

UNIVERSIDADE FEDERAL DO PARANÁ

GLENDIA PROENÇA TRAIN

ANÁLISE DAS ETAPAS DE PRÉ-PROCESSAMENTO E MODELOS DE PREVISÃO DE
VELOCIDADE DE VENTO UTILIZANDO DADOS EÓLICOS E HIDROLÓGICOS

CURITIBA PR

2021

GLEND A PROENÇA TRAIN

ANÁLISE DAS ETAPAS DE PRÉ-PROCESSAMENTO E MODELOS DE PREVISÃO DE
VELOCIDADE DE VENTO UTILIZANDO DADOS EÓLICOS E HIDROLÓGICOS

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de Concentração: *Ciência da Computação*.

Orientador: Fabiano Silva.

CURITIBA PR

2021

“Se é uma boa ideia, vá em frente e faça”

(Grace Hopper)

AGRADECIMENTOS

Aos meus pais, Eliane e Moacir, e à minha irmã, Aline, por todo o apoio, incentivo e confiança para a realização deste trabalho e durante todos os anos da minha graduação.

Ao meu amigo Rodrigo, por confiar que eu conseguiria completar essa última etapa, sempre me motivando e tornando os dias mais leves e divertidos.

Aos demais membros da minha família, por sempre torcerem por mim e celebrarem as minhas conquistas.

Ao meu orientador, Fabiano, por me ajudar a pensar e desenvolver novas ideias e ser compreensivo nos períodos difíceis.

À UFPR, por proporcionar diversas oportunidades que me permitiram evoluir e alcançar os meus objetivos.

RESUMO

Dentre as várias formas de tecnologias renováveis, a energia eólica vem se destacando nos últimos anos, chegando a representar 5% da eletricidade mundial em 2019. Com uma popularidade ainda maior, a energia hidrelétrica ocupa o primeiro lugar entre as energias geradas a partir de tecnologias renováveis, alcançando a marca de 16% da eletricidade global e 64,58% da eletricidade brasileira em 2019. Essas formas de geração de energia apresentam uma grande relevância para reduzir o uso de combustíveis fósseis, que causam danos ao ambiente e à saúde humana. O problema encontrado por muitas pesquisas que visam à previsão energética, com foco em Aprendizado Supervisionado, está na quantidade de dados eólicos disponíveis, pois geralmente são simulados ou escassos, enquanto que os dados hidrológicos existem em grandes quantidades, até mesmo sendo disponibilizados em bases de dados públicas. Com a intenção de estudar esse cenário de forma aprofundada, foram definidos 6 objetivos a serem investigados neste trabalho, que são: construir uma metodologia para a previsão de velocidade de vento utilizando somente os dados eólicos dos meses anteriores, chamada de modo Univariado; desenvolver uma metodologia para prever os dados eólicos utilizando séries temporais hidrológicas diferentes, chamada de modo Multivariado; avaliar o impacto de diferentes técnicas de pré-processamento de séries temporais, além de implementar uma nova forma de extrair características dos dados de vários meses anteriores, chamada de Rede de Representação; comparar 18 modelos diferentes de previsão no modo Univariado e 12 modelos no Multivariado, abrangendo 3 áreas diferentes, com modelos derivados da Teoria de Séries Temporais, Aprendizado de Máquina e Aprendizado Profundo; otimizar os hiperparâmetros de cada um dos modelos e otimizar os valores de pré-processamento para cada um dos modelos otimizados. Uma análise de 3 etapas foi realizada nos resultados, que consistiu em: apresentar um estudo experimental para todos os modelos, além das curvas de predição dos resultados mais relevantes; investigar o impacto de cada um dos pré-processamentos nas previsões e, na última etapa, selecionar as duas melhores configurações de hiperparâmetros e parâmetros de pré-processamento, retrainar os modelos e avaliar os resultados em outras 6 bases de dados com horizontes de previsão de 12 e 60 meses. Os resultados evidenciam que, com exceção de 3 modelos do modo Univariado, todos os modelos conseguiram atingir métricas melhores do que o modelo definido como *baseline*. As formas de pré-processamento que mais se destacaram foram a aplicação de escala e de *Look Back* com mais de 3 meses, o uso equilibrado do *Walk Forward* e a Rede de Representação, que só apresentou resultados significativos nos modelos derivados da Teoria de Séries Temporais. Por fim, os melhores resultados do modo Univariado foram alcançados pelo modelo GRU e pela LSTM no modo Multivariado.

Palavras-chave: Modelos de Regressão. Previsão de dados eólicos. Pré-processamento em série temporal. Aprendizado de Máquina. Aprendizado Profundo. Séries Temporais.

ABSTRACT

Among the various forms of renewable technologies, wind energy has stood out in recent years, reaching 5% of the world's electricity in 2019. With an even greater popularity, hydroelectricity occupies the first place among the energies generated from renewable technologies, reaching the mark of 16% of global electricity and 64.58% of Brazilian electricity in 2019. These forms of energy generation are of great relevance to reduce the use of fossil fuels, which cause damage to the environment and human health. The problem found by many researches aimed at energy forecasting, with a focus on Supervised Learning, is in the amount of wind data available, as they are usually simulated or scarce, while hydrological data exist in large quantities, even being made available in public databases. With the intention of studying this scenario in depth, 6 objectives were defined to be investigated in this work, which are: building a methodology for the wind speed forecasting using only the wind data from previous months, called Univariate mode; developing a methodology to forecast wind data using different hydrological time series, called Multivariate mode; evaluating the impact of different time series pre-processing techniques, in addition to implementing a new way of extracting data characteristics from several previous months, called Representation Network; comparing 18 different forecasting models in Univariate mode and 12 models in Multivariate, covering 3 different areas, with models derived from Time Series Theory, Machine Learning and Deep Learning; optimizing the hyperparameters of each of the models and optimizing the pre-processing values for each of the optimized models. A 3-step analysis was performed on the results, which consisted of: presenting an experimental study for all models, in addition to the prediction curves of the most relevant results; investigating the impact of each of the pre-processing on the predictions and, in the last step, selecting the two best configurations of hyperparameters and pre-processing parameters, retraining the models and evaluating the results in other 6 databases with forecasting horizons of 12 and 60 months. The results show that, with the exception of 3 models in the Univariate mode, all models were able to achieve better metrics than the model defined as baseline. The pre-processing forms that stood out the most were the application of scale and Look Back with more than 3 months, the balanced use of the Walk Forward and the Representation Network, which only presented significant results in the models derived from the Time Series Theory. Finally, the best results in the Univariate mode were achieved by the GRU model and the LSTM in the Multivariate mode.

Keywords: Regression Models. Wind Data Forecasting. Time Series Pre-processing. Machine Learning. Deep Learning. Time Series.

LISTA DE FIGURAS

2.1	Gráfico de exemplo da relação das horas estudadas pelos alunos com as notas obtidas no teste.	29
2.2	Exemplo do armazenamento e representação gráfica da série temporal de vendas mensais de shampoo.	30
2.3	Série temporal do número de ataques cibernéticos em 8 anos.	31
2.4	Exemplo de diagramas apresentando a influência direta mês a mês (a) e mostrando a relação indireta entre meses não consecutivos (b).	33
2.5	Exemplo dos gráficos ACF (esquerda), dos quatro primeiros <i>lags</i> , sendo os dois primeiros mais significativos, e PACF (direita), sendo somente o primeiro <i>lag</i> significativo.	34
2.6	Gráfico da tendência do número de passageiros (em milhares) no transporte aéreo entre os anos de 1949 e 1961.	34
2.7	Gráfico do consumo de energia (em kWh) em 10 dias, iniciando em 13 de setembro de 2017.	35
2.8	Gráficos da série temporal e suas decomposições.	36
2.9	Condições de estacionariedade de uma série temporal. Os gráficos verdes representam séries estacionárias e os vermelhos são as séries não estacionárias, que falharam em alguma das três condições (média, variância e autocorrelação)	37
2.10	Demonstração do resultado do teste ADF em duas séries temporais diferentes, uma estacionária (esquerda) e outra não estacionária (direita). No topo da imagem há a representação gráfica das séries e abaixo os valores obtidos pelo teste. . .	39
2.11	Esquema da transformação de uma série não estacionária (em roxo) em uma série estacionária (em azul) por meio da diferenciação. A inversão da diferenciação também é apresentada na parte inferior direita, em que os valores são somados recursivamente até resultar na série temporal original (em cinza).	40
2.12	Exemplo de adição de <i>Look Back</i> à série temporal com janela de 1 e 3 unidades	42
2.13	Exemplo do processo de <i>Random Walk</i>	43
2.14	Fluxo de execução do método <i>Walk Forward</i>	45

2.15	Exemplo de uso do modelo PMF para predição de série temporal.	47
2.16	Gráfico de exemplo da distribuição de leite em cada mês (esquerda) e gráfico do PACF construído com os dados do problema (direita).	48
2.17	Tabela com os dados do exemplo (tempo, previsão, erro e valores esperados) e gráfico de projeção da quantidade esperada de <i>cupcakes</i> ao longo do tempo (direita).	49
2.18	Gráficos ACF e PACF do exemplo da série temporal referente à produção de lâmpadas.	51
2.19	Gráfico da venda de âncoras gerado com os dados brutos (esquerda) e gráfico da venda de âncoras diferenciado uma vez (direita).	53
2.20	Gráfico do exemplo de venda de <i>donuts</i> de três em três meses de 2014 a 2019.	55
2.21	Gráfico do resíduo ($\epsilon_t = \text{valor esperado} - \text{valor real}$ de cada semana) do exemplo da venda de tickets de cinema.	57
2.22	Gráfico de dados com “ <i>bursty</i> ”, em que os dados mantêm uma volatilidade constante, variam bastante e voltam a ter a volatilidade constante (esquerda) e gráfico em que, quando a volatilidade muda bastante, ela se mantém por um tempo antes de voltar a ser constante (direita).	59
2.23	Gráfico do exemplo de reta encontrada pela Regressão Linear (esquerda) e fórmulas utilizadas para construir o modelo (direita).	61
2.24	Comparação entre um <i>underfitting</i> obtido por um modelo Regressão Linear Simples (esquerda) com um bom ajuste obtido pela Regressão Polinomial (direita).	61
2.25	Representação da Árvore de Regressão da predição de horas de golfe que serão jogadas no dia seguinte (esquerda) tabela com os dados disponíveis para a construção da árvore (direita).	63
2.26	Sequência de passos para o cálculo do SDR.	64
2.27	Comparação entre os métodos de combinação de modelos, inicialmente o caso de modelos individuais (<i>Single</i>), combinação paralela (<i>Bagging</i>) e combinação sequencial (<i>Boosting</i>).	66

2.28	Exemplo do funcionamento de previsão feita pelo modelo de Florestas Aleatórias	68
2.29	Exemplo dos ajustes sequenciais feitos pelo modelo de <i>Gradient Boosting</i> utilizando Árvores de Regressão.	70
2.30	Representação gráfica do <i>Perceptron</i>	71
2.31	Visualização da entrada escala e traduzida.	73
2.32	Arquitetura do modelo MLP.	74
2.33	Exemplo do <i>loop</i> de uma rede recorrente (esquerda) e a forma desenrolada do <i>loop</i> (direita).	75
2.34	Relação de dependência de informações mais recentes (esquerda) e de informações mais distantes (direita) na arquitetura da rede recorrente.	76
2.35	Comparação entre a arquitetura da RNN (esquerda) com a LSTM (direita).	76
2.36	Comparação estrutural entre LSTM (esquerda) e GRU (direita).	78
2.37	Exemplo de séries temporais com baixa Correlação e alta Cointegração.	81
4.1	Localização das 22 usinas hidrelétricas selecionadas das 171 iniciais.	89
4.2	Localização dos sítios presentes nos dados de velocidade de vento.	90
4.3	Diagrama simplificado das etapas da metodologia proposta.	92
4.4	Gráfico dos dados de vazão e velocidade de vento mensais da usina hidrelétrica de Serra Mesa e do sítio de Afrânio, respectivamente.	93
4.5	Exemplo do fluxo das 4 primeiras etapas da metodologia adotada.	94
4.6	Exemplo de aplicação do <i>Look Back</i> de 3 meses no modo Multivariado.	97
4.7	Comparação entre os valores dos 22 aproveitamentos hidrelétricos dos dados hidrológicos com a série temporal de velocidade de vento em Afrânio, no período de janeiro de 1994 a dezembro de 1995.	98
4.8	Arquiteturas implementadas da Rede de Representação no modo Univariado.	100
4.9	Arquitetura da Rede de Representação no modo Multivariado.	101

6.1	Valores esperados e obtidos com o modelo baseline PMF para a série temporal de Afrânio.	114
6.2	Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos modelos ARMA, SARIMA, Árvore de Regressão, <i>Gradient Boosting</i> , LSTM e GRU, no modo Univariado, com horizonte de previsão de 12 meses.	117
6.3	<i>Boxplots</i> anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e <i>boxplot</i> mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Afrânio.	118
6.4	Gráficos das curvas obtidas (em azul) e esperadas (em alaranjado) dos modelos ARCH e GARCH sem o uso da estacionariedade (esquerda) e com o uso da estacionariedade (direita).	120
6.5	Gráficos das curvas esperadas e obtidas pelas três variações do modelo <i>Exponential Smoothing</i> , nos dados de Afrânio, no modo Univariado e com horizonte de previsão de 12 meses.	121
6.6	Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos modelos VAR, VECM, Árvore de Regressão, Florestas Aleatórias, LSTM e GRU, no modo Multivariado, com horizonte de previsão de 12 meses.	124
6.7	Exemplo do impacto da aplicação de <i>Look Back</i> utilizando a Janela Rolante de tamanho 3 com agregação por média.	125
6.8	Aplicação da estacionariedade no modelo AR (modo Univariado).	127
6.9	Aplicação da estacionariedade no modelo VAR (modo Multivariado).	128
6.10	Aplicação da escala no modelo <i>Bagging</i> (modo Univariado).	129
6.11	Aplicação da escala no modelo GRU (modo Multivariado).	130
6.12	Aplicação de diferentes valores de <i>Look Back</i> no modelo de Florestas Aleatórias (modo Univariado).	131
6.13	Aplicação de diferentes valores de <i>Look Back</i> no modelo <i>Bagging</i> (modo Multivariado).	132
6.14	Aplicação do <i>Walk Forward</i> no modelo AR (modo Univariado).	133
6.15	Aplicação do <i>Walk Forward</i> no modelo VECM (modo Multivariado).	134
6.16	Aplicação da Rede de Representação no modelo AR (modo Univariado).	135

6.17	Aplicação da Rede de Representação no modelo VAR (modo Multivariado).	136
6.18	Aplicação da função de ativação na Rede de Representação com o modelo ARIMA (modo Univariado).	137
6.19	Aplicação da função de ativação na Rede de Representação com o modelo VAR (modo Multivariado).	137
6.20	Aplicação de diferentes valores de <i>Look Back</i> na Rede de Representação com o modelo MA (modo Univariado).	139
6.21	Aplicação de diferentes valores de <i>Look Back</i> na Rede de Representação com o modelo VAR (modo Multivariado).	140
6.22	Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos melhores modelos, entre <i>Gradient Boosting</i> e GRU, no modo Univariado, com horizonte de previsão de 12 meses, para os dados de Afrânio, Estreito, <i>Offshore</i> CE e SC.	144
6.23	Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos melhores modelos, entre <i>Gradient Boosting</i> e GRU, no modo Univariado, com horizonte de previsão de 12 meses, para os dados de Palmas, Quilombo e São João.	145
6.24	Gráficos das curvas obtidas e esperadas, no modo Univariado, dos melhores modelos, entre <i>Gradient Boosting</i> e GRU, em todas as bases de dados, com horizonte de previsão de 60 meses, dos dados de Afrânio, Estreito, <i>Offshore</i> CE, <i>Offshore</i> SC, Palmas e Quilombo.	146
6.25	Gráficos das curvas obtidas e esperadas, no modo Univariado, dos melhores modelos, entre <i>Gradient Boosting</i> e GRU, em todas as bases de dados, com horizonte de previsão de 60 meses, dos dados de São João.	147
6.26	Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos melhores modelos, entre LSTM e GRU, no modo Multivariado, com horizonte de previsão de 12 meses, nos dados de Afrânio, Estreito, <i>Offshore</i> CE, <i>Offshore</i> SC, Palmas e Quilombo.	149
6.27	Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos melhores modelos, entre LSTM e GRU, no modo Multivariado, com horizonte de previsão de 12 meses, dos dados de São João.	150
6.28	Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos melhores modelos, entre LSTM e GRU, em todas as bases de dados, com horizonte de	

	previsão de 60 meses, dos dados de Afrânio, Estreito, <i>Offshore</i> CE, <i>Offshore</i> SC, Palmas e Quilombo.	151
6.29	Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos melhores modelos, entre LSTM e GRU, em todas as bases de dados, com horizonte de previsão de 60 meses, dos dados de São João.	152
B.1	Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos AR, MA, ARMA, ARIMA e SARIMA.	174
B.2	Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos ARCH sem transformação de estacionariedade, ARCH estacionário, GARCH sem transformação de estacionariedade e GARCH estacionário.	175
B.3	Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos <i>Simple Exponential Smoothing</i> , <i>Double Exponential Smoothing</i> e <i>Triple Exponential Smoothing</i>	176
B.4	Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos Regressão Linear, Regressão Polinomial, Árvore de Regressão, <i>Bagging</i> , Florestas Aleatórias e <i>Gradient Boosting</i>	177
B.5	Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos <i>Perceptron</i> , MLP, LSTM e GRU.	178
B.6	Gráficos das curvas obtidas e esperadas, no modo Multivariado, pelos modelos VAR, VECM, Regressão Linear, Árvore de Regressão, <i>Bagging</i> e Florestas Aleatórias.	179
B.7	Gráficos das curvas obtidas e esperadas, no modo Multivariado, pelos modelos <i>Gradient Boosting</i> , <i>Perceptron</i> , MLP, LSTM e GRU.	180
C.1	Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos <i>Gradient Boosting</i> e GRU nos dados de Afrânio, Estreito e <i>Offshore</i> CE, com horizonte de previsão de 12 meses.	181
C.2	Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos <i>Gradient Boosting</i> e GRU nos dados de <i>Offshore</i> SC, Palmas e Quilombo, com horizonte de previsão de 12 meses.	182

C.3	Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos <i>Gradient Boosting</i> e GRU nos dados de São João, com horizonte de previsão de 12 meses.	183
C.4	Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos <i>Gradient Boosting</i> e GRU nos dados de Afrânio, Estreito e <i>Offshore</i> CE, com horizonte de previsão de 60 meses.	185
C.5	Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos <i>Gradient Boosting</i> e GRU nos dados de <i>Offshore</i> SC, Palmas e Quilombo, com horizonte de previsão de 60 meses.	186
C.6	Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos <i>Gradient Boosting</i> e GRU nos dados de São João, com horizonte de previsão de 60 meses.	187
C.7	Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de Afrânio, Estreito e <i>Offshore</i> CE, com horizonte de previsão de 12 meses.	188
C.8	Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de <i>Offshore</i> SC e Palmas, com horizonte de previsão de 12 meses.	189
C.9	Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de Quilombo e São João, com horizonte de previsão de 12 meses.	190
C.10	Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de Afrânio, Estreito e <i>Offshore</i> CE, com horizonte de previsão de 60 meses.	192
C.11	Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de <i>Offshore</i> SC, Palmas, Quilombo e São João, com horizonte de previsão de 60 meses.	193
D.1	<i>Boxplots</i> anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e <i>boxplot</i> mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Afrânio.	195

D.2	<i>Boxplots</i> anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e <i>boxplot</i> mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Estreito.	196
D.3	<i>Boxplots</i> anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e <i>boxplot</i> mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de <i>Offshore</i> CE.	197
D.4	<i>Boxplots</i> anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e <i>boxplot</i> mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de <i>Offshore</i> SC.	198
D.5	<i>Boxplots</i> anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e <i>boxplot</i> mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Palmas.	199
D.6	<i>Boxplots</i> anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e <i>boxplot</i> mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Quilombo.	200
D.7	<i>Boxplots</i> anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e <i>boxplot</i> mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de São João.	201

LISTA DE TABELAS

4.1	Tabela com uma amostra dos dados de vazões nas 4 primeiras colunas e os dados de velocidade de vento (simulados devido à privacidade dos dados) na última coluna.	88
5.1	Modelos avaliados no problema de previsão de velocidade de vento.	103
5.2	Referências detalhadas sobre os hiperparâmetros de cada modelo implementado.	105
5.3	Intervalos de hiperparâmetros utilizados na otimização dos modelos derivados da Teoria de Séries Temporais do modo Univariado, sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.	106
5.4	Intervalos de hiperparâmetros utilizados na otimização dos modelos derivados da Teoria de Séries Temporais do modo Multivariado, sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.	107
5.5	Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado de Máquina (Regressão Linear, Regressão Polinomial e Árvore de Regressão), sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento	107
5.6	Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado de Máquina (<i>Bagging</i> e Florestas Aleatórias), sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.	108
5.7	Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado de Máquina (<i>Gradient Boosting</i>), sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.	108

5.8	Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado Profundo (<i>Perceptron</i> e MLP), sendo a primeira coluna o nome do hiperparâmetro e a segunda os valores testados.	109
5.9	Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado Profundo (LSTM e GRU), sendo a primeira coluna o nome do hiperparâmetro e a segunda os valores testados	110
5.10	Intervalos de hiperparâmetros do uso das técnicas de pré-processamento no caso dos modelos derivados da Teoria de Séries Temporais do modo Univariado. .	111
5.11	Intervalos de hiperparâmetros do uso das técnicas de pré-processamento no caso dos modelos de Aprendizado de Máquina e Aprendizado Profundo Univariados e todos os Multivariados. Nesses hiperparâmetros, a Rede de Representação só é utilizada quando não há o uso de <i>Look Back</i> maior que 1, pois a rede considera seu próprio <i>Look Back</i>	111
6.1	Parâmetros de pré-processamento e as métricas (MSE, RMSE, MAPE) obtidas por cada modelo do modo Univariado.	115
6.2	Parâmetros de pré-processamento e as métricas (MSE, RMSE, MAPE) obtidas por cada modelo do modo Multivariado.	122
6.3	Resultados dos melhores modelos, com o aumento de épocas, para todas as bases de dados no modo Univariado, para o horizonte de predição de 12 meses. . .	142
6.4	Valores médios de MSE, RMSE e MAPE dos modelos Univariados, considerando os resultados em todos os sete conjuntos de dados, com horizonte de 12 meses.	143
6.5	Resultados dos melhores modelos, com o aumento de épocas, para todas as bases de dados no modo Multivariado, para o horizonte de predição de 12 meses. .	148
6.6	Valores médios de MSE, RMSE e MAPE dos modelos Multivariados, considerando os resultados em todos os sete conjuntos de dados, com horizonte de 12 meses.	148
A.1	Valores dos melhores hiperparâmetros encontrados para todos os modelos. .	168

C.1	Melhores métricas e parâmetros de pré-processamento dos melhores modelos (<i>Gradient Boosting</i> e GRU) Univariados, com horizonte de previsão de 12 meses.	184
C.2	Melhores métricas e parâmetros de pré-processamento dos modelos <i>Gradient Boosting</i> e GRU univariados, com horizonte de previsão de 60 meses.	187
C.3	Melhores métricas e parâmetros de pré-processamento dos melhores modelos (LSTM e GRU) Multivariados, com horizonte de previsão de 12 meses. . .	191
C.4	Melhores métricas e parâmetros de pré-processamento dos melhores modelos (LSTM e GRU) Multivariados, com horizonte de previsão de 60 meses. . .	194

LISTA DE ACRÔNIMOS

ABEEólica *Associação Brasileira de Energia Eólica.*

ACF *AutoCorrelation Function.*

ADF *Augmented Dickey–Fuller.*

AGL *Above Ground Level.*

ANA *Agência Nacional de Águas e Saneamento Básico.*

ANEEL *Agência Nacional de Energia Elétrica.*

ANN *Artificial Neural Network.*

AR *AutoRegressive.*

ARCH *AutoRegressive Conditional Heteroscedasticity.*

ARFIMA *Autoregressive Fractionally Integrated Moving Average.*

ARIMA *AutoRegressive Integrated Moving Average.*

ARMA *AutoRegressive Moving Average.*

BSO *Brain Storm Optimization.*

CCEE *Câmara de Comercialização de Energia Elétrica.*

COPEL *Companhia Paranaense de Energia.*

CS *Cuckoo Search.*

ERNN *Elman Recurrent Neural Network.*

GARCH *Generalized Autoregressive Conditional Heteroskedasticity.*

GRU *Gated Recurrent Unit.*

IRENA *International Renewable Energy Agency.*

KNN *K-Nearest Neighbors.*

LSTM *Long Short-Term Memory.*

MA *Moving Average.*

MAE *Mean Absolute Error.*

MAPE *Mean Absolute Percentage Error.*

MdAE *Median Absolute Error.*

MdAPE *Median Absolute Percentage Error.*

MLP *MultiLayer Perceptron.*

MRLS *Modelo de Regressão Linear Simples.*

MSE *Mean Squared Error.*

PACF *Partial AutoCorrelation Function.*

PMC *Perceptron MultiCamadas.*

PMF *Persistence Model Forecast.*

ReLU *Rectified Linear Unit.*

RMSE *Root Mean Square Error.*

RNN *Recurrent Neural Network.*

RP *Resilient Propagation.*

SARIMA *Seasonal AutoRegressive Integrated Moving Average.*

SDR *Standard Deviation Reduction.*

SGD *Stochastic Gradient Descent.*

SIN *Sistema Interligado Nacional.*

SVM *Support Vector Machine.*

SVR *Support Vector Regression.*

VAR *Vector Autoregression.*

VECM *Vector Error Correction Model.*

SUMÁRIO

1	INTRODUÇÃO	23
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	APRENDIZADO SUPERVISIONADO	27
2.1.1	CLASSIFICAÇÃO	28
2.1.2	REGRESSÃO	28
2.2	SÉRIES TEMPORAIS	29
2.2.1	AUTOCORRELAÇÃO	32
2.2.2	TENDÊNCIA	34
2.2.3	SAZONALIDADE	35
2.2.4	RUÍDO	35
2.2.5	ESTACIONARIEDADE	36
2.2.6	TESTE ADF	38
2.2.7	DIFERENCIAÇÃO	39
2.2.8	NOTAÇÃO BACKSHIFT	41
2.2.9	LOOK BACK	41
2.2.10	ESCALA MIN MAX	42
2.2.11	WALK FORWARD	43
2.3	MODELOS DE PREVISÃO	46
2.3.1	MODELOS UNIVARIADOS	46
2.3.1.1	PERSISTENCE MODEL FORECAST	46
2.3.1.2	AR	47
2.3.1.3	MA	48
2.3.1.4	ARMA	50
2.3.1.5	ARIMA	52
2.3.1.6	SARIMA	54
2.3.1.7	ARCH	56
2.3.1.8	GARCH	58
2.3.1.9	EXPONENTIAL SMOOTHING	59
2.3.2	MODELOS MISTOS	60
2.3.2.1	REGRESSÃO LINEAR	60
2.3.2.2	REGRESSÃO POLINOMIAL	61
2.3.2.3	ÁRVORE DE REGRESSÃO	62
2.3.2.4	BAGGING	65

2.3.2.5	FLORESTAS ALEATÓRIAS	66
2.3.2.6	GRADIENT BOOSTING	68
2.3.2.7	PERCEPTRON.	70
2.3.2.8	MLP	73
2.3.2.9	LSTM	75
2.3.2.10	GRU	77
2.3.3	MODELOS MULTIVARIADOS	78
2.3.3.1	VAR	78
2.3.3.2	VECM	80
2.4	CONSIDERAÇÕES FINAIS	82
3	TRABALHOS RELACIONADOS	83
3.1	PREVISÃO DE VELOCIDADE DE VENTO COM REDES NEURAS . . .	83
3.2	PREVISÃO DE VELOCIDADE DE VENTO COM APRENDIZADO DE MÁQUINA	84
3.3	PREVISÃO DE DADOS HIDROLÓGICOS VISANDO MODELOS DERIVADOS DA TEORIA DE SÉRIES TEMPORAIS	85
3.4	CONSIDERAÇÕES FINAIS	86
4	METODOLOGIA	87
4.1	BASE DE DADOS	87
4.2	MODO UNIVARIADO	90
4.3	MODO MULTIVARIADO	96
4.4	REDE DE REPRESENTAÇÃO	99
4.5	CONSIDERAÇÕES FINAIS	101
5	EXPERIMENTOS	102
6	RESULTADOS	113
6.1	ANÁLISE DE DESEMPENHO DE TODOS OS MODELOS	113
6.1.1	RESULTADOS DOS MODELOS UNIVARIADOS	114
6.1.2	RESULTADOS DOS MODELOS MULTIVARIADOS	122
6.2	ANÁLISE DAS ETAPAS DE PRÉ-PROCESSAMENTO	126
6.2.1	USO DA ESTACIONARIEDADE	126
6.2.2	USO DE ESCALA	128
6.2.3	USO DE LOOK BACK	130
6.2.4	USO DE WALK FORWARD	133
6.2.5	USO DA REDE DE REPRESENTAÇÃO	134
6.2.6	USO DE FUNÇÃO DE ATIVAÇÃO	136
6.2.7	USO DE LOOK BACK NA REDE DE REPRESENTAÇÃO	138

6.2.8	ANÁLISE DOS PRÉ-PROCESSAMENTOS UTILIZADOS NOS RESULTADOS	141
6.3	ANÁLISE DOS MELHORES MODELOS	142
6.3.1	RESULTADOS DOS MODELOS UNIVARIADOS	142
6.3.2	RESULTADOS DOS MODELOS MULTIVARIADOS	147
6.4	CONSIDERAÇÕES FINAIS	152
7	CONCLUSÃO	154
	REFERÊNCIAS	157
	APÊNDICE A - HIPERPARÂMETROS DE TODOS OS MELHORES MODELOS	168
	APÊNDICE B - MELHORES RESULTADOS EM TODOS OS MODELOS	174
B.1	MODELOS UNIVARIADOS	174
B.2	MODELOS MULTIVARIADOS	179
	APÊNDICE C - MÉTRICAS E GRÁFICOS DOS DOIS MELHORES MODELOS DE CADA MODO	181
C.1	MODELOS UNIVARIADOS COM HORIZONTE DE PREVISÃO DE 12 MESES	181
C.2	MODELOS UNIVARIADOS COM HORIZONTE DE PREVISÃO DE 60 MESES	185
C.3	MODELOS MULTIVARIADOS COM HORIZONTE DE PREVISÃO DE 12 MESES	188
C.4	MODELOS MULTIVARIADOS COM HORIZONTE DE PREVISÃO DE 60 MESES	192
	APÊNDICE D - GRÁFICOS DE DISPERSÃO DE TODAS AS BASES DE DADOS	195
D.1	AFRÂNIO	195
D.2	ESTREITO	196
D.3	OFFSHORE CE	197
D.4	OFFSHORE SC	198
D.5	PALMAS	199
D.6	QUILOMBO	200
D.7	SÃO JOÃO	201

1 INTRODUÇÃO

De acordo com (Ritchie e Roser, 2020), desde a Revolução Industrial, a matriz energética da maioria dos países do mundo tornou-se dominada por combustíveis fósseis, gerando implicações importantes para o clima global, bem como para a saúde humana. Como uma forma de solução para reduzir a quantidade de CO₂ e a poluição do ar, as energias renováveis têm se destacado ao redor do mundo nos últimos anos. Um exemplo da ampla adoção desta estratégia ocorreu em 2019, em que cerca de 11% da energia primária global veio de tecnologias renováveis.

