter que identificar dados anômalos em combinação com propriedade temporais.

As abordagens de detecção de anomalias estão fundamentadas em modelos e predições de dados passados. Num primeiro momento, pode-se intuir que detecção de anomalias trata-se de um problema de classificação, ou seja, separa-se os dados em duas classes: anomalias e não anomalias. Nessa linha, pode-se supor que os algoritmos de aprendizado de máquina poderiam lidar com esse problema. No entanto, é provável que esses algoritmos não funcionem adequadamente, pois geralmente existe um problema de desbalanceamento drástico entre as duas classes: os dados anômalos são muito mais raros do que os dados normais (MEHROTRA; MOHAN; HUANG, 2017).

2.4.1 Métodos para Detecção de Outliers Isolados em Séries Temporais Univariadas

Nesta revisão, abordaremos métodos especificamente para detecção de outliers isolados em séries temporais univariadas, pois será esse o contexto sob análise nesta pesquisa. Outlier isolado é o caso mais comum de detecção de outliers na área de séries temporais. Assim, nesta subseção serão apresentados métodos para detecção desse tipo de outliers em séries temporais univariadas (BLÁZQUEZ-GARCÍA *et al.*, 2020). O problema de detecção de outliers em séries temporais univariadas também pode ser formulado como um problema onde se busca identificar

quando os parâmetros de uma série temporal são alterados (MEHROTRA; MOHAN; HUANG, 2017).

Duas características chave desses métodos são:

- Temporalidade: alguns métodos consideram o fator temporal inerente aos dados enquanto outros métodos ignoram essa informação. Nesse caso, os métodos aplicados não apresentarão os mesmos resultados caso os dados sejam embaralhados. Um subgrupo dos métodos que consideram a temporalidade utiliza uma janela de tempo. Nesse caso, os mesmos resultados são obtidos quando os dados são embaralhados dentro das janelas, mas não quando toda a série é embaralhada.
- Dados correntes (*stream*) ou não correntes: algumas técnicas conseguem detectar outliers à medida que os valores são captados ou produzidos. Nesse grupo, alguns métodos utilizam outliers um modelo fixo ao longo de toda a produção dos dados, enquanto outros atualização os modelos de detecção de outliers ao obter o novo dado.

De acordo com Gupta *et al.* (2014), diversos métodos de detecção de outliers tem sido propostos com o intuito de se detectar outliers isolados em séries temporais. Esses métodos de detecção de outliers, segundo Blázquez-García *et al.* (2020), podem ser organizados em diferentes categorias, tais como métodos baseados em modelos, métodos baseados em densidade e métodos que utilizam histogramização. Nesse subseção serão considerados os métodos baseados em modelos, pois são os utilizados na etapa experimental da presente pesquisa. Os métodos baseados em modelos podem ser divididos em duas subcategorias: modelos de estimação e modelos de predição.

Um outlier isolado pode ser definido como uma observação que desvia significantemente de seu valor esperado. Portanto, dada uma série univariada, um ponto no tempo t pode ser declarado como um outlier se a distância para seu valor esperado for maior do que um predeterminado limite τ

$$|x_t - \hat{x}_t| > \tau \tag{2.26}$$

onde x_t é o dado observado, e $\hat{x}_t$ é seu valor esperado. Os métodos de detecção de outliers que usam a estratégia da equação 2.26 são chamados de métodos baseados em modelos e é a abordagem mais abordada na literatura. Mesmo que cada método compute o valor esperado $\hat{x}_t$ e o limite τ de formas diferentes, todos eles são baseados no ajuste de um modelo.

Os métodos baseados em modelos de estimação são aqueles em que o valor esperado $\hat{x}_t$ é obtido com o uso tanto de observações anteriores como subsequentes a x_t. Se $\hat{x}_t$ é obtido somente por meio de valores anteriores a x_t, então tem-se que o método é baseado em modelos de predição. Na prática, a principal diferença entre estimação e predição é que as técnicas dos métodos baseados em modelos de predição podem ser aplicadas em séries temporais correntes (*streaming data*) porque são capazes de determinar se um novo dado é outlier ou não tão logo ele

é obtido. No caso de métodos de estimação, isso poderia ser feito se somente x_t fosse utilizado para calcular o valor esperado.

Alguns outros métodos de detecção de outliers univariados analisam os resíduos obtidos a partir do ajuste de diferentes modelos, tais como decomposição STL, ARIMA e modelos de regressão linear, para identificação dos ouliers. Embora esses modelos possam também ser utilizados para predição, nesse caso, os outliers são detectados por meio do resíduo. Assim, uma vez que o modelo selecionado é ajustado, são aplicados testes de hipóteses sobre o resíduo para detectar os outliers. Nesse caso, busca-se evidências para rejeitar a hipótese nula (um valor extremo não é um outlier) e dar suporte à hipótese alternativa de que um valor extremo é um outlier.

Em contraste com os modelos de estimação, os métodos baseados em modelos de predição ajustam um modelo à série temporal e obtêm $\hat{x}_t$ utilizando somente dados passados, ou seja, sem utilizar x_y ou qualquer outro valor posterior a ele (BLÁZQUEZ-GARCÍA *et al.*, 2020). Para Aggarwal (2017) e Inoue *et al.* (2017), a detecção de outliers isolados está tipicamente baseada no desvio dos valores da série em relação a um valor esperado ou predito. Ainda segundo Blázquez-García *et al.* (2020), valores que são bastante diferentes do predito são identificados como outliers. Esses métodos geralmente conseguem lidar com dados correntes. Assim, a detecção de outliers por meio desses métodos está bastante relacionada com o problema de predições de séries temporais, uma vez que os valores são declarados como outliers com base em valores preditos (AGGARWAL, 2017).

Dentro dessa categoria, pode-se utilizar modelos fixos que não são capazes de se adaptarem a mudanças que ocorrem na série de dados ao longo do tempo. Existem ainda outros métodos que usam modelos autoregressivos ou modelos ARIMA nos quais é possível encontrar intervalos de confiança para as predições ao invés de somente pontos de estimação. Dessa forma, esses métodos implicitamente definem o valor de τ. Outros métodos dessa categoria adaptam os modelos de predição à medida que os dados evoluem, retreinando os modelos. Esse métodos podem envolver desde modelos de predição simples, como médias móveis, como também modelos mais complexos, como os modelos ARIMA. Os modelos adotados por esses métodos podem retreinar seus modelos periodicamente ou somente quando algum dado novo é obtido (BLÁZQUEZ-GARCÍA *et al.*, 2020).

CAPÍTULO

3

METODOLOGIA

O objetivo deste trabalho envolveu estudar um modelo de detecção de outliers baseado nas capacidades preditivas das redes neurais LSTM e GRU. A diferença entre os valores preditos e os valores observados foram calculados como erros de predição e utilizados para detectar outliers em dados não vistos de três séries temporais univariadas de contexto econômico. Como linha de base para comparações, foi utilizado o modelo estatístico SARIMA. Primeiramente, utilizou-se um valor limite específico para detecção de outliers, calculado a partir dos erros de predição do conjunto de treinamento. Num segundo momento, os modelos foram testados com todos os valores limites possíveis para detecção de outliers.

3.1 Descrição das Séries Temporais

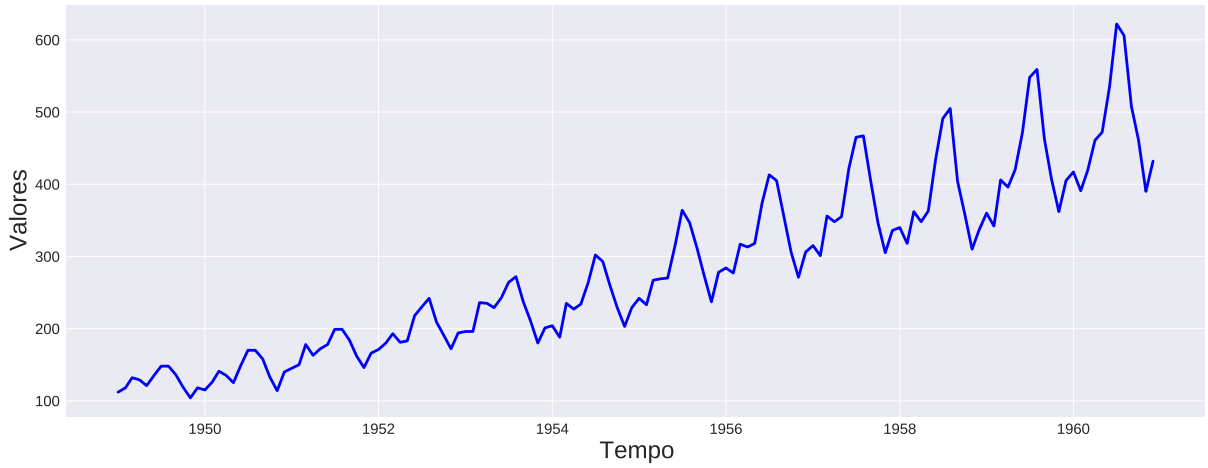
Para elaboração deste trabalho foram utilizadas três séries temporais univariadas reais e públicas, com observações mensais. Neste capítulo, inicialmente será feita uma descrição dessas séries, mostrando suas principais características tais como tendência, sazonalidade e estacionariedade. Em seguida, será feita uma descrição da processo experimental, detalhando os procedimentos utilizados para condução desta pesquisa.

3.1.1 International Airline Passengers

Contém dados mensais sobre o número total de passageiros internacionais nos Estados Unidos, no período de janeiro de 1949 a dezembro de 1960. No total, a série possui 144 observações, conforme ilustrado na Figura 9.

A decomposição dessa série nas componentes tendência e sazonalidade, além dos resíduos, pode ser vista na Figura 10. Pode-se observar, pelo gráfico superior na figura, que a série apresenta uma tendência de crescimento ao longo do tempo. O gráfico central mostra a componente sazonalidade. É possível notar que existe um padrão de sazonalidade, ou seja, a

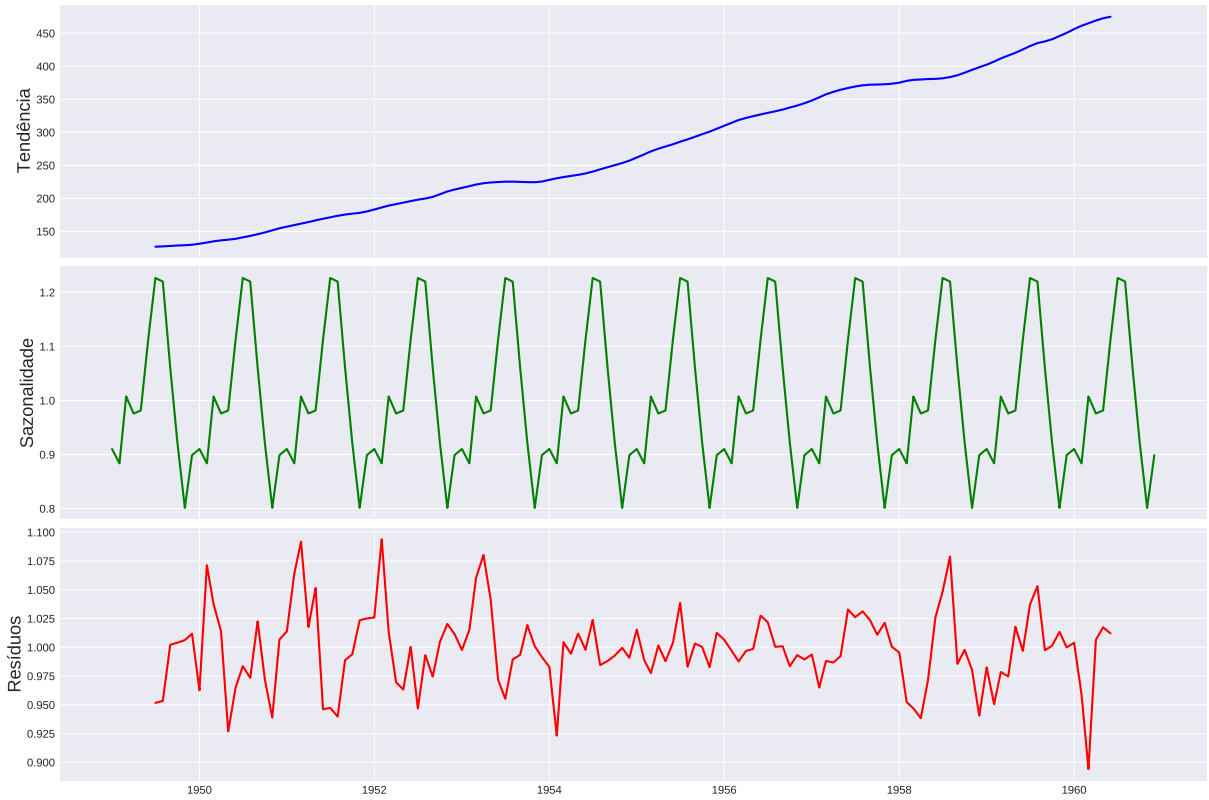
Figura 9 – Dados mensais do número total de passageiros.



Fonte: Elaborada pelo autor.

demanda aumenta e decresce nos mesmos períodos para todos as anos da série.

Figura 10 – Decomposição da série temporal "International Airline Passengers".

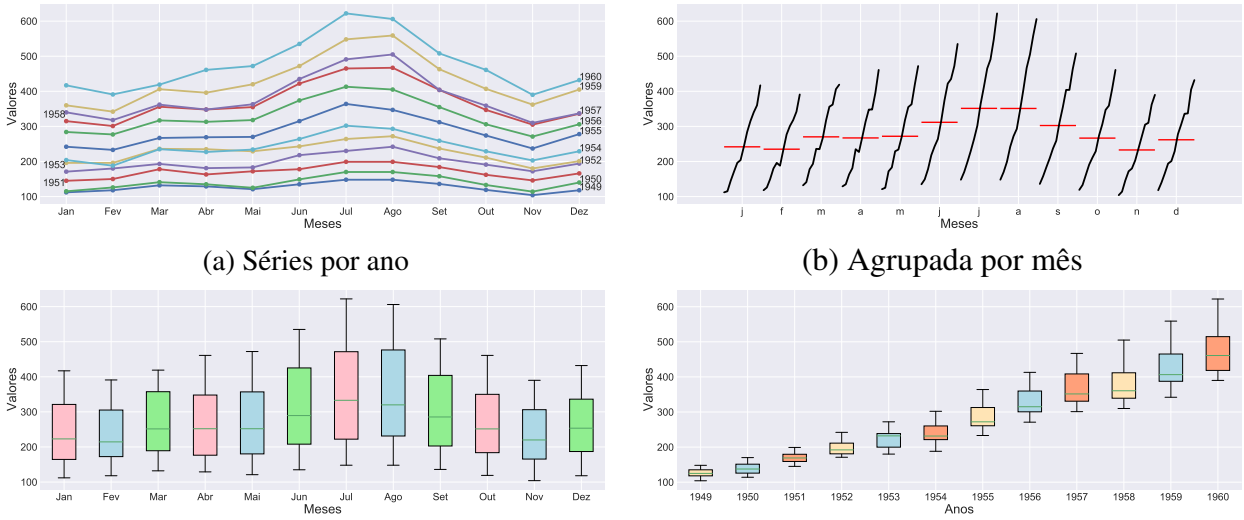


Fonte: Elaborada pelo autor.

O conjunto de gráficos apresentados na Figura 11 ajuda a compreender melhor o comportamento da série temporal. No gráfico da Figura 11 (a), cada linha representa um ano da

série. Percebe-se que a demanda apresenta praticamente o mesmo padrão para todos os anos: o número de passageiros aumenta de janeiro a julho, depois torna-se decrescente até novembro e em dezembro volta a aumentar. No gráfico da Figura 11 (b), é possível visualizar os dados agrupados por mês, evidenciando um aumento da volatilidade nos meses de junho a agosto, quando a demanda é maior. É possível notar que a média nesses meses é mais alta, denotada pelos traços vermelhos. O gráfico de caixas da Figura 11 (c) também mostra a existência desse mesmo comportamento. O gráfico de caixas da Figura 11 (d) apresenta a distribuição dos dados ao longo dos anos. Nesse gráfico, é possível notar que a volatilidade não é constante, aumentando ao longo dos anos.

Figura 11 – Gráfico mensais e anuais referentes à série temporal ”Passengers”.



(a) Séries por ano

(b) Agrupada por mês

(c) Valores agrupados por mês.

(d) Valores agrupados por ano.

Fonte: Elaborada pelo autor.

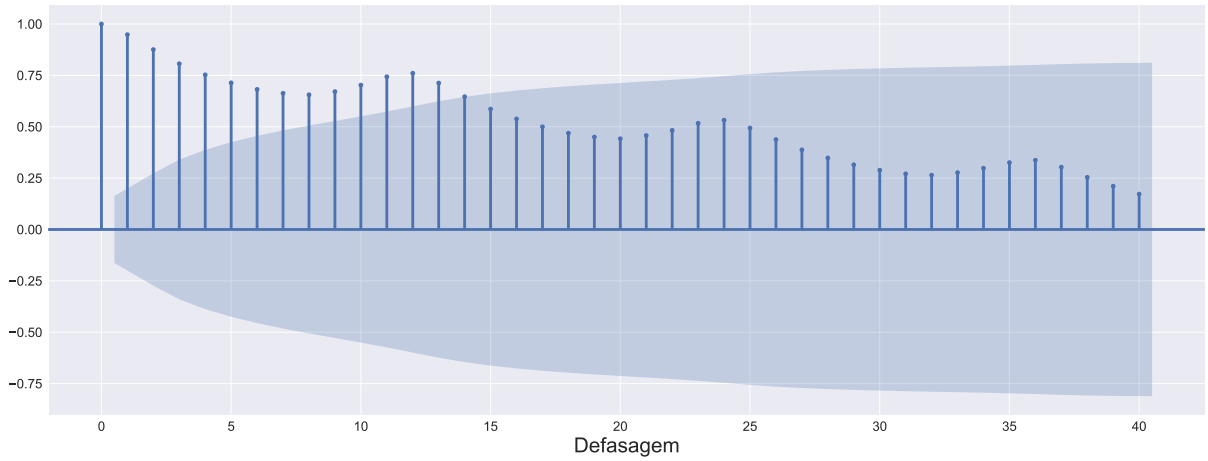
O gráfico da função de autocorrelação, detalhado na Seção 2.1.2 da revisão teórica, também auxilia na compreensão do comportamento de uma série temporal. A Figura 12 apresenta o gráfico de autocorrelação da série temporal em questão. O decaimento lento nas correlações das defasagens mostra que a série possui tendência, enquanto que o formato ”ondular” ocorre devido à sazonalidade.

Os padrões de tendência e sazonalidade indicam que essa série não é estacionária, o que é confirmado pelo teste "Augmented Dickey-Fuller"(ADF), com p-value igual a 0.9918. Portanto, a hipótese nula de não estacionariedade não pode ser rejeitada.

3.1.2 Milk Production

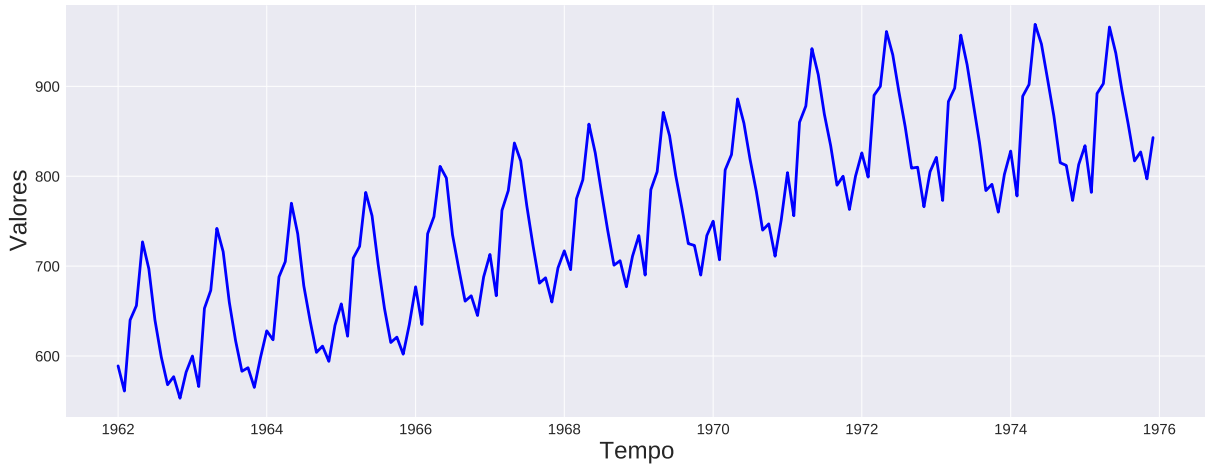
Média mensal da produção de leite por vaca em libras (*pounds*) ao longo de 14 anos, indo de janeiro de 1962 a dezembro de 1975. No total, a série contém 168 observações, conforme ilustrado na Figura 13.

Figura 12 – Gráfico de autocorrelação das defasagens - série temporal "International Airline Passengers".



Fonte: Elaborada pelo autor.

Figura 13 – Dados mensais da produção de leite (em pounds).

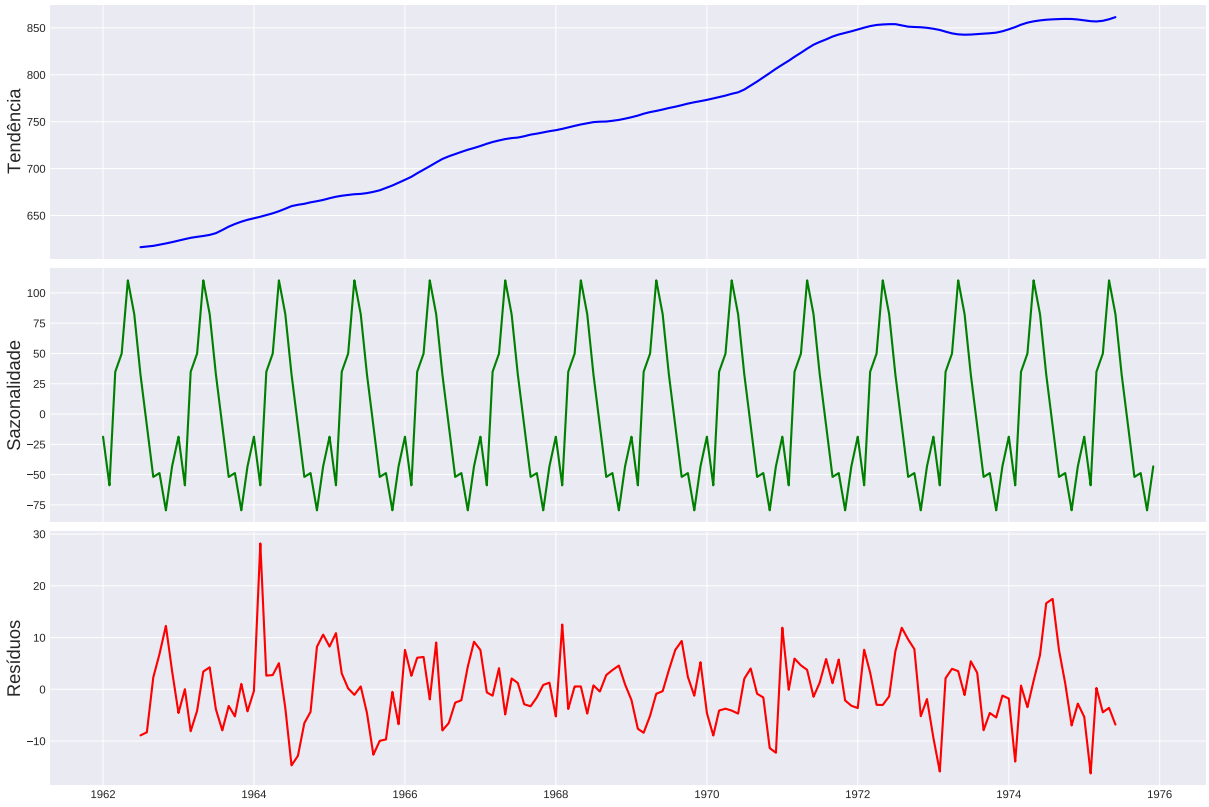


Fonte: Elaborada pelo autor.

A série mostra uma tendência de crescimento ao longo do tempo e um padrão de sazonalidade com maior produção de leite em determinados meses do ano. Ao contrário da série apresentada anteriormente, nessa série a volatilidade se apresenta aparentemente constante ao longo do tempo. A decomposição da série em tendência e sazonalidade mostra esses padrões com maior clareza, conforme é possível visualizar na Figura 14.

A Figura 15 mostra um conjunto de gráficos que detalham mais as características dessa série temporal. O gráfico da Figura 15 (a), no qual cada linha representa um ano da série temporal, mostra um comportamento padronizado ao longo dos anos, com um decréscimo inicial na produção de leite, um aumento até o mês de maio, depois um decréscimo novamente até o mês de novembro, seguido de um aumento em dezembro. O gráfico da Figura 15 (b) mostra os dados

Figura 14 – Decomposição da śerie temporal ”Milk Production”.



Fonte: Elaborada pelo autor.

agrupados por mês. Nesse formato, é possível visualizar que a volatilidade é aparentemente constante. Além disso, as médias demarcadas com um traço vermelho para cada grupo de meses confirmam a tendência da série apresentada no gráfico anterior. Tanto o padrão sazonal quanto o padrão de tendência são também confirmados pelo gráfico de caixas agrupado por mês da Figura 15 (c). O gráfico de caixas da Figura 15 (d) mostra a evolução dos dados ano a ano, no qual é possível observar que a volatilidade mantém-se constante ao longo do tempo.

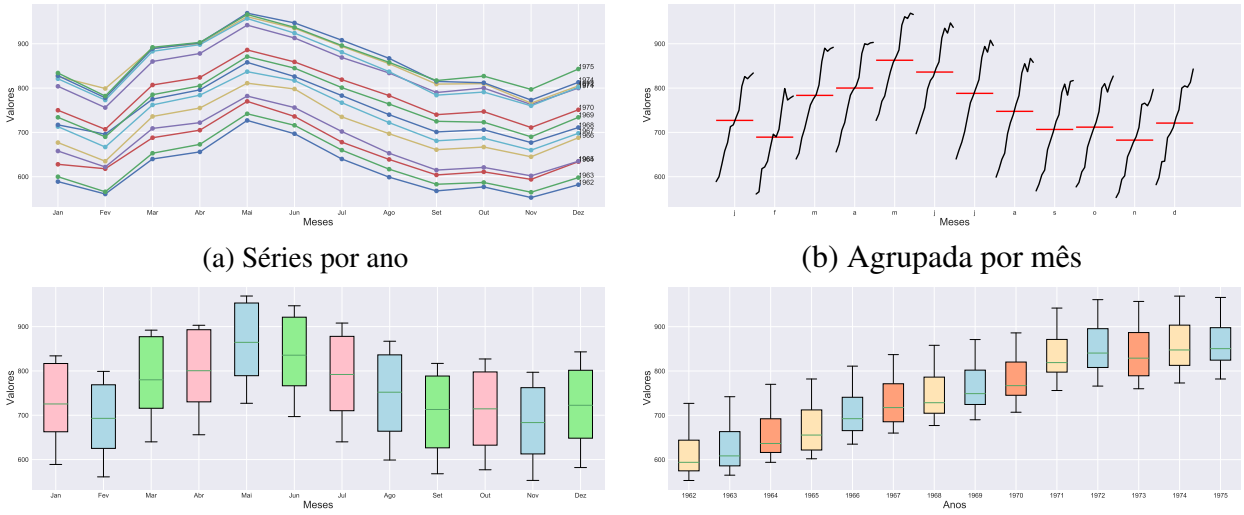
A Figura 16 mostra o gráfico da função de autocorrelação dessa série. Semelhante à série anterior, o decaimento lento nas correlações das defasagens mostra que a série possui tendência, enquanto que o formato ”ondular” revela padrões de sazonalidade da série.

Os padrões de tendência e sazonalidade da série indicam que ela não é estacionária, o que é confirmado pelo teste ADF, com p-value igual a 0.6274. Portanto, também nesse caso, a hipótese nula de não estacionariedade não pode ser rejeitada.

3.1.3 Beer Production

Média mensal da produção de cerveja ao longo de 39 anos, de janeiro de 1956 a dezembro de 1994. No total, a série contém 468 observações. A fig 17 mostra o gráfico dessa série temporal.

Figura 15 – Gráfico mensais e anuais referentes à série temporal ”Milk Production”.



(a) Séries por ano

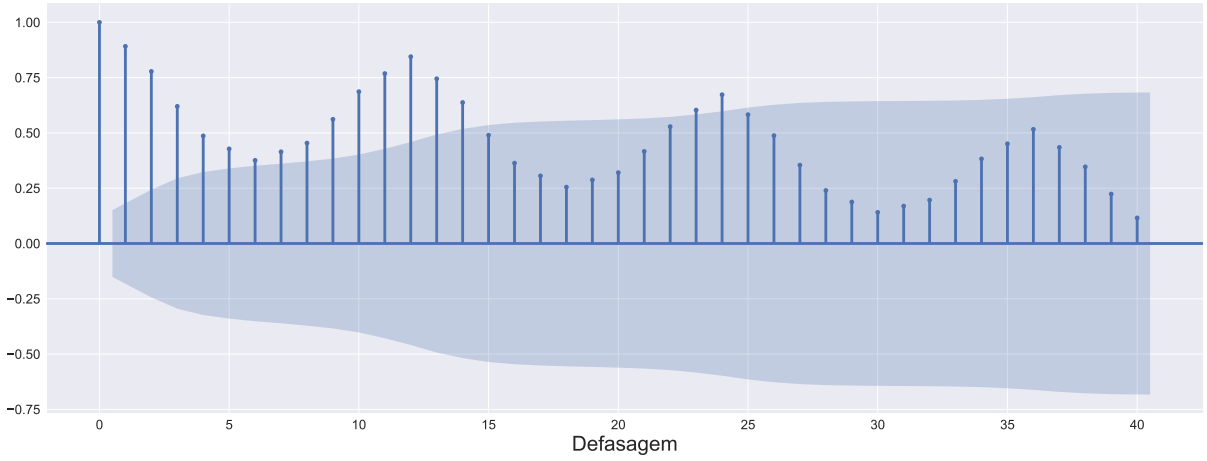
(b) Agrupada por mês

(c) Valores agrupados por mês.

(d) Valores agrupados por ano.

Fonte: Elaborada pelo autor.

Figura 16 – Gráfico de autocorrelação das defasagens - série temporal "Milk Production".

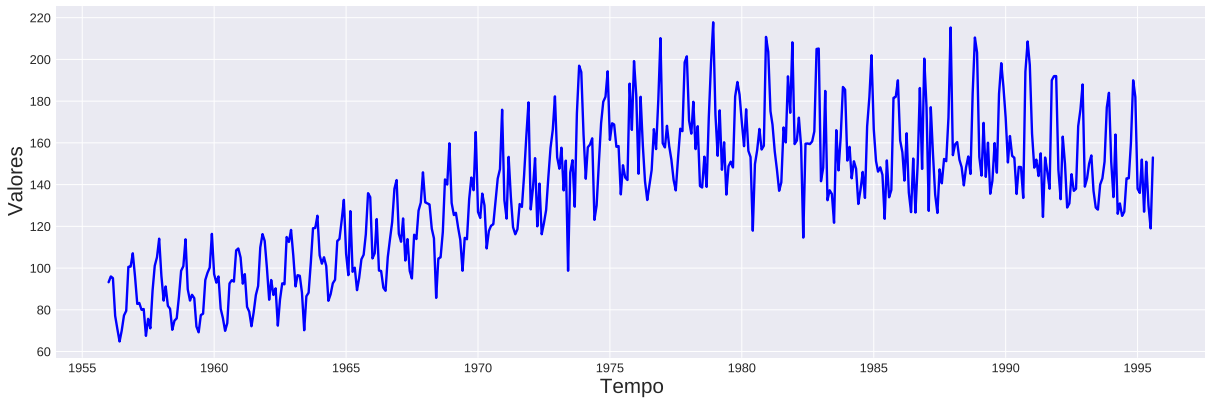


Fonte: Elaborada pelo autor.

A decomposição dessa série nas componentes tendência, sazonalidade e resíduos (Figura 18) mostra uma tendência de crescimento até por volta do início de 1975. Após esse período a produção mantém-se numa faixa constante até por volta do início de 1990, e depois apresenta um leve declínio. A componente sazonal apresenta-se forte, mas consistente. Percebe-se também um aumento no ruído por volta de 1974.

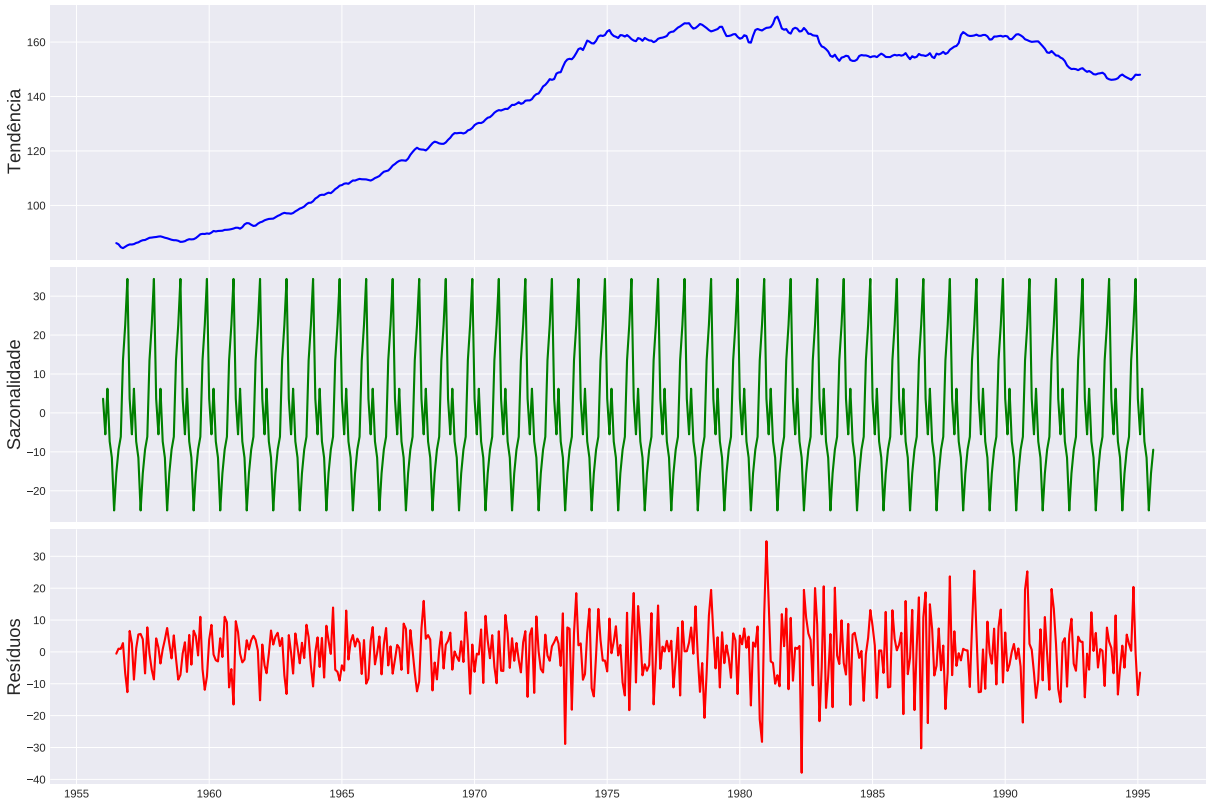
Na Figura 19 é possível visualizar um conjunto de gráficos que apresentam algumas características dessa série temporal. O gráfico da Figura 19 (a), no qual cada linha representa um ano da série temporal, no geral mostra um comportamento padronizado ao longo dos anos, ou

Figura 17 – Dados mensais da produção de cerveja.



Fonte: Elaborada pelo autor.

Figura 18 – Decomposição da śerie temporal "Beer Production".

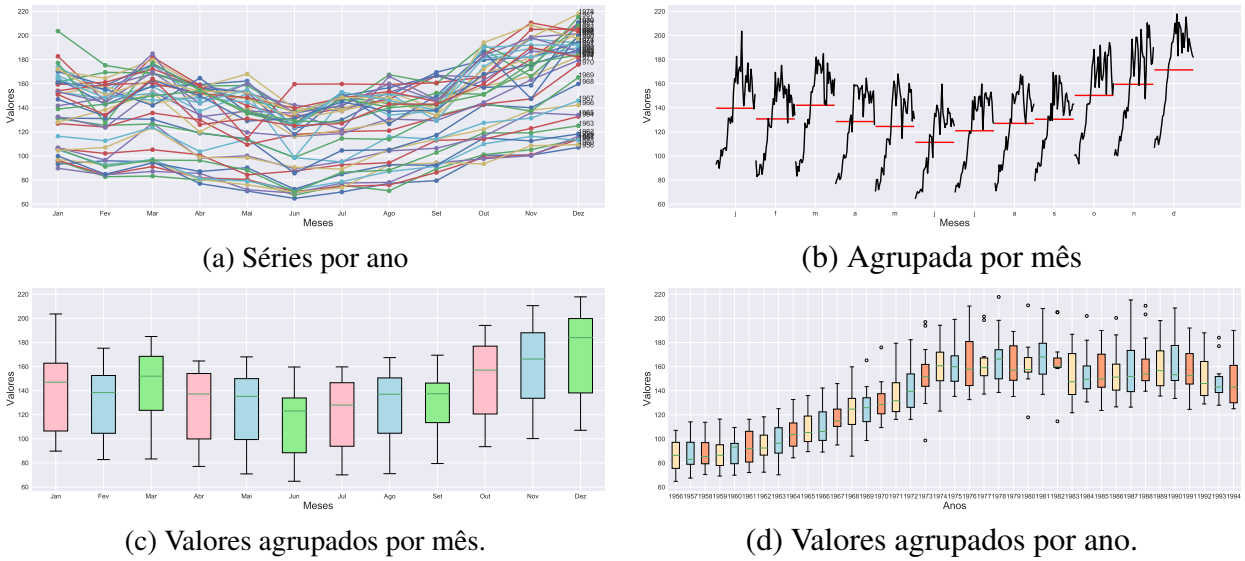


Fonte: Elaborada pelo autor.

seja, a produção decai até o mês de junho e depois aumenta até dezembro. No entanto, alguns anos seguem padrões diferentes, fazendo com que a volatilidade não mantenha-se constante ao longo do tempo. Os gráficos (b) e (c) da Figura 19 mostram os dados agrupados por mês. As médias, tanto do gráfico de linhas quanto do gráfico de caixas confirmam a tendência dessa série, descrita anteriormente. O gráfico de caixas da Figura 19 (d) mostra a evolução dos níveis de

produção de cerveja ano a ano. É possível visualizar que a volatilidade mantém-se constante por alguns anos e varia consideravelmente em outros, não apresentando um padrão específico.

Figura 19 – Gráfico mensais e anuais referentes à série temporal ”Beer Production”.



(a) Séries por ano

(b) Agrupada por mês

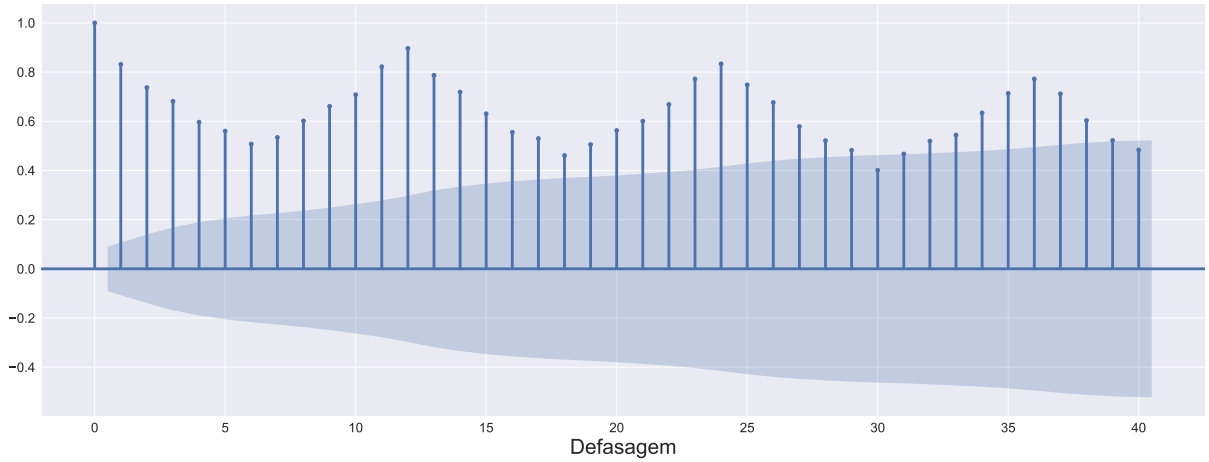
(c) Valores agrupados por mês.