Dentre as várias formas de tecnologias renováveis, a energia hidrelétrica é a que mais se sobressai, representando 16% da eletricidade global e 64,58% da eletricidade brasileira em 2019 (Ritchie e Roser, 2020). O segundo lugar é ocupado pela energia eólica, que vem ganhando popularidade dentre as técnicas de geração de energia renovável. Em 2019, 2% da energia global e 5% da eletricidade mundial vieram do vento. No Brasil, a partir de 2006, se iniciou o investimento em energia eólica, indo de 0,06% a 9,36%, em 2020, de produção de eletricidade a partir dos parques eólicos (Ritchie e Roser, 2020).

Observando os dados da IRENA (*International Renewable Energy Agency*), a tendência de capacidade instalada explorando a energia eólica só aumenta ao longo dos anos, apresentando um comportamento bastante crescente de 2019 para 2020 no Brasil (IRENA, 2020). Considerando esse cenário, muitos trabalhos têm surgido ao longo dos anos para ajudar a prever a quantidade ou velocidade de vento em locais com potencial para instalação de parques eólicos ou até mesmo para fazer um planejamento da quantidade de energia que será gerada nos próximos meses.

A maioria das pesquisas atuais se baseiam em técnicas de Regressão, seguindo a ideia do Aprendizado Supervisionado, pertencentes à área de Aprendizado de Máquina e Aprendizado Profundo, como em (Del Ser et. al, 2021), que utiliza algoritmos aleatorizados, como as técnicas de *Bagging* e *Boosting*, e (Alvarez et al, 2020), que explora modelos como SVM (*Support Vector Machine*) e Redes Neurais. Mesmo com a popularidade dessas áreas, ainda há o investimento em modelos estatísticos para realizar essas previsões, como o modelo ARIMA (*AutoRegressive Integrated Moving Average*) presente em (Swaroop e Reddy, 2018).

Em geral, o problema encontrado em grande parte dessas pesquisas se encontra na quantidade de dados eólicos disponíveis, que não abrangem muitos anos, pois o investimento

maior neste tipo de energia ocorreu somente depois dos anos 2000, diferentemente dos dados hidrológicos que tiveram bastante atenção desde antes de 1965 (Ritchie e Roser, 2020). Além disso, considerando o cenário do Brasil, ao se comparar com a quantidade de usinas hidrelétricas, o número de parques eólicos é bem menor. São 217 usinas hidrelétricas em operação, com 713 centrais geradoras hidrelétricas e 425 pequenas centrais hidrelétricas (ANEEL, 2019), enquanto, de acordo com a Associação Brasileira de Energia Eólica (ABEEólica), o Brasil conta com 695 parques eólicos (ABEEólica, 2021).

Além das circunstâncias da quantidade de dados eólicos, o tipo desses dados requer um pré-processamento cuidadoso, pois eles pertencem à categoria de Séries Temporais, em que cada valor dos dados possui uma forte relação com o tempo. Considerando o contexto de Séries Temporais, existem várias formas de pré-processamentos, como a transformação de estacionariedade, escala e quanto do passado da série será utilizado para realizar uma predição (chamado de *Look Back*), por exemplo. Essas técnicas visam ajudar os modelos de previsão a diminuir os erros ao fornecer dados de entrada mais significativos (Oguntayo, 2020).

Depois da preparação dos dados, inúmeros modelos de previsão podem ser escolhidos para serem treinados. Geralmente, os modelos mais populares pertencem a uma das três categorias, sendo os derivados da Teoria de Séries Temporais, como AR (*Autoregressive*), MA (*Moving Average*), ARMA (*AutoRegressive Moving Average*) e ARIMA (*AutoRegressive Integrated Moving Average*), os advindos da área de Aprendizado de Máquina, como a Regressão Linear ou algoritmos de *Ensemble*, como *Bagging* e *Gradient Boosting* e, por fim, os modelos de Aprendizado Profundo, com as Redes Neurais profundas e recorrentes, como LSTM (*Long Short-Term Memory*) e GRU (*Gated Recurrent Unit*).

Neste trabalho foram definidos 6 objetivos, sendo o primeiro deles criar uma metodologia capaz de prever o último ano de dados mensais eólicos utilizando os meses dos 18 anos passados de dados de velocidade de vento. Essa metodologia foi chamada de Univariada, pois utilizou somente uma variável para prever os dados eólicos. A segunda meta foi descobrir se os dados hidrológicos, de 22 aproveitamentos hidrelétricos diferentes, poderiam ser capazes de prever as velocidades de vento, processo que foi chamado de Multivariado (22 variáveis de característica). Dessa forma, se fosse possível prever os dados eólicos utilizando dados hidrológicos, os modelos de previsão não precisariam enfrentar o desafio de realizar predições com uma pequena quantidade de dados, reduzindo as possibilidades de *underfitting*.

O terceiro objetivo se concentrou em testar a influência das diferentes técnicas de pré-processamento, como a Estacionariedade das séries temporais, *Look Back* e *Walk Forward*, além de propor uma nova maneira de combinar os dados dos meses anteriores e gerar uma nova característica, método chamado de Rede de Representação. Após analisar as técnicas de pré-processamento, o próximo passo foi comparar os diferentes modelos das três áreas já mencionadas, com a intenção de descobrir se os modelos de Aprendizado Profundo sempre alcançam resultados significativamente superiores do que os modelos derivados da Teoria de Séries Temporais e Aprendizado de Máquina.

Os modelos avaliados neste trabalho foram: AR (*AutoRegressive*), MA (*Moving Average*), ARMA (*AutoRegressive Moving Average*), ARIMA (*AutoRegressive Integrated Moving Average*), SARIMA (*Seasonal ARIMA*), ARCH (*Autoregressive Conditional Heteroskedasticity*), GARCH (*Generalized ARCH*), *Exponential Smoothing*, VAR (*Vector AutoRegression*) e VECM (*Vector Error Correction Model*) na área de modelos derivados da Teoria de Séries Temporais; Regressão Linear, Regressão Polinomial, Árvore de Regressão, *Bagging*, Florestas Aleatórias e *Gradient Boosting*, nos modelos da área de Aprendizado de Máquina e *Perceptron*, MLP (*MultiLayer Perceptron*), LSTM (*Long Short-Term Memory*) e GRU (*Gated Recurrent Unit*), nos modelos do campo de Aprendizado Profundo.

Com a preparação dos dados e com os modelos implementados, as duas últimas etapas retratam duas otimizações diferentes, uma para os hiperparâmetros de cada modelo e outra para os valores dos parâmetros de pré-processamento. Assim, foi feita uma *Grid Search* com várias combinações de valores de pré-processamento e, para cada combinação, foi realizada a otimização dos hiperparâmetros, que contou com três otimizadores diferentes. Para os modelos derivados da Teoria de Séries Temporais foi realizada outra *Grid Search*, o Skopt foi utilizado para os modelos de Aprendizado de Máquina e, por fim, nos modelos da área de Aprendizado Profundo, foi aplicado o otimizador Talos. Essas otimizações foram feitas em uma base de validação e avaliadas com as métricas MSE, RMSE e MAPE.

Por fim, abrangendo todos os objetivos definidos, foi realizada uma análise de 3 etapas para compreender os resultados. Na primeira etapa foi realizado um estudo de todos os modelos executados, tanto no modo Univariado quanto no Multivariado. No segundo passo foram selecionados exemplos de uso de cada um dos pré-processamentos e feita uma investigação do impacto que eles podem apresentar nos resultados finais das previsões. Na última etapa, as duas melhores configurações de hiperparâmetros e de valores de

pré-processamento foram escolhidas para cada modo, os modelos foram retreinados e avaliados em outras 6 bases de dados com horizontes de predição de 12 e 60 meses.

O trabalho foi organizado da seguinte maneira: o capítulo 2 abrange toda a fundamentação teórica, iniciando nos conceitos de Classificação e Regressão dentro da área de Aprendizado Supervisionado, seguindo para as propriedades, conceitos, fundamentos e pré-processamentos das Séries Temporais e, por fim, descrevendo brevemente os modelos que foram utilizados neste trabalho. O capítulo 3 apresenta três trabalhos relacionados, sendo os dois primeiros sobre previsão de dados eólicos e o terceiro sobre dados hidrológicos. No capítulo 4 é explicada a metodologia, compreendendo a apresentação da base de dados, dos dois modos, Univariado e Multivariado, e a explicação sobre a Rede de Representação proposta. Em seguida, o capítulo 5 se refere aos experimentos, contendo todos os modelos implementados, os hiperparâmetros avaliados e os parâmetros de pré-processamento utilizados. Logo após é descrito o capítulo 6, com os resultados, contendo uma análise de todos os modelos, de cada método de pré-processamento e dos melhores modelos em bases de dados diferentes. Por último, no capítulo 7, é apresentada a conclusão de toda a pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os conceitos estudados para a realização do trabalho, iniciando com uma breve descrição dos problemas de Aprendizado Supervisionado, na seção 2.1, passando para os conceitos, definições, fundamentos, propriedades e pré-processamentos comumente utilizados na área de Séries Temporais, presentes na seção 2.2, seguindo para os modelos de previsão em séries temporais Univariadas (somente uma série temporal) e Multivariadas (várias séries temporais), buscando explorar modelos derivados da Teoria de Séries Temporais, de Aprendizado de Máquina e de Aprendizado Profundo, apresentados na seção 2.3.

2.1 APRENDIZADO SUPERVISIONADO

O aprendizado supervisionado é o sub-ramo mais comum do Aprendizado de Máquina atualmente. O nome de aprendizagem “supervisionada” vem da ideia de que treinar o algoritmo é como ter um professor supervisionando o processo de aprendizagem o tempo todo. O primeiro passo para iniciar o treinamento é criar o conjunto de treinamento, que consiste em emparelhar as entradas com as saídas corretas. Durante o treino, o algoritmo buscará padrões nas entradas que se correlacionam com as saídas desejadas. Após o processo de aprendizagem, o algoritmo receberá novas entradas não vistas (conjunto de teste) e determinará qual a saída mais adequada para aquela entrada (Wilson, 2019).

De forma simplificada, um algoritmo de Aprendizado Supervisionado pode ser escrito como na equação 2.1, onde Y é a saída prevista que é determinada por uma função de mapeamento que atribui uma saída a um valor de entrada x . A função utilizada para conectar as características de entrada a uma saída prevista é criada pelo modelo de Aprendizado de Máquina durante o treinamento (Wilson, 2019).

$$Y = f(x) \tag{2.1}$$

A aprendizagem supervisionada pode ser dividida em duas subcategorias: classificação e regressão.

2.1.1 CLASSIFICAÇÃO

Na classificação, durante o treinamento, o algoritmo recebe dados atribuídos a uma determinada classe. Então, o trabalho do algoritmo de classificação é ler o exemplo de entrada e atribuir a ele uma classe na qual se encaixa, usando como base os dados de treinamento fornecidos. Um exemplo desta subcategoria é o algoritmo que determina se um email é *spam* ou não, o que é chamado de classificação binária. Nesse caso, o algoritmo recebe dados de treinamento com emails dos dois tipos, encontra as correlações dos dados com as classes e cria uma função de mapeamento, como a apresentada na equação 2.1. Então, quando o algoritmo receber um novo email, o modelo utilizará esta função para determinar se o email é *spam* ou não. Dentre os algoritmos de classificação, os mais conhecidos são: Classificadores Lineares, SVMs, Árvores de Decisão, kNN e Florestas Aleatórias (Wilson, 2019).

2.1.2 REGRESSÃO

A regressão é um processo estocástico preditivo em que o modelo tenta encontrar a relação entre as variáveis dependentes e independentes. Assim, o objetivo de um algoritmo de regressão é prever um número contínuo, como vendas, salários ou pontuações em testes. Matematicamente, a Regressão Linear pode ser descrita pela equação 2.2, onde $x[i]$ são as características dos dados e onde $w[i]$ e b são os parâmetros aprendidos durante o treinamento (Wilson, 2019).

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \dots + w[i] * x[i] + b \quad (2.2)$$

Um exemplo do uso da regressão pode ser um modelo que tente determinar a nota do teste de um aluno com base em quantas horas ele estudou na semana do teste. Como o problema só possui uma característica, o resultado da regressão vai assumir a forma de uma reta. Um modelo possível é apresentado pela Figura 2.1 (Wilson, 2019).

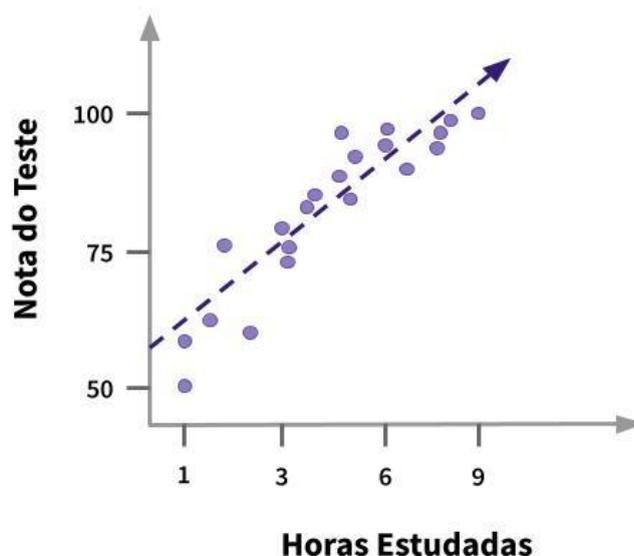


Figura 2.1: Gráfico de exemplo da relação das horas estudadas pelos alunos com as notas obtidas no teste.

FONTE: adaptado de Wilson (2019).

Na figura acima, há uma correlação positiva entre as horas estudadas (variável independente) e a pontuação final do aluno (variável dependente). Com esse modelo, é possível dizer a nota de um aluno que estuda por 5 horas apenas observando a linha obtida pela regressão. Existem vários algoritmos de regressão, sendo os mais comuns: Regressão Linear, Regressão Logística e Regressão Polinomial (Wilson, 2019).

2.2 SÉRIES TEMPORAIS

Uma série temporal é um conjunto ordenado de observações que possuem uma relação com o tempo (Tyagi, 2020a). Esses dados são indicadores de alguma atividade que ocorre em um determinado período de tempo. Assim, as séries costumam ser representadas pelo tempo no eixo x e a magnitude no eixo y (Sen, 2019). Na Figura 2.2 é apresentada uma amostra da base de dados que descreve o número de vendas mensais de shampoo por um período de 3 anos. Nessa imagem é possível observar a forma como as séries temporais são armazenadas, sendo a unidade de tempo (meses, no exemplo) indicada em uma coluna e o valor da série (número de vendas) em outra coluna (Brownlee, 2016a).

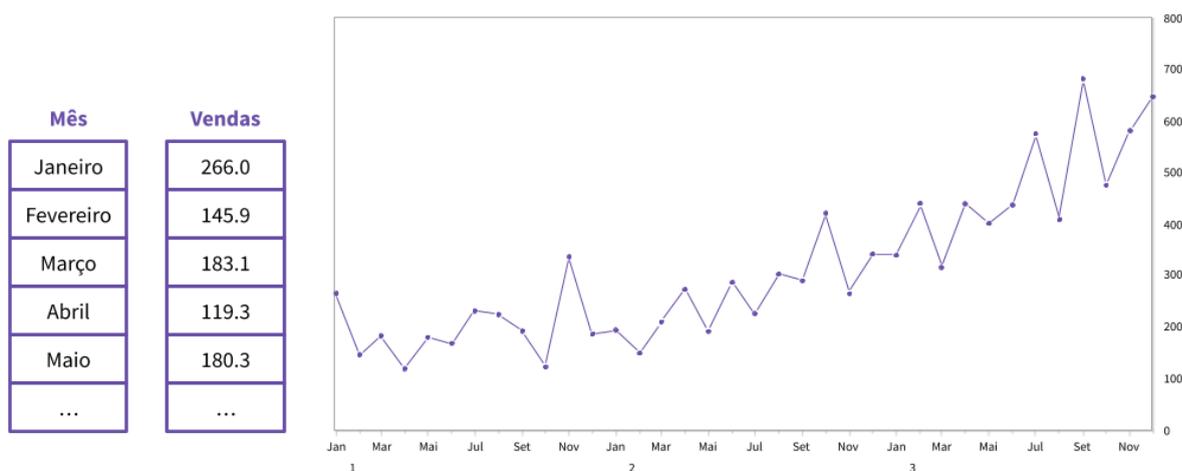


Figura 2.2: Exemplo do armazenamento e representação gráfica da série temporal de vendas mensais de shampoo.

FONTE: adaptado de Brownlee (2016a).

Um aspecto importante das séries temporais é a consistência do intervalo de tempo. Esse intervalo pode ser definido como milissegundos, segundos, minutos, horas, dias, semanas, meses, anos ou até mesmo uma década ou século. Na área de Cibersegurança, um exemplo dessa relação pode ser uma série sobre o número de ataques cibernéticos em um ano para prever a intensidade futura de ataques, como apresentado pela Figura 2.3 (Sen, 2019).

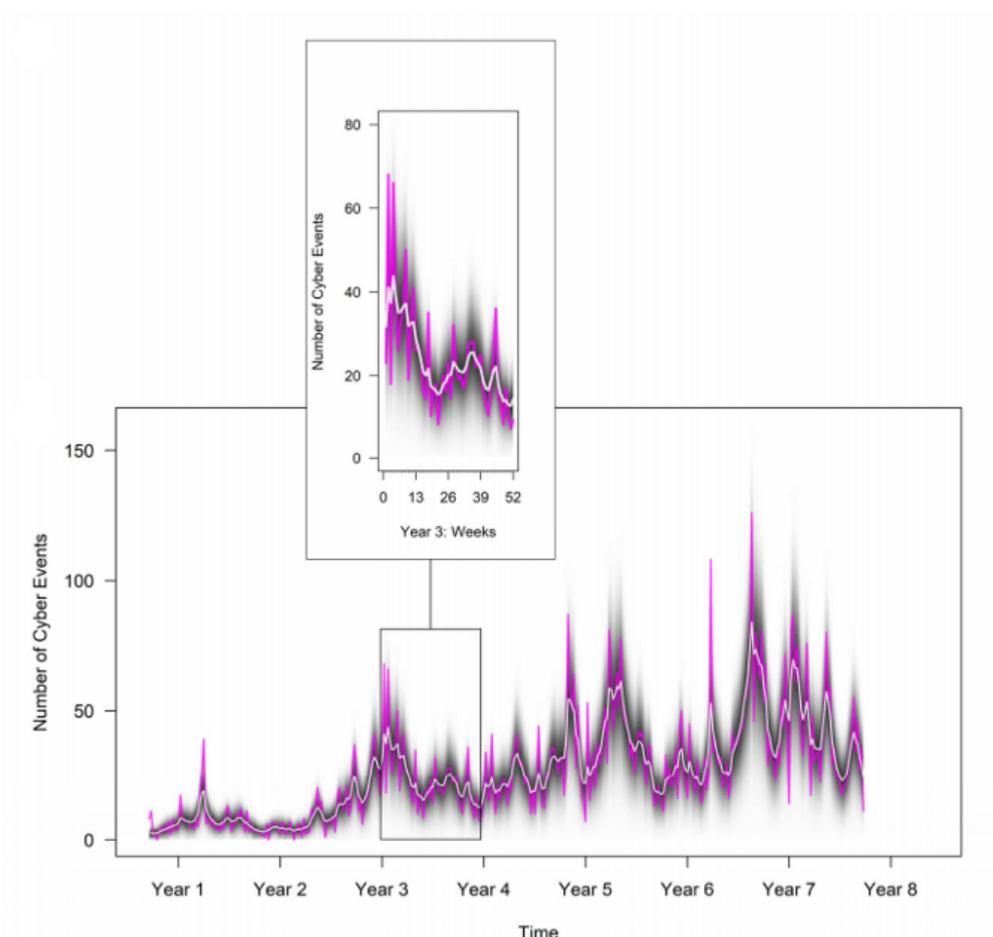


Figura 2.3: Série temporal do número de ataques cibernéticos em 8 anos.

FONTE: Sen (2019).

Junto com a ideia de séries temporais, há o conceito de Análise de Séries Temporais, que se refere ao estudo dos desenhos formados pelos dados das séries temporais. Essa análise envolve a interpretação dos dados e o cálculo das mudanças que ocorrem ao longo do tempo (Sen, 2019). Dentro desse conceito, existem dois métodos, chamados de Univariado e Multivariado. O método Univariado se refere a somente uma variável que está mudando ao longo do tempo, por exemplo, uma série temporal sobre o número de pacotes sendo recebidos em uma rede ao longo do tempo. Já o Multivariado possui várias variáveis passando por alterações ao longo do tempo. Um exemplo similar seria, ao invés de ter o número de pacotes, a série teria informações sobre o endereço IP de origem, endereço IP de destino, porta de origem e porta de destino que são utilizados com mais frequência ao longo do tempo (Sen, 2019).

Dentro da área de Análise de Séries Temporais existem várias noções e definições para ajudar a entender melhor o comportamento das séries. Nas próximas subseções serão apresentados os principais conceitos pertencentes às séries temporais.

2.2.1 AUTOCORRELAÇÃO

A Autocorrelação pode ser vista como uma medida de correlação interna em uma série temporal, como uma forma de medir e explicar a associação interna entre observações. É possível ter uma associação muito forte e positiva, de que a série temporal em um ponto será igual a um ponto em algum tempo no futuro. Ou pode ser uma associação muito forte e negativa, em que a série em um ponto será completamente diferente em um ponto no futuro. O intervalo usado para quantificar a autocorrelação varia entre +1 e -1, sendo +1 uma forte associação positiva, -1 uma forte associação negativa e 0 indicando nenhuma associação (Smith, 2020).

Nesse contexto, dois métodos muito utilizados para compreender a relação entre os diferentes tempos de uma série temporal são o ACF (*Autocorrelation Function*) e o PACF (*Partial Autocorrelation Function*). O ACF descreve quão bem o valor presente está relacionado com seus valores anteriores, considerando todos os componentes presentes na série como tendência, sazonalidade e resíduo, que serão apresentados nas seções 2.2.2, 2.2.3 e 2.2.4. Já o PACF encontra correlação dos resíduos (o que permanece após a remoção dos efeitos que já são explicados pelos tempos anteriores) com o próximo valor de tempo (Salvi, 2019). Para facilitar o entendimento desses métodos será apresentado o exemplo proposto por (Ramkumar, 2019a).

O cenário se resume em um vendedor de salmão que deseja prever o preço médio mensal do salmão, sendo que as únicas informações que ele possui são os preços médios do salmão nos meses anteriores. Vários fatores podem afetar o preço do salmão, como o tempo e o regulamento de pesca, mas nesse exemplo, o fator determinante será o preço do salmão nos meses anteriores. Assim, se o preço foi alto no mês passado, provavelmente vai ser alto esse mês. De forma similar, se o preço foi alto há um ano, no mesmo mês, provavelmente vai ser alto esse mês.

Considerando esse cenário, existem duas formas de relações entre os preços. A primeira é apresentada na Figura 2.4 a), em que há uma influência indireta do preço de janeiro em março, pois o preço em janeiro afeta o preço em fevereiro que afeta o preço em março. Já

na Figura 2.4 b), o preço em janeiro afeta o preço em março de forma direta. Essa relação pode ser explicada por um festival que ocorre a cada dois meses. Assim, o vendedor pode querer lucrar mais e aumentar o preço do salmão somente nesses meses. Dessa forma, se o festival acontecesse em janeiro, o preço do salmão deste mês influencia o preço do salmão no próximo mês do festival, que seria em março.

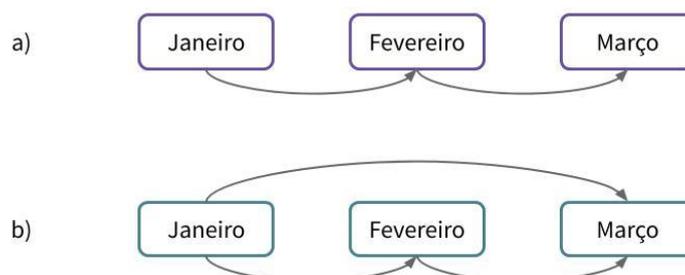


Figura 2.4: Exemplo de diagramas apresentando a influência direta mês a mês (a) e mostrando a relação indireta entre meses não consecutivos (b).

FONTE: adaptado de Ramkumar (2019a).

De uma forma simplificada, o ACF considera as duas relações, direta e indireta, para calcular a autocorrelação. Geralmente, o ACF é calculado usando o coeficiente de correlação *Pearson*. Já o PACF observa somente a relação direta. Ele pode ser importante para definir se a variável de janeiro é uma boa preditora da variável em março, desconsiderando os efeitos indiretos que podem afetar essa observação, como a variável de fevereiro. Matematicamente, o PACF pode ser obtido dos coeficientes de uma regressão que depende do número de períodos passados (*lags*) que serão considerados, da soma do valor da variável de um determinado mês (S_t) multiplicado por um coeficiente (Φ_{ij}) com o valor do erro ϵ_t , sendo t o mês atual. A equação 2.3 mostra a fórmula da regressão com *lag* igual a 2. Os coeficientes desta equação determinam o PACF, pois representam as relações diretas entre as variáveis.

$$S_t = \Phi_{21} S_{t-1} + \Phi_{22} S_{t-2} + \epsilon_t \quad (2.3)$$

Os gráficos do ACF e do PACF podem ser observados na Figura 2.5, em que o eixo x representa os *lags* e o eixo y a autocorrelação do *lag* em relação ao mês atual. Somente as autocorrelações acima da barra de erro são significativas, pois o que está dentro dessas barras não possui evidência suficiente para ser considerado como diferente de zero (Ramkumar, 2019a).

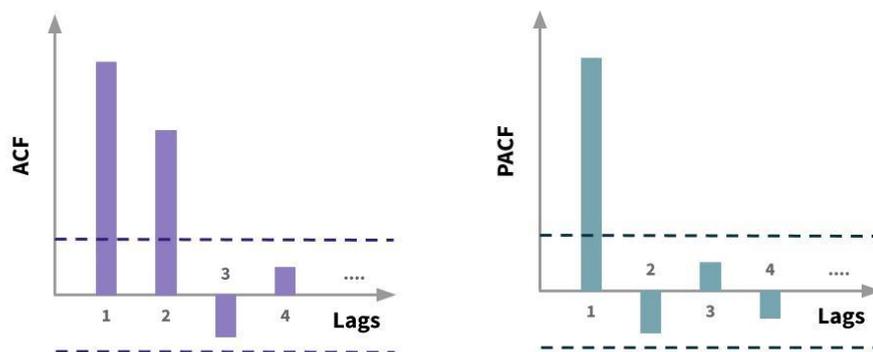


Figura 2.5: Exemplo dos gráficos ACF (esquerda), dos quatro primeiros *lags*, sendo os dois primeiros mais significativos, e PACF (direita), sendo somente o primeiro *lag* significativo.

FONTE: adaptado de Ramkumar (2019a).

2.2.2 TENDÊNCIA

A tendência de uma série temporal é definida como um padrão de crescimento/decrescimento da variável em um certo período de tempo (Oliveira, 2019). Quando a série temporal mostra um padrão geral mais alto, a tendência é considerada alta, quando o padrão é mais baixo, a tendência é baixa. No caso em que a série não possui movimentos altos nem baixos a tendência é chamada de horizontal (Sen, 2019).

Existem vários testes específicos para a identificação da tendência, como o Teste de Wald e o de Cox-Stuart. No entanto, uma técnica muito utilizada é o ajuste de uma Regressão Linear Simples para a identificação da inclinação da reta de tendência. Na Figura 2.6, referente ao número de passageiros internacionais no transporte aéreo entre os anos de 1949 e 1961, é possível observar uma tendência crescente na série ao longo dos anos (Oliveira, 2019).

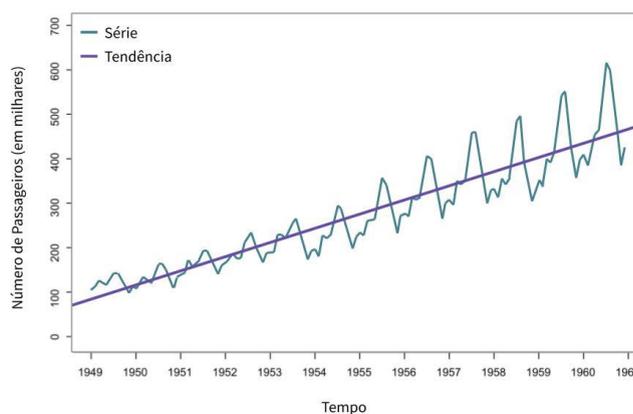


Figura 2.6: Gráfico da tendência do número de passageiros (em milhares) no transporte aéreo entre os anos de 1949 e 1961.

FONTE: adaptado de Oliveira (2019).