(d) Valores agrupados por ano.

Fonte: Elaborada pelo autor.

A Figura 20 mostra o gráfico da função de autocorrelação dessa série. Semelhante às anteriores, é possível notar um decaimento lento nas correlações conforme a defasagem aumenta, evidenciando que a série possui tendência. Ao mesmo tempo, o formato ”ondular” das correlações revela que a série possui padrões de sazonalidade.

Figura 20 – Gráfico de autocorrelação das defasagens - série temporal "Beer Production".



Fonte: Elaborada pelo autor.

Os padrões observados anteriormente indicam que ela não é estacionária, o que é confirmado pelo teste ADF, com p-value igual a 0.1776. Igualmente às duas séries anteriores, a

hipótese nula de não estacionariedade não pode ser rejeitada.

O Quadro 2 resume o comportamento das séries temporais descritas anteriormente com relação às suas principais características. Na tabela, os *p*-values indicam se uma série é estacionária ou não. *P*-values abaixo de 0.05 indicam que a série é estacionária.

Quadro 2 – Resumo das características das séries temporais selecionadas para este estudo. Os *p*-values foram obtidos com o teste ADF, indicando se as séries são estacionárias ou não.

Série	Tamanho	Tendência	Sazonalidade	Volatilidade	p-value
Passengers	144	Alta forte	Anual	Crescente	0.992
Milk	168	Alta leve	Anual	Constante	0.627
Beer	468	Alta leve e lateral	Anual	Instável	0.178

Fonte: Dados da pesquisa.

3.2 Processo Experimental

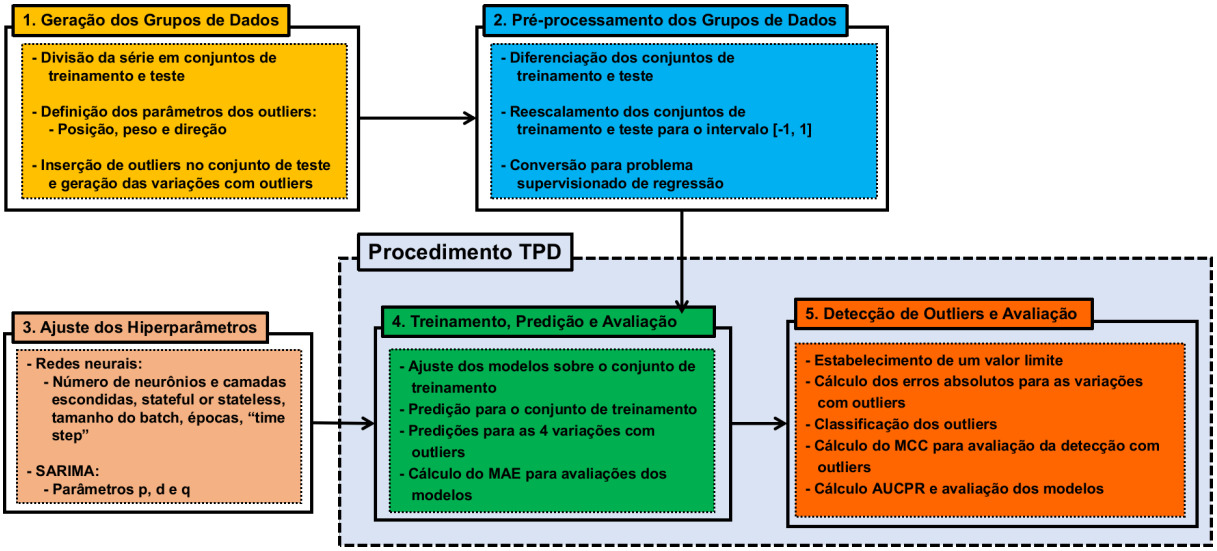
Os experimentos que visam alcançar os objetivos deste trabalho envolvem detectar outliers em três séries temporais univariadas por meio de predições realizadas pelos modelos SARIMA, LSTM e GRU. Esse processo foi executado em cinco etapas:

- Etapa 1: Geração de grupos de dados;
- Etapa 2: Pré-processamento dos grupos de dados;
- Etapa 3: Ajuste dos hiperparâmetros dos modelos;
- Etapa 4: Treinamento, predições e avaliação dos modelos de predição; e
- Etapa 5: Detecção de outliers e avaliação dos modelos de detecção.

As cinco etapas listadas acima estão subdivididas em procedimentos e atividades, detalhados na próxima Subseção. Porém, antes de uma descrição mais detalhada do Processo Experimental, convém apresentar uma visão geral para facilitar sua compreensão conforme Figura 21.

A Etapa 1 tem como objetivo a geração de três grupos de dados, cada um gerado a partir de uma das três séries temporais abordadas neste estudo. Esses grupos de dados são denominados por Grupo de Dados Passengers, Grupo de Dados Milk e Grupo de Dados Beer. A geração do grupo de dados deu-se da seguinte forma: primeiramente, a série temporal foi dividida em conjunto de treinamento e conjunto de teste. Os outliers foram, então, gerados em quatro versões diferentes e inseridos em quatro cópias distintas do conjunto de teste, originando, assim, quatro variações diferentes de conjuntos de testes com outliers. Não foram inseridos outliers no conjunto

Figura 21 – Esquema de execução do processo experimental adotado neste trabalho.



Fonte: Elaborada pelo autor.

de treinamento. Portanto, cada grupo de dados foi composto por um conjunto de treinamento e quatro variações do conjunto de teste, cada qual com uma versão diferente de outliers inseridos. Um esquema simplificado da Etapa 1 e a geração dos grupos de dados pode ser visto na Figura 22.

Figura 22 – Geração dos grupos de dados na Etapa 1.

Fonte: Elaborada pelo autor.

Na Etapa 2, os grupos de dados foram preparados para serem processados pelas redes neurais LSTM e GRU. Nesta etapa houve também a conversão dos dados para um problema de aprendizado supervisionado de regressão. Para o modelo SARIMA, nenhuma preparação nos dados foi necessária. Na Etapa 3, foram realizados procedimentos que visaram identificar os hiperparâmetros apropriados para os modelos, considerando as especificidades das séries temporais abordadas neste estudo.

As Etapas 4 e 5, por terem suas atividades executadas de forma sequencial e contínua, formam um único procedimento denominado como "Procedimento de Treinamento, Predição e

Detecção (TPD)", executado sobre os grupos de dados gerados e preprocessados nas Etapas 1 e 2. Este procedimento pode ser visto com maiores detalhes em formato de algoritmo (Algoritmo 1).

Algoritmo 1 – Procedimento TPD

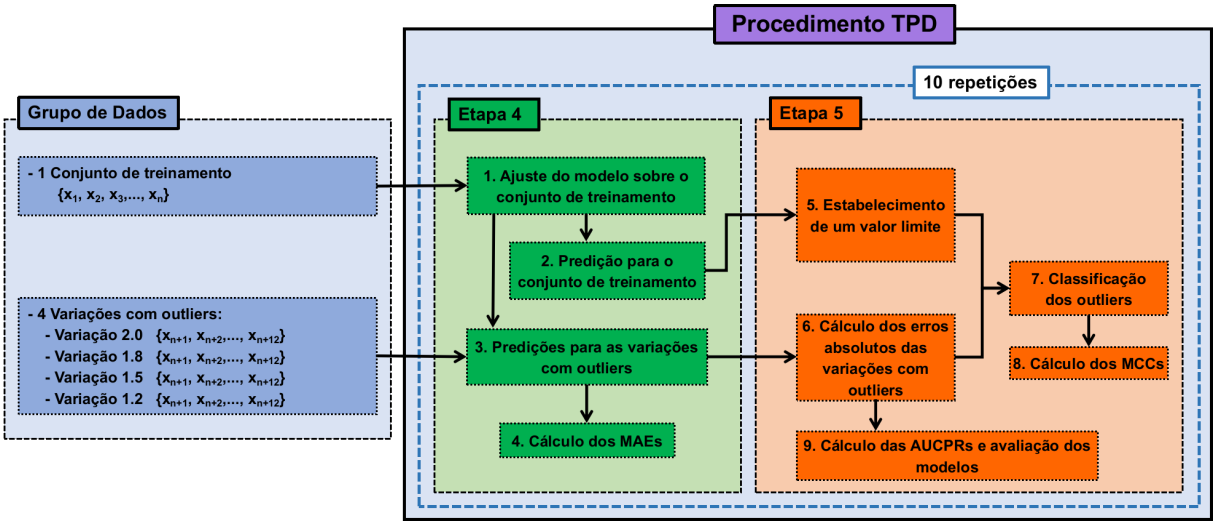
Para cada Grupo de Dados: Passengers, Milk e Beer:
Para cada modelo: SARIMA, LSTM e GRU:
Repetir 10 vezes:
Atividades da Etapa 4:
→ Treinamento do modelo sobre o conjunto de treinamento;
→ Predição para o conjunto de treinamento;
→ Predições para as 4 variações com outliers;
→ Cálculo do MAE para avaliação dos modelos;
Atividades da Etapa 5:
→ Estabelecimento de um valor limite ("threshold");
→ Cálculo dos erros absolutos para as variações com outliers;
→ Classificação dos outliers;
→ Cálculo do MCC para avaliação da detecção de outliers;
→ Cálculo da área abaixo da curva *precision-recall* e avaliação dos modelos.

Para avaliação das predições foi utilizado o "Erro Médio Absoluto" (MAE - do inglês *Mean Absolute Error*); para a avaliação da detecção de outliers foi utilizada a métrica Matthew Correlation Coefficient (MCC); e para a avaliação geral dos modelos foi utilizada a Área Abaixo da Curva *Precision-Recall* (AUCPR). Essas métricas serão detalhadas na próxima Subseção.

O Esquema da Figura 23 mostra como é executado o Procedimento TPD. Para cada grupo de dados e modelo, são executadas as atividades das Etapas 4 e 5: (1) O modelo é treinado sobre o conjunto de treinamento; (2) O modelo treinado é utilizado para realizar as predições para o conjunto de treinamento; (3) O modelo treinado é utilizado para realizar as predições para as 4 variações com outliers; (4) São calculados os MAEs para as predições sobre as variações com outliers; (5) É estabelecido, a partir do erro absoluto máximo entre o conjunto de treinamento e sua predições, um valor limite que será utilizado para classificar os outliers; (6) São calculados os erros absolutos entre as variações com outliers e suas predições; (7) Esses erros absolutos são comparados com o valor limite e classificados os outliers; (8) São calculados os MCCs para as detecções de outliers; (9 e 10) A partir dos MAEs calculados no item 6, são geradas as curvas Precision-Recall e calculadas as AUCPRs para todos as detecções realizadas. As atividades 1 a 10 são repetidas 10 vezes para cada grupo de dados, com diferentes inicializações dos modelos. Para as análises dos resultados, discussão e comparações, foram utilizadas as médias e desvios padrão das métricas geradas nas 10 repetições.

A Figura 24 apresenta um esquema resumido do Processo Experimental: a Etapa 1 gera um grupo de dados para cada série temporal, compostos pelo um conjunto de treinamento e quatro variações com outliers do conjunto de teste; a Etapa 2 preprocessa os grupos de dados, convertendo-os em dados de entrada para as redes neurais; a Etapa 3 ajusta os hiperparâmetros

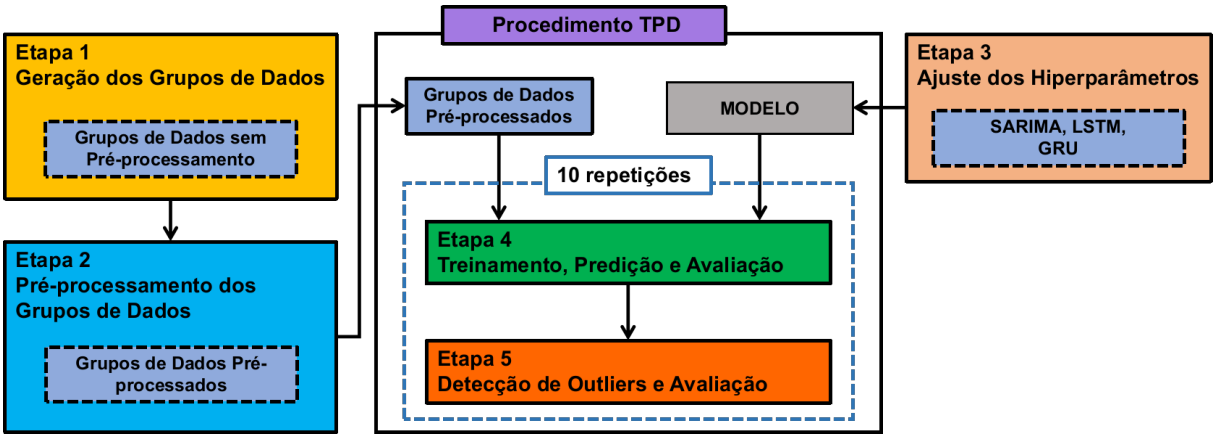
Figura 23 – Aplicação do Procedimento TPD sobre os grupos de dados.



Fonte: Elaborada pelo autor.

do modelos; as Etapas 4 e 5 formam o Procedimento TPD, cujas atividades são repetidas 10 vezes para cada grupo de dados/modelo.

Figura 24 – As cinco etapas do Processo Experimental e o Procedimento TPD.



Fonte: Elaborada pelo autor.

A seguir, será apresentada uma descrição detalhada das etapas, procedimentos e atividades do Processo Experimental.

3.2.1 As Cinco Etapas do Processo Experimental

3.2.1.1 Etapa 1: Geração dos Grupos de Dados

A Etapa 1 tem como entrada uma série temporal e como saída um "Grupo de Dados". Ao final desta etapa, três grupos de dados foram gerados, cada um referente a uma das três séries

temporais abordadas neste estudo: “Grupo de Dados Passengers”, “Grupo de Dados Milk” e “Grupo de Dados Beer”. O primeiro procedimento realizado nesta etapa foi a divisão da série temporal em dois conjuntos de dados: conjunto de treinamento e conjunto de teste. O conjunto de teste foi composto pelas últimas 12 observações da série original e, para as 3 séries originais, o período compreendido foi de um ano completo (janeiro a dezembro). O conjunto de treinamento foi composto por todas as observações da série temporal anteriores às 12 observações do conjunto de teste. Em seguida, prosseguiu-se com a geração e inserção dos outliers, introduzidos apenas nos conjuntos de teste.

Os parâmetros utilizados para geração dos outliers foram: quantidade de outliers, posição de inserção, peso e direção. Arbitrariamente, foi decidido que os conjuntos de teste receberiam dois outliers, inseridos na quinta e nona posições. Para geração dos outliers, foi calculado o desvio padrão do conjunto de treinamento e multiplicado por um peso que determinou o quão distante o outlier ficaria de seu valor original. O valor resultante foi, então, multiplicado por -1 ou 1, determinando a direção do outlier, ou seja, se ele seria somado ou subtraído do valor real do conjunto de teste.

Foram geradas 4 variações do conjunto de teste, sendo que cada variação teve 2 outliers inseridos, um na quinta e outro na nona posição, com direções 1 e -1, respectivamente. O que diferenciou uma variação da outra foi o valor do parâmetro peso atribuído aos outliers. Esse parâmetro assumiu o mesmo valor para os dois outliers inseridos em uma mesma variação, porém assumiu valores diferentes entre as variações. As configurações das quatro variações com outliers é igual para os três grupos de dados, e são mostrada na tabela 1.

Tabela 1 – Configuração das quatro variações com outliers de um grupo de dados.

Nome da Variação	Qtd de Outliers	Posições	Direções	Peso
Variação com outliers 2.0	2	5a e 9a	1 e -1	2.0
Variação com outliers 1.8	2	5a e 9a	1 e -1	1.8
Variação com outliers 1.5	2	5a e 9a	1 e -1	1.5
Variação com outliers 1.2	2	5a e 9a	1 e -1	1.2

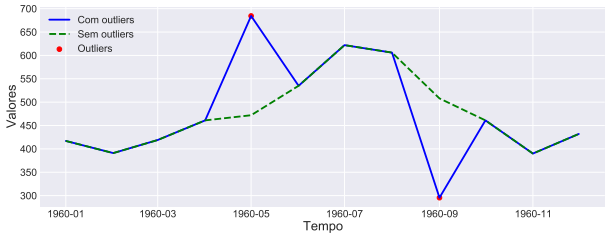
Fonte: Dados da pesquisa.

Ao longo do trabalho, essas variações são denominadas por termos como “variação do conjunto de teste com outliers” ou “variações com outliers”, para se referir às variações em geral, ou acrescentando-se a esses termos o valor do peso atribuído como, por exemplo, “variação do conjunto de teste com outliers 2.0” ou “variação com outliers 1.8”, para se referir a uma variação específica.

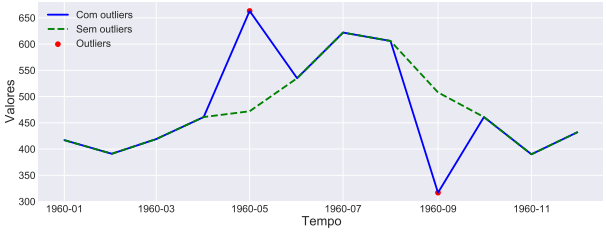
As Figuras 25, 26 e 27 mostram os gráficos das variações com outliers dos Grupos de Dados Passengers, Milk e Beer, respectivamente. No gráfico, a linha azul sólida representa a variação com outliers. Os dois pontos vermelhos representam os outliers inseridos na quinta e nona posições. A linha verde tracejada mostra o conjunto de teste original, antes da inserção de

outliers.

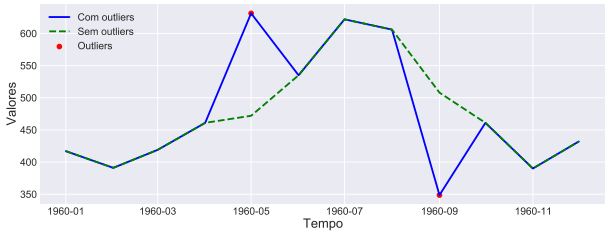
Figura 25 – As 4 variações com outliers do Grupo de Dados Passengers.



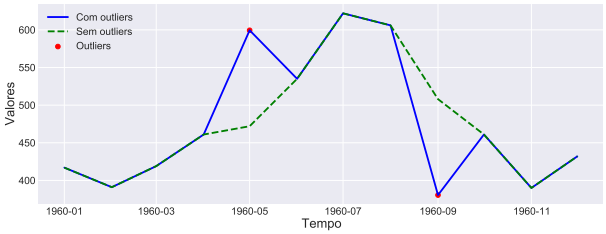
(a) Variação com outliers 2.0



(b) Variação com outliers 1.8



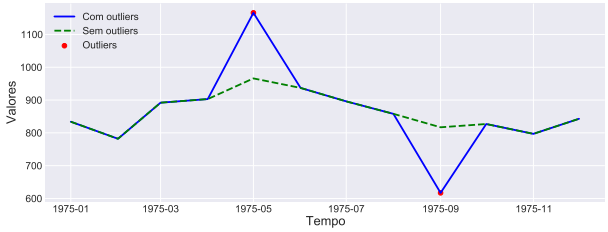
(c) Variação com outliers 1.5



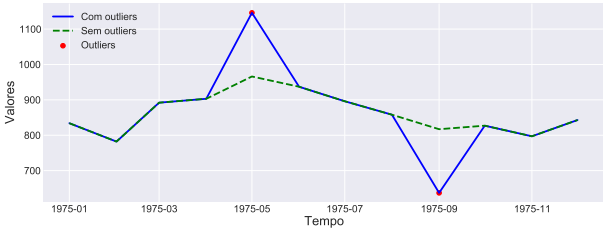
(d) Variação com outliers 1.2

Fonte: Elaborada pelo autor.

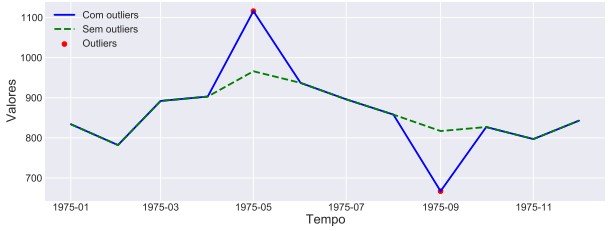
Figura 26 – As 4 variações com outliers do Grupo de Dados Milk.



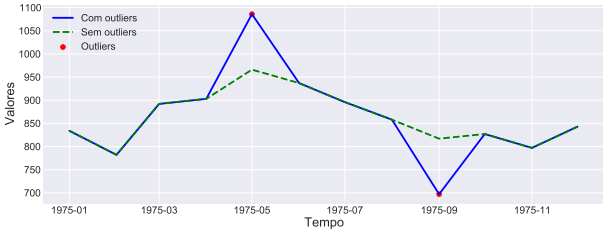
(a) Variação com outliers 2.0



(b) Variação com outliers 1.8



(c) Variação com outliers 1.5

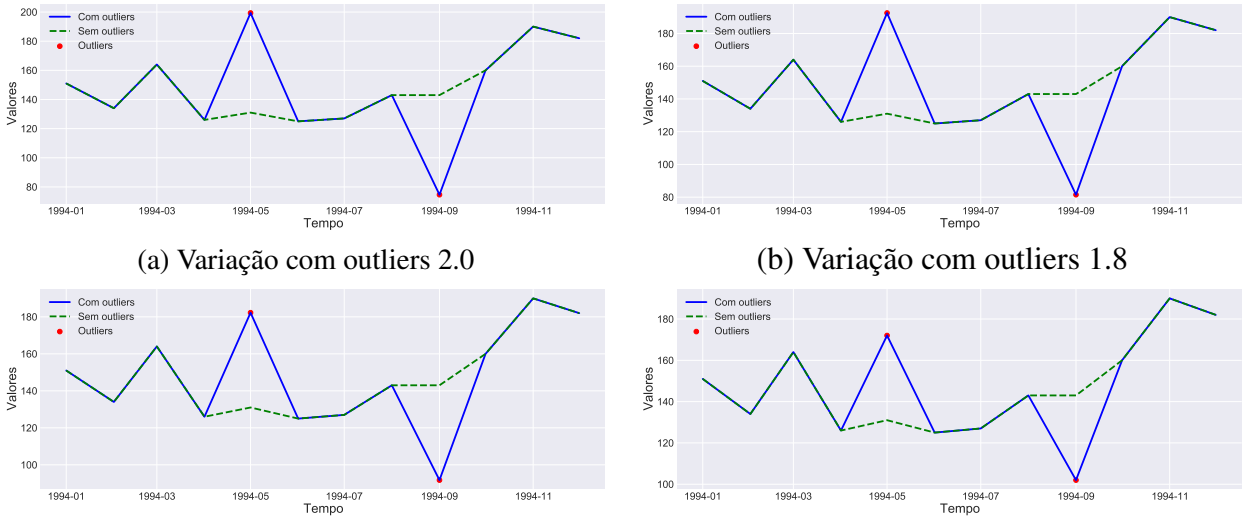


(d) Variação com outliers 1.2

Fonte: Elaborada pelo autor.

A Figura 28 mostra um esquema simplificado da Etapa 1, com a geração do grupo de dados e sua composição. Conforme já mencionado, cada grupo de dados é composto por um conjunto de treinamento e quatro variações do conjunto de teste com outliers, sendo que o que se alterou de uma variação para outra foi o peso atribuído aos outliers.

Figura 27 – As 4 variações com outliers do Grupo de Dados Beer.



(a) Variação com outliers 2.0

(b) Variação com outliers 1.8

(c) Variação com outliers 1.5

(d) Variação com outliers 1.2

Fonte: Elaborada pelo autor.

Figura 28 – A geração dos grupos de dados na Etapa 1, com o conjunto de treinamento e as variações com outliers.

Fonte: Elaborada pelo autor.

3.2.1.2 Etapa 2: Pré-processamento dos Grupos de Dados

A Etapa 2 tem como entrada os grupos de dados gerados na Etapa 1 e como saída grupos de dados preprocessados, convertidos em dados de entrada para as redes neurais. O modelo SARIMA não exige nenhum pré-processamento prévio dos dados. Para esse modelo, os grupos de dados foram utilizados com seus valores iniciais, sem nenhuma transformação prévia.

Para os modelos LSTM e GRU, os grupos dados precisaram receber alguns tratamentos antes de serem inseridos nas redes neurais. Primeiramente, os conjuntos de treinamento e teste foram aproximados para séries estacionárias por meio de diferenciação de primeira ordem. Isso foi necessário porque as séries abordadas neste estudo são séries não estacionárias, conforme mostrado nas descrição das séries temporais (Seção 3.1). Em seguida, os valores dos conjuntos de treinamento e teste foram reescalados para o intervalo -1 e 1. Vale ressaltar que os parâmetros para

transformar os valores foram aprendidos no conjunto de treinamento e somente posteriormente essa transformação foi aplicada sobre os valores do conjunto de teste.

Ainda, para utilização das redes neurais LSTM e GRU, foi necessário converter os dados para um problema supervisionado de regressão. Nos casos de série temporais, uma forma de se construir um problema de regressão é gerar uma sequência com valores da série, por exemplo, t_1, t_2, t_3 e estabelecendo o valor seguinte a essa sequência, t_4, como saída. Neste trabalho, o conjunto de treinamento foi convertido em sequências com 12 valores, sendo a saída o valor seguinte. Assim, por exemplo, tem-se a primeira sequência $s_1 = \{t_1, t_2, ..., t_{12}\}$, e sua saída t_{13}; a segunda sequência $s_2 = \{t_2, t_3, ..., t_{13}\}$ e sua saída t_14, e assim por diante. As sequências geradas foram armazenadas em uma matriz X como vetores linha, que formam as sequências de entrada do problema de regressão. E os valores de saída foram armazenados em um vetor y.

Para o conjunto de teste, o mesmo procedimento foi adotado. Porém, para formação das sequências com 12 valores foi necessário utilizar valores prévios do conjunto de treinamento. Assim, a primeira sequência utilizou os 12 últimos valores do conjunto de treinamento, tendo como saída o primeiro valor do conjunto de teste. A segunda sequência utilizou os últimos 11 valores do conjunto de treinamento e o primeiro valor do conjunto de teste, tendo como saída o segundo valor do conjunto de teste, e assim por diante.

3.2.1.3 Etapa 3: Ajuste dos Hiperparâmetros dos Modelos

A Etapa 3 foi realizada paralelamente às Etapas 1 e 2. Esta etapa teve como objetivo identificar hiperparâmetros adequados tanto para as redes neurais quanto para o modelo SARIMA. No caso das redes neurais LSTM e GRU, para implementação dessas redes foram utilizadas as bibliotecas TensorFlow versão 2.3.1 e Keras versão 2.4.3. Keras é um API desenvolvido sobre o TensorFlow que facilita a implementação de redes neurais. Essas bibliotecas permitem ajustar uma infinidade de hiperparâmetros que determinarão tanto a arquitetura como também a forma com que a rede neural será treinada. Definir e ajustar esses hiperparâmetros não é uma atividade simples (REIMERS; GUREVYCH, 2017). Em muitos casos, emprega-se uma busca do tipo "força bruta" utilizando-se um intervalo com tamanho considerável de possibilidades para cada hiperparâmetro, o que pode ter um alto custo computacional. Ou pode-se também empregar uma busca aleatória pelos hiperparâmetros mas, nesse caso, nem sempre os melhores valores acabam sendo avaliados. Em ambos os casos, mantém-se registros dos desempenhos alcançados e, ao final, seleciona-se a arquitetura que obteve melhor resultado.

Neste trabalho, porém, optou-se por iniciar essa busca a partir de uma arquitetura já pré-definida. Abbasimehr, Shabani e Yousefi (2020) realizaram um estudo com o objetivo de detectar uma arquitetura precisa de rede neural LSTM em uma situação econômica: prever a demanda de vendas para uma empresa de móveis. Devido à semelhança com as séries temporais abordadas neste trabalho, que também referem-se a situações econômicas, optou-se por iniciar os ajustes dos hiperparâmetros a partir dessa arquitetura. A Tabela 2 mostra a arquitetura da rede

LSTM proposta pelos autores.

Tabela 2 – Hiperparâmetros da arquitetura base adotada para ajuste final da rede neural LSTM.

Hiperparâmetros	Valores
Time step	12
Número de camadas escondidas	2
Número de neurônios em cada camada	64
Taxa de dropout	0.1
Tamanho do batch	1
Número de épocas	500

Fonte: Abbasimehr, Shabani e Yousefi (2020).

Para o ajuste dos hiperparâmetros e definição da arquitetura das redes utilizadas neste trabalho, desenvolveu-se um procedimento que foi aplicado às três séries temporais originais. Esse procedimento foi realizado paralelamente às outras etapas do Processo Experimental e, conforme mostra a Figura 21, ele conecta-se diretamente à Etapa 4, quando os modelos são treinados e as predições são realizadas utilizando-se a arquitetura de rede definida nesta Etapa 3. Esse procedimento, listado abaixo, foi aplicado somente à rede LSTM e, assim como em Gao *et al.* (2020), devido à similaridade e para facilitar comparações, a arquitetura encontrada para essa rede foi também adotada para a rede GRU. O procedimento adotado foi:

1. A série foi dividida em três conjuntos: treinamento, validação e teste. O conjunto de teste compreendeu as últimas 12 observações da série, e o conjunto de validação, as 12 observações anteriores às do conjunto de teste. Somente os conjuntos de treinamento e validação foram utilizados neste procedimento de ajustes dos hiperparâmetros.

2. O modelo de Abbasimehr, Shabani e Yousefi (2020) foi implementado como modelo base com uma única alteração: inicialmente, o parâmetro "state" da rede foi ajustado como "True". Nesse caso, a rede passa a ser considerada "stateful", implicando que a memória interna da camada aprendida a partir de uma sequência será utilizada na próxima sequência. Em outras palavras, quando o parâmetro "stateful" é definido "True", o último estado para cada sequência no índice *i* de um lote (*batch*) será utilizado como estado inicial para a sequência no índice *i* do lote seguinte. Se ajustado como "False", significa que o estado será reiniciado a cada sequência.

3. Dessa vez, a rede foi testada com o parâmetro "state" ajustado como "False", e a rede pode ser considerada "stateless". Os resultados foram comparados com os da etapa anterior.

4. Em seguida, avaliou-se o número de neurônios das camadas escondidas da LSTM. Foram testados os valores: 32, 128 e 256, e comparados com o valor original de 64.

5. Por último, foram avaliados os seguintes valores para “time steps”: 2, 4, 6, 8, 10. Os resultados foram comparados com o resultado encontrado quando utilizado o “time steps” da arquitetura inicial da rede (12), conforme definida pelos autores.

Com relação à determinação do número de épocas, foram utilizadas as funções EarlyStoping e ModelCheckpoint da biblioteca Keras. Existem formas diferentes para uso dessas funções. Neste trabalho, a primeira função monitorou o valor da função de perda avaliada no conjunto de validação, iterando mais 50 vezes a cada valor mínimo encontrado. Se dentro dessas próximas 50 iterações nenhum novo valor mínimo é encontrado, o processo é interrompido. A segunda função, ”ModelCheckpoint”, foi responsável por salvar o modelo que obteve o menor erro, ou seja, aquele correspondente ao menor erro encontrado, e não o último modelo, gerado após as 50 iterações.

Em todos os casos, a métrica utilizada para avaliação do desempenho e seleção da melhor arquitetura foi o ”Erro Médio Absoluto” (MAE), cuja equação será dada mais adiante. As arquiteturas encontradas para as três séries temporais originais foram idênticas, e bastante semelhantes à encontrada por Abbasimehr, Shabani e Yousefi (2020). A única diferença é que, neste caso, as redes ”stateful” apresentaram melhores resultados e, para os autores, os melhores resultados foram obtidos com redes ”stateless”. A Tabela 3 mostra como ficaram definidos os parâmetros para as redes neurais.

Tabela 3 – Parâmetros definidos para as redes neurais LSTM e GRU.

Camadas	Neurons	Batch	Iterações	Stateful	Time step
2	64	1	entre 1 e 500	True	12

Fonte: Dados da pesquisa.

Convém ressaltar que o parâmetro "Shuffle", que define se as observações da base de dados serão ou não embaralhadas ao entrarem na rede neural, foi definido como "False"para ambas as redes. Por padrão, quando se lida com séries temporais, cujos valores observados possuem correlação entre si, esse parâmetro é sempre definido como "False". Ou seja, os valores da série temporal não são embaralhados para não se perder as informações que a sequência oferece.

Com relação ao modelo SARIMA, foi necessário definir os parâmetros p, d e q, que determinam a ordem do modelo AR, a ordem de diferenciação a que será submetida a série temporal e a ordem do modelo MA, respectivamente. Para definição dos parâmetros p e q foi utilizado um procedimento conhecido como ”grid search”, o qual testa diversas combinações de valores para esses parâmetros. O parâmetro d foi fixado em 1. O conjunto de parâmetros selecionado foi aquele que apresentou o menor AIC (Akaike Information Criterion). O AIC é uma medida utilizada para avaliação de modelos SARIMA descrita na Seção 2.2. Os parâmetros resultantes do ”grid search” que apresentaram os melhores AIC podem ser vistos na Tabela 4.

Tabela 4 – Parâmetros definidos para o modelo SARIMA.

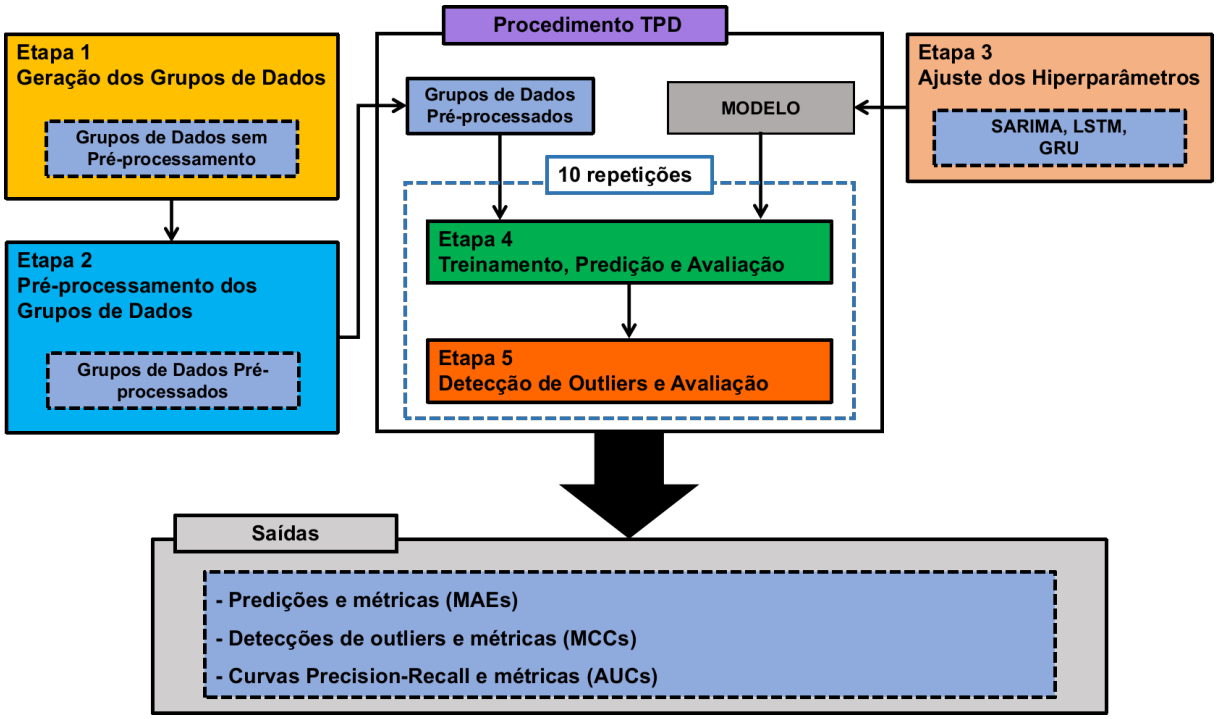
	Ordem						
Série	AR (p)	Dif (d)	MA (q)	Saz AR	Saz Dif	Saz MA	Sazonalidade
Passengers	4	1	2	1	1	1	12
Milk	4	1	3	0	1	1	12
Beer	4	1	4	0	1	1	12

Fonte: Dados da pesquisa.

3.2.1.4 Etapa 4: Treinamento, Predições e Avaliação dos Modelos de Predição

Conforme já mencionado, as Etapas 4 e 5 formam um procedimento único denominado Procedimento TPD (pode ser revisto no Algoritmo 1). Como entrada, esse procedimento recebe um grupo de dados preprocessado, e como saída tem-se: as predições e suas métricas (MAEs); as detecções de outliers e suas métricas (MCCs); e as Curvas Precision-Recall e suas métricas (AUCPRs). O esquema completo do Procedimento TPD pode ser revisto na Figura 23. Um esquema relacionando todas as etapas do Processo Experimental, o Procedimento TPD e suas saídas pode ser visto na Figura 29.

Figura 29 – As cinco etapas do Processo Experimental, o Procedimento TPD e suas saídas.



Fonte: Elaborada pelo autor.

A Etapa 4 tem como entrada um grupo de dados preprocessado e um modelo (SARIMA, LSTM ou GRU). As saídas da Etapa 4 - as predições para o conjunto de treinamento e as predições para as variações com outliers - servirão como entradas para a Etapa 5. Na próxima

Subseção será mostrado como essas saídas foram utilizadas na Etapa 5. Na Etapa 4 foram desenvolvidas 4 atividades, listadas abaixo:

- Ajuste do modelo sobre o conjunto de treinamento;
- Predição para o conjunto de treinamento;
- Predições para as 4 variações com outliers;
- Cálculo do MAE para avaliação dos modelos.

Dado um grupo de dados e um modelo, primeiramente, esse modelo foi ajustado sobre o conjunto de treinamento do grupo de dados. Em seguida, com o modelo ajustado, foram realizadas as predições para o conjunto de treinamento e para as 4 variações com outliers do conjunto de teste. As predições sobre as 4 variações com outliers foram realizadas de forma "um passo à frente", com retreinamento do modelo a cada passo. Para as redes LSTM e GRU, o comprimento da sequência de entrada para o problema de regressão, que também é um hiperparâmetro para as redes neurais, foi de 12 valores, conforme definido na Subseção 3.2.1.3, referente ao ajuste dos hiperparâmetros. Na predição "um passo à frente", isso significa que a predição de um determinado valor no tempo t_i é realizada por uma sequência composta pelos 12 valores anteriores a t_i, ou seja $\{t_{i-12}, t_{i-11}, ..., t_{i-1}\}$.

Dessa forma, por exemplo, para a predição do valor referente ao tempo t_1 correspondente à primeira posição de uma determinada variação com outliers, foram utilizados os últimos 12 valores do conjunto de treinamento. Para as predições a partir do segundo passo em t_2, a sequência de entrada para a rede neural passa a utilizar valores que não são somente do conjunto de treinamento, mas também das variações do conjunto de teste com outliers. No caso de t_2, a sequência com 12 valores anteriores será composta pelos últimos 11 valores do conjunto de treinamento e o valor referente ao tempo t_1 da variação com outliers. Para a predição do tempo t_3, a sequência tomará os últimos 10 valores do conjunto de treinamento e os valores nos tempos t_1 e t_2 da variação com outliers, e assim por diante, conforme esquematizado abaixo.

$$
\begin{aligned}
&\{t_{1-12}, t_{1-11}, t_{1-10}, ..., t_{1-1}\} \rightarrow t_1 \\
&\{t_{1-11}, t_{1-10}, t_{1-9}, ..., t_1\} \rightarrow t_2 \\
&\{t_{1-10}, t_{1-9}, t_{1-8}, ..., t_1, t_2\} \rightarrow t_3 \\
&\{t_{1-9}, t_{1-8}, t_{1-7}, ..., t_1, t_2, t_3\} \rightarrow t_4 \\
&\vdots \\
&\{t_{1-1}, t_1, t_2, t_3, ..., t_{11}\} \rightarrow t_{12}
\end{aligned}
$$

Ao se utilizar valores das variações com outliers na sequência de entrada para o modelo realizar as predições, surgem duas possibilidades: esses valores podem ser os valores reais da variação com outliers, inclusive os outliers inseridos na quinta e nona posições; ou quando um

determinado valor da variação com outliers for classificado como outlier, pode-se desprezar o valor real da variação e utilizar o valor predito para aquela posição.