2.2.3 SAZONALIDADE

O componente de sazonalidade pode ser definido como padrões de comportamento que se repetem em específicas épocas do ano (Oliveira, 2019). Por exemplo, o consumo de energia tem sazonalidade diurna, pois todo dia há um pico pela tarde e os pontos mais baixos são durante o começo e o final do dia, como é apresentado pela Figura 2.7 (Peixeiro, 2019a).

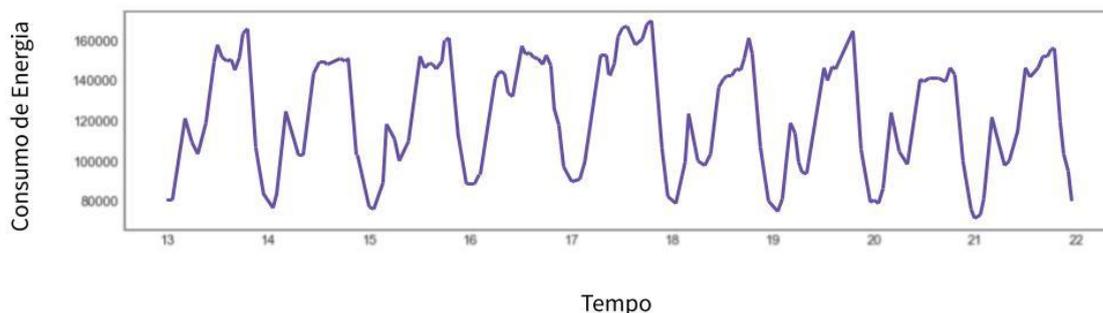


Figura 2.7: Gráfico do consumo de energia (em kWh) em 10 dias, iniciando em 13 de setembro de 2017.
FONTE: adaptado de Peixeiro (2019a).

A sazonalidade, muitas vezes, pode ser identificada de maneira visual. Uma das técnicas formais utilizadas é a identificação de autocorrelações significativas de determinada ordem. Por exemplo, se existe uma sazonalidade anual em uma série, a autocorrelação de ordem 12 geralmente é significativa (Oliveira, 2019).

2.2.4 RUÍDO

As séries temporais podem exibir uma variedade de padrões e geralmente é vantajoso dividir uma série temporal em vários componentes. Existem vários métodos para decompor uma série temporal, como o método aditivo ou o multiplicativo, que conseguem separar a série em três partes, que são: o componente sazonal, o componente de tendência e o componente restante, chamado de ruído (Hyndman e Athanasopoulos, 2018). Então, o ruído é o crescimento ou decréscimo aleatório nos valores da série. Um exemplo de decomposição pode ser observado na Figura 2.8 (Sen, 2019).

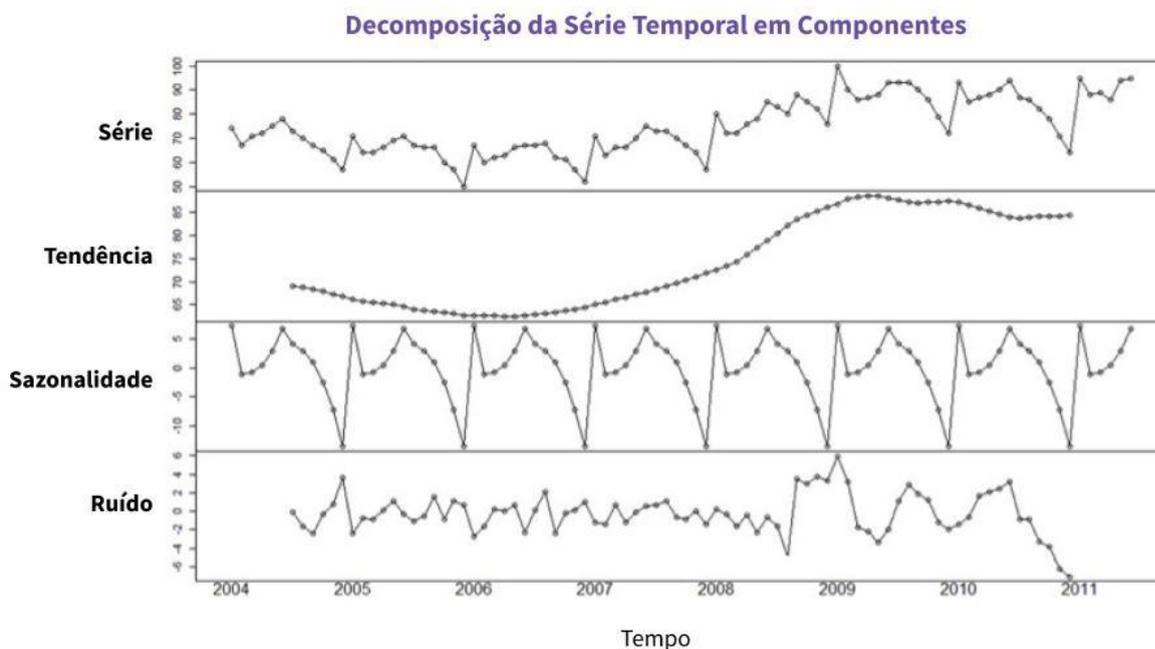


Figura 2.8: Gráficos da série temporal e suas decomposições.
 FONTE: adaptado de Sen (2019).

2.2.5 ESTACIONARIEDADE

Outro elemento importante pertencente ao tema de séries temporais é a estacionariedade. Uma série temporal é considerada estacionária quando suas propriedades estatísticas não mudam com o tempo. Isso quer dizer que a média, variância e autocorrelação são igualmente distribuídas sobre o tempo (Sen, 2019).

A Figura 2.9 demonstra os 3 critérios que precisam ser satisfeitos para a série ser considerada como estacionária. Os gráficos verdes representam uma série estacionária e os gráficos vermelhos apresentam uma série não estacionária que falhou em um dos três critérios de estacionariedade. Na primeira linha, é possível observar que o gráfico vermelho não é estacionário pois a média aumenta ao longo do tempo. Na segunda linha, a variância não é constante no gráfico vermelho. E, por fim, na última linha, é possível notar que a covariância pode ser vista como em função do tempo, pois, a partir de um ponto, ela se espalha à medida que o tempo aumenta (Sen, 2019).

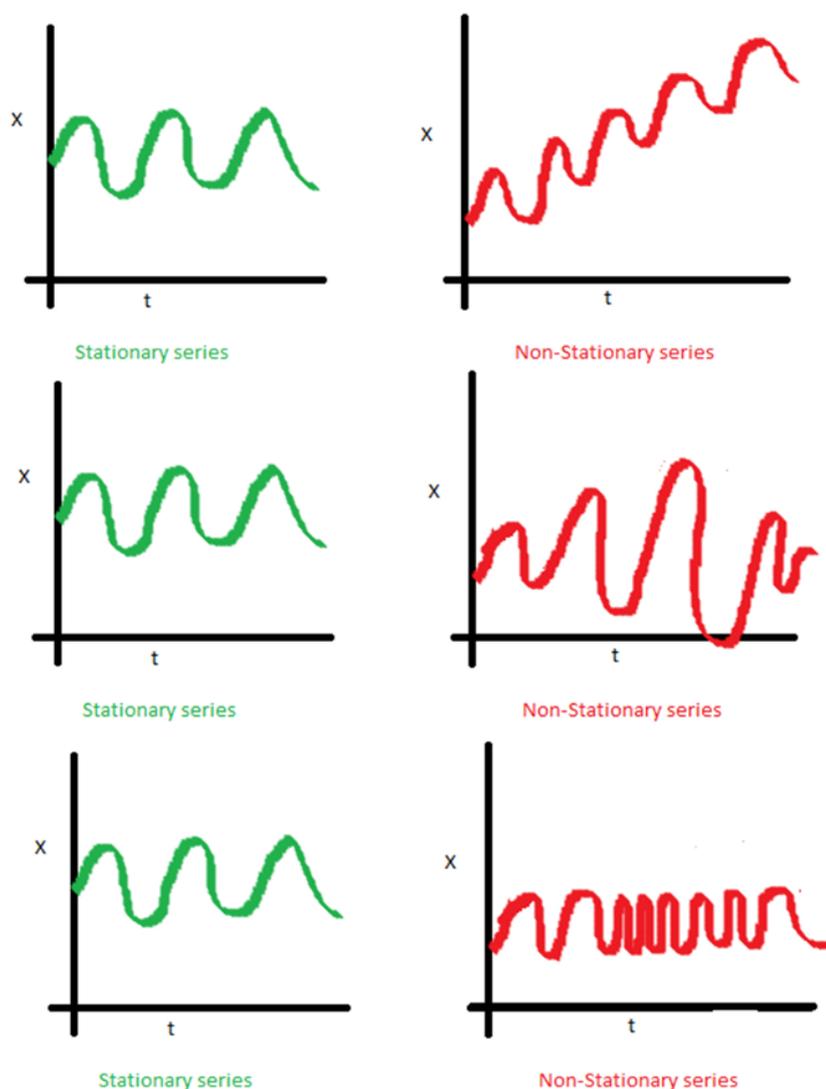


Figura 2.9: Condições de estacionariedade de uma série temporal. Os gráficos verdes representam séries estacionárias e os vermelhos são as séries não estacionárias, que falharam em alguma das três condições (média, variância e autocorrelação).

FONTE: Sen (2019).

A estacionariedade é vista como um pré-requisito dos modelos clássicos de previsão derivados da Teoria de Séries Temporais. O motivo por trás dessa exigência se apoia no fato de que, ao criar um modelo de previsão, as propriedades estatísticas da série temporal (média, variância e a autocorrelação) não devem ser diferentes das propriedades que estão sendo observadas atualmente. Se a série não for estacionária, os modelos e previsões com base nessas propriedades não atingiriam um resultado preciso, pois essas propriedades teriam mudado (Sen, 2019). Em resumo, dados não-estacionários podem assumir qualquer forma, o que não contribui para modelos de previsão que tentam encontrar determinados padrões nessas propriedades (Gupta, 2019).

2.2.6 TESTE ADF

Na seção anterior, foi apresentado o conceito de estacionariedade e mencionada a importância da série ser estacionária ao criar modelos de previsão baseados nas propriedades de séries temporais. O teste ADF (*Augmented Dickey–Fuller Test*) é um teste de hipótese nula muito utilizado para testar se uma determinada série temporal é, ou não, estacionária (Gupta, 2019).

A ideia do teste ADF se baseia no fato de que qualquer série temporal possui um fluxo de direção que depende de algumas características associadas a valores numéricos chamados de raiz (Gupta, 2019). De forma simples, uma série pode ser escrita por vários monômios (expressões com um único termo), sendo que cada monômio corresponde a uma raiz. Se uma dessas raízes for igual a 1, então essa é uma raiz unitária e pode ser a causa da não estacionariedade de uma série (Glen, 2016). Assim, um processo pode ser estacionário se e somente se todos os valores característicos são menores que 1.

Ao usar o teste ADF na verificação de estacionariedade, ele terá vários parâmetros e saídas, sendo os principais o *p-value* e o nível de significância. A hipótese nula é definida como a afirmação da existência de uma raiz única. Assim, se o *p-value* é menor ou igual ao nível de significância então a hipótese nula é rejeitada e a série é estacionária. Caso contrário, não há evidências suficientes para rejeitar a hipótese nula, então a série é considerada não-estacionária (Gupta, 2019). Dois exemplos do uso do teste ADF são apresentados na Figura 2.10, sendo a série da esquerda estacionária com *p-value* de 0 menor do que o nível de significância 0,05, e a série da direita não estacionária (média não é constante), com *p-value* de 0,9585 maior do que o nível de significância 0,05.

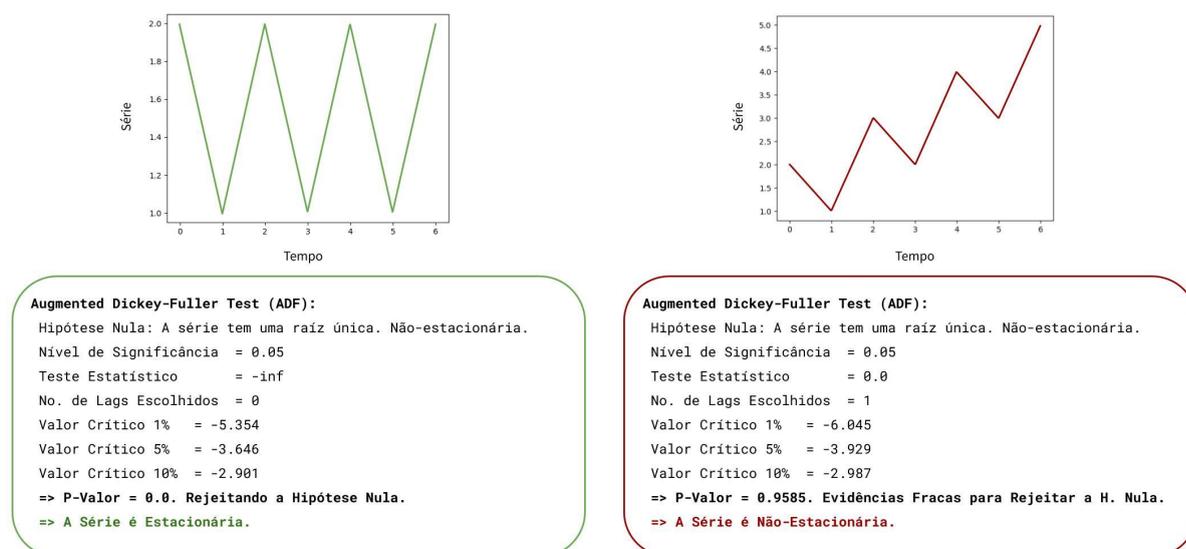


Figura 2.10: Demonstração do resultado do teste ADF em duas séries temporais diferentes, uma estacionária (esquerda) e outra não estacionária (direita). No topo da imagem há a representação gráfica das séries e abaixo os valores obtidos pelo teste.

FONTE: elaborado pela autora (2021).

2.2.7 DIFERENCIAÇÃO

Nas seções anteriores, foi mencionada a importância de utilizar séries estacionárias e também como testar a estacionariedade de uma série. Mas caso uma série não seja estacionária, ela precisará ser transformada de modo a satisfazer as 3 condições de estacionariedade. Para isso, o método mais utilizado é a diferenciação. A diferenciação pode ajudar a estabilizar a média da série temporal com o objetivo de reduzir a tendência e a sazonalidade (Brownlee, 2017a).

O funcionamento desse método é simples, se resumindo a subtrair a observação anterior da observação atual da série. As diferenças entre observações consecutivas são chamadas de diferenças *lag-1*, mas podem ser ajustadas para se adequar a uma estrutura temporal específica, como séries com um componente sazonal onde o *lag* pode ser o período de sazonalidade (Brownlee, 2017a).

Após a aplicação de uma diferenciação ainda existe a possibilidade da série continuar não estacionária, como em casos que a tendência é não linear. Portanto, o processo de diferenciação pode ser repetido mais de uma vez até que todas as dependências temporais sejam removidas. O número de vezes que a diferenciação é realizada é chamado de ordem de diferença (Brownlee, 2017a).

Um exemplo é apresentado na Figura 2.11, onde a série em roxo representa a série original, não estacionária, e nela é aplicada uma diferenciação, resultando na série em azul, como é possível observar no esquema à esquerda da imagem. Para inverter essa transformação, inicialmente é utilizado o primeiro valor da série original, somado com o primeiro valor da série estacionária. Após isso, o resultado é somado com o próximo valor da série azul e assim por diante. No fim, os resultados das somas geram a série original, representada em cinza.

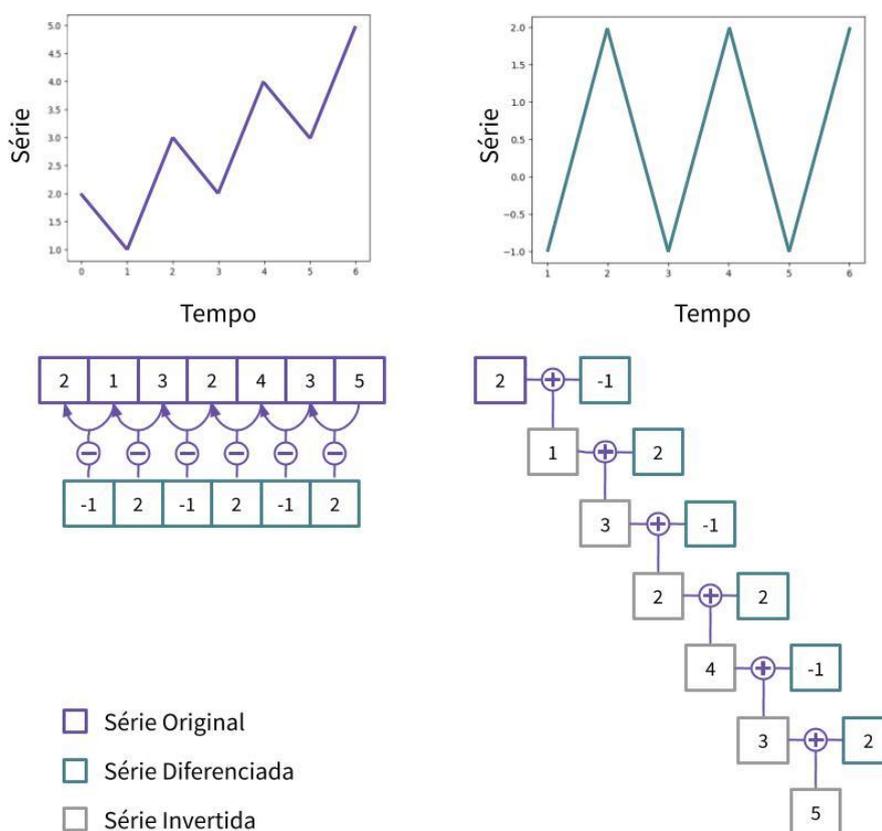


Figura 2.11: Esquema da transformação de uma série não estacionária (em roxo) em uma série estacionária (em azul) por meio da diferenciação. A inversão da diferenciação também é apresentada na parte inferior direita, em que os valores são somados recursivamente até resultar na série temporal original (em cinza).

FONTE: elaborado pela autora (2021).

2.2.8 NOTAÇÃO BACKSHIFT

O operador *Backshift* é uma forma de notação utilizada quando o problema que está sendo resolvido utiliza séries temporais com *lags*. Na equação 2.4 (a), B , operando em y_t (observação do tempo t), tem o efeito de deslocar o dado um período para trás. A equação 2.4 (b) mostra o efeito de duas aplicações de B em y_t e a última equação apresenta a notação para dados com deslocamento de 1 ano (Hyndman e Athanasopoulos, 2018).

$$\begin{aligned}
 (a) \quad B y_t &= y_{t-1} \\
 (b) \quad B(B y_t) &= B^2 y_t = y_{t-2} \\
 (c) \quad B^{12} y_t &= y_{t-12}
 \end{aligned} \tag{2.4}$$

A notação *Backshift* é conveniente para descrever o processo de diferenciação de uma série. A primeira diferença pode ser descrita como apresentada pela equação 2.5 (a), já a segunda diferença, pela equação 2.5 (b). A importância dessa notação se destaca quando as diferenciações são combinadas, pois o operador pode ser tratado utilizando regras algébricas comuns (Hyndman e Athanasopoulos, 2018). O conhecimento desta notação será necessário nas próximas seções para descrever as equações dos modelos derivados da Teoria de Séries Temporais, como o SARIMA (*Seasonal Autoregressive Integrated Moving Average*).

$$\begin{aligned}
 (a) \quad y'_t &= y_t - y_{t-1} = y_t - B y_t = (1 - B) y_t \\
 (b) \quad y''_t &= y_t - 2y_{t-1} + y_{t-2} = (1 - 2B + B^2) y_t = (1 - B)^2 y_t
 \end{aligned} \tag{2.5}$$

2.2.9 LOOK BACK

Para que as séries temporais sejam adequadas para serem utilizadas em modelos de previsão, um dos primeiros passos no pré-processamento é transformar uma sequência de observações em múltiplos exemplos (dados e *targets*) para que o modelo supervisionado possa aprender. Uma das formas de resolver essa questão é utilizar a observação do tempo t como *target* (rótulo) do tempo $t-1$ (Brownlee, 2018a). Nesse caso, o próximo exemplo seria utilizar a observação do tempo $t+1$ como *target* do tempo t e assim por diante, como apresentado pela Figura 2.12, no lado esquerdo.

Além desse método, existe a possibilidade de utilizar mais de uma observação para prever um determinado *target*, transformando o problema de Univariado para Multivariado.

No caso anterior, o *target* do tempo t só estava relacionado com uma observação (do tempo $t-1$), mas ele poderia estar relacionado com três observações. Assim, a observação do tempo t seria *target* das observações $t-1$, $t-2$ e $t-3$, como apresentado à direita da Figura 2.12 (Brownlee, 2018a). A quantidade de observações relacionadas com o *target* é conhecida como “*Look Back*”.

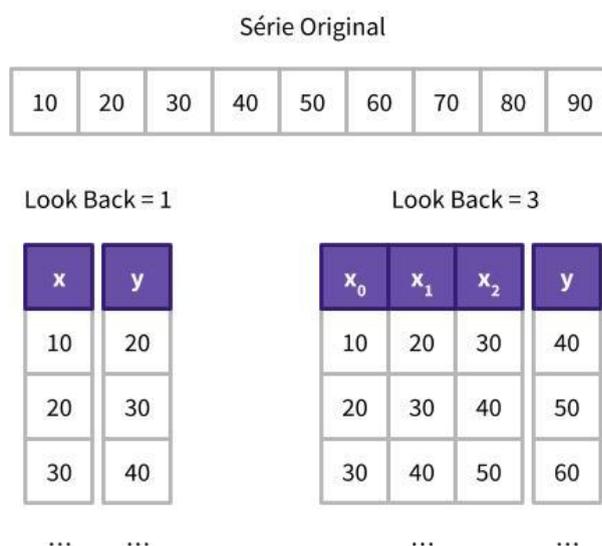


Figura 2.12: Exemplo de adição de *Look Back* à série temporal com janela de 1 e 3 unidades.
 FONTE: elaborado pela autora (2021).

2.2.10 ESCALA MIN MAX

A escala *MinMax* é uma técnica de normalização, que tem o objetivo de escalar os dados para um intervalo fixo, geralmente entre 0 e 1. O custo de ter essa limitação de intervalo é a eliminação de desvios padrões menores o que pode suprimir o efeito de *outliers*, que são dados que se distanciam radicalmente dos outros (Raschka, 2020). Essa transformação é apresentada pela equação 2.6 (Scikit-Learn, 2021), na qual os dados são inicialmente padronizados utilizando os valores mínimos e máximos do conjunto de dados e, logo após, são escalados, contando com duas constantes (MIN e MAX), que indicam o intervalo fixo em que se deseja escalar os dados.

$$dados_padronizados = \frac{dados - \min(dados)}{\max(dados) - \min(dados)}$$

$$dados_escalados = dados_padronizados * (MAX - MIN) + MIN \quad (2.6)$$

2.2.11 WALK FORWARD

O principal objetivo dos modelos de previsão de séries temporais é realizar previsões corretas sobre o futuro. Os métodos rápidos e poderosos de treinamento advindos da área de Aprendizado de Máquina, como Validação Cruzada *K-Fold*, não funcionam para séries temporais, pois eles ignoram componentes temporais inerentes ao problema devido à necessidade de embaralhar os dados. Mas ainda existem vários métodos para treinar esses modelos de forma a manter os dados sequenciais, sendo *Walk Forward* o mais utilizado (Brownlee, 2016b). Na prática, o método *Walk Forward* treina novamente o modelo toda vez que novos dados se tornam disponíveis. Esse método não é caro para bases de dados pequenas, mas pode se tornar um problema para bases com muitos dados devido à grande quantidade de retreino (Srivastava, 2015).

O objetivo do *Walk Forward* é utilizar o máximo de dados para realizar uma boa previsão, pois modelos de séries temporais têm grandes dificuldades para prever dados em um espaço de tempo muito grande. Essa dificuldade tem relação com o termo *Random Walk*, que é um processo em que os objetos se afastam de sua posição inicial ao se moverem aleatoriamente. Para exemplificar, é possível considerar o cenário em que uma peça se move aleatoriamente em um tabuleiro de xadrez e que o objetivo é prever qual posição do tabuleiro ela estará no próximo movimento (Figura 2.13). Inicialmente, no tempo 0, é possível dizer exatamente onde a peça está. Já no tempo 1, a peça pode estar em um dos 8 quadrados em volta da posição do tempo anterior, o que diminui a probabilidade de acerto para $1/8$. Quanto mais o tempo passa sem novos dados, mais difícil fica a previsão (Srivastava, 2015).



Figura 2.13: Exemplo do processo de *Random Walk*.

FONTE: Srivastava (2015).

Para ajudar o modelo de previsão a ter mais informações para realizar a predição, o método de *Walk Forward* costuma ser utilizado e seu funcionamento é apresentado na Figura 2.14. O método inicia com o modelo de previsão sendo treinado com a base de treinamento (1). Após o treinamento terminar, um exemplo do conjunto de teste é dado ao modelo e ele retorna uma previsão, chamada de “Resultado Obtido”. Essa previsão é armazenada, pois representa parte da saída do modelo (2). O diferencial do *Walk Forward* se mostra no passo 3. Ao invés de seguir realizando as previsões do conjunto de teste, a base de treinamento é incrementada com o dado e o rótulo do exemplo do conjunto de teste que acabou de ser previsto pelo modelo. Então, com essa nova base de treinamento (4), o modelo é re-treinado (5) e o próximo exemplo do conjunto de teste é dado ao modelo para ser previsto (6). Esse processo se repete até todos os dados de teste receberem uma previsão (Brownlee, 2016b).

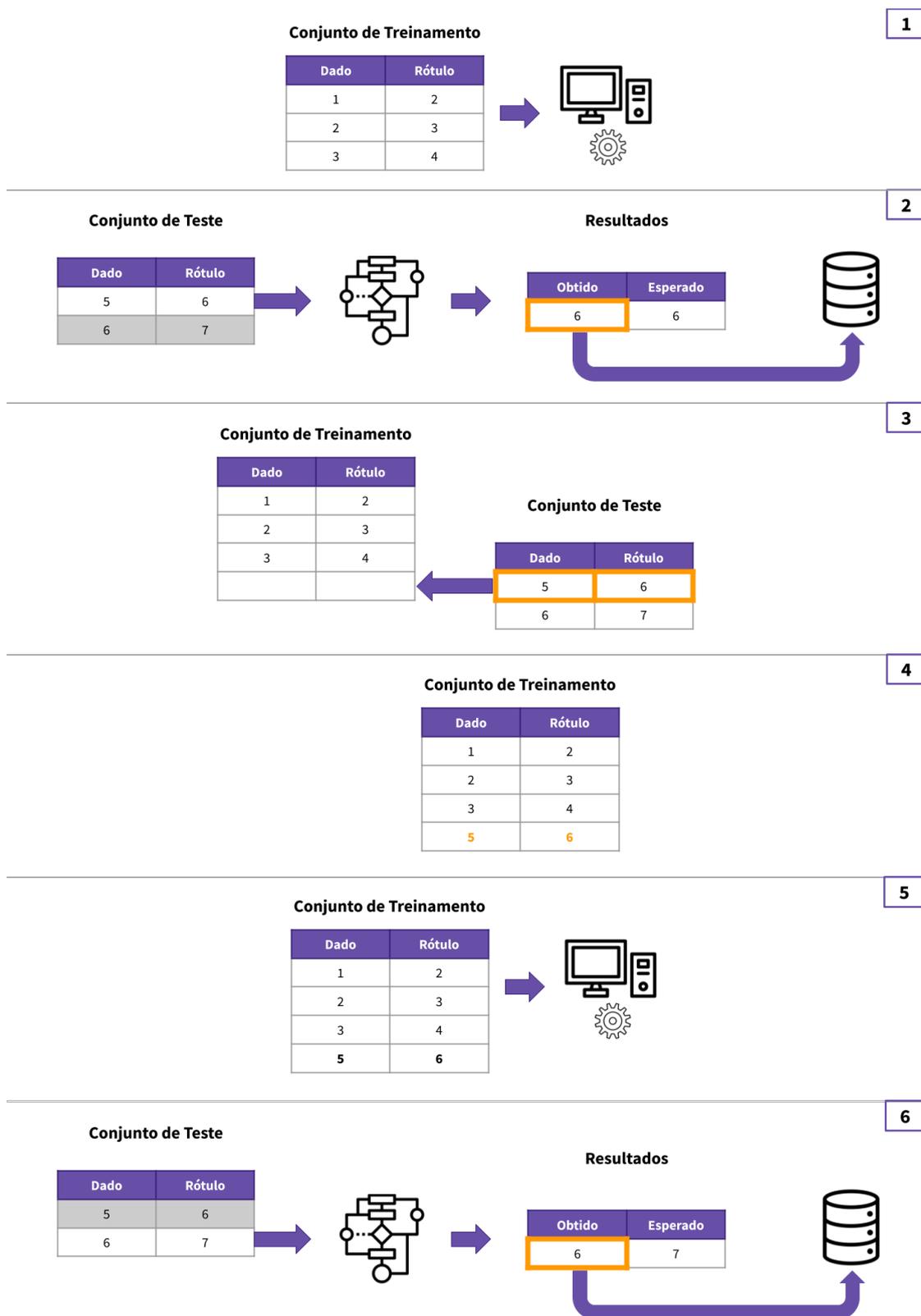


Figura 2.14: Fluxo de execução do método *Walk Forward*.

FONTE: elaborado pela autora (2021).

2.3 MODELOS DE PREVISÃO

Nas próximas subseções serão descritos os modelos de previsão considerados neste trabalho. Esses modelos foram classificados em dois modos, chamados de Univariado e Multivariado, que se referem à quantidade de características utilizadas para realizar as previsões. Além dessa classificação, os modelos foram categorizados de acordo com a área em que se originaram, definindo assim modelos derivados da Teoria de Séries Temporais (AR, MA, ARMA, ARIMA, SARIMA, ARCH, GARCH, *Exponential Smoothing*, VAR e VECM), advindos da área de Aprendizado de Máquina (Regressão Linear, Regressão Polinomial, Árvore de Regressão, *Bagging*, Florestas Aleatórias e *Gradient Boosting*) e, por fim, de Aprendizado Profundo (*Perceptron*, MLP, LSTM e GRU).

2.3.1 MODELOS UNIVARIADOS

Nesta seção serão apresentados os modelos que consideram apenas uma variável para realizar as previsões, chamados de Univariados. Os modelos Univariados selecionados para comporem esta pesquisa foram: *Persistence Model Forecast*, AR, MA, ARMA, ARIMA, SARIMA, ARCH, GARCH e *Exponential Smoothing*, que serão brevemente descritos nas subseções abaixo.

2.3.1.1 PERSISTENCE MODEL FORECAST

O Modelo PMF (*Persistence Model Forecast*) é o modelo mais simples para previsão de séries temporais, geralmente servindo como base para avaliar outros modelos. O funcionamento do PMF se resume a utilizar a observação do tempo anterior $y(t-1)$ como previsão para o tempo atual $\hat{y}(t)$ (Brownlee, 2017b). Matematicamente, o PMF pode ser representado pela equação 2.7 (Brownlee, 2017b). Um exemplo visual do comportamento do modelo é apresentado na Figura 2.15 na qual a série predita é apenas a série original deslocada por uma unidade de tempo.

$$PMF: \hat{y}(t) = y(t - 1) \quad (2.7)$$

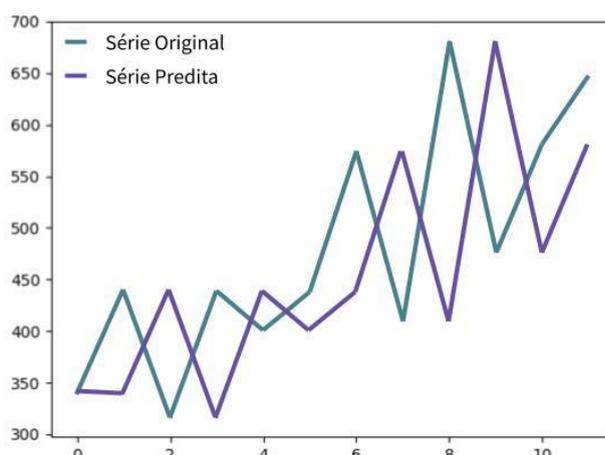


Figura 2.15: Exemplo de uso do modelo PMF para predição de série temporal.
 FONTE: adaptado de Brownlee (2017b).

2.3.1.2 AR

O modelo AR (*AutoRegressive Model*) é um modelo derivado da Teoria de Séries Temporais utilizado para descrever processos que variam ao longo do tempo (Pardoe, 2020). Considerando uma sequência de observações, a ideia deste modelo é prever o próximo passo usando uma função linear composta pelas observações anteriores (Brownlee, 2018b). Desta forma, é feita uma regressão linear, mas diferente da regressão comum, o AR é baseado nos valores passados da própria sequência. Por exemplo, seria como prever o valor de hoje se baseando no valor de ontem, da semana anterior ou do mês passado.

Matematicamente, o modelo pode ser definido pela equação 2.8 (Hyndman e Athanasopoulos, 2018), onde p é a ordem do modelo, X_t é a previsão do tempo t realizada pelo modelo, c é uma constante, β_1, \dots, β_p são os parâmetros do modelo e $\varepsilon_1, \dots, \varepsilon_p$ são os ruídos (*White Noise*).

$$AR(p): X_t = c + \sum_{i=1}^p \beta_i X_{t-i} + \varepsilon_t \quad (2.8)$$

Para entender melhor o funcionamento do modelo, pode-se pensar em um trabalhador responsável por distribuir leite por todo o país, como no exemplo proposto por (Ramkumar, 2019b). Esse trabalhador deseja saber qual a quantidade necessária de leite que precisa ser produzida mês a mês para que consiga satisfazer todos os seus clientes. Ele não quer produzir muito leite, porque esse leite pode estragar e não quer produzir pouco leite a ponto de não

conseguir cumprir todos os seus pedidos. O objetivo do trabalhador é saber a quantidade exata de leite que deve ser carregada no caminhão no mês atual.

Então, para aplicar o modelo AR nesse problema, o primeiro passo é gerar um gráfico no qual o eixo x é o tempo em meses e o eixo y é a quantidade de leite entregue em cada mês. Na esquerda da Figura 2.16 é possível observar um padrão nos últimos meses. A quantidade de leite necessária aumenta até um pouco depois da metade do ano, depois diminui e termina em um platô. Para definir esse problema matematicamente, X_t será a quantidade de leite necessária no mês atual t , sendo assim, X_{t-1} é a quantidade de leite do mês passado ($t-1$) e X_{t-12} a quantidade de leite necessária há 12 meses. O objetivo é prever X_t , usando X_{t-x} , que são os valores dos meses passados.

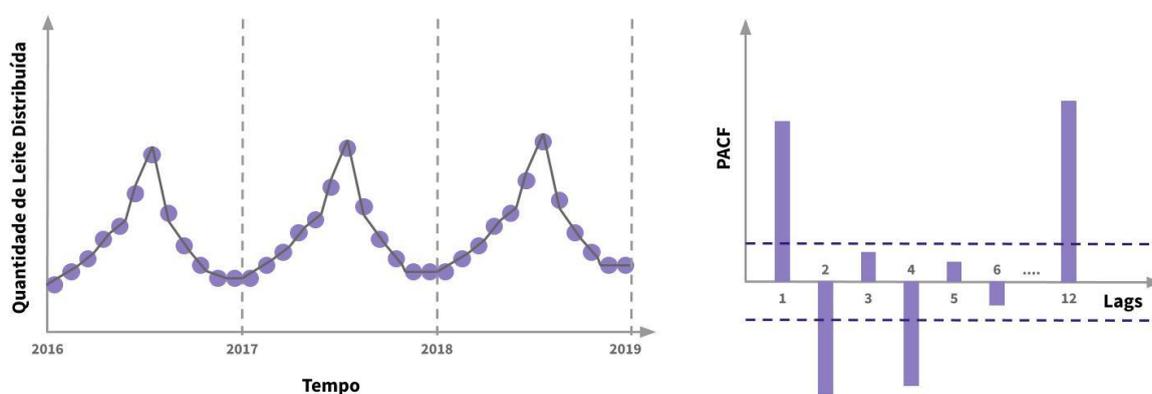


Figura 2.16: Gráfico de exemplo da distribuição de leite em cada mês (esquerda) e gráfico do PACF construído com os dados do problema (direita).

FONTE: adaptado de Ramkumar (2019b).

O segundo passo é descobrir quais *lags* são importantes para esse problema. De acordo com o gráfico PACF presente na Figura 2.16, os *lags* 1, 2, 4 e 12 são significativamente diferentes de zero o que indica que podem ter uma boa correlação com o que está sendo predito. Então, um bom modelo para este problema seria:

$$X_t = c + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \beta_4 X_{t-4} + \beta_{12} X_{t-12} + \varepsilon_{t-1} \quad (2.9)$$

2.3.1.3 MA

O modelo MA (*Moving Average Model*) também é um modelo derivado da Teoria de Séries Temporais e, ao invés de utilizar os valores passados da sequência para realizar a regressão, o MA considera os erros de previsão anteriores para construir o modelo (Hyndman e Athanasopoulos, 2018). Esse modelo é uma das ferramentas de análise técnica mais

populares no ramo de investimentos (Capital, 2020), e pode ser definido pela equação 2.10, onde q é a ordem do modelo, X_t é a previsão do tempo t realizada pelo modelo, c é uma constante, Φ_1, \dots, Φ_q são os parâmetros do modelo e $\epsilon_1, \dots, \epsilon_q$ são os ruídos (*White Noise*) (Hyndman e Athanasopoulos, 2018).

$$MA(q): X_t = c + \sum_{i=1}^q \phi_i \epsilon_{t-i} + \epsilon_t \quad (2.10)$$

Para exemplificar o funcionamento do modelo, pode-se imaginar o seguinte cenário: todo mês um professor de uma universidade faz um jantar e convida os estudantes de graduação para participar, com a condição de que cada estudante deve trazer 10 *cupcakes*. O professor acredita que um determinado aluno sempre traz a quantidade errada de *cupcakes*, mesmo o aluno tendo a certeza de que entrega os 10 *cupcakes* em todo jantar. Assim, o objetivo é usar o modelo MA para saber quantos *cupcakes* o aluno deve levar para que o professor não diga que ele trouxe a quantidade errada (Ramkumar, 2019c).

Na Figura 2.17, na esquerda, há uma tabela que indica o tempo (em meses), a quantidade prevista e esperada de *cupcakes* para cada jantar e o erro entre essas quantidades.

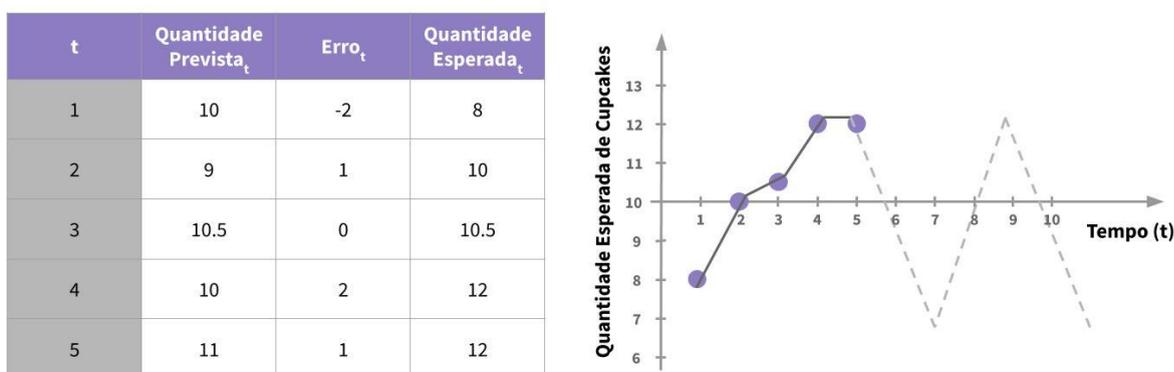


Figura 2.17: Tabela com os dados do exemplo (tempo, previsão, erro e valores esperados) e gráfico de projeção da quantidade esperada de *cupcakes* ao longo do tempo (direita).

FONTE: adaptado de Ramkumar (2019c)

Por simplicidade, será definido um modelo MA(1), onde somente o erro do tempo imediatamente anterior é considerado. Então, no primeiro jantar, o estudante levou 10 *cupcakes*, mas o professor disse que ele trouxe 2 a menos, então, para o professor, o estudante acabou trazendo 8. Para prever o segundo jantar, deve ser considerada a média c como sendo 10, e Φ_1 , que é o coeficiente multiplicativo do erro do mês passado, será 0,5, ou seja, o modelo será ajustado de forma a considerar 50% do erro do *lag* anterior. A previsão do

segundo jantar é mostrada na equação 2.11 e o mesmo processo pode ser executado para os próximos jantares.

$$\begin{aligned} X_2 &= c + \phi_1 \epsilon_1 \\ X_2 &= 10 + 0,5.(-2) \\ X_2 &= 9 \end{aligned} \quad (2.11)$$

O nome do modelo, Média Móvel, tem relação com o fato de que o valor esperado está sempre centrado na média, sendo, no caso do exemplo, o valor 10. Assim, os valores esperados flutuam em volta dessa média, como é possível observar no gráfico da Figura 2.17. Para determinar a ordem deste modelo, o gráfico da Função de Autocorrelação (ACF) dos dados pode ser utilizado.

2.3.1.4 ARMA

O modelo ARMA (*AutoRegressive Moving Average Model*) pode ser considerado como a fusão dos dois modelos apresentados anteriormente (Peixeiro, 2020). A parte AR envolve considerar a influência das observações do passado no presente, já a parte MA se refere à modelagem do presente utilizando os erros passados de previsão (Feijó, 2015).

A definição formal do modelo é apresentada pela equação 2.12, em que p é a ordem do modelo AR, q é a ordem do modelo MA, X_t é a previsão do tempo t realizada pelo modelo, c é uma constante, β_1, \dots, β_p são os parâmetros do modelo AR, Φ_1, \dots, Φ_q são os parâmetros do modelo MA e $\epsilon_1, \dots, \epsilon_q$ são os ruídos (Peixeiro, 2020).

$$ARMA(p, q): X_t = c + \epsilon_t + \sum_{i=1}^p \beta_i X_{t-i} + \sum_{i=1}^q \phi_i \epsilon_{t-i} \quad (2.12)$$

Para explicar o funcionamento do modelo será apresentado o exemplo proposto por (Ramkumar, 2019d). O cenário é composto por um vendedor de lâmpadas com o desafio de descobrir quantas lâmpadas devem ser produzidas para cumprir a demanda do mês. Para resolver esse problema é possível modelar o número de lâmpadas usando um modelo ARMA, com p e q iguais a 1, por simplicidade. A equação do modelo (2.13) é apresentada abaixo, sendo (a) a definição formal e (b) a fórmula usada para a previsão. A diferença entre as equações se resume ao termo ϵ_t , que é o erro do tempo t . Como o objetivo é prever o tempo t , não é possível saber o erro do tempo t , então ele foi removido da equação.

$$\begin{aligned}
 (a) \quad & ARMA(1, 1): X_t = c + \beta_1 X_{t-1} + \phi_1 \epsilon_{t-1} + \epsilon_t \\
 (b) \quad & ARMA(1, 1): X_t = c + \beta_1 X_{t-1} + \phi_1 \epsilon_{t-1}
 \end{aligned}
 \tag{2.13}$$

Por fim, este modelo poderia ser ajustado de forma a encontrar os melhores valores de p e q para o problema que se deseja resolver. Para isso, pode-se utilizar os gráficos ACF e PACF. Considerando o exemplo anterior, se a produção de lâmpadas dos meses anteriores gerassem os gráficos da Figura 2.18, seria possível ajustar p como sendo 1 pelo gráfico PACF e q como sendo 2 pelo gráfico ACF.

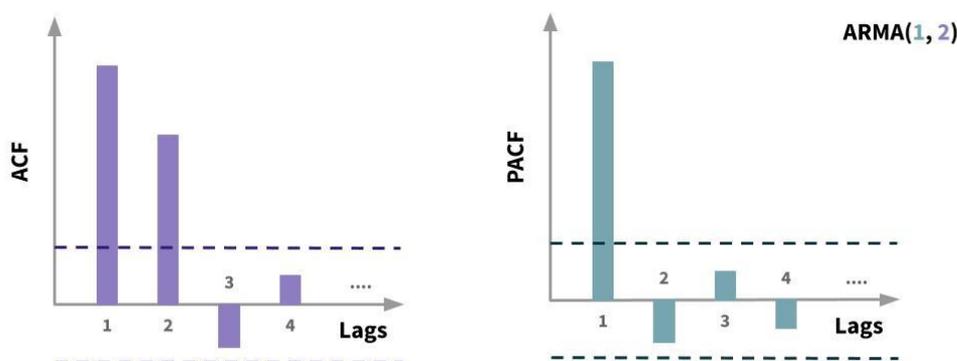


Figura 2.18: Gráficos ACF e PACF do exemplo da série temporal referente à produção de lâmpadas.

FONTE: adaptado de Ramkumar (2019d).

Observando o conjunto de fórmulas apresentadas nesta seção, é possível notar que o modelo ARMA pode ser transformado em um modelo AR, caso seu primeiro parâmetro p seja igual a 0, e em MA, caso o parâmetro q receba o valor 0. Existem vários questionamentos sobre qual o cenário ideal para utilizar cada um desses modelos. A ideia mais intuitiva seria implementar e executar todos eles e realizar uma análise dos resultados. Por mais que sejam modelos simples, nem sempre essa execução será rápida, pois o tamanho do conjunto de dados e a quantidade de *lags* considerados pode afetar diretamente no tempo de processamento. Alguns autores incentivam a análise visual dos dados para facilitar a escolha dos modelos (Alam, 2020).

De acordo com (Masum, 2020), os gráficos ACF e PACF devem ser considerados para definir o modelo. Quando o gráfico ACF diminui gradualmente e o PACF tem uma queda acentuada depois de p *lags* significativos, a melhor escolha é um modelo AR. Quando ocorre o oposto, o ACF mostra uma queda acentuada após um certo número q de *lags* e o PACF mostra uma tendência gradual decrescente, a melhor opção é o modelo MA. Caso os gráficos ACF e PACF demonstrem um padrão decrescente gradual, o processo ARMA deve ser considerado.

2.3.1.5 ARIMA

O modelo ARIMA (*AutoRegressive Integrated Moving Average Model*) é uma generalização do modelo ARMA com a adição da noção de integração. O “I” pertencente ao acrônimo representa o uso de diferenciação das observações com o objetivo de tornar a série estacionária. Essa diferenciação pode ser, por exemplo, a subtração de uma observação do tempo t menos a observação do tempo $t-1$. Dessa forma, a vantagem desse modelo é que não é necessário tratar o problema da não-estacionariedade previamente, como nos modelos anteriores, pois o próprio modelo corrige os dados (Brownlee, 2020a). Esse método possui 3 parâmetros (Brownlee, 2020a):

- **p**: número de observações anteriores incluídas no modelo, também chamado de ordem de *lag*.
- **d**: número de vezes que as observações são diferenciadas, também chamado de grau de diferenciação.
- **q**: tamanho da janela da média móvel, também chamado de ordem da Média Móvel.

Com a junção de vários componentes, a definição matemática do ARIMA acaba sendo dividida em etapas. Inicialmente a série é diferenciada d vezes, sendo o resultado chamado de z_t . Quando d é igual a 1, o valor de z_t resulta da subtração da observação da série no tempo $t+1$ (X_{t+1}) menos a observação do tempo t (X_t). Para valores de d maiores do que 1, z_t pode ser obtido de forma recursiva, como apresentado na equação 2.14 (Ramkumar, 2019e).

$$\begin{aligned} z_t &= X_{t+1} - X_t \quad \text{se } d = 1 \\ z'_t &= z_{t+1} - z_t = X_{t+2} - X_{t+1} - X_{t+1} - X_t \quad \text{se } d = 2 \\ &\dots \end{aligned} \tag{2.14}$$

A fórmula geral é apresentada na equação 2.15, em que c é uma constante, β_1, \dots, β_p são os parâmetros do modelo AR, Φ_1, \dots, Φ_q são os parâmetros do modelo MA e $\varepsilon_1, \dots, \varepsilon_q$ são os ruídos (Hyndman e Athanasopoulos, 2018). A diferença dessa fórmula é que, ao invés de prever o valor do real da série, ela prediz o valor da série diferenciada. A inversão deste valor diferenciado será apresentada posteriormente.

$$ARIMA(p, d, q): z_t = c + \beta_1 z_{t-1} + \dots + \beta_p z_{t-p} + \phi_1 \varepsilon_{t-1} + \dots + \phi_q \varepsilon_{t-q} + \varepsilon_t \tag{2.15}$$

Para resumir o funcionamento do modelo será utilizado o exemplo proposto por (Ramkumar, 2019e). A ideia central se baseia em um vendedor de âncoras. Assim, o número de âncoras que são vendidas todo mês é dado por uma variável a_t , sendo t o mês atual. Então, o objetivo deste vendedor é fazer uma previsão de quantas âncoras serão vendidas no próximo mês. Para isso, o primeiro passo é gerar um gráfico do número de âncoras vendidas (eixo y) pelo tempo (eixo x), como apresentado à esquerda na Figura 2.19.

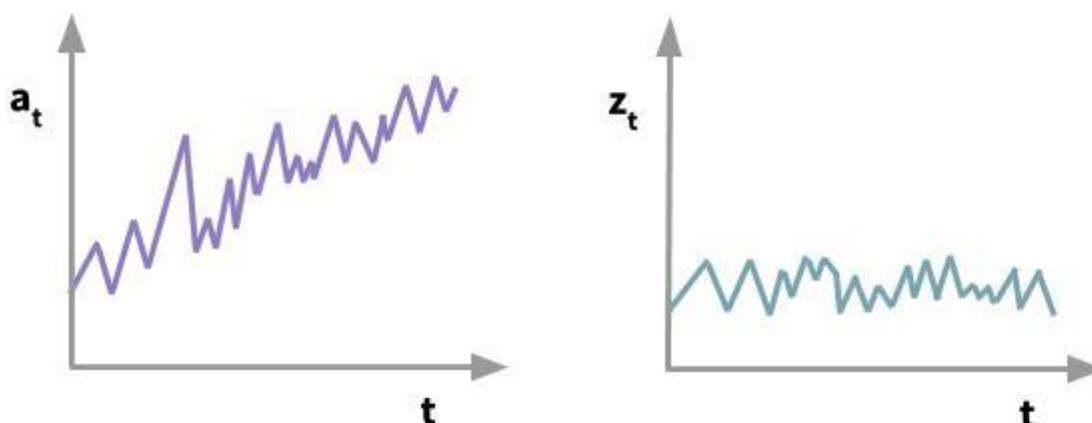


Figura 2.19: Gráfico da venda de âncoras gerado com os dados brutos (esquerda) e gráfico da venda de âncoras diferenciado uma vez (direita).

FONTE: adaptado de Ramkumar (2019e).

Ao analisar as observações em roxo, pode-se perceber que seria inviável utilizar um modelo AR, MA ou ARMA, pois esses modelos necessitam de uma série estacionária, mas no gráfico, a média não é constante e sim crescente. Então, o uso do modelo ARIMA é indicado quando a série parece estacionária, mas não é, pois violou alguma das condições de estacionariedade (Ramkumar, 2019e).

Como a série não é estacionária, ao invés de prever o próximo tempo da própria série, será predita a diferença de um tempo $t+1$ com um tempo anterior t . Assim, será criada uma nova série, chamada z_t , que será $z_t = a_{t+1} - a_t$. O resultado desta transformação é apresentado à direita da Figura 2.19. É possível observar que agora a série tem uma média constante e as outras propriedades de estacionariedade se mantêm.

Então, para realizar a previsão, será utilizado o modelo ARIMA(p, d, q) básico, ou seja, ARIMA(1, 1, 1). Como d é 1, só será aplicada uma transformação na série, então complementando a equação com AR e MA, $z_t = c + \beta_1 z_{t-1} + \Phi_1 \varepsilon_{t-1} + \varepsilon_t$. Como, no fim, o objetivo não é prever a diferença e sim o número real de âncoras que serão vendidas, a_k pode

ser recuperada pela equação 2.16 abaixo, mas deve haver a certeza de que a_0, a_1, \dots, a_l , estão disponíveis (com $k > l$).

$$z_t = \beta_1 z_{t-1} + \phi_1 \epsilon_{t-1} + \epsilon_t$$

$$ARIMA(1, 1, 1): a_k = \sum_{i=1}^{k-l} z_{k-i} + a_l \quad (2.16)$$

2.3.1.6 SARIMA

O modelo SARIMA (*Seasonal AutoRegressive Integrated Moving Average*) é uma extensão do modelo ARIMA que suporta séries temporais univariadas com componentes sazonais. Além de manter os parâmetros do modelo ARIMA, este modelo adiciona 3 novos parâmetros para especificar a auto-regressão (AR), diferenciação (I) e a média móvel (MA) para o componente sazonal da série, além de contar com o parâmetro adicional que indica o período da sazonalidade da série (Brownlee, 2020b). Dessa forma, os parâmetros são (Brownlee, 2020b):

- **p**: ordem da auto-regressão;
- **d**: ordem de diferenciação;
- **q**: ordem da média móvel;
- **P**: ordem da auto-regressão sazonal;
- **D**: ordem da diferenciação sazonal;
- **Q**: ordem da média móvel sazonal;
- **m**: quantidade de tempo para um único período sazonal.

Para exemplificar o funcionamento do modelo será utilizado o conteúdo proposto por (Ramkumar, 2019f). O cenário se resume em um vendedor de *donuts* que registra, a cada três meses, o número de *donuts* que foram vendidos. O histórico desses registros é apresentado na Figura 2.20, na qual é possível observar uma sazonalidade, pois há um padrão em formato de “W” que se repete todo ano.

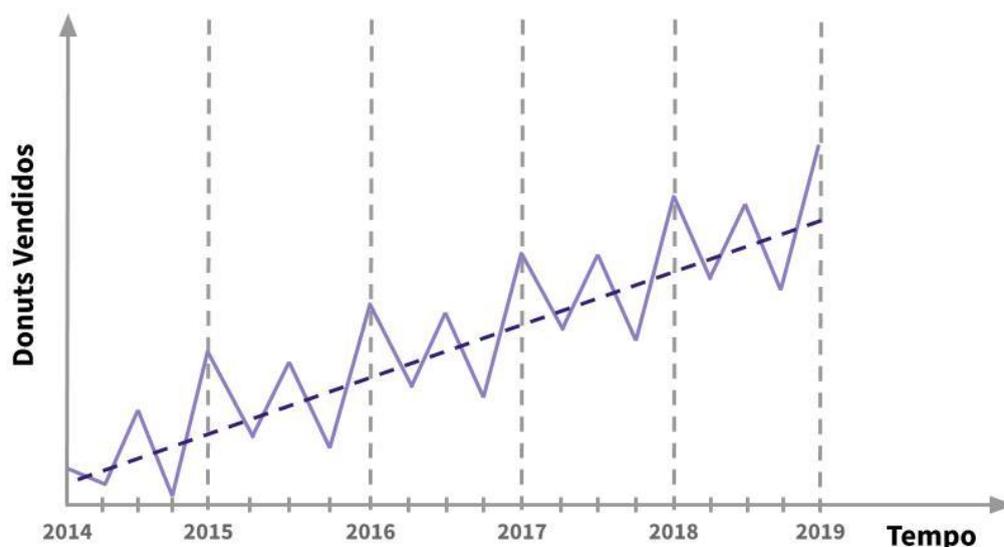


Figura 2.20: Gráfico do exemplo de venda de *donuts* de três em três meses de 2014 a 2019.

FONTE: adaptado de Ramkumar (2019f).

Com esse cenário, é possível justificar o uso do modelo SARIMA com base em três evidências:

- 1) Existe sazonalidade;
- 2) Há indícios que os valores atuais dependem dos valores passados, pois há picos a cada 4 meses, condição necessária para os modelos AR e MA;
- 3) Há uma tendência crescente na série, o que indica a sua não-estacionariedade que pode ser resolvida com a diferenciação.

Como solução, é possível utilizar o modelo $SARIMA(p, d, q)(P, D, Q)_m$ com as ordens iguais a 1, por simplicidade, e m sendo 4, pois o padrão repetitivo da série da Figura 2.20 mostra uma sazonalidade que ocorre a cada 4 meses. Então, usando o operador *Backshift*, o modelo gera a equação 2.17:

$SARIMA(1, 1, 1)(1, 1, 1)_4$:

$$(1 - \phi_1 B)(1 - \phi_1 B^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B)(1 + \theta_1 B^4)\epsilon_t \quad (2.17)$$

Onde:

- $(1 - B)y_t$: é a integração da série, ou seja, a 1ª diferença pois d é igual a 1;
- $(1 - B^4)y_t$: é a integração da série sazonal, que é a 4ª diferença pois m é igual a 4;
- $(1 - \phi_1 B)y_t$: é a parte AR da série;

- $(1 - \phi_1 B^4)y_t$: é a parte AR da série sazonal;
- $(1 + \theta_1 B)\epsilon_t$: é a parte MA da série;
- $(1 + \theta_1 B^4)\epsilon_t$: é a parte MA da série sazonal.

Por questões de compreensibilidade, não será apresentada a equação genérica do modelo. A equação geral segue a lógica da fórmula usada no exemplo apresentado acima, e pode ser encontrada no livro em (Box et al, 2016).

2.3.1.7 ARCH

O modelo ARCH (*AutoRegressive Conditional Heteroskedasticity*) propõe uma maneira de tratar a mudança de volatilidade, ou variância, que causa problemas de previsão ao utilizar modelos semelhantes ao ARIMA (Brownlee, 2018c). Este modelo foi criado no contexto de problemas na área de economia e finanças relacionados com o quanto os investimentos ou estoques aumentam (ou diminuem) em um período de tempo (PennState, 2020).

Especificamente, o método ARCH reconhece a diferença entre a variância incondicional e a condicional, permitindo que a condicional mude ao longo do tempo em função de erros passados (Brownlee, 2018c). Semelhante aos outros modelos, o acrônimo do ARCH também resume o modelo (Ramkumar, 2019g):

- **AR** (*AutoRegressive*): o valor atual pode ser expresso como uma função dos valores anteriores;
- **C** (*Conditional*): informa que a variância é baseada em erros passados;
- **H** (*Heteroskedasticity*): implica que a série mostra uma variância incomum (mudanças na variância).

Para exemplificar o funcionamento do modelo será utilizado o material proposto por (Ramkumar, 2019g). O cenário é composto por um gerente de cinema que quer modelar o número de *tickets* que são vendidos toda semana dentro de 1 ano. Como primeiro passo, o gerente treina o melhor modelo possível para seus dados, seja AR, MA ou ARIMA. Então, após executar esse modelo, os resíduos (erros de cada semana) são agrupados no gráfico abaixo (Figura 2.21).

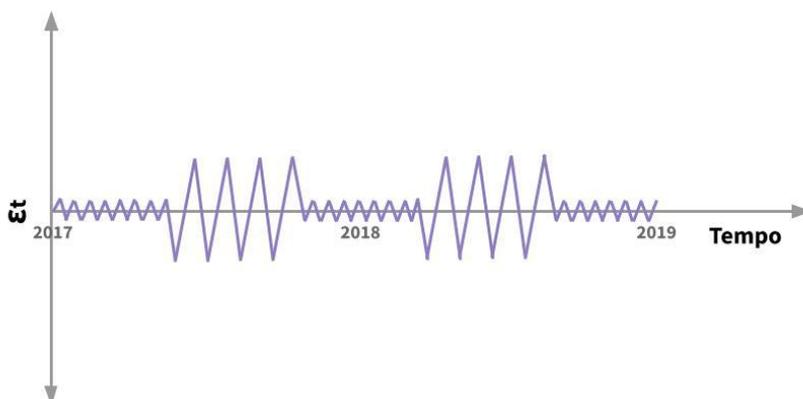


Figura 2.21: Gráfico do resíduo ($\epsilon_t = \text{valor esperado} - \text{valor real}$ de cada semana) do exemplo da venda de *tickets* de cinema.

FONTE: adaptado de Ramkumar (2019g).