Para o modelo SARIMA, não é requerida uma sequência de entrada para realização das predições, como acontece no caso das redes neurais. Neste caso, para realização das predições de forma ”um passo à frente”, o modelo SARIMA é retreinado novamente até o passo atual, t_i, e em seguida é realizada a predição para o passo seguinte t_{i+1}. No entanto, mesmo para o modelo SARIMA, tem-se também as mesmas duas possibilidades mencionadas acima: para predições realizadas a partir do segundo passo do conjunto de teste, t_2, os valores referentes à variação com outliers utilizados para retreinamento do modelo poderão ser valores reais da variação ou os valores que foram preditos nos passos anteriores, quando os valores da variação foram classificados como outliers.

Dessa forma, dois métodos de predição foram desenvolvidos e empregados neste trabalho, baseados nas duas possibilidades colocadas acima. Esses métodos estão descritos a seguir:

- **Método de Predição com Valores Reais (VR)**: Nesse método, para se fazer as predições posteriores, utilizou-se somente os valores reais das variações do conjunto de teste com outliers, mesmo quando esses valores eram os outliers inseridos na Etapa 1;
- **Método de Predição com Valores Corrigidos (VC)**: Nesse método, a cada passo da predição, o valor da variação com outliers foi classificado como outlier ou não. Essa classificação foi feita utilizando-se um ”valor limite” (o erro absoluto máximo encontrado entre os valores preditos e os valores reais do conjunto de treinamento). A cada predição sobre o conjunto de teste com outliers em t_i, calculou-se o erro absoluto entre essa predição e seu valor real correspondente (em t_i). Caso classificado como outlier, ou seja, caso o erro absoluto nesse passo ultrapassasse o valor limite previamente estabelecido, o valor utilizado para a próxima predição em t_{i+1} seria o valor predito no passo anterior, em t_i. E caso o valor real em t_i fosse classificado como sendo ”não outlier”, o valor utilizado para a próxima predição, em t_{i+1}, seria o valor real da variação com outliers em t_i.

Conforme já mencionado anteriormente, a métrica selecionada para a avaliação das predições realizadas pelos dois métodos descritos acima foi o ”Erro Médio Absoluto” (MAE). Essa métrica é dada pela equação 3.1.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i|. \tag{3.1}$$

3.2.1.5 Etapa 5: Detecção de Outliers e Avaliação dos Modelos na Detecção de Outliers

Na Etapa 5 é dada continuidade no Procedimento TPD, iniciado na etapa anterior. As entradas para a Etapa 5 foram os resultados obtidos pelas atividades da Etapa 4: a predição para o conjunto de treinamento foi utilizada para realizar o cálculo do valor limite que servirá como

um limiar para a detecção de outliers; e as predições para as variações com outliers, realizadas por meio dos dois métodos de predição, foram utilizadas para o cálculo dos erros absolutos, que permitiram a classificação de outliers. Na Etapa 5 foram desenvolvidas 5 atividades, listadas abaixo:

- Estabelecimento de um valor limite;
- Cálculo dos erros médios para as variações com outliers;
- Classificação dos outliers;
- Cálculo do MCC para avaliação da detecção de outliers; e
- Cálculo da área abaixo da curva precision-recall e avaliação dos modelos.

Assim, a primeira atividade da Etapa 5 foi estabelecer um ”valor limite”, utilizado como um limiar para a detecção de outliers. O valor limite é o erro absoluto máximo encontrado entre o conjunto de treinamento e suas predições. Em seguida, foram calculados os erros absolutos entre as variações com outliers e suas predições, para os dois métodos de predição (VR e VC). Foi considerado outlier todo valor da variação com outliers cujo erro absoluto em relação à sua predição era maior do que o ”valor limite” previamente estabelecido. Foi gerado um vetor com 0 nas posições classificadas como ”não outliers” e 1 nas posições classificadas como outliers. Foi também gerado um vetor com o valor 1 na quinta e nona posições (onde foram inseridos os outliers) e 0 nas posições restantes. Esses dois vetores compostos por 0’s e 1’s foram comparados sendo, então, possível determinar o desempenho dos modelos na detecção de outliers quando se tem um valor limite pré-estabelecido. O procedimento de detecção de outliers, neste caso, caracteriza-se como um problema de classificação desbalanceado. Para avaliação dos modelos quanto à detecção de outliers, foi utilizada a métrica Matthews Correlation Coefficient (MCC), adequada para problemas de classificação com dados desbalanceados (SOKOLOVA; LAPALME, 2009). Essa métrica será detalhada mais abaixo.

Em seguida, utilizou-se a curva precision-recall (detalhada mais abaixo) e a área abaixo dessa curva (AUCPR) para avaliar o desempenho geral dos modelos quanto à detecção de outliers. Primeiramente, os erros absolutos entre as predições e o conjunto de teste foram normalizados. Em seguida, criou-se um vetor com cem valores entre 0 e 1. Esses valores foram, então, utilizados como os ”valores limites” na detecção de outliers e, para cada um desses valores limites foram calculadas as métricas precision e recall. Com os valores dessas duas métricas foram geradas as curvas precision-recall e calculadas as áreas abaixo dessas curvas.

Os procedimentos para detecção de outliers conforme desenvolvido neste trabalho caracterizam-se como problemas supervisionados de classificação. Para avaliação desses problemas, recomendam-se métricas como precision, recall e Matthews Correlation Coefficient (MCC) (SOKOLOVA; LAPALME, 2009). Essas métricas são indicadas quando a classe positiva

é mais importante do que a classe negativa. Como neste trabalho o objetivo é detectar as classes positivas, ou seja, as observações em uma série temporal classificadas como outliers, essas métricas passam a ser apropriadas. Além disso, deve ser levado em consideração que problemas de detecção de outliers geralmente são problemas desbalanceados, ou seja, a classe negativa é consideravelmente maior do que a classe positiva. De acordo com Fernández *et al.* (2018), nesse tipo de problema, métricas como a acurácia não são apropriadas, pois não levam a questão dos desbalanceamento dos dados em consideração. Por outro lado, métricas como precision e recall são bastante apropriadas para problemas com dados desbalanceados (HE; MA, 2013).

Para auxiliar no cálculo dessas métricas, e também do MCC (descrito mais abaixo) foi utilizada a matriz de confusão. A matriz de confusão é uma maneira conveniente de sumarizar a performance de um classificador, tratando-se de uma tabulação cruzada entre os valores reais das classes e as predições (BRZEZINSKI *et al.*, 2018; FERNÁNDEZ *et al.*, 2018). Um exemplo de matriz confusão é mostrado no Quadro 3. Na figura, VP significa verdadeiro positivo, VN

Quadro 3 – Exemplo de uma matriz de confusão.

		Valores Preditos	
		Positive	Negative
Valores Reais	Positive	VP	FN
	Negative	VP	VN

Fonte: Elaborada pelo autor.

significa verdadeiro negativo, FP siginifica falso positivo e FN significa falso negativo.

A métrica Precision avalia a fração de observações corretamente classificadas entre as classificadas como positivas, enquanto que Recall é a fração de observações positivas corretamente classificadas como positivas (BRANCO; TORGO; RIBEIRO, 2015; FERNÁNDEZ *et al.*, 2018; HE; MA, 2013). Em outras palavras, precision é o número de observações positivas corretamente classificadas dividido pelo número de observações rotuladas pelo classificador como positivas, e recall é o número de observações positivas corretamente classificadas dividido pelo número de observações positivas no conjunto de dados (BRZEZINSKI *et al.*, 2018; SOKOLOVA; LAPALME, 2009). Precision e recall tornam possível a avaliação da performance de um classificador em relação à classe minoritária (HE; MA, 2013) e são computadas a partir da matriz de confusão, conforme mostram as equações 3.2 e 3.3 (BRANCO; TORGO; RIBEIRO, 2015).

$$precision = \frac{VP}{VP + FP} \tag{3.2}$$

$$recall = \frac{VP}{VP + FN} \tag{3.3}$$

Em problemas desbalanceados, o objetivo passa a ser aprimorar a recall sem prejudicar a precision. Esse objetivo é, no entanto, muitas vezes conflitantes, uma vez que para aumentar

os VP para a classe minoritária, o número de FP frequentemente também aumenta, resultando numa precision reduzida (HE; MA, 2013). Essas métricas são utilizadas para a construção da curva precision-recall.

O MCC é uma medida que vem do campo da bioinformática, onde desbalanceamento de classes ocorre com frequência. Essa medida leva em consideração todos os valores da matriz de confusão, considerando erros e acertos na classificação de ambas as classes, conforme equação 3.4. O valor resultante MCC varia entre -1 e 1. Quanto mais próximo de -1, pior é o classificador. 0 indica que as classificações são semelhantes a escolhas aleatórias. E quanto mais próximo de 1, melhor o classificador (BRZEZINSKI *et al.*, 2018; FERNÁNDEZ *et al.*, 2018).

$$MCC = \frac{VP \times VN + FP \times FN}{\sqrt{(VP+FP)(VP+FN)(VN+FP)(VN+FN)}} \tag{3.4}$$

O MCC foi considerado neste trabalho pois diversos autores recomendam essa métrica para problemas desbalanceados (BRZEZINSKI *et al.*, 2018; BALDI *et al.*, 2000; BEKKAR; DJEMAA; ALITOUCHE, 2013).

A curva precision-recall (PRC, do inglês *Precision-Recal Curve*) também é indicada quando se tem dados desbalanceados (BOYD; ENG; PAGE, 2013; DAVIS; GOADRICH, 2006; SAITO; REHMSMEIER, 2015), sendo uma alternativa à curva ROC (do inglês *Receiver Operating Curve*), mais indicada quando as classes são balanceadas (DAVIS; GOADRICH, 2006; GOADRICH; OLIPHANT; SHAVLIK, 2006). A PRC é uma ferramenta de avaliação para classificação binária que permite a visualização do desempenho do modelo (classificador) para todos os valores limites possíveis (BOYD; ENG; PAGE, 2013), priorizando a capacidade do classificador em identificar a classe minoritária. Dessa forma, essa curva ignora os ”verdadeiros negativos” (TN), os quais são o grupo dominante em um problema desbalanceado (OZENNE; SUBTIL; MAUCORT-BOULCH, 2015).

A PRC é construída plotando-se pares (pontos) de valores das métricas precision e recall, que são obtidos utilizando-se diferentes ”valores limites” em um classificador (BOYD; ENG; PAGE, 2013). Após exibir esses pontos no espaço da PRC, pode-se calcular a área abaixo da curva (BOYD; ENG; PAGE, 2013; DAVIS; GOADRICH, 2006; OZENNE; SUBTIL; MAUCORT-BOULCH, 2015). Essa área é utilizada como um medida geral da capacidade do classificador em identificar a classe minoritária (BOYD; ENG; PAGE, 2013; OZENNE; SUBTIL; MAUCORT-BOULCH, 2015), não relacionada a nenhum valor limite específico (BOYD; ENG; PAGE, 2013), sendo útil também para avaliação e comparação entre diversos classificadores (SAITO; REHMSMEIER, 2015). A linha de base (y) da PRC é determinada pela proporção de positivos (P) e negativos (N) de forma que

$$y = P/(P+N). \tag{3.5}$$

Por exemplo, tem-se $y = 0.5$ para uma distribuição balanceada de classes, mas $y = 0.09$ para um distribuição desbalanceada com uma proporção de P:N igual a 1:10. Devido à essa

flexibilidade da linha de base, a área abaixo da PRC também muda de acordo com a proporção P:N (SAITO; REHMSMEIER, 2015).

CAPÍTULO

4

RESULTADOS E DISCUSSÃO

Conforme descrito na Seção 3.2, o Processo Experimental deste trabalho envolveu cinco etapas, conforme listadas abaixo:

- Etapa 1: Geração dos grupos de dados;
- Etapa 2: Pré-processamento dos grupos de dados;
- Etapa 3: Ajuste dos hiperparâmetros dos modelos;
- Etapa 4: Treinamento, predições e avaliação dos modelos de predição; e
- Etapa 5: Detecção de outliers e avaliação dos modelos de detecção.

A Etapa 1 gerou três grupos de dados, um para cada série temporal abordada neste trabalho. A Etapa 2 preprocessou os grupos de dados, convertendo-os em dados de entrada para as redes neurais. A Etapa 3 ocorreu paralelamente e ajustou os hiperparâmetros dos modelos. As Etapas 4 e 5 formaram um único procedimento (Procedimento TPD) que visou realizar os treinamentos dos modelos, as predições e a detecção de outliers. Os resultados das Etapas 1 a 3 foram apresentados no capítulo anterior, mais especificamente nas Subseções 3.2.1.1, 3.2.1.2 e 3.2.1.3. Neste capítulo, serão apresentados os resultados e análises referentes ao Procedimento TPD (Etapas 4 e 5).

O Procedimento TPD pode ser revisto no Algoritmo 1. As atividades relativas às Etapas 4 e 5 do Procedimento TPD foram repetidas 10 vezes para cada modelo (SARIMA, LSTM e GRU). As análises, comparações e discussão apresentados neste capítulo consideraram as médias e os desvios padrão das métricas calculadas ao longo dessas repetições.

4.1 Resultados da Etapa 4: Treinamento, Predição e Avaliação dos Modelos de Predição

Com a Etapa 4, é iniciada a execução do Procedimento TPD. Esse procedimento, conforme já descrito, é composto pelas Etapas 4 e 5 do Processo Experimental, e foi executado sobre os grupos de dados gerados e preprocessados nas Etapas 1 e 2. A Etapa 4 foi subdividida em 4 atividades e a Etapa 5 em 5 atividades. Essas atividades foram executadas de forma sequencial e contínua sobre os grupos de dados, repetidas 10 vezes para cada modelo. As quatro atividades que compõem a Etapa 4 estão descritas com detalhes na Subseção 3.2.1.4. Resumidamente, na primeira atividade foi realizado o treinamento do modelo. Na segunda atividade foi realizada a predição para o conjunto de treinamento e calculados os MAEs, apresentados na Tabela 5. Em seguida, na terceira atividade, foram realizadas as predições para as quatro variações com outliers, utilizando-se os dois métodos de predição: o Método VR e o Método VC. Na quarta atividade, foram calculados os MAEs entre os valores preditos e os valores reais das variações com outliers. As médias desses MAEs, calculadas a partir das 10 repetições, são apresentadas na Tabela 6.

Tabela 5 – MAEs entre os valores preditos e observados dos conjuntos de treinamentos dos três grupos de dados.

Grupo de Dados	Modelos		
	SARIMA	LSTM	GRU
Passengers	9.01	7.75	8.13
Milk	13.21	4.82	4.57
Beer	7.02	6.42	5.27

Fonte: Dados da pesquisa.

Os MAEs para os conjuntos de treinamentos dos três grupos de dados também são apresentadas no gráfico de barras da Figura 30. Para os conjuntos de treinamento de cada grupo de dados, os modelos que obtiveram as melhores médias dos MAEs foram:

- Grupo de Dados Passengers: Modelo LSTM (7.75);
- Grupo de Dados Milk: Modelo GRU (4.57);
- Grupo de Dados Beer: Modelo GRU (5.27).

As Figuras 31, 32 e 33 apresentam as médias dos MAEs para as variações com outliers dos Grupos de Dados Passengers, Milk e Beer, respectivamente. Nos gráficos, as linhas sólidas mostram as médias dos MAEs e as áreas sombreadas mostram os desvios padrão. A cor azul refere-se ao Método VR e a cor verde refere-se ao Método VC. No caso do modelo SARIMA, os gráficos não mostram o desvio padrão pois esse método não apresentou aleatoriedade nos

Tabela 6 – Médias dos MAEs para as predições das variações com outliers dos três grupos de dados, para os dois métodos de predição: Método de Predição com Valores Reais (VR) e Método de Predição com Valores Corrigidos (VC).

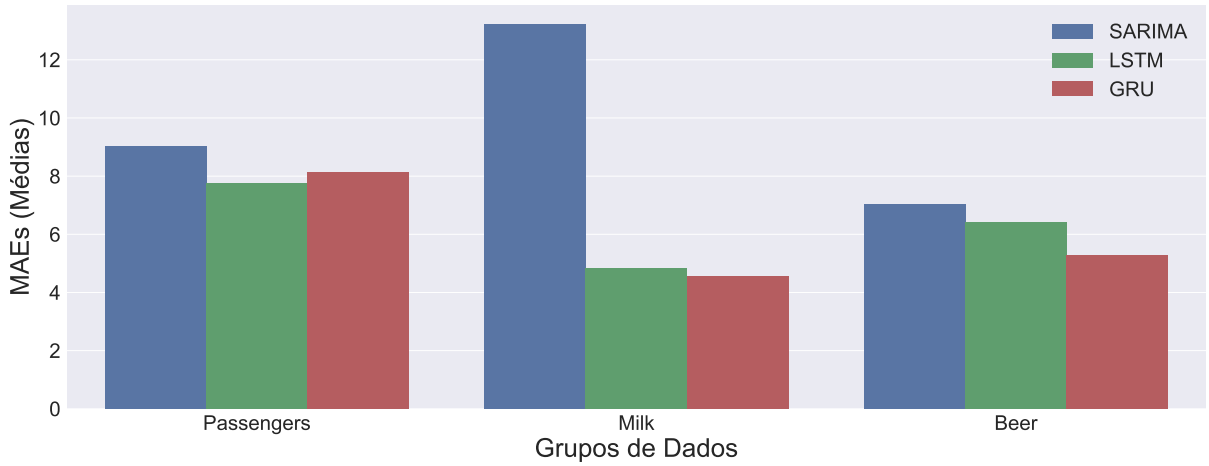
Grupo de Dados	Variações	Métodos	Modelos		
			SARIMA	LSTM	GRU
Passengers	2.0	VR	86.23	62.82	47.77
		VC	59.05	58.69	47.66
	1.8	VR	77.75	62.02	43.60
		VC	55.51	57.02	46.02
	1.5	VR	61.13	52.40	40.26
		VC	50.20	53.05	39.13
	1.2	VR	50.91	47.88	32.77
		VC	44.89	46.69	33.69
Milk	2.0	VR	79.99	54.57	52.49
		VC	37.32	57.22	50.81
	1.8	VR	70.70	51.97	50.70
		VC	33.99	52.90	48.97
	1.5	VR	61.11	46.43	46.61
		VC	28.99	47.17	44.90
	1.2	VR	50.69	41.86	40.07
		VC	24.00	45.68	38.82
Beer	2.0	VR	14.82	20.27	21.12
		VC	14.78	19.88	21.20
	1.8	VR	13.49	17.98	20.02
		VC	13.64	19.43	19.97
	1.5	VR	11.46	16.98	17.99
		VC	11.92	16.80	17.35
	1.2	VR	9.75	15.28	15.76
		VC	9.75	14.91	15.97

Fonte: Dados da pesquisa.

resultados entre as dez repetições. Percebe-se que, para todos os casos, os erros decrescem à medida que os pesos dos outliers decaem, mostrando que os modelos preditivos utilizados neste estudo são sensíveis a ruídos nas séries temporais, e que, portanto, as predições de séries temporais com outliers cujos valores são menos distorcidos apresentam melhores desempenhos. O valor de ”p-value” que aparece nos gráficos é resultado do teste estatístico que foi realizado entre os dois métodos de predição e será detalhado abaixo.