No começo de cada ano a volatilidade do erro é bem baixa, mas por volta do meio do ano, ela aumenta consideravelmente. A cada inverno, o modelo escolhido faz uma péssima predição, porque os erros mudam significativamente. Conforme o final do ano se aproxima, a volatilidade volta a ficar baixa, o que indica que a predição foi boa. Esse acontecimento pode ser justificado por vários motivos, como por exemplo, nos meses de inverno são lançados vários filmes, sendo alguns deles muito bons ou muito ruins, e então é muito difícil saber quantos *tickets* serão vendidos pois existem várias variáveis que não foram consideradas. E também pode acontecer de que no começo e no fim de cada ano a quantidade de filmes lançados não seja tão grande ou eles possuam uma qualidade consistente, o que facilita a predição.

Então, observando a Figura 2.21, existe volatilidade (H), a variância dos dados não é fixa sobre o tempo, ela é baseada no ponto do gráfico que está sendo analisado, ou seja, ela é alta em alguns momentos e baixa em outros o que indica que a volatilidade é condicionada à algum fator (C) e, como o padrão de variância se repete, isso indica que a volatilidade depende de seus valores passados (AR). Assim, é possível modelar um ARCH(1), onde ϵ_t é o erro que será modelado, σ_t^2 é a volatilidade de hoje, σ_{t-1}^2 é a volatilidade de ontem, α_1 e α_2 são os coeficientes e ω_t é o ruído (*White Noise*):

$$\begin{aligned} \text{var}(\epsilon_t) &= \sigma_t^2 = \alpha_0 + \alpha_1 \sigma_{t-1}^2 \\ \epsilon_t &= \omega_t \sqrt{\alpha_0 + \alpha_1 \epsilon_{t-1}^2} \end{aligned} \quad (2.18)$$

A equação 2.18 modela o quanto terá de variância hoje (quantificando o quão fora da média os dados estão) usando uma função do quão fora da média os dados estavam no dia,

semana ou mês anterior. Ao aplicar o quadrado na equação da variância é possível chegar em uma equação que modela o erro, como é apresentado na segunda fórmula da equação 2.18. Assim, o ARCH(1) modela o erro de hoje com base no erro de ontem, o ARCH(2) modela o erro de hoje baseado no erro dos dois dias anteriores e assim por diante (Ramkumar, 2019g).

Observando o modelo, a intuição é que se o erro de ontem (seja positivamente ou negativamente) foi alto, o erro de hoje também será alto. Se o erro de ontem foi baixo, a raiz será próxima de zero e o erro de hoje será pequeno. Então, é possível aplicar o quadrado, chegando na equação final (2.19) do ARCH(1):

$$\epsilon_t^2 = \omega_t^2(\alpha_0 + \alpha_1 \epsilon_{t-1}^2) = \omega_t^2 \alpha_0 + \omega_t^2 \alpha_1 \epsilon_{t-1}^2 \quad (2.19)$$

Matematicamente, o modelo ARCH(m) pode ser definido pela equação 2.20 abaixo:

$$ARCH(m) = var(\epsilon_t | \epsilon_{t-1}, \dots, \epsilon_{t-m}) = \sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \dots + \alpha_m \epsilon_{t-m}^2 \quad (2.20)$$

Como nos modelos anteriores, a ordem do ARCH também pode ser encontrada pelo gráfico de Autocorrelação, mas a diferença é que ao invés de usar os dados, o gráfico é gerado a partir dos resíduos (Ramkumar, 2019g).

2.3.1.8 GARCH

O modelo GARCH (*Generalized AutoRegressive Conditional Heteroskedasticity*) é uma generalização do modelo ARCH, em que além dos valores dos erros passados, as variâncias passadas também são utilizadas. Matematicamente, o modelo é definido pela equação 2.21, onde ϵ_t é o erro que será modelado, σ_t^2 é a volatilidade do tempo t e $\alpha_0, \dots, \alpha_m$ e β_0, \dots, β_n são os coeficientes (PennState, 2020).

$$GARCH(m, n) = \sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \dots + \alpha_m \epsilon_{t-m}^2 + \beta_1 \sigma_{t-1}^2 + \dots + \beta_n \sigma_{t-n}^2 \quad (2.21)$$

Um dos problemas do modelo ARCH ocorre quando os dados não têm “*bursty*”, ou seja, os dados inicialmente estão mantendo uma volatilidade constante e, em algum momento, variam bastante e depois voltam ao comportamento anterior, como é mostrado na esquerda da Figura 2.22. Mas se os dados, ao variarem bastante, se manterem altos por um tempo e depois caírem e se manterem baixos, o ARCH começa a ter problemas para fazer as previsões (Ramkumar, 2019h). Um exemplo disso é apresentado à direita da Figura 2.22.

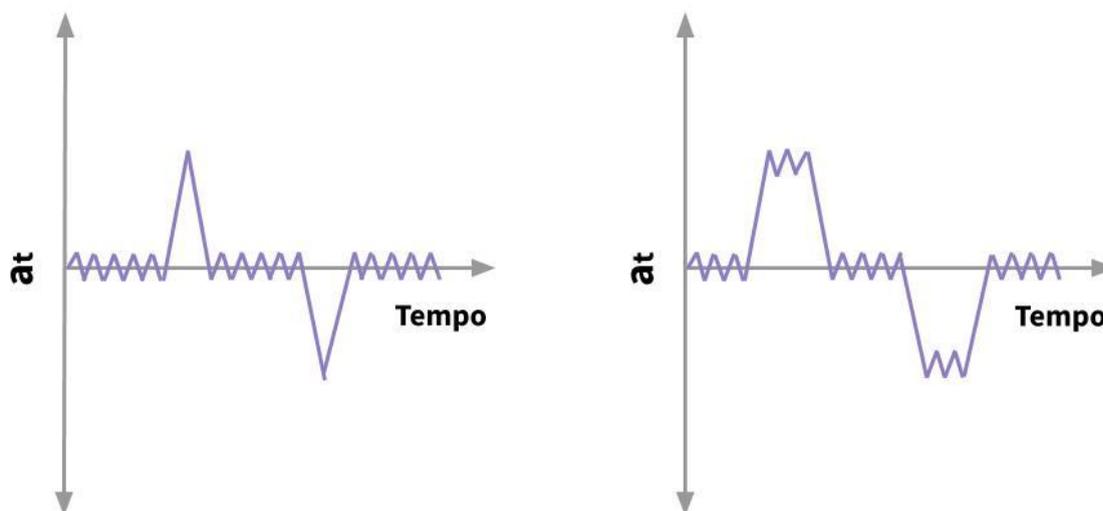


Figura 2.22: Gráfico de dados com “bursty”, em que os dados mantêm uma volatilidade constante, variam bastante e voltam a ter a volatilidade constante (esquerda) e gráfico em que, quando a volatilidade muda bastante, ela se mantém por um tempo antes de voltar a ser constante (direita).

FONTE: adaptado de Ramkumar (2019h).

Ao analisar a equação 2.21, é possível notar que o modelo GARCH é afetado pelo valor da série de ontem mas também pela volatilidade de ontem. Assim, se a série temporal foi bem alta ontem e a volatilidade foi alta também, é provável que seja bem volátil hoje. Com σ_{t-1} é possível propagar a volatilidade pelo tempo ao invés de descartar depois de um ou dois períodos (Ramkumar, 2019h). Esse modelo é representado por GARCH(p, q) porque ele leva em conta p períodos de tempo da série e q períodos da volatilidade (Ramkumar, 2019h).

2.3.1.9 EXPONENTIAL SMOOTHING

A ideia do modelo *Exponential Smoothing* é assumir que o futuro será parecido com o passado recente. A lógica do modelo é bem parecida com o modelo MA, mas há o acréscimo de um peso decrescente para cada observação. Dessa forma, menos importância é dada às observações à medida que há um afastamento do presente (Peixeiro, 2019b).

Este modelo pode assumir vários níveis. O nível mais básico é o *Simple Exponential Smoothing* que é apresentado na equação 2.22, onde α é o fator de suavização que pode assumir valores entre 0 e 1. Esse parâmetro determina o quão rápido o peso diminui para as observações prévias.

$$y = \alpha x_t + (1 - \alpha)y_{t-1} \quad \text{sendo } t > 0 \quad (2.22)$$

Os outros níveis se baseiam na ideia de aplicar esta técnica recursivamente, adicionando outros fatores de suavização. Um deles é o *Double Exponential Smoothing*, que costuma ser utilizado quando há uma tendência na série temporal, pois permite ajustar a predição atual com base na tendência do período anterior, ajudando a eliminar defasagens. Já o *Triple Exponential Smoothing* estende o *Double Exponential Smoothing*, adicionando um fator de suavização sazonal, permitindo ajustar o valor da predição de acordo com a tendência e com a sazonalidade dos períodos anteriores. As equações dessas duas últimas variações podem ser encontradas em (Peixeiro, 2019b).

2.3.2 MODELOS MISTOS

Nesta seção serão apresentados os modelos que podem ser utilizados tanto no modo Univariado quanto no modo Multivariado, que são os modelos das áreas de Aprendizado de Máquina e Aprendizado Profundo. Dessa forma, uma ou mais características podem ser dadas como entrada e o modelo irá gerar uma saída correspondente. Os modelos que serão descritos nas subseções abaixo são: Regressão Linear, Regressão Polinomial, Árvore de Regressão, *Bagging*, Florestas Aleatórias, *Gradient Boosting*, *Perceptron*, MLP, LSTM e GRU.

2.3.2.1 REGRESSÃO LINEAR

O modelo de Regressão Linear (*Linear Regression*) é uma representação matemática da Análise de Regressão, que estuda a relação entre uma variável chamada de “dependente” e outras variáveis chamadas de “independentes”. Esse modelo é conhecido como Modelo de Regressão Linear Simples (MRLS), em que se define uma relação linear entre a variável dependente e uma variável independente, como é apresentado na Figura 2.23. Se forem incorporadas diversas variáveis independentes no modelo, ele recebe o nome de Modelo de Regressão Linear Múltipla (Rodrigues, 2015).

Na figura abaixo, Y_i é a variável dependente, X_i é a variável independente, β_1 é o coeficiente angular e β_0 é a interceptação da reta com o eixo y, chamado de coeficiente linear. No gráfico ao lado é apresentada a linha reta, obtida pela Regressão Linear, que melhor se adequa à distribuição dos dados. Essa técnica é simples, rápida e útil quando a relação entre o *target* e as variáveis independentes não é complexa, ou quando não há dados suficientes disponíveis (Tyagi, 2020b).

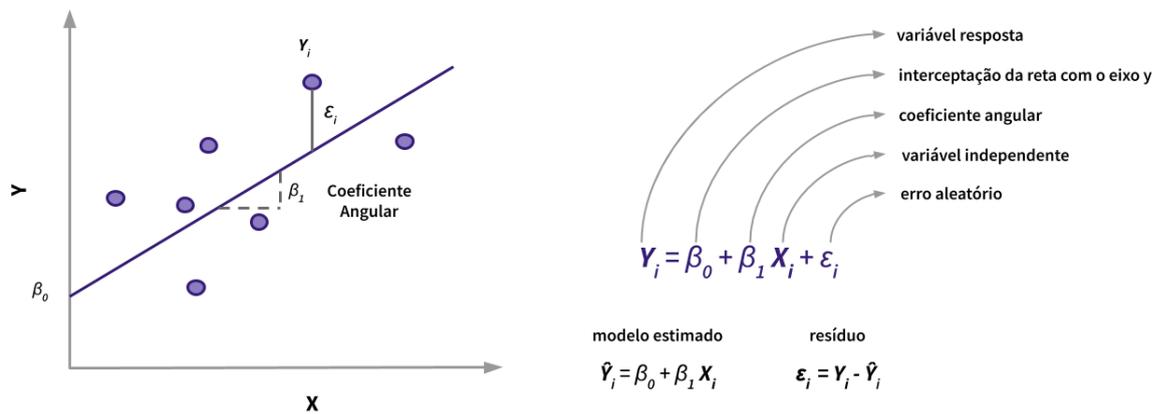


Figura 2.23: Gráfico do exemplo de reta encontrada pela Regressão Linear (esquerda) e fórmulas utilizadas para construir o modelo (direita).

FONTE: adaptado de Tyagi (2020b).

2.3.2.2 REGRESSÃO POLINOMIAL

A Regressão Polinomial (*Polynomial Regression*) é uma forma de Análise de Regressão na qual a relação entre as variáveis independentes e dependentes são modeladas em um polinômio de n -ésimo grau. Dessa forma, ela é considerada um tipo especial da Regressão Linear, pois uma equação polinomial é ajustada aos dados com uma relação curvilínea entre as variáveis dependentes e independentes, não necessitando que essas variáveis tenham uma relação linear no conjunto de dados (Abhigyan, 2020), como é apresentado pela comparação da Figura 2.24.

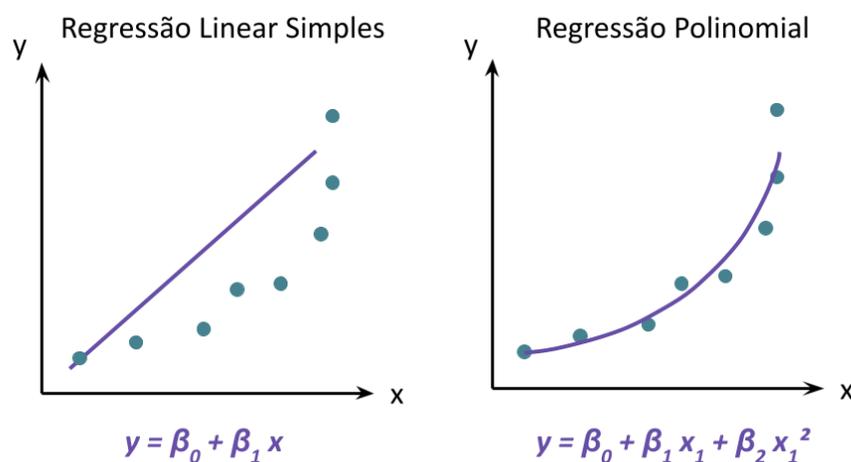


Figura 2.24: Comparação entre um *underfitting* obtido por um modelo Regressão Linear Simples (esquerda) com um bom ajuste obtido pela Regressão Polinomial (direita).

FONTE: adaptado de Abhigyan (2020).

À medida que o grau do polinômio é aumentado, o desempenho do modelo tende a aumentar, mas o aumento em excesso pode gerar *overfitting* e um alto custo computacional para treinar o modelo. Matematicamente, a Regressão Polinomial pode ser entendida pela evolução das fórmulas na equação 2.23 abaixo, sendo a primeira uma Regressão Linear Simples, a segunda uma Regressão Linear Múltipla e, por fim, a Regressão Polinomial Simples (Abhigyan, 2020).

$$\begin{aligned}
 \text{Regressão Linear Simples: } y &= \beta_0 + \beta_1 x_1 \\
 \text{Regressão Linear Múltipla: } y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \\
 \text{Regressão Polinomial Simples: } y &= \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \dots + \beta_n x_1^n \quad (2.23)
 \end{aligned}$$

Para uma Regressão Polinomial Múltipla, as equações podem ser obtidas pela matriz da equação 2.24. Na maioria dos casos, a função de custo deste tipo de regressão é baseada no Erro Quadrado Médio (*MSE - Mean Squared Error*) e a Descida de Gradiente (*Gradient Descent*) é utilizada como otimizador para os parâmetros que minimizam a função de custo (Abhigyan, 2020).

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \quad (2.24)$$

2.3.2.3 ÁRVORE DE REGRESSÃO

O modelo de Árvore de Decisão para o problema de Regressão, também chamado de *Regression Tree*, faz parte do conjunto de algoritmos de Aprendizado de Máquina. A representação deste modelo é uma árvore, onde o nó raiz e os nós internos representam as variáveis (X) e o ponto de divisão que os valores de uma variável passam. Os nós folhas representam a variável de saída (y) e são eles que definem a predição (Seif, 2018).

A intuição da construção da árvore se resume em utilizar o conjunto de instâncias de treinamento como entrada, decidir quais atributos do conjunto são melhores para dividir as decisões, realizar a separação do conjunto de dados de acordo com as variáveis e repetir o processo até que todas as instâncias de treinamento sejam categorizadas (Seif, 2018).

Para exemplificar o funcionamento do modelo, será utilizado o exemplo apresentado em (Dubey, 2020). O problema é centrado em um jogador de golfe que gostaria de prever quantas horas de golfe serão jogadas no dia seguinte. Os dados disponíveis para criar o modelo são apresentados à direita na Figura 2.25.

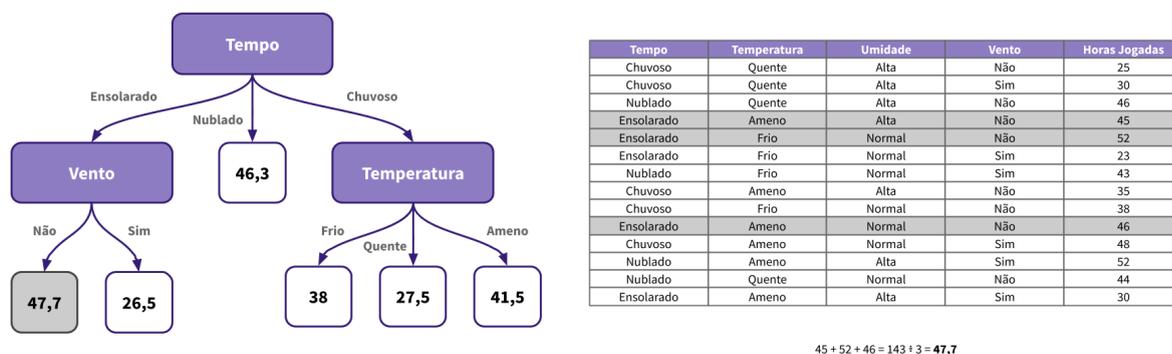


Figura 2.25: Representação da Árvore de Regressão da previsão de horas de golfe que serão jogadas no dia seguinte (esquerda) tabela com os dados disponíveis para a construção da árvore (direita).

FONTE: adaptado de Dubey (2020).

O primeiro passo para a construção da árvore é descobrir qual nó raiz, entre as quatro variáveis (Tempo, Temperatura, Umidade e Vento), faz uma boa divisão inicial dos dados. Para isso, pode-se usar o método em que é necessário encontrar o atributo que produza a maior redução do desvio padrão (*SDR - Standard Deviation Reduction*), que leva em consideração o desvio padrão do *target* (y) e do atributo (X) em relação ao *target*. Considerando a variável Tempo, seu SDR pode ser obtido da seguinte forma (Figura 2.26):

Calcular a média de todas as horas jogadas (variável Y)

$$\overline{\text{horas_jogadas}} = \frac{\sum_{i=1}^n \text{horas_jogadas}_i}{n} = \frac{25 + 30 + \dots + 44 + 30}{14} = 39.8$$

Calcular o desvio padrão das horas jogadas

$$S(\text{horas_jogadas}) = \sqrt{\frac{\sum_{i=1}^n (\text{horas_jogadas}_i - \overline{\text{horas_jogadas}})^2}{n}} = \sqrt{\frac{(25 - 39.8)^2 + (30 - 39.8)^2 + \dots + (30 - 39.8)^2}{14}} = 9.32$$

Calcular o desvio padrão para cada valor da variável que está sendo observada

$$\begin{aligned} \overline{\text{tempo_chuvoso}} &= \frac{25 + 30 + 35 + 38 + 48}{5} = 35.2 \\ S(\text{Tempo} = \text{Chuvoso}) &= \sqrt{\frac{(25 - 35.2)^2 + \dots + (48 - 35.2)^2}{5}} = 7.78 \\ S(\text{Tempo} = \text{Nublado}) &= 3.49 \\ S(\text{Tempo} = \text{Ensolarado}) &= 10.87 \end{aligned}$$

Calcular o desvio padrão do target em relação a variável que está sendo observada

$$\begin{aligned} S(\text{horas_jogadas}, \text{tempo}) &= \sum_{c \in \text{tempo}} P(c)S(c) \\ S(\text{horas_jogadas}, \text{tempo}) &= P(\text{tempo} = \text{ensolarado}) \times S(\text{tempo} = \text{ensolarado}) + \\ &\quad P(\text{tempo} = \text{nublado}) \times S(\text{tempo} = \text{nublado}) + \\ &\quad P(\text{tempo} = \text{chuvoso}) \times S(\text{tempo} = \text{chuvoso}) \\ S(\text{horas_jogadas}, \text{tempo}) &= \frac{5}{14} \times 10.87 + \frac{4}{14} \times 3.49 + \frac{5}{14} \times 7.78 = 7.66 \end{aligned}$$

Calcular o SDR em relação às variáveis que estão sendo observadas

$$SDR(\text{horas_jogadas}, \text{tempo}) = S(\text{horas_jogadas}) - S(\text{horas_jogadas}, \text{tempo}) = 9.32 - 7.66 = 1.66$$

Figura 2.26: Sequência de passos para o cálculo do SDR.

FONTE: adaptado de Dubey (2020).

No último passo dessa figura, o SDR também deveria ser calculado para as outras variáveis como Temperatura, Umidade e Vento, mas é considerado que o SDR da variável Tempo obteve o maior valor, por simplicidade. Com esse resultado, a variável Tempo se torna o nó raiz da árvore, gerando 3 novos ramos para dividir os dados: Ensolarado, Nublado e Chuvoso. Assim, são escolhidos todos os dados em que a variável Tempo tem o valor de “Ensolarado”, por exemplo, e, dentre esses dados, o processo anterior é repetido. Essa

sequência de passos é realizada até que um critério de parada seja atingido, como a profundidade da árvore, número de nós folhas (Campos, 2017) ou um determinado Coeficiente de Variação (Dubey, 2020). A árvore completa pode ser observada à esquerda na Figura 2.25.

Após a construção da árvore, para realizar uma predição, as características da entrada são testadas e um caminho é percorrido pela árvore até chegar em um nó folha. Diferentemente da Árvore de Decisão para classificação, no cenário de regressão, o valor de predição é a média de todos os *targets* de treinamento que detêm características correspondentes àquele caminho dentro da árvore. Por exemplo, ao construir a árvore do exemplo anterior, quando uma entrada tem a variável de Tempo como sendo “Ensolarado”, só é necessário testar a característica de Vento para ter a predição, tanto Temperatura quanto Umidade não são necessárias para dividir os dados. Assim, se a entrada tem Tempo como sendo “Ensolarado” e não tem Vento, é feita uma média com os valores de *target* dos exemplos de treinamento com Umidade alta e normal e com Temperatura quente, amena e fria, como é evidenciado em cinza na tabela dos dados.

Esse modelo tem a vantagem de conseguir aprender relacionamentos complexos e não lineares, além de ser simples de interpretar e entender. Os pontos negativos se concentram na possibilidade da árvore estar sujeita a grandes ajustes excessivos, pois árvores completas podem ser muito grandes e conter estruturas desnecessárias, além do alto consumo de tempo e de memória para conjuntos de dados volumosos (Seif, 2018).

2.3.2.4 BAGGING

O modelo *Bootstrap Aggregation*, ou *Bagging*, é um método de conjunto que combina as respostas de vários algoritmos de Aprendizado de Máquina para realizar previsões com maior precisão do que um modelo individual. Os passos do funcionamento do *Bagging* são (Brownlee, 2016c):

- 1) Criar muitas sub-amostras aleatórias do conjunto de dados, com substituição;
- 2) Treinar um modelo em cada uma (por exemplo, uma árvore de regressão para cada amostra);
- 3) Dado um novo conjunto de dados, calcular a previsão média do conjunto de modelos.

Alguns modelos individuais podem ter uma alta variância, como as Árvores de Regressão. Nelas, a estrutura da árvore é sensível aos dados em que foi treinada, pois se os

dados forem alterados a estrutura da árvore resultante pode ser bem diferente, o que pode gerar previsões diferentes também. Então, em casos em que há a necessidade de diminuir a variância, o *Bagging* é utilizado (Brownlee, 2016c).

Por exemplo, se em um problema de previsão existissem cinco árvores de regressão, que previssem a resposta como 1, 1, 2, 1 e 2, ao realizar o *Bagging* com elas, a previsão seria $(1 + 1 + 2 + 1 + 2) \div 5$ que resulta em 1. Assim, há uma menor preocupação com as árvores individuais em questões como *overfitting*, profundidade e poda. Os únicos parâmetros necessários para utilizar esse modelo são o número de amostras e o número de árvores a serem incluídas (Brownlee, 2016c).

Na Figura 2.27, é possível observar a diferença entre o *Bagging* e um modelo individual. Na primeira imagem, o modelo *Single* treina somente um modelo no conjunto de dados. Já na segunda imagem, o *Bagging* treina vários modelos em amostras diferentes do conjunto de dados de forma paralela e depois combina os resultados. Na última imagem, o modelo de *Boosting*, que será apresentado posteriormente, também treina vários modelos, mas de forma sequencial.

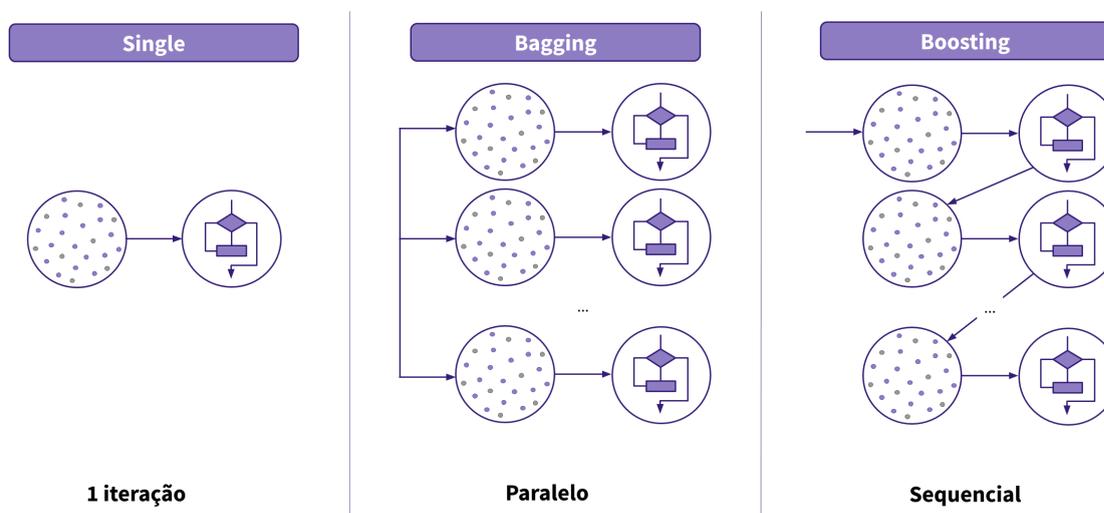


Figura 2.27: Comparação entre os métodos de combinação de modelos, inicialmente o caso de modelos individuais (*Single*), combinação paralela (*Bagging*) e combinação sequencial (*Boosting*).

FONTE: adaptado de Brownlee (2016c).

2.3.2.5 FLORESTAS ALEATÓRIAS

O modelo de Florestas Aleatórias (*Random Forests*) é apresentado como uma melhoria do *Bagging* de Árvores de Regressão. No método explicado anteriormente, as árvores escolhem em qual variável dividir os dados usando um algoritmo guloso, que tenta minimizar o erro. Dessa forma, as árvores construídas podem ter muitas semelhanças estruturais

implicando em uma alta correlação em suas previsões. Ao utilizar a técnica de combinação de modelos de predição, o melhor cenário é aquele em que as previsões dos sub-modelos são fracamente correlacionadas, pois assim cada sub-modelo consegue avaliar os dados de uma perspectiva diferente (Brownlee, 2016c).

O método de Florestas Aleatórias altera esse algoritmo visando diminuir a correlação entre essas árvores dos sub-modelos. Ao invés de analisar todas as características para selecionar um ponto ideal de divisão dos dados, esse modelo altera o algoritmo de aprendizado de forma que só sejam consideradas as características pertencentes a uma amostra aleatória de variáveis. O número de características que podem ser pesquisadas em cada ponto de divisão (m) deve ser especificado como um parâmetro para o algoritmo. Geralmente, bons valores para m são obtidos pela raiz quadrada do número de características dos dados ou da divisão desse número por 3 (Brownlee, 2016c).

A Figura 2.28 exemplifica o processo de predição desse modelo. Inicialmente, a entrada de teste é fornecida a cada árvore e, em cada uma delas, um caminho é percorrido e uma predição encontrada. Todas as predições obtidas são agrupadas em uma média que será a resposta final do modelo. O processo é quase o mesmo de um *Bagging* de Árvores de Regressão, mas a diferença fundamental é que, nas Florestas Aleatórias, apenas um subconjunto de características é selecionado aleatoriamente do total e a melhor característica do subconjunto é usada para dividir cada nó em uma árvore, ao contrário do *Bagging*, onde todas as características são consideradas para a divisão do nó.

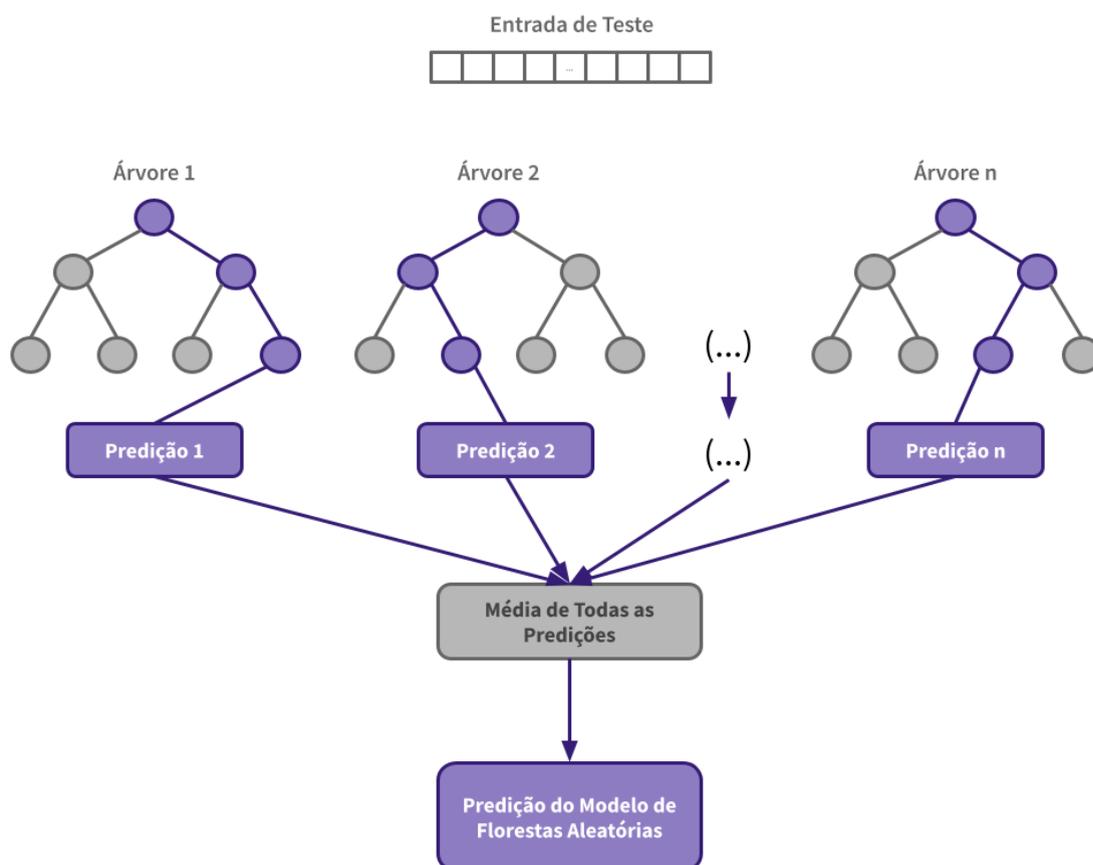


Figura 2.28: Exemplo do funcionamento de previsão feita pelo modelo de Florestas Aleatórias.

FONTE: adaptado de Miller (2018).

2.3.2.6 GRADIENT BOOSTING

Antes de descrever o funcionamento do algoritmo de *Gradient Boosting*, é necessário apresentar o conceito de *Boosting*, que é um método para converter preditores fracos em preditores fortes (Singh, 2018). Nessa técnica, os preditores não são utilizados de forma independente e paralela, mas sim sequencialmente, como mostrado na última imagem da Figura 2.27. Dessa forma, os preditores subsequentes podem aprender com os erros dos preditores anteriores, resultando em menos tempo para chegar perto das previsões reais (Grover, 2017).

O *Gradient Boosting* é uma técnica de *Boosting* pensada para problemas de classificação e regressão, que produz um modelo de previsão na forma de um conjunto de modelos de previsão fracos, como Árvores de Decisão, kNN, *Perceptrons*, entre outros. Analisando da perspectiva de definições matemáticas, o algoritmo visa minimizar uma função de perda e atualizar as novas previsões com base no erro usando o SGD (*Stochastic Gradient*

Descent). Como exemplo, a função de perda poderia ser a equação 2.25, onde y_i é o i -ésimo target e y_i^p é a i -ésima predição (Grover, 2017).

$$MSE: L(y_i, y_i^p) = \sum (y_i - y_i^p)^2 \quad (2.25)$$

Para atualizar as predições com o SGD as fórmulas da equação 2.26 são utilizadas, onde α é a taxa de aprendizado e o conteúdo do somatório é a soma dos resíduos. Assim, as previsões são atualizadas de modo que a soma dos resíduos seja próxima de zero e os valores preditos sejam suficientemente próximos dos valores reais (Grover, 2017).

$$y_i^p = y_i^p + \alpha * \frac{\delta}{\delta y_i^p} \sum (y_i - y_i^p)^2$$

$$y_i^p = y_i^p - \alpha * 2 * \sum (y_i - y_i^p) \quad (2.26)$$

Para exemplificar essa atualização sequencial das previsões, é apresentada abaixo a visualização disponibilizada por (Grover, 2017). Observando a Figura 2.29, os pontos roxos (esquerda) são as saídas (y) em relação às entradas (x), a linha preta (esquerda) mostra os valores preditos pela Árvore de Regressão que está dentro do modelo *Gradient Boosting*, os pontos azuis (direita) mostram os resíduos (y) em relação a entrada (x) e as iterações indicam a ordem sequencial de ajuste do modelo. Analisando todas as iterações, é possível notar que, na vigésima iteração, os resíduos estão aleatoriamente distribuídos ao redor de 0 e as previsões estão bem perto dos valores verdadeiros, o que indica que é um bom ponto de parada para o modelo. Já na iteração 50, os resíduos continuam baixos e aleatórios, mas o modelo faz previsões bem mais complexas, o que pode indicar a ocorrência de *overfitting*.

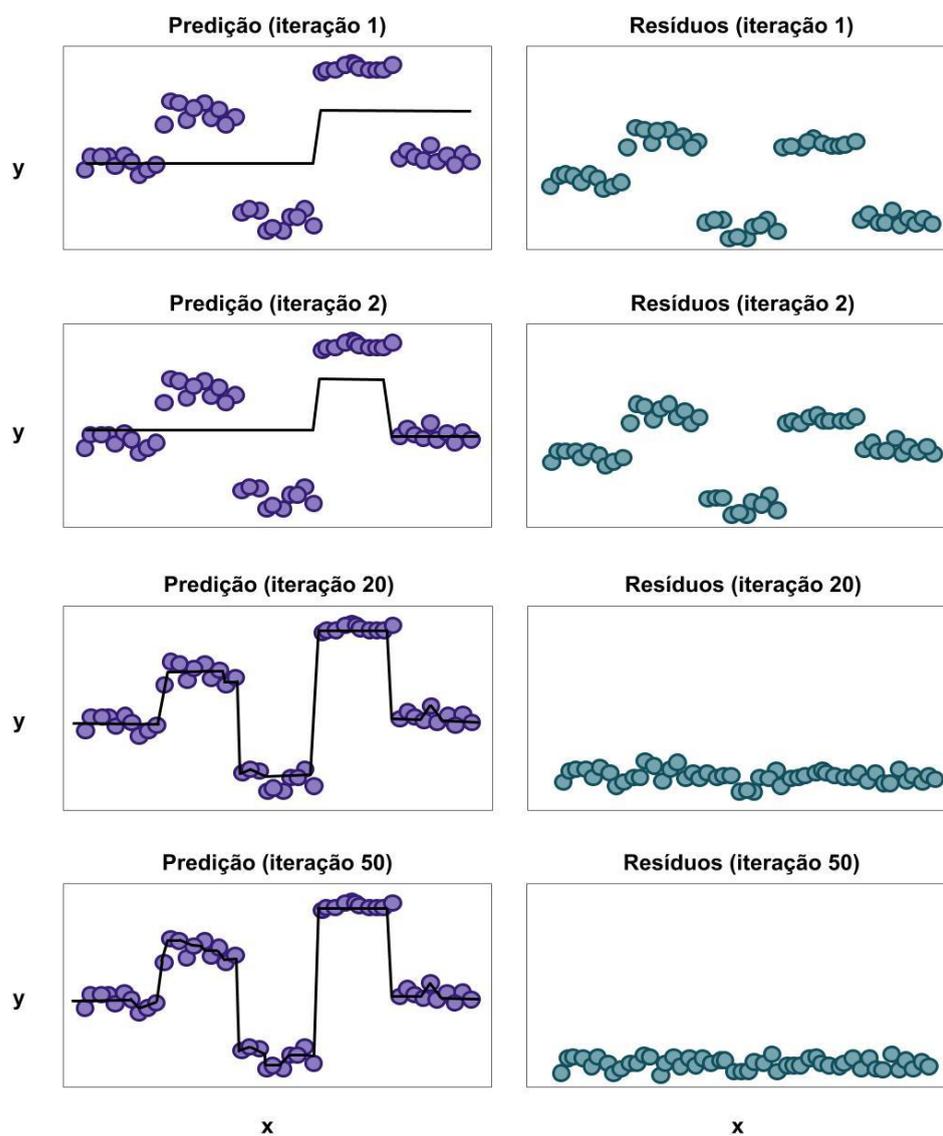


Figura 2.29: Exemplo dos ajustes sequenciais feitos pelo modelo de *Gradient Boosting* utilizando Árvore de Regressão.

FONTE: adaptado de Grover (2017).

2.3.2.7 PERCEPTRON

O modelo *Perceptron* foi desenvolvido nas décadas de 1950 e 1960 pelo cientista Frank Rosenblatt, inspirado em trabalhos anteriores de Warren McCulloch e Walter Pitts. Em resumo, esse modelo é um classificador linear que recebe várias entradas (x_1, x_2, \dots, x_n) e produz uma única saída (Data Science Academy, 2019). A visualização gráfica do modelo pode ser conferida na Figura 2.30.

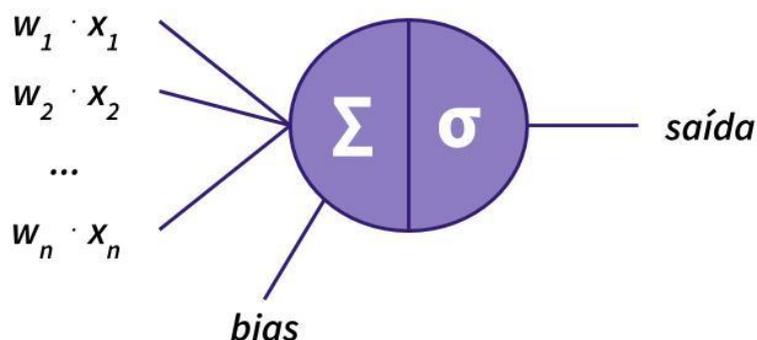


Figura 2.30: Representação gráfica do *Perceptron*.

FONTE: adaptado de Hill (2018).

Rosenblatt introduziu a ideia de pesos, que expressam a importância das respectivas entradas para a saída. Assim, para definir a entrada do modelo é feita uma soma ponderada entre as entradas (x_1, x_2, \dots, x_n) e esses pesos (w_1, w_2, \dots, w_n) (Data Science Academy, 2019). Além dos pesos, o *Perceptron* também pode contar com a ideia do *Bias*, também conhecido como viés, que é um valor escalar adicionado à soma ponderada realizada anteriormente. Juntando o *bias* com a soma ponderada obtém-se a entrada total do *Perceptron*, como é apresentado pelas duas primeiras fórmulas da equação 2.27 (Hill, 2018).

Após receber a entrada, o modelo aplica uma função de ativação, que pode ligar ou não o *Perceptron*, no caso de um problema de classificação binária, ou normalizar o valor de saída se for um problema de regressão. Em resumo, a função de ativação é usada para mapear a entrada para os valores necessários para o problema, como $(0, 1)$ ou $(-100, 100)$ (Sharma, 2017). Geralmente, a função de ativação é definida como σ , o que pode ser observado na última fórmula da equação 2.27 (Hill, 2018).

$$\begin{aligned} \text{Entrada: } & \sum_{i=0}^n w_i \cdot x_i \\ \text{Entrada Total: } z &= \left(\sum_{i=0}^n w_i \cdot x_i \right) + \text{bias} \\ \text{Saída: } & \sigma(z) \end{aligned} \tag{2.27}$$

Com a introdução dos pesos e do *bias*, se torna necessário descobrir uma forma de encontrar seus valores, de forma automática, durante o treinamento. Para isso, além do processo de *Feed Forward* para conseguir a saída, também é feito o processo de *Backpropagation*, geralmente utilizando o algoritmo de *Stochastic Gradient Descent* (SGD)

para atualizar esses parâmetros. O SGD é calculado na função de custo, que também é fundamental para esse modelo (Hill, 2018). As fórmulas para atualizar os pesos e os *bias* podem ser encontradas na equação 2.28, onde W^+ é o conjunto dos novos pesos (já atualizados), W é o conjunto de pesos utilizados para gerar as saídas, ∇C é o gradiente dos custos, b^+ é o novo *bias*, b é o *bias* antigo, ∇b é o gradiente do *bias* e η é a constante da taxa de aprendizado (Hill, 2018).

$$Pesos : \begin{matrix} W^+ \\ \begin{bmatrix} w_1^+ \\ w_2^+ \\ \vdots \\ w_n \end{bmatrix} \end{matrix} = \begin{matrix} W \\ \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \end{matrix} - \eta \begin{matrix} \nabla C \\ \begin{bmatrix} \frac{\delta C}{\delta w_1} \\ \frac{\delta C}{\delta w_2} \\ \vdots \\ \frac{\delta C}{\delta w_n} \end{bmatrix} \end{matrix}$$

$$Bias : b^+ = b - \eta \frac{\nabla C}{\nabla b} \quad (2.28)$$

A ideia do *Perceptron* traz uma certa expressividade para o modelo, pois com os conceitos de pesos e *bias* é possível escalar e traduzir os valores de entrada. Considerando o cenário em que se deseja aproximar uma função de um conjunto de pontos, ao escalar a entrada, ou seja, fazer a soma ponderada com o conjunto de pesos, é possível comprimir ou expandir a função pelo eixo x . Já ao traduzir a entrada, isto é, adicionar o *bias*, é possível mover a função pelo eixo x (Hill, 2018). Essa ideia pode ser visualizada na Figura 2.31, sendo $\sigma(x)$ a função sigmóide.

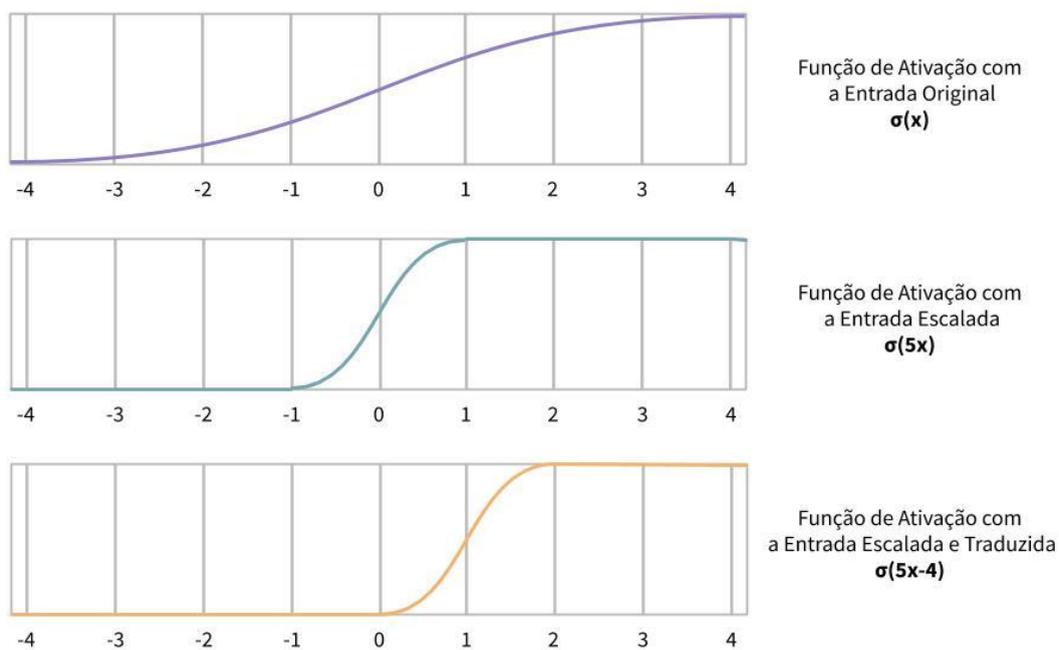


Figura 2.31: Visualização da entrada escala e traduzida.
 FONTE: adaptado de Hill (2018).

2.3.2.8 MLP

O modelo *Perceptron* Multicamadas (PMC ou MLP - *Multi Layer Perceptron*) é uma extensão do modelo *Perceptron* com uma ou mais camadas de neurônios, e cada camada pode ser composta por vários neurônios. A arquitetura geral do modelo é apresentada na Figura 2.32. Os círculos roxos indicam as entradas que foram passadas para a rede, os círculos brancos são as camadas ocultas (que estão entre a entrada e a saída) e os círculos cinzas representam a camada de saída. As camadas ocultas recebem esse nome porque não é possível prever a saída desejada nas camadas intermediárias (Moreira, 2018).

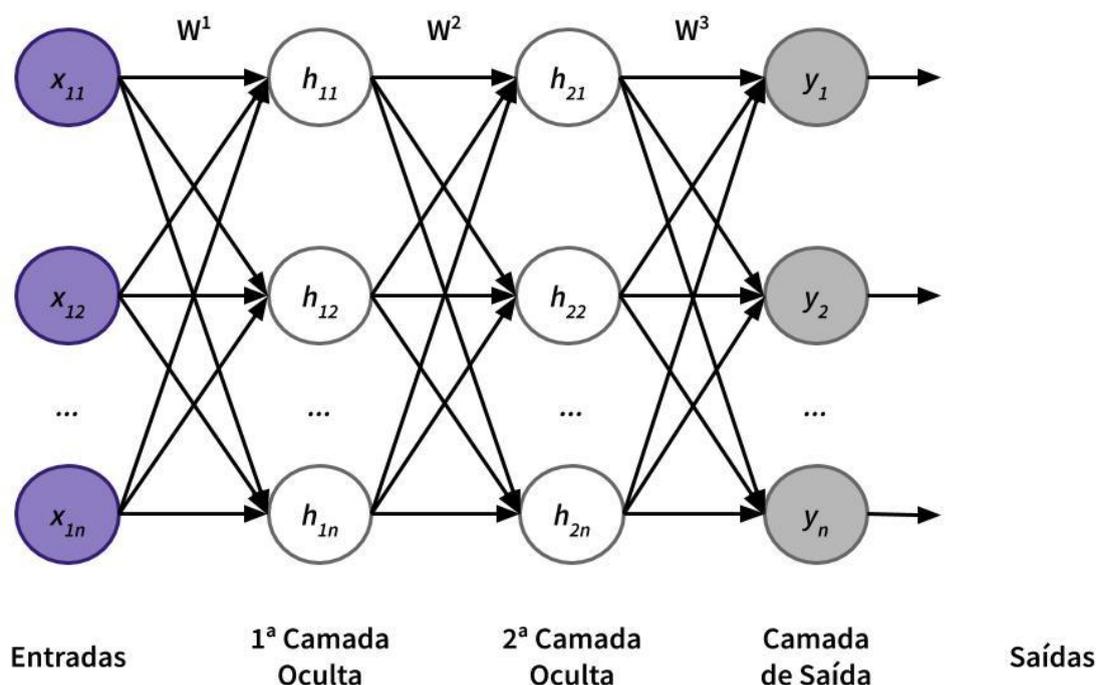


Figura 2.32: Arquitetura do modelo MLP.
 FONTE: adaptado de Moreira (2018).

O treinamento da rede MLP geralmente é feito utilizando o algoritmo de Retropropagação (*Backpropagation*). Para dar uma maior intuição ao funcionamento do aprendizado do modelo, o *Backpropagation* pode ser resumido em 4 passos (Moreira, 2018):

- 1) **Inicialização:** nesse passo são atribuídos valores aleatórios para os pesos iniciais e *bias*. Geralmente esses valores são pequenos e uniformemente distribuídos.
- 2) **Ativação:** são calculados os valores dos neurônios das camadas ocultas e da camada de saída, usando o método de *Feed Forward* apresentado na seção do *Perceptron*.
- 3) **Treinamento dos pesos:** nessa etapa é calculado o erro da rede pela função de custo, e são feitos os cálculos de derivadas e atualização dos pesos como mostrado pela equação 2.28. Com os pesos e *bias* do fim da rede atualizados, o erro e as derivadas da camada anterior são calculados e os parâmetros atualizados. Esse processo ocorre recursivamente até chegar na primeira camada oculta.
- 4) **Iteração:** o método é repetido a partir do passo 2 até que o critério de parada pré-definido seja satisfeito.

A vantagem de usar gradientes é que eles ajustam os pesos que estão precisando de uma mudança a mais, e os pesos com menos necessidade passam por alterações menores (Hill, 2018). Atualmente, existem vários algoritmos para realizar o aprendizado da rede, como SGD, RMSprop, Adam, Adadelta, Adagrad, Nadam, entre outros (Keras, 2020). Detalhes mais específicos sobre os cálculos dos gradientes podem ser encontrados em (Hill, 2018).

2.3.2.9 LSTM

Antes de descrever as redes *Long Short-Term Memory* (LSTM), é necessário mencionar as Redes Neurais Recorrentes (RNNs). Diferentemente das redes básicas como a MLP, além de lembrarem o que aprenderam durante o treinamento, as redes recorrentes tomam decisões do presente considerando o que aconteceu no passado (Venkatachalam, 2019). As RNNs podem utilizar um ou mais vetores de entrada e produzir um ou mais vetores de saída e os resultados são influenciados não apenas pelos pesos aplicados nas entradas, mas também por um vetor de Estado Oculto, que representa o contexto construído pelas entradas e saídas anteriores. Portanto, a mesma entrada pode produzir uma saída diferente dependendo das entradas anteriores de dados (Venkatachalam, 2019).

No diagrama abaixo (Figura 2.33), a rede neural *A* olha para alguma entrada x_t e produz um valor h_t . Um *loop* permite que as informações sejam transmitidas de uma etapa da rede para a próxima. Essa rede pode ser visualizada como várias cópias da mesma rede, cada uma passando uma mensagem a um sucessor, como se o *loop* fosse desenrolado (Olah, 2015).

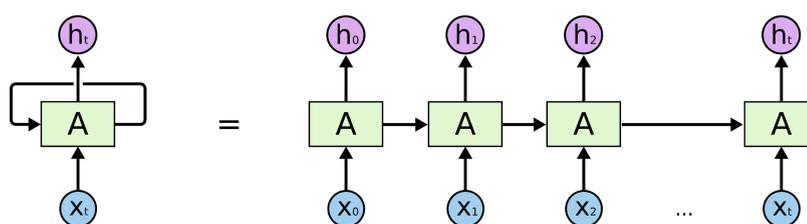


Figura 2.33: Exemplo do *loop* de uma rede recorrente (esquerda) e a forma desenrolada do *loop* (direita).

FONTE: Olah (2015).

Essa arquitetura de rede apresenta a vantagem de conectar informações anteriores à previsão que está sendo feita no presente, mas essa característica tem suas limitações. Para exemplificar, será considerado o cenário de um modelo de linguagem natural que tenta prever a próxima palavra com base nas anteriores. Se o modelo tenta prever a última palavra da frase “as nuvens estão no céu”, não é necessário nenhum contexto adicional além do início da frase. Nesse caso, a RNN tem um bom desempenho pois a lacuna entre as informações

relevantes e a palavra que se quer prever é pequena. Mas se a rede tentar prever a última palavra no texto “Eu cresci na França Eu falo francês”, só é possível prever que a próxima palavra é o nome de um idioma se baseando nas informações recentes. Nesse caso, a lacuna entre as informações relevantes e o ponto onde elas são necessárias é muito grande, o que prejudica o desempenho da RNN. A Figura 2.34 apresenta a relação da dependência de informações mais recentes em comparação com a dependência de informações mais distantes dentro da arquitetura (Olah, 2015).

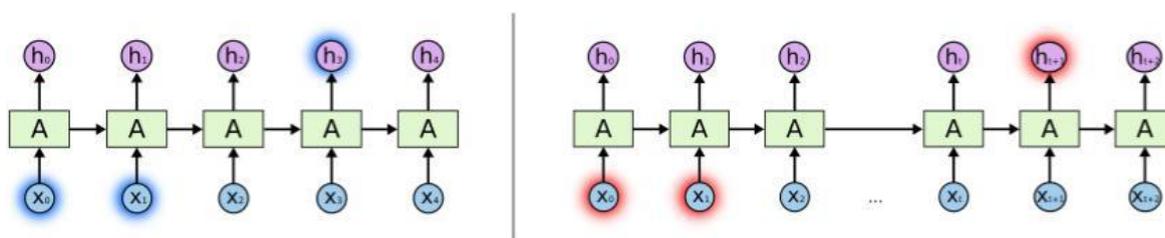


Figura 2.34: Relação de dependência de informações mais recentes (esquerda) e de informações mais distantes (direita) na arquitetura da rede recorrente.

FONTE: Olah (2015).

Para resolver esse problema, foi criado um tipo especial de RNN capaz de aprender as dependências de longo prazo, as LSTMs. As LSTMs também têm a forma de uma cadeia de módulos repetidos, como as RNNs, mas a estrutura desses módulos é diferente. Ao invés de ter uma única camada de Rede Neural, existem quatro camadas interagindo entre si. A Figura 2.35 mostra a diferença entre a estrutura do módulo da RNN e da LSTM (Olah, 2015).

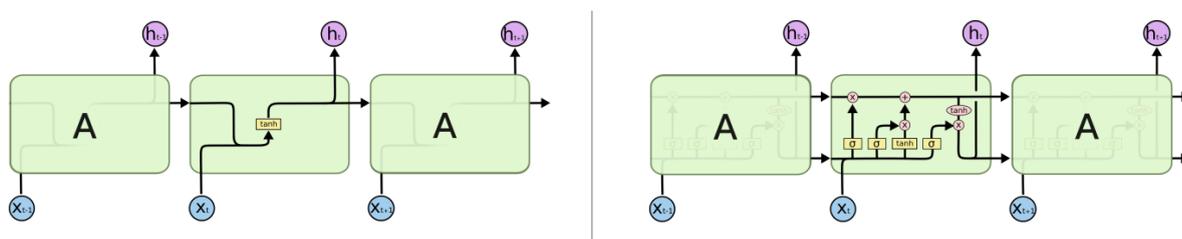


Figura 2.35: Comparação entre a arquitetura da RNN (esquerda) com a LSTM (direita).

FONTE: (Olah, 2015).

A primeira etapa da rede LSTM se resume a decidir quais informações devem ser jogadas fora do estado da célula. Essa decisão é feita por uma camada sigmóide (σ) chamada de *Forget Gate Layer*. Ela avalia o estado anterior da célula e a entrada e gera um valor entre 0 e 1, sendo que 1 representa “manter completamente as informações” e 0 “jogar fora todas as

informações”. Essa etapa ajuda a rede a selecionar as informações que são realmente relevantes (Olah, 2015).

A segunda etapa é composta por dois passos, e tem o objetivo de decidir quais novas informações serão armazenadas no estado da célula. O primeiro passo é a camada sigmóide chamada de *Input Gate Layer*, que decide quais valores serão atualizados. Em seguida, uma camada *tanh* (Tangente Hiperbólica) cria um vetor de novos valores candidatos que podem ser adicionados ao estado da célula. Então, esses dois passos são combinados para criar a atualização para o estado (Olah, 2015).

A última etapa é focada em decidir o que será produzido de saída. Primeiro, é executada uma camada sigmóide que decide quais partes do estado da célula serão utilizados para produzir a saída, chamada de *Output Gate Layer*. Após isso, o estado da célula passa por meio de uma *tanh* e o resultado é multiplicado pela saída da camada sigmóide (Olah, 2015).

Tanto a parte matemática quanto o funcionamento mais completo das redes RNN e LSTM podem ser encontrados em (Olah, 2015), (Karim, 2018) e (Phi, 2018).

2.3.2.10 GRU

O modelo GRU (*Gated Recurrent Unit*) é de uma geração mais recente de Redes Neurais do que a LSTM, mas possui uma estrutura bem semelhante. Considerando o modelo anterior, a GRU passou por modificações na estrutura, substituiu o estado da célula pelo estado oculto para transferir informações e definiu somente duas portas, uma de reinicialização e uma de atualização (Phi, 2018). Ambas portas se baseiam na ideia de aplicar a sigmóide sobre a soma da entrada multiplicada pelos pesos, referentes à porta, com o estado oculto multiplicado pelos próprios pesos, também referentes à porta (Kostadinov, 2017).

A primeira porta, chamada de *Update Gate*, atua de forma semelhante às portas *Input Gate* e *Forget Gate* da LSTM. Ela decide quais informações devem ser ignoradas e quais novas informações devem ser consideradas (Phi, 2018). Em resumo, essa porta ajuda o modelo a determinar o quanto da informação do passado (das iterações anteriores) precisa ser passada para o futuro (Kostadinov, 2017). A segunda e última porta, chamada de *Reset Gate*, também ajuda na etapa de decidir quantas informações anteriores devem ser esquecidas (Phi, 2018). Essa porta é utilizada pelo modelo para decidir quanta informação do passado deve ser esquecida (Kostadinov, 2017). A comparação estrutural entre as duas redes pode ser observada na Figura 2.36.

As redes GRU possuem menos operações de *Tensor* do que as LSTMs, portanto, elas são um pouco mais rápidas de treinar. Mas além dessa característica, não há nenhuma evidência que faça a GRU ser superior ou inferior à LSTM (Phi, 2018).

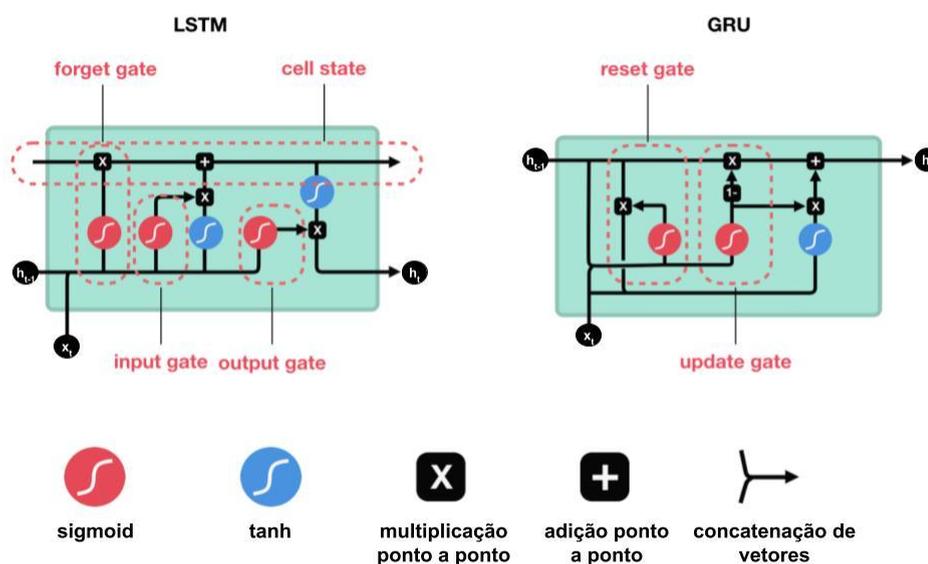


Figura 2.36: Comparação estrutural entre LSTM (esquerda) e GRU (direita).