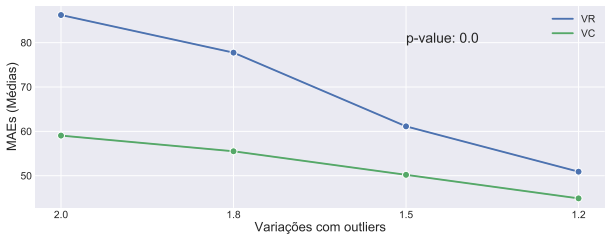
Conforme já mencionado e detalhado na Subseção 3.2.1.4, dois métodos de predição foram utilizados neste trabalho: o Método VR, que utilizou valores da variação com outliers para realizar as predições seguintes; e o Método VC, que utilizou os valores preditos para as predições posteriores, quando o valor da variação foi classificado como outlier. Uma pergunta que buscou-se responder neste estudo foi se haveria diferença entre os resultados apresentados por esses dois métodos. Para verificar essa questão, foi utilizado o Teste t de Student, o qual

Figura 30 – Médias dos MAEs para as predições dos conjuntos de treinamentos para os três grupos de dados.

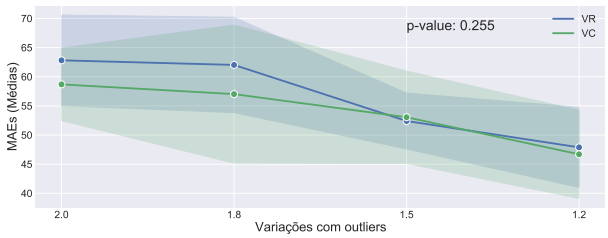


Fonte: Elaborada pelo autor.

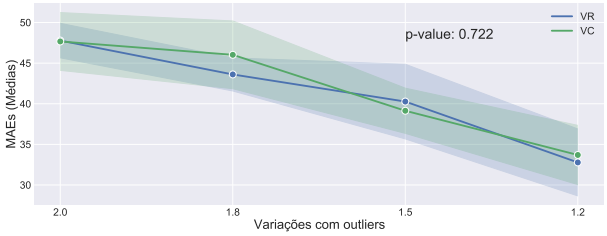
Figura 31 – Médias dos MAEs para as variações com outliers do Grupo de Dados Passengers, com os dois métodos de predição.



(a) SARIMA



(b) LSTM

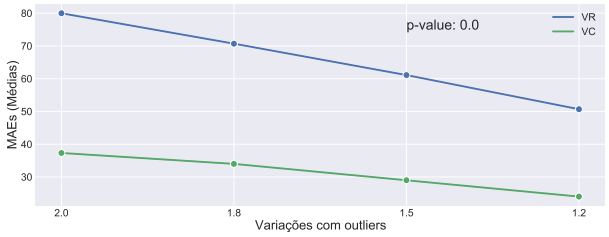


(c) GRU

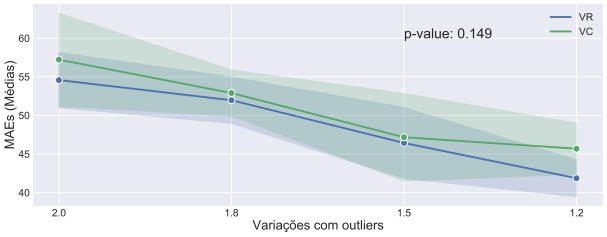
Fonte: Elaborada pelo autor.

verifica se duas amostras independentes são oriundas de uma mesma distribuição. Nesse teste, a hipótese nula é de que as duas amostras são provenientes de uma mesma distribuição, e que não há diferença estatisticamente significativa entre elas. Na presente situação, indicaria que não há diferença estatisticamente significativa entre os resultados apresentados pelos dois métodos de predição. As duas amostras consideradas foram os MAEs das predições para as variações com outliers do Método VR e do Método VC, calculados ao longo das 10 repetições do Procedimento TPD. Isso foi feito para os três grupos de dados e para os três modelos. Os resultados dos Testes t são apresentados na Tabela 7.

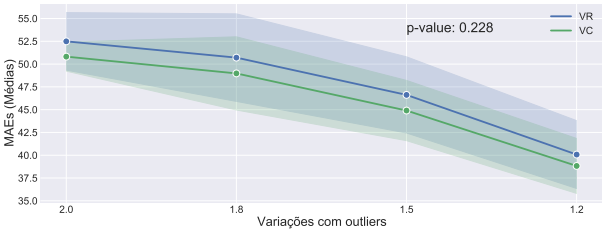
Figura 32 – Médias dos MAEs para as variações com outliers do Grupo de Dados Milk, com os dois métodos de predição.



(a) SARIMA



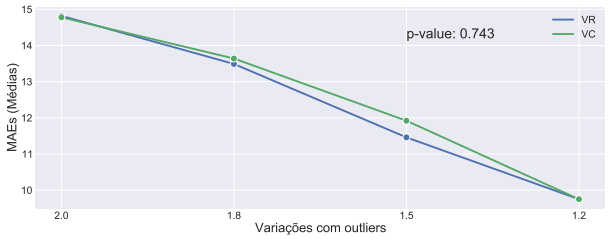
(b) LSTM



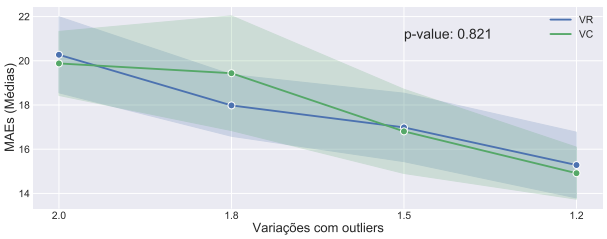
(c) GRU

Fonte: Elaborada pelo autor.

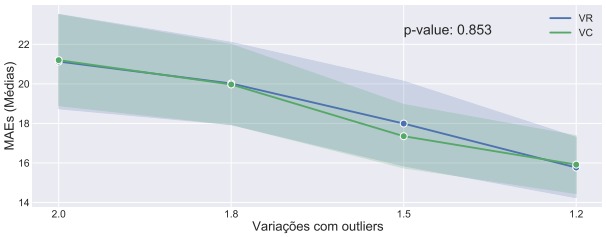
Figura 33 – Médias dos MAEs para as variações com outliers do Grupo de Dados Beer, com os dois métodos de predição.



(a) SARIMA



(b) LSTM



(c) GRU

Fonte: Elaborada pelo autor.

O nível de significância adotado foi $\alpha = 0.05$. Na tabela, quando $p \leq 0.05$, rejeita-se a hipótese nula, indicando que há diferença estatisticamente significativa entre os resultados alcançados pelos dois métodos de predição; e quando $p > 0.05$, a hipótese nula não deve ser rejeitada, indicando que os resultados alcançados pelos dois métodos de predição não possuem diferença estatisticamente significativa. Em outras palavras, neste caso, não há diferença entre os resultados apresentados pelos dois métodos e ambos podem ser utilizados indistintamente.

Tabela 7 – Teste t de Student realizado para cada grupo de dados sobre os resultados (MAEs) obtidos pelos dois métodos de predição: VR e VC.

	SARIMA		LSTM		GRU	
Grupo de Dados	t-stats	$p-$value	t-stats	$p-$value	t-stats	$p-$value
Passengers	6.995	0.000	1.147	0.255	-0.358	0.722
Milk	17.957	0.000	-1.457	0.149	1.214	0.228
Beer	-0.329	0.743	-0.227	0.821	0.186	0.853

Fonte: Dados da pesquisa.

As duas situações que apresentaram diferença estatisticamente significativa entre os dois métodos de predição foram para os Grupos de Dados Passengers e Milk, com o modelo SARIMA. Em todas as outras situações, não houve diferença estatisticamente significativa entre os dois métodos de predição, indicando que os dois métodos de predição podem ser utilizados indistintamente.

4.2 Resultados da Etapa 5: Detecção de Outliers e Avaliação dos Modelos de Detecção

A Etapa 5 é composta por 5 atividades que dão continuidade às atividades da Etapa 4 e ao Procedimento TPD. A Etapa 4 recebeu como entrada um grupo de dados e um modelo, o modelo foi treinado e, em seguida, foram realizadas predições para o conjunto de treinamento e para as variações com outliers. Na Etapa 5, a predição para o conjunto de treinamento foi utilizada para se estabelecer um valor limite, um limiar utilizado na classificação de outliers, e as predições para as variações com outliers foram utilizadas tanto para a classificação de outliers, como também para a construção das curvas precision-recall, o que permitiu a avaliação geral dos modelos de detecção de outliers. Essas atividades foram repetidas 10 vezes para cada grupo de dados/modelo. Primeiramente serão apresentados os resultados referentes à detecção de outliers com um valor limite específico e, em seguida, será apresentada a avaliação geral dos modelos quanto à detecção de outliers.

4.2.1 Detecção de Outliers com Valor Limite Específico

A primeira atividade realizada na Etapa 5 foi o estabelecimento de um valor limite, utilizado como um limiar na classificação de outliers. O valor limite foi, então, comparado com os erros absolutos calculados entre as variações do conjunto de teste com outliers e suas predições. Foram considerados outliers os valores das variações com outliers cujos erros absolutos ultrapassaram o valor limite. Conforme detalhado na Seção 3.2.1.5, essa atividade gerou um vetor composto por 1's nas posições classificadas como outliers e 0's nas posições restantes. Foi também gerado um vetor com 0's em todas as posições, exceto na quinta e nona posições onde,

na Etapa 1, foram deliberadamente inseridos os outliers. Trata-se, portanto, de um problema de classificação, conforme também explicado na Seção 3.2.1.5, sendo o MCC a métrica selecionada para avaliação dos modelos de detecção. A Tabela 8 mostra as médias dos MCCs, calculadas a partir das dez repetições, referentes à detecção de outliers nos três grupos de dados.

Tabela 8 – Médias dos MCCs para as detecções de outliers dos três grupos de dados, para os dois métodos de predição: Método de Predição com Valores Reais (VR) e Método de Predição com Valores Corrigidos (VC). Para a variação 1.2 do Grupo de Dados Beer, o modelo SARIMA não conseguiu detectar outliers.

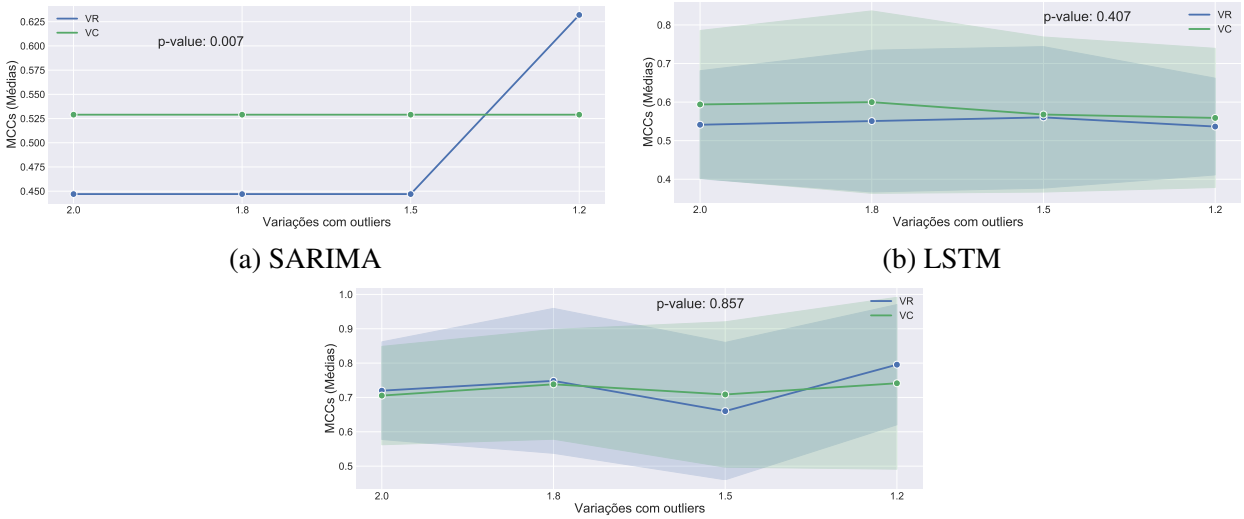
Grupo de Dados	Variações	Métodos	Modelos		
			SARIMA	LSTM	GRU
Passengers	2.0	VR	0.45	0.54	0.72
		VC	0.53	0.59	0.70
	1.8	VR	0.45	0.55	0.75
		VC	0.53	0.59	0.74
	1.5	VR	0.45	0.56	0.67
		VC	0.53	0.57	0.70
	1.2	VR	0.63	0.54	0.79
		VC	0.53	0.56	0.74
Milk	2.0	VR	0.45	0.63	0.60
		VC	1.00	0.51	0.60
	1.8	VR	0.45	0.61	0.58
		VC	1.00	0.49	0.57
	1.5	VR	0.38	0.63	0.54
		VC	1.00	0.54	0.61
	1.2	VR	0.38	0.58	0.59
		VC	1.00	0.53	0.56
Beer	2.0	VR	1.00	0.95	0.78
		VC	1.00	0.94	0.76
	1.8	VR	1.00	0.91	0.75
		VC	1.00	0.92	0.72
	1.5	VR	0.67	0.91	0.65
		VC	0.67	0.85	0.77
	1.2	VR	–	0.67	0.66
		VC	–	0.71	0.67

Fonte: Dados da pesquisa.

As Figuras 34, 35 e 36 mostram as médias dos MCCs para as variações com outliers dos Grupos de Dados Passengers, Milk e Beer, respectivamente. Nos gráficos, as linhas sólidas mostram as médias dos MCCs e as áreas sombreadas mostram os desvios padrão. A cor azul refere-se ao Método VR e a cor verde refere-se ao método VC. O modelo SARIMA não apresenta variabilidade em seus resultados, pois os valores obtidos em todas as 10 aplicações são os mesmo, uma vez que não há aleatoriedade na aplicação desse modelo. Assim como nos gráficos das médias dos MAEs, é possível visualizar nos gráficos os ”p-values” resultantes dos testes estatísticos realizados entre os dois métodos de predição.

Pelos gráficos, é possível perceber que, para a maioria dos casos, o MCC mantém-se relativamente estável ao longo das variações dos pesos dos outliers. Ou seja, o desempenho na detecção de outliers parece não ter sofrido muita influência do peso dado aos outliers (2.0, 1.8, 1.5 ou 1.2). Parece que isso não ocorre somente com o modelo SARIMA em duas situações: para as variações do Grupo de Dados Passengers, com o método VR, na qual o MCC salta de 0.45 (na variação 1.5) para 0.63 (na variação 1.2); e para as variações do Grupo de Dados Beer, com os dois métodos de predição, onde o MCC decresce de 1.0 (na variação 1.8) para 0.67 (na variação 1.5). Além disso, na maioria dos casos, parece não haver grande diferença nos desempenhos (nos MCCs) entre os dois métodos de predição. Os casos onde essa diferença aparece mais nitidamente ocorrem com o modelo SARIMA, com as variações dos Grupos de Dados Passengers e Milk. Para as variações do Grupo de Dados Milk, o método de predição VC apresentou MCC igual a 1.00 para todas as séries com outliers, enquanto que para o método VR o MCC variou de 0.38 a 0.45. No caso das variações para o Grupo de Dados Beer, parece não haver diferença estatisticamente significativa entre os MCCs dos dois métodos de predição para nenhum modelo. Porém, para as variações com outliers desse Grupo de Dados, o modelo SARIMA conseguiu detectar outliers somente para as séries com pesos 2.0, 1.8 e 1.5, com praticamente os mesmos valores de MCCs para os dois métodos.

Figura 34 – Médias dos MCCs para as detecções de outliers do Grupo de Dados Passengers, com os dois métodos de predição.



(a) SARIMA

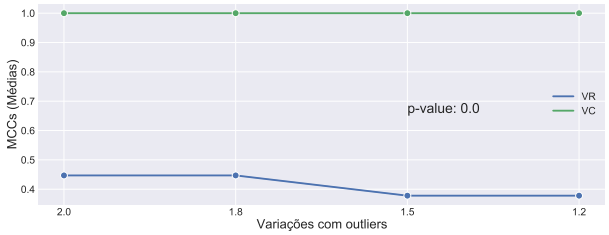
(b) LSTM

(c) GRU

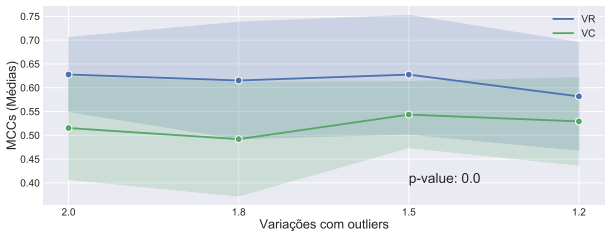
Fonte: Elaborada pelo autor.

Semelhante ao que foi feito com as predições, buscou-se conhecer se houve diferença significativa no desempenho da detecção de outliers (MCCs), dados os dois métodos de predição (VR e VC). Para a verificação dessa questão, foi realizado o Teste t de Student com os valores de MCCs coletados ao longo das 10 repetições do Procedimento TPD para os dois métodos de predição. Na Tabela 9 é possível visualizar os resultados desse teste. Da mesma forma que

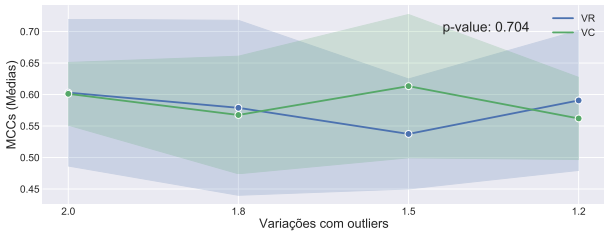
Figura 35 – Médias dos MCCs para as detecções de outliers do Grupo de Dados Milk, com os dois métodos de predição.



(a) SARIMA



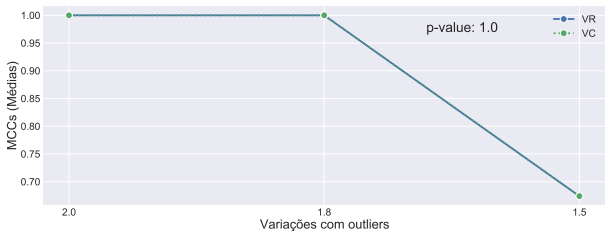
(b) LSTM



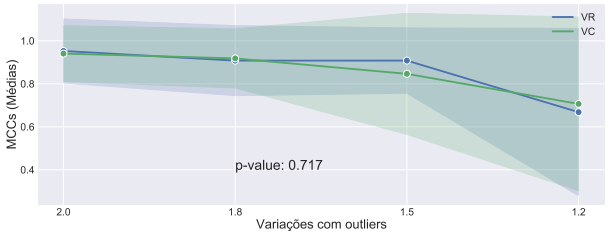
(c) GRU

Fonte: Elaborada pelo autor.

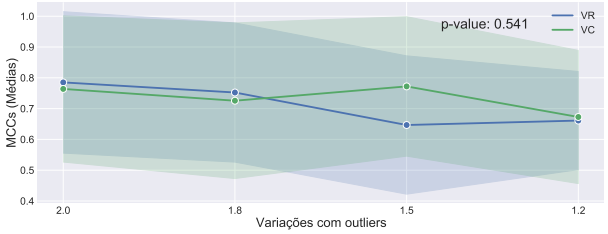
Figura 36 – Médias dos MCCs para as detecções de outliers do Grupo de Dados Beer, com os dois métodos de predição.



(a) SARIMA



(b) LSTM



(c) GRU

Fonte: Elaborada pelo autor.

anteriormente, o nível de significância adotado foi $\alpha = 0.05$.

Somente em três casos houve diferença significativa entre as detecções, dados os dois métodos de predição: para as variações dos Grupos de Dados Passengers e Milk, com o modelo SARIMA; e para as variações do Grupo de Dados Milk, com o modelo LSTM. Para o Grupo de Dados Passengers, o Método VC apresentou melhor desempenho para todas as variações, exceto para a variação 1.2, para a qual o Método VR apresentou maior MCC. Para o Grupo

Tabela 9 – Teste t de Student para os resultados dos dois métodos de predição na detecção de outliers.

	SARIMA		LSTM		GRU	
Série	t-stats	$p-$value	t-stats	$p-$value	t-stats	$p-$value
Passengers	-2.787	0.007	-0.833	0.407	0.181	0.857
Milk	-106.34	0.000	4.046	0.000	-0.382	0.704
Beer	0.000	1.00	0.364	0.717	-0.614	0.541

Fonte: Dados da pesquisa.

de Dados Milk com o modelo SARIMA, o Método VC apresentou melhor desempenho para todas as variações, e com o modelo LSTM, o Método de Predição VR apresentou melhor desempenho para todas as variações. Para todos os outros casos, o Teste t demonstrou não haver diferença estatisticamente significativa na detecção de outliers, dados os dois métodos de predição. Percebe-se, considerando-se os testes realizados para as predições (MAEs) e os realizados para as detecções (MCCs), que os dois métodos mostraram-se indiferentes nos mesmos casos, exceto para as variações do Grupo de Dados Milk com o modelo LSTM, onde, para as predições não houve diferença e para as detecções de outliers houve diferença entre os resultados apresentados pelos dois métodos.

4.2.2 Avaliação dos Modelos de Detecção de Outliers

A partir das predições realizadas na atividade anterior, foram calculados os erros absolutos entre as variações com outliers e suas predições e geradas as Curvas Precision-Recall para os modelos em todos os cenários. As Curvas Precision-Recall mostram o desempenho dos modelos com relação à detecção de outliers para todas as possibilidades de valores limites, conforme detalhado em 3.2.1.5. Com as curvas geradas, foi possível calcular também as AUCPRs, apresentadas na Tabela 10.

A AUCPR igual a 1.00 indica que existe pelo menos um valor limite que permite classificar corretamente todos os outliers. Nesse caso, os erros absolutos estão distribuídos de tal forma que é possível separar somente os erros relacionados aos outliers. Portanto, para todas as variações com outliers dos Grupos de Dados Milk e Beer, existem valores limites que permitiriam identificar corretamente todos os outliers. Para as variações com outliers do Grupo de Dados Passengers, isso seria possível somente com o método VC. Com o método VR, nenhum modelo alcançou AUCPR igual a 1.0.

As curvas precision-recall para os Grupos de Dados Passengers, Milk e Beer estão nas Figuras 37, 38 e 39, respectivamente. Cada figura apresenta quatro gráficos, um para cada variação com outliers, e, para cada variação com outliers, são apresentadas as curvas precision-recall para os dois métodos de predição e para os três modelos. Além disso, os gráficos mostram a ”baseline” que basicamente é a proporção da classe positiva em relação à classe negativa, no conjunto de teste.

Tabela 10 – AUCPRs das curvas Precision-recall para os três grupos de dados.

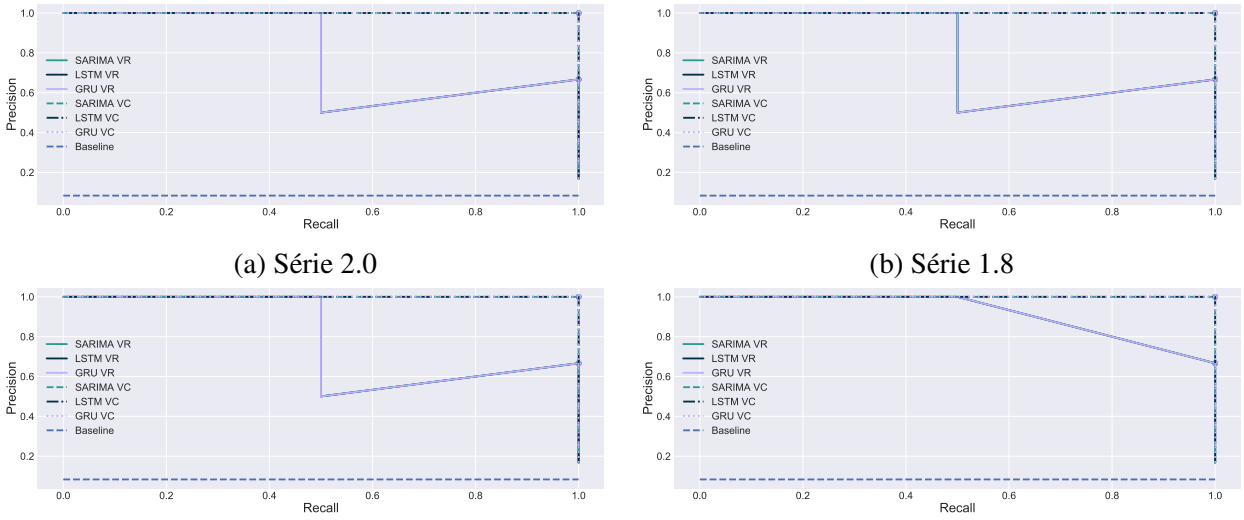
Grupo de Dados	Variações	Métodos	Modelos		
			SARIMA	LSTM	GRU
Passengers	2.0	VR	0.79	0.79	0.79
		VC	1.00	1.00	1.00
	1.8	VR	0.79	0.79	0.79
		VC	1.00	1.00	1.00
	1.5	VR	0.79	0.79	0.79
		VC	1.00	1.00	1.00
	1.2	VR	0.92	0.92	0.92
		VC	1.00	1.00	1.00
Milk	2.0	VR	1.00	1.00	1.00
		VC	1.00	1.00	1.00
	1.8	VR	1.00	1.00	1.00
		VC	1.00	1.00	1.00
	1.5	VR	1.00	1.00	1.00
		VC	1.00	1.00	1.00
	1.2	VR	1.00	1.00	1.00
		VC	1.00	1.00	1.00
Beer	2.0	VR	1.00	1.00	1.00
		VC	1.00	1.00	1.00
	1.8	VR	1.00	1.00	1.00
		VC	1.00	1.00	1.00
	1.5	VR	1.00	1.00	1.00
		VC	1.00	1.00	1.00
	1.2	VR	1.00	1.00	1.00
		VC	1.00	1.00	1.00

Fonte: Dados da pesquisa.

As curvas precision-recall que se estendem até a extremidade superior direita dos gráficos mostram que foi possível, nesses casos, alcançar precision e recall igual a 1.0. Corroborando com o que foi dito anteriormente, isso significa que existe pelo menos um valor limite com o qual foi possível classificar corretamente todos os outliers (recall) e que todos os valores classificados como outliers realmente eram outliers (precision). Pode-se dizer que os modelos de detecção de outliers (ou classificadores de outliers) que se enquadram nessa situação alcançaram desempenho máximo.

Por outro lado, algumas curvas não se estenderam até a extremidade superior direita do gráfico. São as curvas referentes ao modelo SARIMA com o método de predição VR, para o Grupo de Dados Passengers. As curvas precision-recall mostram que foi possível, com pelo menos um valor limite, obter recall igual a 1.0. Isso significa que existe pelo menos um valor limite capaz de identificar e classificar todos os verdadeiros outliers. No entanto, quando se obtém esse resultado para o recall, o precision é menor do que 1.0, o que indica que haverá falsos positivos entre os valores classificados como outliers. Em outras palavras, nesses casos, todos os

Figura 37 – AUCPRs das curvas Precision-recall para o Grupo de Dados Passengers.



(a) Série 2.0

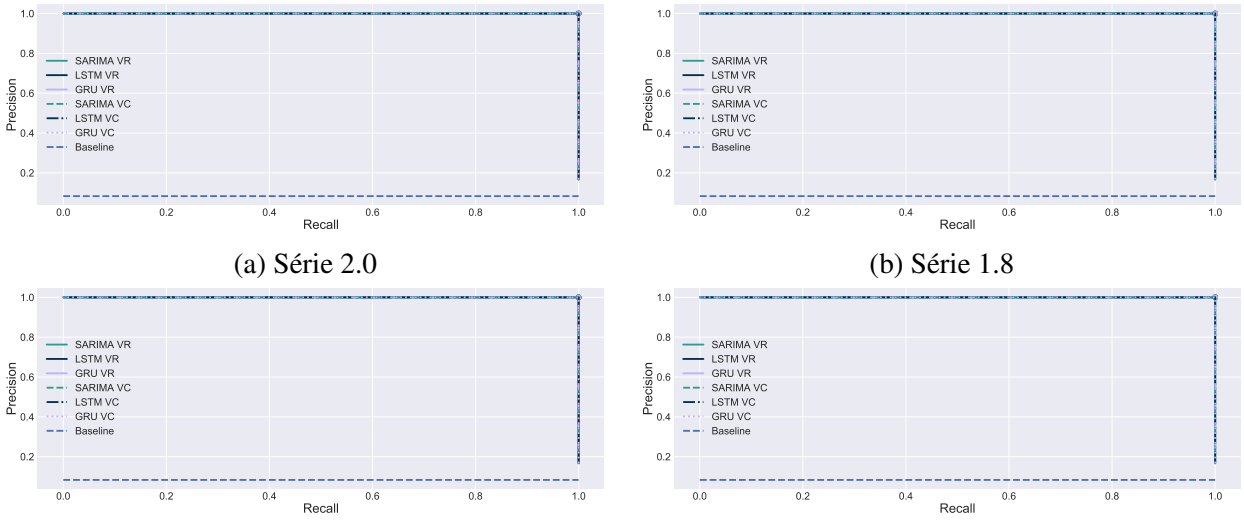
(b) Série 1.8

(c) Série 1.5

(d) Série 1.2

Fonte: Elaborada pelo autor.

Figura 38 – AUCPRs das curvas Precision-recall para o Grupo de Dados Milk.



(a) Série 2.0

(b) Série 1.8

(c) Série 1.5

(d) Série 1.2

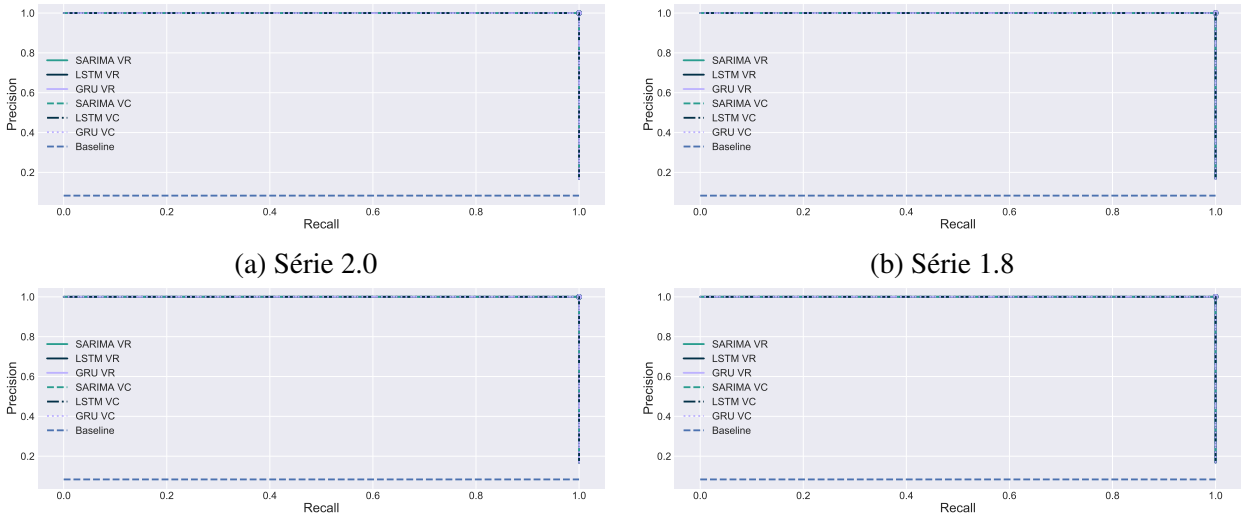
Fonte: Elaborada pelo autor.

verdadeiros outliers serão identificados e classificados como tal (recall = 1.0), porém, entre os valores classificados como outliers, haverão valores que não são outliers e que foram, portanto, classificados erroneamente (precision < 1.0).

4.3 Discussão

Um resumo dos melhores resultados obtidos para as atividades de predição e detecção de outliers para cada grupo de dados pode ser visto no Quadro 4. A seguir, são discutidos os

Figura 39 – AUCPRs das curvas Precision-recall para o Grupo de Dados Beer.



(a) Série 2.0

(b) Série 1.8

(c) Série 1.5

(d) Série 1.2

Fonte: Elaborada pelo autor.

resultados com relação a cada modelo e tarefa, iniciando com os experimentos de predição e finalizando com os resultados relacionados à detecção de anomalias.

Quadro 4 – Melhores resultados alcançados pelos modelos para cada grupo de dados.

	Grupo de Dados		
Tarefas	Passengers	Milk	Beer
Predição	GRU (1.2 / VR)	SARIMA (1.2 / VC)	SARIMA (1.2 / VC)
Detecção outliers	GRU (1.2 / VR)	SARIMA (Todas / VC)	SARIMA (2.0, 1.8 / VR, VC)

Com relação aos resultados de predição, no Grupo de Dados Passengers, o modelo que melhor se ajustou ao conjunto de treinamento foi a rede neural LSTM (MAE=7.75). Nesse cenário, o modelo que apresentou melhor desempenho foi o GRU, para a variação 1.2 com o Método de Predição VR (MAE=32.77). No geral, para esse grupo de dados, em todos os casos em que foi utilizado o Método de Predição VR, as redes neurais LSTM e GRU obtiveram melhor desempenho do que o modelo SARIMA. No entanto, com o Método de Predição VC, o modelo SARIMA, no geral, saiu-se melhor do que a rede LSTM, mas não superou o modelo GRU. Com relação aos dois métodos de predição, para o modelo SARIMA, o Método VC apresentou melhor desempenho para todas as variações com outliers, indicando que esse método é mais apropriado para esse modelo. Para a rede LSTM, o Método VC apresentou resultado superior para a maioria dos casos. Já para a rede neural GRU, essa relação é inconclusiva, pois em alguns casos o resultado melhora ao utilizar o Método VC e em outros piora, sendo possível constatar que essa variação nos resultados entre os dois métodos de predição não está associada aos pesos dos outliers. Para esse grupo de dados, o modelo GRU, indiferentemente do método de predição adotado, apresentou melhor desempenho em todos os casos. A série temporal Passengers possui

144 observações, apresentando tendência de alta forte, sazonalidade anual e volatilidade não constante, aumentando ao longo do tempo. A tendência de alta forte e a volatilidade crescente do Grupo de Dados Passengers podem ter sido os fatores que favoreceram o modelo GRU a obter o melhor desempenho.

Para o Grupo de Dados Milk, o modelo que melhor se ajustou aos dados de treinamento foi a rede neural GRU (MAE=4.57). O melhor desempenho na predição foi apresentado pelo modelo SARIMA, para a variação 1.2 com o Método de Predição VC (MAE=20.00). Para esse grupo de dados, quando utilizado o Método de Predição VR, as redes neurais LSTM e GRU apresentaram melhor desempenho em relação ao modelo SARIMA para todas as variações com outliers. Com o Método de Predição VC, o modelo SARIMA saiu-se melhor em todos os casos, em relação às duas redes neurais. Além disso, o modelo SARIMA apresentou melhores resultados com o Método VC, com uma redução significativa do MAE em todos os casos com relação ao VR. Com a rede LSTM ocorreu o oposto, ou seja, esse modelo apresentou melhor desempenho com o Método VR em todos os casos. Já a rede GRU desempenhou-se melhor com o Método de Predição VC em todos os casos. Para esse grupo de dados, o modelo SARIMA com o Método VC apresentou desempenho superior em todos os casos. A série temporal Milk possui 168 observações e apresenta uma leve tendência de alta, sazonalidade anual e volatilidade constante ao longo de toda a série. A tendência não acentuada e a volatilidade constante pode ter contribuído para o que o modelo SARIMA alcançasse o melhor desempenho nesse caso.

Para o Grupo de Dados Beer, o modelo que melhor se ajustou ao conjunto de treinamento foi a rede neural GRU (MAE=5.27). Já para as predições, o Modelo SARIMA apresentou o melhor desempenho, para a variação 1.2 com o Método de Predição VC (MAE=9.75). O modelo SARIMA apresentou melhor desempenho em todos os casos, para os dois métodos de predição. Além disso, o desempenho desse modelo não se alterou significativamente entre os métodos de predição VR e VC. Entre as duas redes neurais, o modelo LSTM saiu-se melhor que o modelo GRU para todos os casos. Com relação ao modelo LSTM entre os dois métodos de predição, observa-se que em alguns casos o desempenho melhora e em outros piora, sem relação direta com os pesos adotados nos outliers. O modelo SARIMA apresentou melhor desempenho para todas as variações com outliers desse grupo de dados, para os dois métodos de predição. A série temporal Beer possui 476 observações, apresentando tendência de alta leve por um período de tempo, passando depois para uma tendência lateral e sazonalidade anual. A volatilidade dessa série mantém-se constante por alguns períodos, com mudanças abruptas em períodos curtos intercalados. Nesse caso, a volatilidade mais constante pode ter contribuído com o desempenho superior apresentados pelo modelo SARIMA.

Não foi possível fazer nenhuma associação entre os resultados alcançados pelos modelos e a sazonalidade das séries temporais pois essa é igual para as três séries. Também não foi possível fazer nenhuma associação entre o comprimento das séries e os desempenhos dos modelos pois, além de, no geral, as três séries poderem ser consideradas como de curto comprimento, o modelo

SARIMA obteve melhor desempenho com a série mais longa (Beer) como também com uma série mais curta (Milk). Percebe-se também que, em nenhum caso, o modelo que melhor se ajustou aos dados de treinamento obteve o melhor desempenho nas predições. Isso corrobora com Hyndman e Athanasopoulos (2018), quando afirmam que um modelo que se ajusta bem aos dados não necessariamente apresentará bom desempenho nas predições, indicando dificuldade de generalização. Além disso, segundo Parmezan, Souza e Batista (2019), o desempenho das redes neurais recorrentes depende ”pesadamente” da quantidade de dados disponível. Dessa forma, as séries de comprimentos mais curtos abordadas neste estudo podem ter favorecido o modelo estatístico SARIMA. Também convém ressaltar que, assim como demonstrado por Gao *et al.* (2020), as redes GRU, embora com estruturas mais simples, demonstraram, no geral, desempenho similar às redes LSTM na tarefa de predição.

Com relação à detecção de outliers, para o Grupo de Dados Passengers, o melhor desempenho ocorreu com o modelo GRU, variação com outliers 1.2, pelo método VR (MCC=0.79). Para esse grupo de dados, o modelo GRU apresentou melhor desempenho em todos os casos, com os dois métodos de predição. Para o modelo SARIMA e LSTM, no geral, os melhores resultados foram alcançados pelo Método de predição VC. Já o modelo GRU, no geral, o Método de Predição VR apresentou melhores resultados. Nesse caso, percebe-se que o modelo que obteve o melhor desempenho na predição, alcançou também o melhor desempenho na detecção de outliers. Inclusive, o melhor desempenho tanto na predição quanto para a detecção ocorreram para a mesma variação com outliers e pelo mesmo método de predição (1.2, VR).

Para o Grupo de Dados Milk, o modelo SARIMA apresentou melhor desempenho para todas as variações com outliers, pelo Método VC (MCC=1.00). Para o modelo LSTM, o Método VR apresentou melhores resultados em relação ao Método VC. No caso do modelo GRU, os resultados de VR e VC variam sem relação com os pesos dos outliers. Nesse caso, observa-se que o modelo SARIMA obteve melhor desempenho tanto na predição quanto na detecção de outliers. A variação com outliers 1.2 aparece com o melhor resultado na predição pelo método VC e também entre os melhores desempenhos na detecção.

Para o Grupo de Dados Beer, o Modelo SARIMA também apresentou o melhor desempenho, mas somente para as variações 2.0 e 1.8 (MCC=1.00), para dois métodos de predição. Para as variações 1.2 e 1.5, o modelo LSTM apresentou melhor desempenho para os dois métodos de predição. O modelo SARIMA não conseguiu detectar outliers na variação 1.2. Considerando somente os modelos LSTM e GRU, o primeiro apresentou melhor desempenho para todas as variações com outliers desse grupo de dados. O modelo GRU apresentou o pior desempenho entre os três modelos para todos os casos. Portanto, para esse grupo de dados, o modelo SARIMA apresentou melhor desempenho para as variações 2.0 e 1.8, para os dois métodos de predição, e o modelo LSTM apresentou melhor desempenho para as variações 1.5 e 1.2, também para os dois métodos de predição. Nesse caso, o modelo LSTM destacou-se nas variações que apresentam maior dificuldade de detecção de outliers, nas quais os pesos são menores e os outliers

encontram-se mais próximos dos valores reais das séries temporais.