FONTE: adaptado de Phi (2018).

2.3.3 MODELOS MULTIVARIADOS

Por fim, a última modalidade de modelos que será apresentada são os modelos Multivariados, que utilizam mais de uma variável, ou característica, para realizar as previsões. Já foram descritos modelos Multivariados na seção anterior mas, nesta seção, a diferença se resume ao fato dos modelos serem utilizados exclusivamente para problemas com várias variáveis de entrada. Assim, os modelos VAR e VECM serão descritos nas subseções abaixo.

2.3.3.1 VAR

Uma das limitações dos modelos Univariados já apresentados é a imposição de uma relação unidirecional, pois a variável de previsão é influenciada pelas variáveis predictoras, mas não vice-versa. Entretanto, existem muitos casos em que o inverso também deve ser permitido, onde todas as variáveis se afetam entre si. O modelo VAR (*Vector Autoregressive*) é uma generalização do modelo Univariado AR para prever um vetor de séries temporais. Esse modelo permite as relações mencionadas anteriormente, sendo que as variáveis são tratadas de forma simétrica, considerando que todas elas se influenciam entre si (Hyndman e Athanasopoulos, 2018).

Matematicamente, o modelo compreende uma equação por variável no sistema. O lado direito de cada equação inclui uma constante e os *lags* de todas as variáveis do sistema. Para simplificar, será apresentada a equação do modelo VAR(1) bidimensional (2.29) abaixo, onde $\varepsilon_{1,t}$ e $\varepsilon_{2,t}$ representam o *White Noise*, o coeficiente $\Phi_{ii,l}$ captura a influência do l -ésimo *lag* da variável y_i , enquanto o coeficiente $\Phi_{ij,l}$ captura a influência do l -ésimo *lag* da variável y_j em y_i (Hyndman e Athanasopoulos, 2018).

$$\begin{aligned} y_{1,t} &= c_1 + \Phi_{11,1} y_{1,t-1} + \Phi_{12,1} y_{2,t-1} + \varepsilon_{1,t} \\ y_{2,t} &= c_2 + \Phi_{21,1} y_{1,t-1} + \Phi_{22,1} y_{2,t-1} + \varepsilon_{2,t} \end{aligned} \quad (2.29)$$

Para explicar o funcionamento do modelo será utilizado o exemplo proposto por (Ramkumar, 2019i). O cenário se resume em um vendedor de frutas que oferece somente dois produtos, maçãs ou bananas. O objetivo desse vendedor é prever quantas bananas e maçãs serão vendidas em um determinado mês. Os dados disponíveis para resolver o problema são as quantidades de maçãs e bananas vendidas no mês passado.

Considerando o modelo AR, é possível modelar, tanto as bananas (b_t) quanto as maçãs (m_t), como sendo a quantidade vendida da fruta no mês passado. Mas a parte diferente é que o número de maçãs que foram vendidas no mês passado pode afetar o número de bananas que serão vendidas no mês atual. Por exemplo, se foram vendidas muitas maçãs no mês passado, pode ser que os clientes estejam menos inclinados a comprar bananas, então elas se tornarão menos populares e, assim, a venda de bananas no mês atual será menor do que o usual. Da mesma forma, o número de bananas vendidas mês passado pode afetar o número de bananas vendidas esse mês. Então, como os dados disponíveis são de *lag* igual a 1, o modelo possível para solucionar o problema seria um VAR(1) (equação 2.30):

$$\begin{aligned} m_t &= c_{11,1} m_{t-1} + c_{12,1} b_{t-1} + \varepsilon_{m,t} \\ b_t &= c_{21,1} b_{t-1} + c_{22,1} m_{t-1} + \varepsilon_{b,t} \end{aligned} \quad (2.30)$$

Como o nome diz, “*Vector*” *Autoregression*, esse modelo costuma ser representado utilizando vetores e matrizes. Considerando o exemplo acima, no formato vetorial a equação para previsão seria (equação 2.31):

$$\text{Sendo } f_t = [m_t, b_t] e \quad \epsilon_t = [\epsilon_{m,t}, \epsilon_{b,t}] e$$

$$C = \begin{pmatrix} c_{11,1} & c_{12,1} \\ c_{21,1} & c_{22,1} \end{pmatrix}$$

$$\text{Então } f_t = C f_{t-1} + \epsilon_t \quad (2.31)$$

A forma geral do modelo, para k variáveis e p lags é apresentada na equação 2.32 abaixo (Lütkepohl, 2005). A diferença desta forma para a equação utilizada no exemplo se resume às constantes (c_1, \dots, c_k) que foram definidas como 0 e o conteúdo da matriz C agora passa a se chamar $a^{lag}_{i,j}$.

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a_{1,1}^1 & a_{1,2}^1 & \cdots & a_{1,k}^1 \\ a_{2,1}^1 & a_{2,2}^1 & \cdots & a_{2,k}^1 \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1}^1 & a_{k,2}^1 & \cdots & a_{k,k}^1 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a_{1,1}^p & a_{1,2}^p & \cdots & a_{1,k}^p \\ a_{2,1}^p & a_{2,2}^p & \cdots & a_{2,k}^p \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1}^p & a_{k,2}^p & \cdots & a_{k,k}^p \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix} \quad (2.32)$$

2.3.3.2 VECM

Nos modelos de séries temporais é comum o uso de diferenciação para tornar as séries estacionárias com o objetivo de viabilizar ou facilitar a previsão. O problema dessa técnica é que ela ignora relações possivelmente importantes a longo prazo entre os níveis (períodos de tempo) (Maitra, 2019). Uma alternativa é testar se as regressões entre os níveis são confiáveis, o que é chamado de Cointegração. A intuição por trás da Cointegração se baseia na ideia de que o grupo de séries temporais, dadas como entrada ao modelo, seguem caminhos aleatórios ao longo do tempo. Dessa forma, é possível pensar em Cointegração como um forma de descobrir quais séries tendem a “andar aleatoriamente juntas” e cuja propagação (diferença entre as duas séries em cada intervalo de tempo) é estacionária. A informação extraída deste conceito diz que, embora as séries se movam independentemente, a distância média entre elas permanece relativamente constante (Goldkamp, 2020). Na Figura 2.37 é mostrado um exemplo onde as duas séries não possuem muita semelhança em cada período de tempo, mas possuem alta Cointegração, como pode-se observar pela diferença que se mantém entre as séries.

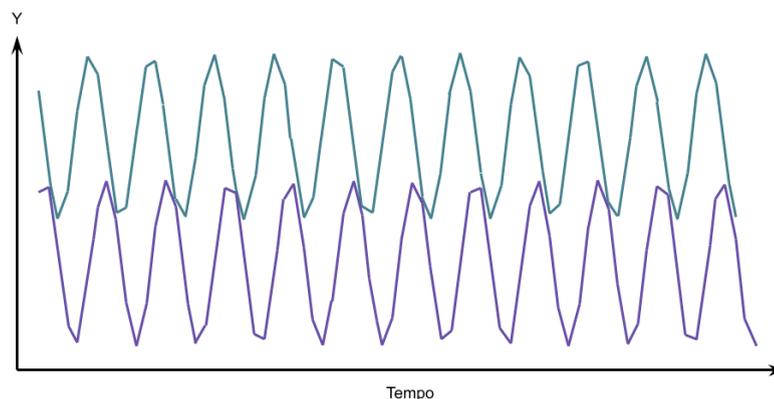


Figura 2.37: Exemplo de séries temporais com baixa Correlação e alta Cointegração.
 FONTE: adaptado de Goldkamp (2020).

O método de Cointegração, seja o Procedimento *Engle-Granger* ou o Teste *Johansen*, forma uma série estacionária sintética a partir da combinação de duas ou mais séries não estacionárias (Goldkamp, 2020). Assim, se houver Cointegração na série, então o Modelo Vetorial de Correção de Erros (VECM - *Vector Error Correction Model*), que combina níveis e diferenças pode ser estimado invés de um VAR (Maitra, 2019). Mais formalidades sobre a teoria e os procedimentos de Cointegração podem ser encontrados em (Goldkamp, 2020).

Matematicamente, o modelo VECM pode ser definido pela equação 2.33, onde Δx é a primeira diferença das variáveis no vetor x , Π_i é uma matriz de coeficientes de relações de cointegração, Γ é uma matriz de coeficientes dos *lags* de variáveis diferenciadas de x , d é um vetor de termos determinísticos e C sua matriz de coeficientes correspondente, p é a ordem de *lag* do modelo em sua forma VAR e ϵ é um termo de erro com média zero. Pela equação, a única diferença entre o modelo VAR e o VECM é o termo de correção de erro Πx_{t-1} , que captura o efeito de como a taxa de crescimento de uma variável em x muda se uma das variáveis se desviar de seu valor de equilíbrio (Mohr, 2019).

$$\Delta x_t = \Pi x_{t-1} + \sum_{l=1}^{p-1} \Gamma_l \Delta x_{t-l} + C d_t + \epsilon_t \quad (2.33)$$

Os passos da previsão com o modelo VECM se resumem a determinar o número de *lags* que serão considerados, realizar o teste Granger de causalidade, realizar o teste de grau de Cointegração, realizar a estimação com o VECM e fazer a decomposição das variáveis (Mohr, 2019). Mais informações e exemplos sobre o modelo podem ser encontrados em (Maitra, 2019).

2.4 CONSIDERAÇÕES FINAIS

Nesta seção foram explicados os conceitos principais para o entendimento do problema, explorando a área de Aprendizado Supervisionado, na seção 2.1, composta pelos dois métodos principais, que são a Classificação, seção 2.1.1, e a Regressão, seção 2.1.2. Na seção 2.2 foram estudados os conceitos, fundamentos, propriedades e pré-processamentos mais relevantes quando se trata de séries temporais, que foram a Autocorrelação, Tendência, Sazonalidade, Ruído, Estacionariedade, Teste ADF, Diferenciação, Notação *Backshift*, *Look Back*, Escala *Min Max* e *Walk Forward*, indo da seção 2.2.1 até a 2.2.11, respectivamente. Os modelos apresentados seguiram a organização das categorias de modelos Univariados (seção 2.3.1), modelos Mistos (2.3.2) e modelos Multivariados (2.3.3). Em todos os tópicos buscou-se trazer uma breve descrição, as equações matemáticas e uma visão mais intuitiva, com uso de exemplos ou imagens, com o objetivo de facilitar o entendimento da metodologia e das análises dos resultados que serão apresentadas posteriormente neste trabalho.

3 TRABALHOS RELACIONADOS

Nesta seção serão apresentadas três pesquisas semelhantes com o problema escolhido para este trabalho, sendo as duas primeiras relacionadas com a predição de velocidade de vento e a terceira com modelos de previsão no contexto de dados hidrológicos. A primeira pesquisa explorou o uso de Redes Neurais Artificiais para realizar a predição de uma estação alvo com base em outras estações próximas. A segunda utilizou um algoritmo de Aprendizado de Máquina, em conjunto com técnicas de otimização de hiperparâmetros, para obter previsões de velocidade de vento em uma base de dados pequena e, por fim, a última pesquisa explorou 5 modelos simples de regressão, e suas 26 combinações, para descobrir a vazão média de 10 anos à frente, utilizando uma grande quantidade de dados, com aproximadamente 90 anos de séries temporais com 600 estações diferentes.

3.1 PREVISÃO DE VELOCIDADE DE VENTO COM REDES NEURAI

Neste trabalho (Bilgili et al., 2007), o foco foi em predizer a média mensal de vento de uma determinada localidade (chamada de *target station*) utilizando a média de velocidade de vento das localidades próximas (*reference stations*). Os dados consistiam em séries temporais de velocidade de vento (m/s) por hora, em 8 localidades na região mediterrânea oriental da Turquia (Dörtyol, Antakya, Samandağ, Karatas, İskenderun, Yumurtalık, Adana e Mersin), iniciando em 1992 e indo até 2001. Para determinar quais estações seriam utilizadas como referência para uma estação alvo foram realizados cálculos de correlação de duas em duas estações. Assim, se a correlação entre a estação alvo e a estação da vez fosse maior do que 0,5 ela seria utilizada como estação de referência para aquela estação alvo.

O modelo escolhido foi uma ANN (*Artificial Neural Network*), consistindo de uma camada de entrada, duas camadas ocultas e uma camada de saída. A média mensal de velocidades das estações de referência assim como os valores delas em um determinado mês foram utilizados como entrada da rede. Na primeira camada oculta não foi utilizada uma função de ativação, mas na segunda camada oculta e na camada de saída, a *Logistic Sigmoid Transfer Function* (logsig) e a *Linear Transfer Function* (purelin) foram utilizadas, respectivamente. O algoritmo de *Resilient Propagation* (RP) foi utilizado para treinar a estimativa da média mensal de velocidade de vento da estação alvo. Foram testados vários

intervalos de quantidade de neurônios, sendo o mínimo de 6 e o máximo de 12 em cada camada oculta da rede.

Para avaliar os resultados foi utilizada a função MAPE (*Mean Absolute Percentage Error*), obtendo valores entre 4,49% a 14,13% para todas as estações treinadas, sendo testadas em um espaço de tempo de 2 anos (2000 a 2001). Como conclusão do trabalho, foi apresentada a afirmação de que a ANN é uma poderosa ferramenta para prever a velocidade de vento de uma estação alvo dependendo da correlação entre ela e as estações de referência.

3.2 PREVISÃO DE VELOCIDADE DE VENTO COM APRENDIZADO DE MÁQUINA

Neste trabalho (Jiang et al., 2015), o cenário se baseou na importância das previsões confiáveis de velocidade de vento de curto prazo (10 minutos e 1 hora), pois elas desempenham um papel prático e crucial nos sistemas de conversão de energia eólica, como o sistema dinâmico de controle de turbinas eólicas e as programações do sistema de potência. Com a relevância do cenário definida, foi proposto um modelo híbrido inteligente para previsão de velocidade de vento de curto prazo, tendo sua metodologia baseada na análise da correlação cruzada, um modelo de regressão com vetores de suporte (SVR - *Support Vector Regression*) e o acoplamento de algoritmos de otimização para a definição de hiperparâmetros, como o BSO (*BrainStorm Optimization*) e o *Cuckoo Search* (CS).

Os dados utilizados são advindos de um parque eólico da China, contendo 4 séries temporais de velocidade de vento (m/s) com intervalos de 10 minutos de 4 turbinas de vento instaladas em 4 sítios diferentes. Após feita a correlação cruzada entre as séries, as velocidades de vento dos dois primeiros sítios foram selecionadas e foi realizada uma agregação por média de 1 hora de intervalo, restando 258 pares de dados para treinamento e 30 pares para teste. Para comparar a eficiência dos métodos de otimização, nessas velocidades de vento foi aplicado o BSO juntamente com o SVR. As outras duas séries restantes de velocidade de vento foram mantidas com o intervalo de 10 minutos e o otimizador utilizado foi o CS. Por questões de comparação, a rede ERNN (*Elman Recurrent Neural Network*), que é uma forma modificada das redes neurais, com uma camada interna de neurônios auto-recorrentes, foi implementada. Assim, para cada série, 3 métodos diferentes foram testados e avaliados com as métricas MAE, MSE e MAPE.

Como resultado, os experimentos com os otimizadores alcançaram os melhores resultados, com MSE de 0,52 e 1,23 e MAPE de 4,56% e 7,35% nas velocidades de vento

agregadas por 1 hora e de 0,83 e 0,61 e 7,13% e 6,01% nas duas séries restantes. O modelo SVR obteve resultados próximos à rede ERNN, com MSE de 1,03 e 1,75 e MAPE de 9,59% e 13,49% nas duas primeiras séries temporais e 0,89 e 0,98 e 8,20% e 8,15% nas duas últimas.

3.3 PREVISÃO DE DADOS HIDROLÓGICOS VISANDO MODELOS DERIVADOS DA TEORIA DE SÉRIES TEMPORAIS

Neste trabalho (Papacharalampous e Tyrallis, 2020), foi ressaltada a importância da entrega de previsões hidrológicas para a gestão de água urbana e agrícola, geração de energia hidrelétrica, proteção e gestão de enchentes, mitigação e alívio de secas e planejamento e o controle das bacias hidrológicas. Tendo em vista esses pontos, foi proposta uma nova metodologia, simples e flexível, para a previsão de séries temporais hidrológicas.

As séries temporais homogêneas de vazão média anual com mais de 90 anos foram selecionadas do arquivo do *Global Streamflow Indices and Metadata*, contando com 600 estações que cobrem grandes partes da América do Norte e da Europa. Essas séries foram dadas como entrada para 5 modelos de previsão simples e 26 variações entre eles, combinando os 5 modelos de dois em dois até cinco em cinco. Os modelos escolhidos contam com o *Naive*, que utiliza o *lag* anterior da série como previsão, o *Simple Exponential Smoothing*, *Complex Exponential Smoothing*, ARFIMA (*Automatic Fractionally Integrated Moving Average*) e o *Prophet* do Facebook.

A metodologia seguiu 7 passos, sendo o primeiro a seleção dos primeiros 80 anos das séries hidrológicas, separando em Globo (todas as estações), Região A (estações da América do Norte) e Região B (estações da Europa). O segundo passo foi o treinamento dos 5 modelos com esse período de dados. No terceiro passo, as previsões dos modelos foram obtidas para os últimos 10 anos de dados. A próxima etapa se resumiu a combinar as previsões obtidas pelos modelos de todas as 26 formas possíveis, utilizando a mediana como combinador. Em seguida, as métricas MAE (*Mean Absolute Error*), MAPE (*Mean Absolute Percentage Error*), MdAE (*Median Absolute Error*), MdAPE (*Median Absolute Percentage Error*) e RMSE (*Root Mean Squared Error*) foram calculadas para todas as previsões realizadas. Nos dois últimos passos as métricas foram ranqueadas e foi computada a melhora relativa de todos os métodos em relação ao modelo *Naive*.

Como resultado, foi possível chegar à conclusão de que, ao combinar os modelos *Naive*, ARFIMA e *Prophet* houve uma melhora relativa de 18,9% de RMSE em relação ao modelo *Naive* individual. Dos 3 agrupamentos de dados, no Globo, o MAE foi de 13,24%, o

MAPE de 13,45% e o RMSE de 12,73%, na Região A, o MAE foi de 13,03%, o MAPE de 13,52% e o RMSE de 12,81 e na Região B o MAE foi de 13,59%, o MAPE de 13,59% e o RMSE de 12,43%, todos para a mesma combinação, reunindo as previsões do *Naive* com o *Simple Exponential Smoothing*, o ARFIMA e o *Prophet*.

3.4 CONSIDERAÇÕES FINAIS

Nesta seção foram brevemente descritas três pesquisas diferentes sobre previsão de séries temporais, sendo duas delas sobre dados eólicos e uma sobre dados hidrológicos. A escolha das pesquisas visou abranger os dois tipos de dados, a importância e dificuldade de realizar regressões em séries temporais, a quantidade de períodos da série (com muitos e poucos dados) e diferentes modelos de previsão, contemplando a área de Aprendizado de Máquina, com o SVR, Aprendizado Profundo, com a Rede Neural, e métodos derivados da Teoria de Séries Temporais, com os dois tipos de *Exponential Smoothing* e o ARFIMA. Além disso, dentre as métricas observadas, foi possível notar que o MAPE variou de, aproximadamente, 4,50% chegando ao máximo de 14,13%, ao considerar os resultados de todas as pesquisas.

4 METODOLOGIA

Considerando a importância e relevância dos dados eólicos e hidrológicos, além da variedade de métodos de pré-processamento e de modelos de previsão de séries temporais, o objetivo desta pesquisa não se restringiu somente a encontrar um método de previsão de séries temporais que melhor se adequasse aos dados, mas também teve a finalidade de analisar e comparar vários métodos utilizados em todo o processo de previsão. Assim, o propósito desta seção é conectar todos os métodos já descritos na fundamentação teórica, gerando várias metodologias para previsão de séries temporais de velocidade de vento.

Além de explorar os pré-processamentos e modelos, este trabalho visou descobrir se, ao utilizar dados hidrológicos para prever as séries eólicas, era possível obter resultados semelhantes, ou melhores, do que recorrer somente aos dados de velocidade de vento para realizar as predições. Para isso, foi necessário dividir a pesquisa em dois modos diferentes, o modo Univariado, que utiliza somente os dados de velocidade de vento para realizar as predições, e o modo Multivariado, que emprega os dados hidrológicos como características para prever as séries temporais eólicas.

Nesta seção, serão apresentados todos os passos para a solução do problema de previsão de séries temporais de velocidade de vento, iniciando com a apresentação da base de dados, na seção 4.1, seguindo para o detalhamento da sequência de etapas da metodologia no modo Univariado, presente na seção 4.2, e no modo Multivariado, na seção 4.3. Na última seção, 4.4, é descrita a nova etapa de pré-processamento proposta, chamada de Rede de Representação.

4.1 BASE DE DADOS

Foram utilizados dois conjuntos de dados nesta pesquisa, um relacionado às vazões de várias usinas e outro contendo informações sobre a velocidade de vento. As séries de vazão foram extraídas dos dados do *deck* fornecido pela Câmara de Comercialização de Energia Elétrica (CCEE) e correspondem às usinas hidrelétricas do SIN (Sistema Interligado Nacional). Os dados compreendem 171 aproveitamentos hidrelétricos, com as vazões em m^3/s sendo fornecidas a cada mês no período de 1994 a 2013 (CCEE, 2017). Dos 171 aproveitamentos hidrelétricos foram selecionados 22 por métodos estatísticos realizados pelo

grupo do projeto Lynx UFPR/LACTEC/COPEL. A Tabela 4.1 mostra uma pequena amostra dos dados, apresentada pelas primeiras 4 colunas.

Já as séries de velocidade de vento vêm da empresa espanhola VORTEX. São séries de velocidades simuladas (em m/s) e abrangem diferentes alturas (de 50 a 150 metros AGL - *Above Ground Level*). Os dados VORTEX foram cedidos pela Copel no âmbito do projeto Lynx do instituto LACTEC e foram compartilhados para a realização desta pesquisa. Dessa forma, as séries temporais de velocidade de vento são dados privados. A última coluna da Tabela 4.1 apresenta uma simulação desses dados, com o objetivo de mostrar o formato das informações utilizadas.

Tabela 4.1: Tabela com uma amostra dos dados de vazões nas 4 primeiras colunas e os dados de velocidade de vento (simulados devido à privacidade dos dados) na última coluna.

FONTE: elaborado pela autora (2021).

Data	270 - Serra Mesa	191 - Cana Brava	253 - Sao Salvador	...	Vento em Afrânio
jan/94	1228	1310	1601	...	1.23456789
fev/94	949	1057	1212	...	2.34567891
mar/94	2157	2291	2472	...	3.45678912
abr/94	1233	1295	1496	4.56789123
mai/94	636	725	803	...	5.67891234
jun/94	482	542	583	...	6.78912345
jul/94	380	436	472	...	7.89123456
ago/94	288	341	372	...	8.91234567
set/94	224	269	291	...	9.12345678
out/94	233	279	292	...	1.23456789
nov/94	437	519	566	...	2.34567891
dez/94	759	864	970	...	3.45678912
jan/94	1049	1154	1338	...	4.56789123
...	
dez/13	1114	1256	1387	...	5.67891234

Das 171 usinas hidrelétricas, foram selecionadas 22, com base em cálculos de correlação feitos pelos pesquisadores do projeto Lynx. A distribuição das usinas escolhidas é apresentada pela Figura 4.1 abaixo. As localizações foram baseadas nas coordenadas presentes em (ANA, 2014), (Sinop, 2020), (COPEL, 2018), (UHE Teles Pires, 2020), (São Manoel Energia, 2018) e (Norte Energia, 2011).

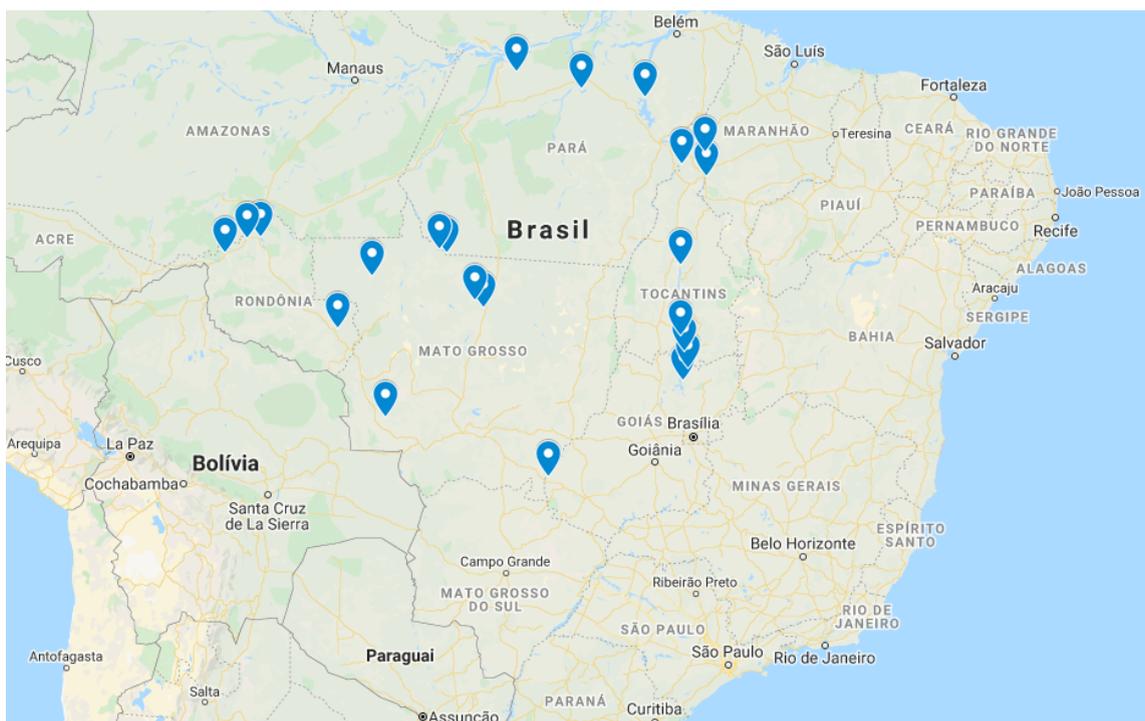


Figura 4.1: Localização das 22 usinas hidrelétricas selecionadas das 171 iniciais.

FONTE: elaborado pela autora (2021).

As usinas escolhidas foram: Serra Mesa (Goiás), Cana Brava (Goiás), São Salvador (Tocantins), Peixe Angical (Tocantins), Lajeado (Tocantins), Estreito Toc (Tocantins-Maranhão), Serra Quebrada (Tocantins-Maranhão), Tucuruí (Pará), Santa Isabel (Tocantins-Pará), Couto Magalhães (Goiás-Mato Grosso), Dardanelos (Mato Grosso), Rondon II (Rondônia), Samuel (Rondônia), Guaporé (Mato Grosso), Jirau (Rondônia), Santo Antônio (Rondônia), Sinop (Mato Grosso), Colíder (Mato Grosso), Teles Pires (Mato Grosso-Pará), São Manoel (Mato Grosso-Pará), Curuá-Una (Pará) e Pimental (Pará). Todas as 22 usinas selecionadas localizam-se nos rios Amazonas, Tocantins e seus afluentes.

Já os sítios de velocidade de vento abrangiam 7 localizações diferentes, que são apresentadas pela Figura 4.2. As localizações foram aproximadas de acordo com os locais indicados nos próprios dados.



Figura 4.2: Localização dos sítios presentes nos dados de velocidade de vento.

FONTE: elaborado pela autora (2021).

Além desses dados, foram recebidos dados já estacionários, no caso das vazões, do grupo de pesquisa do projeto Lynx. Os procedimentos adotados para tornar os dados originais em estacionários podem ser encontrados no artigo (Detzel et al., 2011). Com os dados estabelecidos, o problema que esta pesquisa visou resolver se concentrou em prever a série temporal de velocidade de vento de Afrânio, utilizando os valores anteriores da própria série, ou os dados hidrológicos das 22 usinas.

4.2 MODO UNIVARIADO

Como mencionado no início da seção, foram definidos dois modos de uso dos dados, sendo um deles o modo Univariado. Neste modo, foram utilizados exclusivamente os dados referentes à velocidade de vento, tornando o problema de previsão dependente de apenas uma série temporal. Além de tentar descobrir se prever dados eólicos, utilizando somente a velocidade de vento de vários meses anteriores, resultaria em um bom preditor, esse modo também permitiu o uso de métodos de previsão baseados na Teoria de Séries Temporais, como AR, MA, ARIMA e ARCH.

A Figura 4.3 apresenta um diagrama simplificado com o fluxo de métodos aplicados ao problema. No topo do diagrama, considera-se o uso da Rede de Representação, que será explicada posteriormente. Por simplicidade, nesta subseção será considerado somente o caminho da esquerda, onde não há o uso da Rede de Representação. Assim, a maioria dos métodos presentes no diagrama são referentes ao pré-processamento dos dados, ou seja, nas alterações realizadas na série temporal de velocidade de vento antes de alimentar alguns dos modelos de previsão com ela. Como o objetivo deste trabalho também visava analisar o impacto dos métodos de pré-processamento, várias combinações diferentes foram aplicadas

aos dados. Analisando a Figura 4.3, é possível observar a presença, ou ausência, de todos os métodos já mencionados, como:

- **Estacionariedade:** que tem o objetivo de tornar, ou não, os dados brutos em estacionários;
- **Escala:** etapa com a finalidade de aplicar, ou não, a escala *MinMax* nos dados;
- **Look Back:** que tem o propósito de aplicar um *Look Back* nos dados, sendo 1 para os modelos baseados na Teoria de Séries Temporais, que utilizam, por definição, somente uma característica, e podendo ser 1, 3, 6, 9 ou 12 para os demais modelos;
- **Walk Forward:** que define se será utilizado, ou não, o método *Walk Forward* para treinar o modelo;
- **Otimização de Parâmetros:** etapa em que ocorrem vários treinamentos diferentes que são avaliados em uma base de validação com o objetivo de encontrar o melhor conjunto de parâmetros para treinar o modelo final;
- **Treinamento do Modelo:** etapa em que acontece o treinamento dos modelos de previsão utilizando os parâmetros encontrados previamente;
- **Avaliação do Modelo:** por fim, a avaliação do modelo consiste em prever o conjunto de teste e calcular o erro.