Percebe-se, assim, que quando utilizados os valores limites específicos para a detecção de outliers, calculados a partir dos erros de predição do conjunto de treinamento, os modelos SARIMA, LSTM e GRU apresentaram desempenhos variados entre os diversos cenários com outliers gerados abordados neste estudo. Inclusive, vale notar que, a despeito da afirmação de Chalapathy e Chawla (2019), Davis, Raina e Jagannathan (2020) de que frequentemente os modelos estatísticos de detecção de outliers falham em capturar completamente a estrutura de dados complexos, neste estudo, constatou-se que o modelo estatístico SARIMA apresentou desempenho superior na detecção de outliers em relação às redes neurais em dois dos três grupos de dados. No entanto, por outro lado, quando utilizadas as curvas precision-recall, que avalia os desempenhos dos modelos com todas as possibilidades de valores limites, os três modelos (SARIMA, LSTM e GRU) apresentaram o mesmo desempenho em todos os cenários e, em alguns casos, esse desempenho foi máximo.

CAPÍTULO

5

CONCLUSÃO

Neste trabalho, foi estudado um modelo de detecção de outliers baseado nas capacidades preditivas das redes neurais LSTM e GRU. A diferença entre os valores preditos e os valores observados foram calculados como erros de predição e utilizados para detectar outliers em dados não vistos de três séries temporais univariadas de contexto econômico. Como linha de base para comparações, foi utilizado o modelo estatístico SARIMA. Primeiramente, utilizou-se um valor limite específico para detecção de outliers, calculado a partir dos erros de predição do conjunto de treinamento. Num segundo momento, os modelos foram testados com todos os valores limites possíveis para detecção de outliers.

Com os resultados obtidos neste estudo, foi possível investigar as capacidades preditivas das redes neurais recorrentes LSTM e GRU e do modelo estatístico SARIMA. Embora as predições foram realizadas com dados previamente coletados, a abordagem aqui desenvolvida pode ser adaptada para predição e detecção de outliers em tempo real. As capacidades preditivas dos modelos foram avaliadas por meio da métrica MAE e, em seguida, comparadas. Foi possível identificar os modelos que melhor desempenharam para cada grupo de dados: a rede GRU apresentou melhor desempenho na predição para o Grupo de Dados Passengers, enquanto que o modelo SARIMA apresentou melhores resultados para os Grupos de Dados Milk e Beer. No entanto, no geral, todos os modelos apresentaram, em pelo menos um cenário, desempenho satisfatório. Além disso, constatou-se que os modelos que melhor se ajustaram ao dados durante o treinamento, não apresentaram os melhores resultados na predição sobre valores não vistos, mas os modelos que apresentaram melhor predição foram os que obtiveram melhores resultados na detecção de outliers.

A eficácia dos modelos de detecção de outliers gerados a partir dos erros de predição foi avaliada em diversos cenários. Para cada série temporal gerou-se um grupo de dados com quatro variações com outliers, nas quais os outliers possuíam pesos diferentes. Com valores limites específicos, os desempenhos entre os modelos se alteraram consideravelmente. No entanto,

quando consideradas todas as possibilidades de valores limites com as curvas precision-recall, os modelos mostraram desempenho idêntico em todos os cenários. Inclusive, para os Grupos de Dados Milk e Beer, em todos os cenários, os três modelos conseguiram detectar corretamente todos os outliers. O Grupo de Dados Passengers, com o Método de Predição VC também foi possível detectar corretamente todos os outliers. Para esse grupo de dados, o Método VR não alcançou a mesma eficácia, mas ainda assim os resultados podem ser considerados satisfatórios. A identificação do valor limite adequado para cada situação mostrou-se um ponto importante na construção de uma metodologia que visa detectar outliers. De fato, os resultados mostraram que todos os modelos tem o mesmo potencial como detectores de outliers, desde que o limiar ótimo, ou o intervalo com limiares ótimos, seja encontrado para cada modelo.

No geral, os resultados obtidos com este trabalho evidenciam que métodos de detecção de outliers que utilizam erros de predições obtidos a partir de redes neurais LSTM e GRU apresentam bons desempenhos e merecem ser mais estudados. Embora o modelo SARIMA tenha se saído melhor tanto na predição quanto na detecção de outliers em dois dos três grupos de dados, as curvas precision-recall mostraram que, considerando-se diversas possibilidades de valores limites, os três modelos apresentaram excelente desempenho na detecção de outliers. Em trabalhos futuros, outros contextos podem ser considerados como, por exemplo, séries temporais mais longas e com diferentes sazonalidades. Pode ser interessante, também, avaliar explicitamente o impacto da tendência e da volatilidade sobre o desempenho de predições e detecção de outliers em séries temporais. Além disso, novas configurações de outliers podem ser consideradas, alterando-se a quantidade de outliers inseridos nos conjuntos de testes, as posições em que foram inseridos e as direções a eles atribuídas. Dessa forma, novos cenários poderão ser criados e utilizados para aprofundar as avaliações e aprimorar o método de detecção de outliers proposto neste trabalho.

REFERÊNCIAS

ABBASIMEHR, H.; SHABANI, M.; YOUSEFI, M. An optimized model using lstm network for demand forecasting. **Computers & Industrial Engineering**, Elsevier, p. 106435, 2020. Citado nas páginas 46, 70, 71 e 72.

AGGARWAL, C. C. An introduction to outlier analysis. In: **Outlier analysis**. [S.l.]: Springer, 2017. p. 1–34. Citado nas páginas 25, 26, 48, 49 e 53.

ALTHELAYA, K. A.; EL-ALFY, E.-S. M.; MOHAMMED, S. Evaluation of bidirectional lstm for short-and long-term stock market prediction. In: IEEE. **2018 9th international conference on information and communication systems (ICICS)**. [S.l.], 2018. p. 151–156. Citado na página 46.

BALA, R.; SINGH, R. P. *et al.* Financial and non-stationary time series forecasting using lstm recurrent neural network for short and long horizon. In: IEEE. **2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)**. [S.l.], 2019. p. 1–7. Citado nas páginas 26, 42 e 45.

BALDI, P.; BRUNAK, S.; CHAUVIN, Y.; ANDERSEN, C. A.; NIELSEN, H. Assessing the accuracy of prediction algorithms for classification: an overview. **Bioinformatics**, Oxford University Press, v. 16, n. 5, p. 412–424, 2000. Citado na página 78.

BEKKAR, M.; DJEMAA, H. K.; ALITOUCHE, T. A. Evaluation measures for models assessment over imbalanced data sets. **J Inf Eng Appl**, v. 3, n. 10, 2013. Citado na página 78.

BIANCHI, F. M.; MAIORINO, E.; KAMPFFMEYER, M. C.; RIZZI, A.; JENSSEN, R. An overview and comparative analysis of recurrent neural networks for short term load forecasting. **arXiv preprint arXiv:1705.04378**, 2017. Citado na página 46.

BLÁZQUEZ-GARCÍA, A.; CONDE, A.; MORI, U.; LOZANO, J. A. A review on outlier/anomaly detection in time series data. **arXiv preprint arXiv:2002.04236**, 2020. Citado nas páginas 48, 49, 51, 52 e 53.

BONTEMPS, L.; MCDERMOTT, J.; LE-KHAC, N.-A. *et al.* Collective anomaly detection based on long short-term memory recurrent neural networks. In: SPRINGER. **International Conference on Future Data and Security Engineering**. [S.l.], 2016. p. 141–152. Citado na página 26.

BOX, G.; JENKINS, G. **Time Series Analysis Forecasting and Control/'Holden Day, San Francisco, California**. 1970. Citado na página 27.

BOX, G. E.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. **Time series analysis: forecasting and control**. [S.l.]: John Wiley & Sons, 2015. Citado nas páginas 34 e 36.

BOYD, K.; ENG, K. H.; PAGE, C. D. Area under the precision-recall curve: point estimates and confidence intervals. In: SPRINGER. **Joint European conference on machine learning and knowledge discovery in databases**. [S.l.], 2013. p. 451–466. Citado na página 78.

BRANCO, P.; TORGO, L.; RIBEIRO, R. A survey of predictive modelling under imbalanced distributions. **arXiv preprint arXiv:1505.01658**, 2015. Citado na página 77.

BROCKWELL, P. J.; DAVIS, R. A. **Introduction to time series and forecasting**. [S.l.]: springer, 2016. Citado nas páginas 29 e 34.

BRZEZINSKI, D.; STEFANOWSKI, J.; SUSMAGA, R.; SZCZCH, I. Visual-based analysis of classification measures and their properties for class imbalanced problems. **Information Sciences**, Elsevier, v. 462, p. 242–261, 2018. Citado nas páginas 77 e 78.

CAO, J.; LI, Z.; LI, J. Financial time series forecasting model based on ceemdan and lstm. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 519, p. 127–139, 2019. Citado na página 45.

CARREÑO, A.; INZA, I.; LOZANO, J. A. Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. **Artificial Intelligence Review**, Springer, p. 1–20, 2019. Citado na página 48.

CHALAPATHY, R.; CHAWLA, S. Deep learning for anomaly detection: A survey. **arXiv preprint arXiv:1901.03407**, 2019. Citado nas páginas 25, 26 e 96.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 41, n. 3, p. 1–58, 2009. Citado nas páginas 25 e 48.

CHATFIELD, C.; XING, H. **The analysis of time series: an introduction with R**. [S.l.]: CRC press, 2019. Citado nas páginas 26, 30, 31, 33, 34 e 35.

CHO, K.; MERRIËNBOER, B. V.; BAHDANAU, D.; BENGIO, Y. On the properties of neural machine translation: Encoder-decoder approaches. **arXiv preprint arXiv:1409.1259**, 2014. Citado na página 44.

CHOI, H. K. Stock price correlation coefficient prediction with arima-lstm hybrid model. **arXiv preprint arXiv:1808.01560**, 2018. Citado na página 46.

CHUNG, J.; GULCEHRE, C.; CHO, K.; BENGIO, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. **arXiv preprint arXiv:1412.3555**, 2014. Citado nas páginas 27 e 44.

DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: **Proceedings of the 23rd international conference on Machine learning**. [S.l.: s.n.], 2006. p. 233–240. Citado na página 78.

DAVIS, N.; RAINA, G.; JAGANNATHAN, K. A framework for end-to-end deep learning-based anomaly detection in transportation networks. **Transportation Research Interdisciplinary Perspectives**, Elsevier, v. 5, p. 100112, 2020. Citado nas páginas 25, 26, 48 e 96.

FERNÁNDEZ, A.; GARCÍA, S.; GALAR, M.; PRATI, R. C.; KRAWCZYK, B.; HERRERA, F. **Learning from imbalanced data sets**. [S.l.]: Springer, 2018. v. 11. Citado nas páginas 77 e 78.

FISCHER, T.; KRAUSS, C. Deep learning with long short-term memory networks for financial market predictions. **European Journal of Operational Research**, Elsevier, v. 270, n. 2, p. 654–669, 2018. Citado na página 46.

GAO, S.; HUANG, Y.; ZHANG, S.; HAN, J.; WANG, G.; ZHANG, M.; LIN, Q. Short-term runoff prediction with gru and lstm networks without requiring time step optimization during sample generation. **Journal of Hydrology**, Elsevier, p. 125188, 2020. Citado nas páginas 44, 45, 71 e 95.

GOADRICH, M.; OLIPHANT, L.; SHAVLIK, J. Gleaner: Creating ensembles of first-order clauses to improve recall-precision curves. **Machine Learning**, Springer, v. 64, n. 1-3, p. 231–261, 2006. Citado na página 78.

GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. Lstm: A search space odyssey. **IEEE transactions on neural networks and learning systems**, IEEE, v. 28, n. 10, p. 2222–2232, 2016. Citado na página 42.

GUPTA, M.; GAO, J.; AGGARWAL, C.; HAN, J. Outlier detection for temporal data. **Synthesis Lectures on Data Mining and Knowledge Discovery**, Morgan & Claypool Publishers, v. 5, n. 1, p. 1–129, 2014. Citado nas páginas 48, 49, 50, 51 e 52.

HANKE, J. E. **Guide for: Business Forecasting**. [S.l.]: Cram101, 2014. Citado nas páginas 34, 36 e 37.

HAWKINS, D. **Identification of Outliers. Springer**. [S.l.]: Netherlands, 1980. Citado nas páginas 25 e 48.

HE, H.; MA, Y. Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons, 2013. Citado nas páginas 77 e 78.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado nas páginas 26 e 42.

HYNDMAN, R.; KOEHLER, A. B.; ORD, J. K.; SNYDER, R. D. **Forecasting with exponential smoothing: the state space approach**. [S.l.]: Springer Science & Business Media, 2008. Citado na página 30.

HYNDMAN, R. J.; ATHANASOPOULOS, G. **Forecasting: principles and practice**. [S.l.]: OTexts, 2018. Citado nas páginas 31, 34, 35, 36, 37, 38 e 95.

INOUE, J.; YAMAGATA, Y.; CHEN, Y.; POSKITT, C. M.; SUN, J. Anomaly detection for a water treatment system using unsupervised machine learning. In: IEEE. **2017 IEEE International Conference on Data Mining Workshops (ICDMW)**. [S.l.], 2017. p. 1058–1065. Citado nas páginas 48 e 53.

KARPATHY, A.; JOHNSON, J.; FEI-FEI, L. Visualizing and understanding recurrent networks. **arXiv preprint arXiv:1506.02078**, 2015. Citado na página 41.

LAPTEV, N.; YOSINSKI, J.; LI, L. E.; SMYL, S. Time-series extreme event forecasting with neural networks at uber. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2017. v. 34, p. 1–5. Citado na página 46.

LAW, R.; LI, G.; FONG, D. K. C.; HAN, X. Tourism demand forecasting: A deep learning approach. **Annals of Tourism Research**, Elsevier, v. 75, p. 410–423, 2019. Citado na página 46.

LIPTON, Z. C.; BERKOWITZ, J.; ELKAN, C. A critical review of recurrent neural networks for sequence learning. **arXiv preprint arXiv:1506.00019**, 2015. Citado nas páginas 40, 41 e 42.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. Statistical and machine learning forecasting methods: Concerns and ways forward. **PloS one**, Public Library of Science San Francisco, CA USA, v. 13, n. 3, p. e0194889, 2018. Citado na página 47.

MALHOTRA, P.; VIG, L.; SHROFF, G.; AGARWAL, P. Long short term memory networks for anomaly detection in time series. In: **Proceedings**. [S.l.: s.n.], 2015. v. 89, p. 89–94. Citado na página 26.

MEHROTRA, K. G.; MOHAN, C. K.; HUANG, H. **Anomaly detection principles and algorithms**. [S.l.]: Springer, 2017. Citado nas páginas 25, 26, 48, 51 e 52.

MELLO, R. F.; PONTI, M. A. **Machine learning: a practical approach on the statistical learning theory**. [S.l.]: Springer, 2018. Citado na página 40.

METCALFE, A. V.; COWPERTWAIT, P. S. **Introductory time series with R**. [S.l.]: Springer, 2009. Citado na página 31.

MILLS, T. C. **Applied Time Series Analysis: A Practical Guide to Modeling and Forecasting**. [S.l.]: Academic Press, 2019. Citado nas páginas 30, 33 e 37.

MURUGAN, P. Learning the sequential temporal information with recurrent neural networks. **arXiv preprint arXiv:1807.02857**, 2018. Citado nas páginas 40 e 42.

ORDÓÑEZ, F. J.; ROGGEN, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 16, n. 1, p. 115, 2016. Citado na página 46.

OZENNE, B.; SUBTIL, F.; MAUCORT-BOULCH, D. The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. **Journal of clinical epidemiology**, Elsevier, v. 68, n. 8, p. 855–859, 2015. Citado na página 78.

PARMEZAN, A. R. S.; SOUZA, V. M.; BATISTA, G. E. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. **Information Sciences**, Elsevier, v. 484, p. 302–337, 2019. Citado nas páginas 42, 47 e 95.

PONTI, M. A.; RIBEIRO, L. S. F.; NAZARE, T. S.; BUI, T.; COLLOMOSSE, J. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In: IEEE. **2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)**. [S.l.], 2017. p. 17–41. Citado na página 40.

PONTI, M. A.; SANTOS, F. P. d.; RIBEIRO, L. S. F.; CAVALLARI, G. B. Training deep networks from zero to hero: avoiding pitfalls and going beyond. **arXiv preprint arXiv:2109.02752**, 2021. Citado na página 40.

REIMERS, N.; GUREVYCH, I. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. **arXiv preprint arXiv:1707.06799**, 2017. Citado na página 70.

RIBEIRO, L. S. F.; BUI, T.; COLLOMOSSE, J.; PONTI, M. Sketchformer: Transformer-based representation for sketched structure. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 14153–14162. Citado na página 40.

SAGHEER, A.; KOTB, M. Time series forecasting of petroleum production using deep lstm recurrent networks. **Neurocomputing**, Elsevier, v. 323, p. 203–213, 2019. Citado nas páginas 42 e 46.

SAITO, T.; REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. **PloS one**, Public Library of Science, v. 10, n. 3, p. e0118432, 2015. Citado nas páginas 78 e 79.

SEZER, O. B.; GUDELEK, M. U.; OZBAYOGLU, A. M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. **Applied Soft Computing**, Elsevier, v. 90, p. 106181, 2020. Citado nas páginas 42 e 46.

SHUMWAY, R. H.; STOFFER, D. S. **Time series analysis and its applications: with R examples**. [S.l.]: Springer, 2017. Citado nas páginas 29, 30, 33, 35 e 36.

SIAMI-NAMINI, S.; TAVAKOLI, N.; NAMIN, A. S. A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. **arXiv preprint arXiv:1911.09512**, 2019. Citado nas páginas 40, 41, 42 e 47.

SINGH, A. **Anomaly detection for temporal data using long short-term memory (lstm)**. 2017. Citado nas páginas 25 e 26.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information processing & management**, Elsevier, v. 45, n. 4, p. 427–437, 2009. Citado nas páginas 76 e 77.

SONG, X.; LIU, Y.; XUE, L.; WANG, J.; ZHANG, J.; WANG, J.; JIANG, L.; CHENG, Z. Time-series well performance prediction based on long short-term memory (lstm) neural network model. **Journal of Petroleum Science and Engineering**, Elsevier, v. 186, p. 106682, 2020. Citado na página 47.

TEALAB, A. Time series forecasting using artificial neural networks methodologies: A systematic review. **Future Computing and Informatics Journal**, Elsevier, v. 3, n. 2, p. 334–340, 2018. Citado na página 46.

TRAN, K. P.; NGUYEN, H. D.; THOMASSEY, S. Anomaly detection using long short term memory networks and its applications in supply chain management. **IFAC-PapersOnLine**, Elsevier, v. 52, n. 13, p. 2408–2412, 2019. Citado nas páginas 25, 26 e 48.

YAO, D.; SHU, X.; CHENG, L.; STOLFO, S. J. Anomaly detection as a service: challenges, advances, and opportunities. **Synthesis Lectures on Information Security, Privacy, and Trust**, Morgan & Claypool Publishers, v. 9, n. 3, p. 1–173, 2017. Citado na página 25.

YOSHIHARA, A.; SEKI, K.; UEHARA, K. Leveraging temporal properties of news events for stock market prediction. **Artif. Intell. Research**, v. 5, n. 1, p. 103–110, 2016. Citado na página 46.

YUNPENG, L.; DI, H.; JUNPENG, B.; YONG, Q. Multi-step ahead time series forecasting for different data patterns based on lstm recurrent neural network. In: IEEE. **2017 14th Web Information Systems and Applications Conference (WISA)**. [S.l.], 2017. p. 305–310. Citado na página 46.

ZHANG, X.; ZOU, Y.; LI, S.; XU, S. A weighted auto regressive lstm based approach for chemical processes modeling. **Neurocomputing**, Elsevier, v. 367, p. 64–74, 2019. Citado na página 45.

ZHU, L.; LAPTEV, N. Deep and confident prediction for time series at uber. In: IEEE. **2017 IEEE International Conference on Data Mining Workshops (ICDMW)**. [S.l.], 2017. p. 103–110. Citado na página 46.


ICMC
USP
SÃO CARLOS