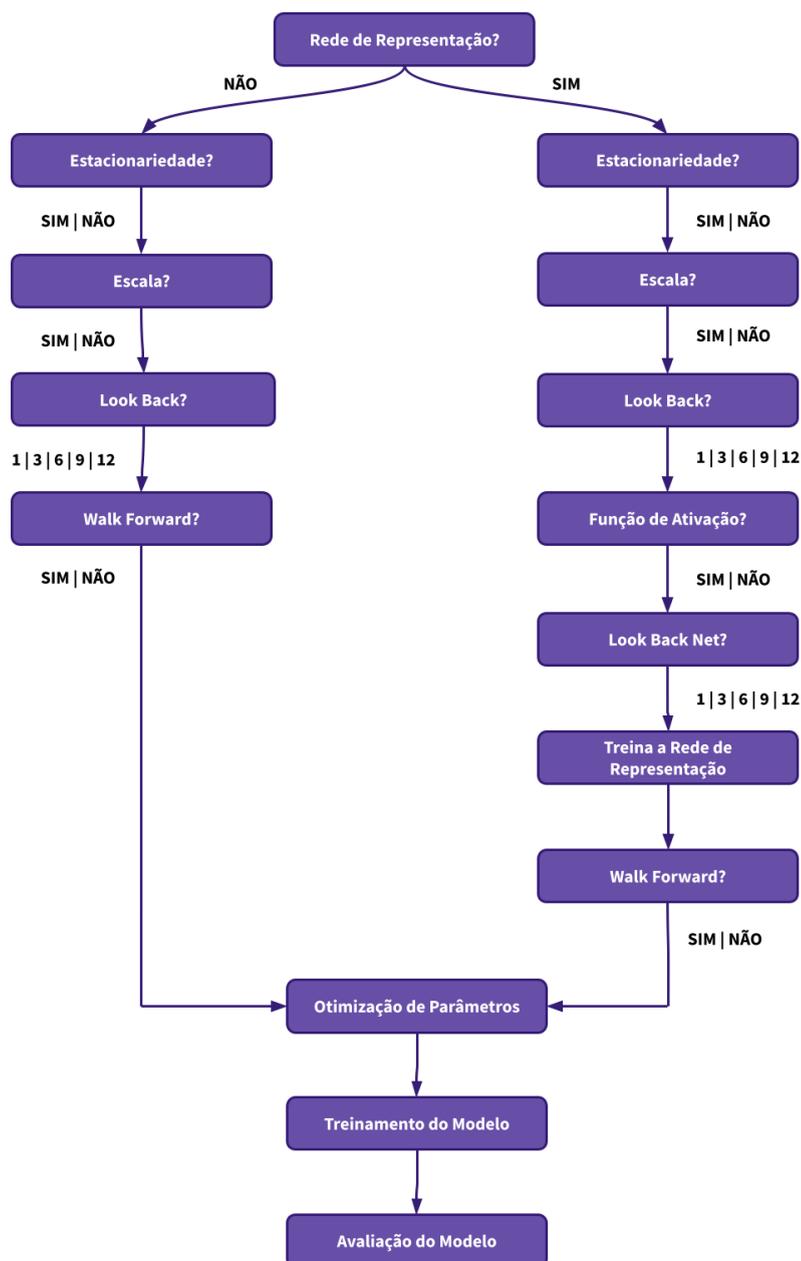


Figura 4.3: Diagrama simplificado das etapas da metodologia proposta.

FONTE: elaborado pela autora (2021).

Para facilitar o entendimento do diagrama, será apresentado somente o fluxo em que há o uso de todas as técnicas mencionadas, exceto a Rede de Representação. O exemplo consiste no fluxo: sem Rede de Representação, Série Estacionária, com Escala, *Look Back* com janela de tamanho 1 e o uso de *Walk Forward*. Iniciando com os dados brutos das séries temporais de velocidade de vento, o primeiro passo consistiu em separar os dados de vento e organizar eles de forma a adaptá-los para o problema de Aprendizado Supervisionado. A organização se resumiu a aplicar o método de *Look Back* com janela de tamanho 1.

O tamanho da janela utilizada neste modo se justifica pelo fato de que os modelos baseados na Teoria de Séries Temporais como AR, MA, ARMA, ARIMA, SARIMA, ARCH e GARCH são pensados para receber dados com apenas uma característica, assim, cada um desses modelos recebeu os dados de vento do mês anterior e tentou prever a velocidade de vento do tempo atual. Já nos modelos derivados de Aprendizado de Máquina e de Aprendizado Profundo, também foram utilizadas janelas de tamanhos 1, 3, 6, 9 e 12, aumentando o número de características mas ainda utilizando somente dados eólicos.

A razão de experimentos com o aumento do número de características pode ser explicada pela importância de considerar mais de 1 mês no passado. Como os dados mostram uma sazonalidade anual, levar em conta os 12 meses anteriores pode contribuir para que o modelo consiga encontrar determinadas relações entre os dados. A Figura 4.4 apresenta um dos gráficos gerados pela primeira etapa de Análise de Séries Temporais. Nos gráficos é possível observar a sazonalidade dos dados, tanto no caso das vazões quanto da velocidade de vento.

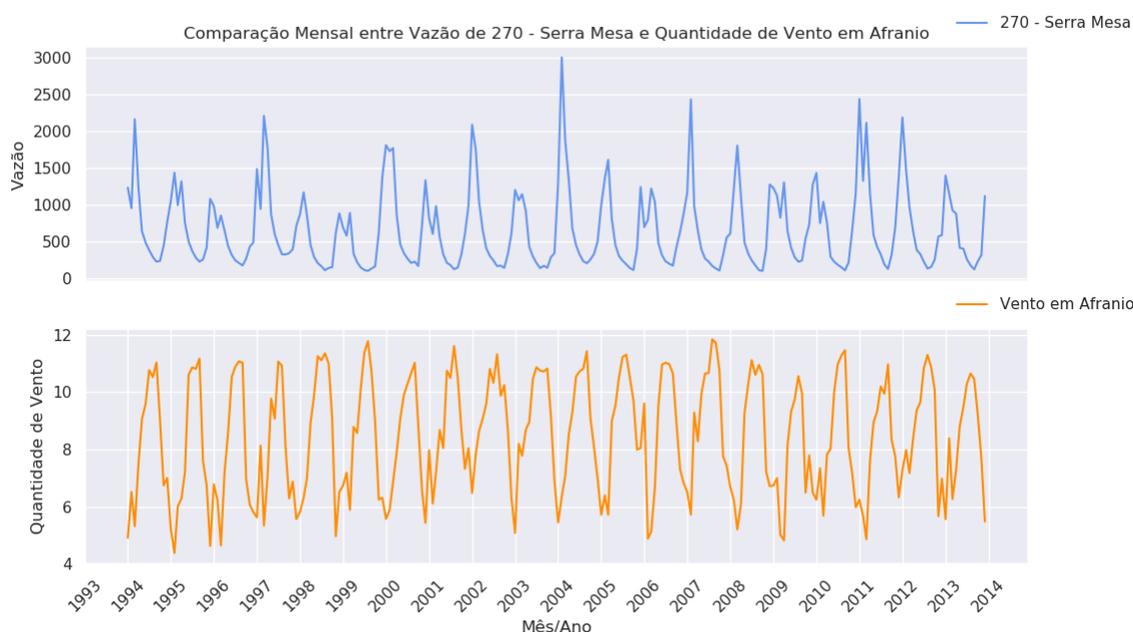


Figura 4.4: Gráfico dos dados de vazão e velocidade de vento mensais da usina hidrelétrica de Serra Mesa e do sítio de Afrânio, respectivamente.

FONTE: elaborado pela autora (2021).

Na primeira transição da Figura 4.5 é apresentada a transformação dos dados brutos em dados ajustados para os modelos de Aprendizado de Supervisionado, sendo a coluna azul da esquerda os dados e a coluna da direita os *targets*. Nesta transição, houve a aplicação do

Look Back com janela de tamanho 1, resultando em um deslocamento da série e na geração dos *targets*, como é explicado na seção 2.2.9.

O próximo passo compreendeu o problema da Estacionariedade. Inicialmente, a série de vento passou pelo Teste ADF para descobrir se os dados originais eram, ou não estacionários. Se não fossem, aplicava-se a diferenciação até que os dados se tornassem estacionários. A segunda transição da Figura 4.5 mostra essa etapa, pois os dados em azul eram não-estacionários. Os dados apresentados em alaranjado são os resultados da primeira diferença aplicada nos dados de treino (esquerda) e nos dados de *target* (direita). Mais detalhes sobre o Teste ADF e a Diferenciação podem ser obtidos nas seções 2.2.6 e 2.2.7, respectivamente.

Em seguida, os dados foram separados em treino e teste, sendo o conjunto de teste composto somente pelos dados do último ano (2013). Tanto os dados de treino quanto os *targets* foram escalados utilizando o método *MinMax*, como apresentado pela terceira transição da Figura 4.5. Foram realizados testes menores nos dados para escolher o valor da escala entre os intervalos (0, 1), (0,5, 1) e (-1, 1), e, o intervalo entre -1 e 1 obteve melhores resultados de RMSE ao utilizar os modelos *Gradient Boosting* e Florestas Aleatórias. Mais informações sobre o método *MinMax* podem ser obtidas na seção 2.2.10.

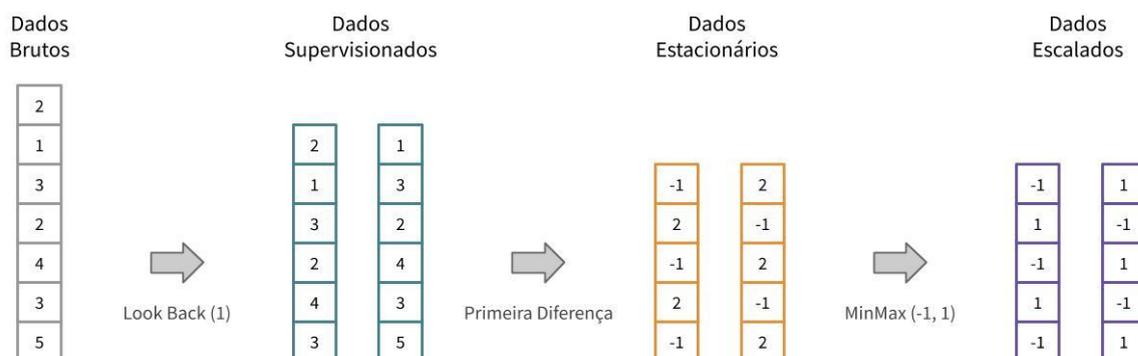


Figura 4.5: Exemplo do fluxo das 4 primeiras etapas da metodologia adotada.

FONTE: elaborado pela autora (2021).

Após todo o processo de pré-processamento dos dados, o próximo passo focou na otimização dos hiperparâmetros do modelo previamente escolhido. Dependendo da categoria do modelo, como baseado na Teoria de Séries Temporais, Aprendizado de Máquina ou Aprendizado Profundo, um otimizador diferente foi utilizado, sendo *Grid Search*, Skopt e Talos, respectivamente. Mesmo com otimizadores diferentes, o fluxo do processo de

otimização seguiu a mesma ideia: definir uma constante com todos os hiperparâmetros do modelo, e todos os valores que se deseja iterar, e estabelecer uma função de criação e treinamento do modelo para executar cada iteração da otimização, com o objetivo de descobrir os melhores hiperparâmetros. Nessa etapa, o melhor hiperparâmetro foi escolhido com base no menor MSE (equação 4.1) encontrado na base de validação definida pelo último mês do conjunto de treino. Como o método *Walk Forward* retreina o modelo várias vezes, por questões de desempenho, em todos os modelos foi utilizado o treinamento padrão (sem *Walk Forward*) para realizar a otimização.

A próxima etapa consistiu em treinar o modelo com a melhor configuração encontrada pelo passo anterior e, por fim, avaliar o modelo. Para isso, foram utilizadas as métricas MSE, MAPE e RMSE, que são as métricas mais comuns encontradas em trabalhos de regressão. As fórmulas são apresentadas abaixo, nas equações 4.1, 4.2 e 4.3, sendo y o valor predito, \hat{y} o valor esperado e n a quantidade de valores preditos no total (Karbhari, 2018).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.1)$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (4.2)$$

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}} \quad (4.3)$$

Além da otimização dos hiperparâmetros, também foi realizada uma otimização dos valores de pré-processamento, pois uma determinada configuração de hiperparâmetros do modelo poderia gerar bons resultados com o uso de um pré-processamento e resultados ruins sem o seu uso. Assim, todas as combinações de valores de parâmetros de pré-processamentos mostradas na Figura 4.3 foram executadas e, em cada iteração, foi feita a otimização de hiperparâmetros do modelo escolhido. Mais informações sobre as otimizações serão apresentadas no próximo capítulo.

Um dos problemas encontrados ocorreu durante o processo de predição, em que foi necessário inverter as transformações de escala e estacionariedade dos dados quando elas estavam ativas. A inversão da escala foi simples, pois todas as informações necessárias foram salvas dentro da classe *Scaler* da biblioteca *Scikit Learn*. Mas a parte desafiadora foi inverter a

estacionariedade obtida com a diferenciação. Toda vez que o resultado da predição de um exemplo da base de teste era obtido, o *target* do exemplo anterior era buscado. Para obter o resultado original bastava somar esses dois valores. O problema de utilizar a diferenciação é que há a necessidade de que a série usada como treinamento seja imediatamente anterior à série utilizada como teste, pela mesma unidade de tempo. No primeiro exemplo de teste, o *target* do último exemplo de treinamento precisava ser conhecido para que fosse possível inverter a estacionariedade. Devido a esse problema, a metodologia que converte os dados em estacionários utilizando a diferenciação foi utilizada somente para encontrar um conjunto de parâmetros que adequava melhor à solução do problema. Após esta etapa, especialistas na área de estatística utilizaram vários métodos para tornar os dados estacionários de forma a excluir a necessidade da inversão. Assim, foram utilizados esses dados para o resultado final do modelo, que foi treinado com os conjuntos de parâmetros pré-selecionados pelas otimizações de hiperparâmetros e de parâmetros de pré-processamento.

4.3 MODO MULTIVARIADO

No modo Multivariado, a sequência de etapas continuou a mesma do modo Univariado (Figura 4.3), a diferença se deu nas implementações dessas etapas, pois, nesse modo, não há somente uma série temporal a ser considerada, mas 22 séries hidrológicas diferentes. Ao invés de utilizar unicamente a série de vento, tanto para dados como para *targets*, agora os 22 aproveitamentos hidrelétricos dos dados de vazões são utilizados como dados para prever as velocidades de vento.

Inicialmente, os dados foram organizados de forma a se adequar ao problema de Aprendizado Supervisionado. Assim, as 22 séries de vazão foram utilizadas como dados e a série de vento como *target*. Nessa fase, diferente do modo anterior, foi aplicado um *Look Back* modificado logo em seguida. Como a quantidade de séries utilizadas é grande, não seria viável concatenar os 22 valores das séries dos meses anteriores. Dessa forma, foi definida uma Janela Rolante que capturou todos os dados de todos os meses de um aproveitamento hidrelétrico, de acordo com valor de *Look Back* indicado, e foi feita uma média desses valores. A Figura 4.6 exemplifica esse processo, apresentando o caso em que o *Look Back* indica para considerar 3 meses. Para cada um dos 22 aproveitamentos, foi feita a média dos últimos 3 valores daquele aproveitamento, como é possível observar na primeira tabela, e o resultado foi emparelhado com o rótulo correspondente, retratado na segunda tabela.

Data	Ap. Hidrelétrico 1	Ap. Hidrelétrico 2	...	Ap. Hidrelétrico 22	Vento
jan/94	1228	1310	...	1601	1.23456789
fev/94	949	1057	...	1212	2.34567891
mar/94	2157	2291	...	2472	3.45678912
abr/94	1233	1295	...	1496	4.56789123

Look Back = 3 | Janela Rolante por Média

Data	Ap. Hidrelétrico 1	Ap. Hidrelétrico 2	...	Ap. Hidrelétrico 22	Vento
mar/94	1445	1553	...	1762	3.45678912
abr/94	1233	1295	...	1496	4.56789123

Figura 4.6: Exemplo de aplicação do *Look Back* de 3 meses no modo Multivariado.

FONTE: elaborado pela autora (2021).

Com os dados organizados, o próximo passo foi a aplicação da Estacionariedade. Nesse modo, cada uma das séries dos aproveitamentos hidrelétricos foi verificada pelo Teste ADF. Se a verificação indicasse não estacionariedade, todas as 22 séries eram diferenciadas juntamente com os *targets*. A razão de diferenciar todas as séries se explica pelo fato de que, ao diferenciar, a série diminui de tamanho, então, para manter todas as séries do mesmo tamanho e conseguir inverter essa diferenciação, todas foram diferenciadas. Detalhes do Teste ADF e da Diferenciação foram explicados nas seções 2.2.6 e 2.2.7.

O próximo passo se resumiu a escalar os dados, que segue as mesmas etapas do modo Univariado, com a diferença de que os dados são escalados de forma independente aos *targets*, pois possuem unidades de medida diferentes. Após esta etapa, os dados são separados em conjunto de treinamento e conjunto de teste, sendo somente os dados do último ano (2013) pertencentes ao conjunto de teste. O conjunto de treinamento também é dividido, separando os dados do último mês para utilizar como validação. Os passos seguintes da metodologia seguem a mesma implementação apresentada na seção anterior.

A tentativa de realizar previsões de velocidade de vento utilizando dados hidrológicos pode ser justificada pela relação inversa (complementaridade hidro-eólica) dos dados eólicos com os dados de vazão. No primeiro gráfico da Figura 4.7 são apresentadas as 22 séries temporais selecionadas de vazão de janeiro de 1994 até dezembro de 1995. No segundo gráfico, é apresentada a série temporal de velocidade de vento de Afrânio durante o mesmo período de tempo das séries hidrológicas. Observando os dois gráficos alinhados, é possível

perceber uma relação inversa, pois no período de julho a outubro de 1994 as séries hidrológicas apresentaram valores bem baixos e a série de vento apresentou valores elevados. O oposto ocorreu no período de janeiro a junho de 1995. O mesmo comportamento ocorreu por todos os anos disponível na base de dados, como é possível observar na Figura 4.4. Dessa forma, o objetivo dos modelos Multivariados foi tentar descobrir se essa relação inversa entre os dados conseguiria contribuir para realizar boas predições.



Figura 4.7: Comparação entre os valores dos 22 aproveitamentos hidrelétricos dos dados hidrológicos com a série temporal de velocidade de vento em Afrânio, no período de janeiro de 1994 a dezembro de 1995.

FONTE: elaborado pela autora (2021).

4.4 REDE DE REPRESENTAÇÃO

Um dos problemas encontrados ao considerar os dados de séries temporais no modo Multivariado foi a impossibilidade de utilizar uma técnica de *Look Back* simples, sem necessidade de média, como no modo Univariado, que concatena as características dos meses anteriores. Existem 22 características para cada mês, tornando inviável concatenar vários meses para prever uma entrada só. Por exemplo, pensando em um *Look Back* de 3 meses, seriam 66 características de entrada, o que tornaria o treinamento do modelo lento e ineficiente. A premissa de que adicionar muita informação na entrada não implica em um melhor desempenho também foi considerada, pois a qualidade dos dados utilizados faz bastante diferença para gerar previsões melhores (Vickery, 2020).

Com esse problema em mente, foi proposta uma arquitetura de Rede Neural, chamada de Rede de Representação, com o objetivo de extrair informações de vários meses (chamados de *Look Back Net*) para realizar a previsão. Essa rede foi adaptada para os dois métodos, Univariado e Multivariado. A Figura 4.8 apresenta a arquitetura da Rede de Representação para os dados de vento (Univariado). Na imagem da esquerda foi projetada uma rede visando preservar o uso de um dado só para realizar a previsão. Assim, os valores dos meses anteriores x_0 (janeiro), x_1 (fevereiro) e x_2 (março) são as entradas, que passam por uma camada de 1 neurônio com alguma função de ativação, e a saída desse neurônio é passada para a camada linear de saída. A saída da última camada se refere ao valor de vento do próximo mês x_3 (abril). A rede da direita apresenta as mesmas propriedades, mas possui 3 neurônios na primeira camada (em roxo) ao invés de um. Essa arquitetura foi construída para seguir o padrão do modo Multivariado, onde o número de meses considerados é igual ao número de características que serão utilizadas.

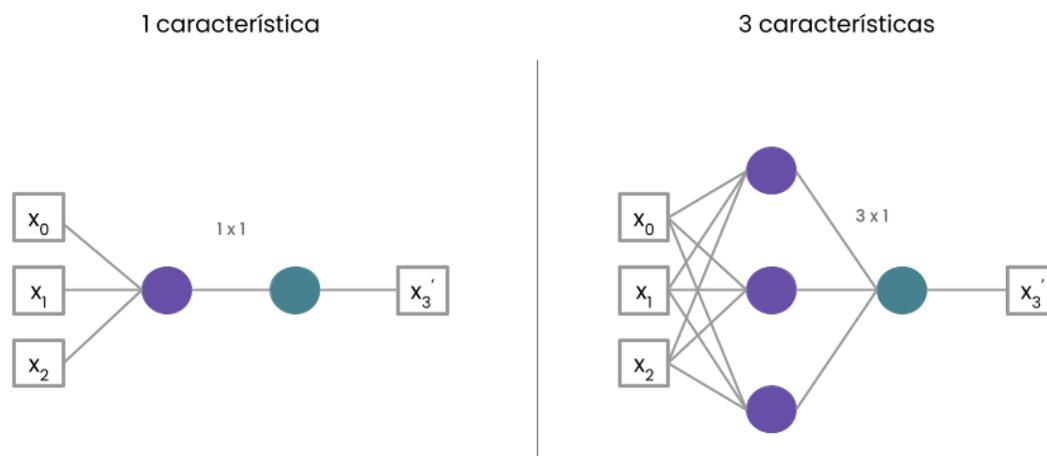


Figura 4.8: Arquiteturas implementadas da Rede de Representação no modo Univariado.

FONTE: elaborado pela autora (2021).

Essa rede foi treinada com a base de treinamento do modelo e, após o término do treino, foram extraídas as saídas da camada em roxo e utilizadas como características. Por exemplo, na imagem da esquerda da Figura 4.8, os dados de entrada dos meses de janeiro, fevereiro e março são transformados em um só, pois a nova característica será essas entradas multiplicadas pelos pesos da primeira camada somada com o *bias*. Nesse momento, a função de ativação é um parâmetro, ou seja, ela pode, ou não, ser considerada na geração da saída. Na imagem da direita o mesmo processo ocorre, mas agora são consideradas as saídas dos 3 neurônios da primeira camada oculta como características.

Já no modo Multivariado, a única diferença é o número de entradas para cada neurônio. Considerando o cenário dos 3 meses, as 22 características de cada mês são concatenadas e dadas como entrada da rede. A extração das 3 características segue o mesmo processo do modo Univariado. A arquitetura pode ser observada na Figura 4.9.

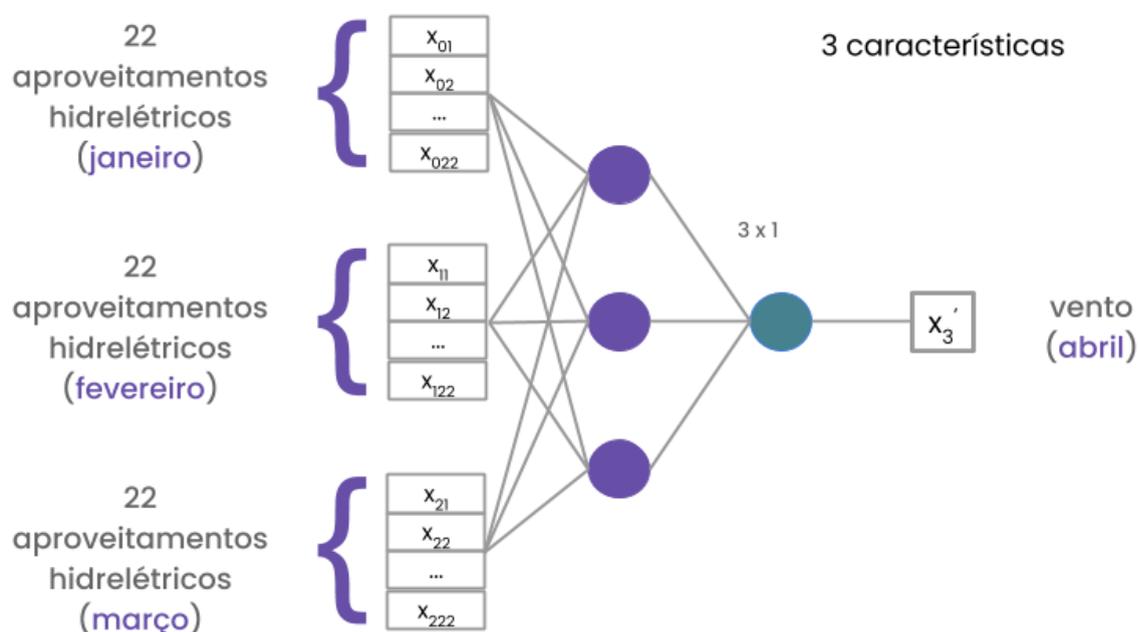


Figura 4.9: Arquitetura da Rede de Representação no modo Multivariado.

FONTE: elaborado pela autora (2021).

A rede foi treinada utilizando a função de custo MSE, o otimizador Adam e 80 épocas. A primeira camada foi otimizada com um conjunto de funções de ativação como a *Sigmoid*, *Tanh*, *ReLU*, *Softmax* e nenhuma, sendo mantida a *Softmax* por apresentar melhores resultados de MSE em testes menores com os modelos VAR e VECM. A última camada (camada de saída) não apresentava função de ativação pois sua saída se referia a um problema de regressão, ou seja, dentre as funções de ativação utilizadas na camada passada, não existia nenhuma que preservasse o intervalo que a saída deveria ter.

4.5 CONSIDERAÇÕES FINAIS

No início desta seção foram apresentadas as bases de dados utilizadas no desenvolvimento da pesquisa, contando com dados de velocidade de vento e com dados de vazão de 22 aproveitamentos hidrelétricos diferentes, respeitando as condições de estacionariedade. Em seguida, foram apresentadas todas as etapas das metodologias implementadas, tanto para o modo Univariado quanto para o Multivariado, e as métricas utilizadas para avaliá-las. Por fim, foi apresentada uma nova proposta de pré-processamento para considerar as informações dos dados anteriores presentes em uma série temporal, técnica chamada de Rede de Representação, que visou aproveitar as informações de pesos e *bias* treinados dentro de uma Rede Neural.

5 EXPERIMENTOS

Nesta seção serão apresentados os experimentos realizados, iniciando com os modelos utilizados em cada modo (Univariado e Multivariado) e seguindo para as otimizações de hiperparâmetros dos modelos e configurações de parâmetros de pré-processamento. Os detalhes sobre os modelos e métodos de pré-processamento já foram apresentados na seção de fundamentação teórica ou na seção de metodologia.

Os modelos escolhidos para realizar as regressões, tanto no modo Univariado quanto no Multivariado, são apresentados na Tabela 5.1. Eles foram divididos em três categorias, sendo AR, MA, ARMA, ARIMA, SARIMA, ARCH, GARCH, *Exponential Smoothing*, VAR e VECM pertencentes ao conjunto de modelos derivados da Teoria de Séries Temporais, Regressão Linear, Regressão Polinomial, Árvore de Regressão, *Bagging*, Florestas Aleatórias e *Gradient Boosting* da categoria de Aprendizado de Máquina e, por fim, o *Perceptron*, MLP, LSTM e GRU advindos da área de Aprendizado Profundo. Exceto na primeira categoria, que necessita de um determinado número de características, todos os modelos de Aprendizado de Máquina e Aprendizado Profundo foram implementados nos dois modos.

Após a implementação de todos os modelos, foi feita uma otimização de hiperparâmetros, com o objetivo de encontrar os melhores modelos em um conjunto de validação composto pelo último mês, ou os últimos 10% de dados (no caso dos modelos de Aprendizado Profundo), do conjunto de treinamento. A otimização foi realizada de 3 formas diferentes, que dependiam da categoria na qual o modelo se encaixava, sendo uma *Grid Search* para os modelos da Teoria de Séries Temporais, o otimizador Skopt para os modelos de Aprendizado de Máquina e o Talos para os modelos de Aprendizado Profundo.

Tabela 5.1: Modelos avaliados no problema de previsão de velocidade de vento.

FONTE: elaborado pela autora (2021).

Categoria	Modelos	Univariado	Multivariado
Teoria de Séries Temporais	AR	✓	✗
	MA	✓	✗
	ARMA	✓	✗
	ARIMA	✓	✗
	SARIMA	✓	✗
	ARCH	✓	✗
	GARCH	✓	✗
	EXP. SMOOTHING	✓	✗
	VAR	✗	✓
	VECM	✗	✓
Aprendizado De Máquina	ÁRVORE DE DECISÃO	✓	✓
	REGRESSÃO LINEAR	✓	✓
	REGRESSÃO POLINOMIAL	✓	✓
	FLORESTAS ALEATÓRIAS	✓	✓
	BAGGING	✓	✓
	GRADIENT BOOSTING	✓	✓
Aprendizado Profundo	PERCEPTRON	✓	✓
	MLP	✓	✓
	LSTM	✓	✓
	GRU	✓	✓

A *Grid Search* (também conhecida como Busca em Grade) é uma técnica de ajuste que tenta calcular os valores ideais de hiperparâmetros. É uma busca exaustiva realizada sobre os valores de parâmetros específicos de um modelo que pode economizar tempo, esforço e recursos (Malik, 2020). De forma simples, considerando um modelo, os hiperparâmetros que se deseja ajustar são separados e, para cada um deles, é definido um intervalo ou uma lista de valores de interesse para serem testados. Então, o método realiza todas as combinações de

valores de hiperparâmetros e treina o modelo com cada uma. A cada iteração da *Grid Search*, uma métrica é calculada para avaliar o quão bom é o conjunto de valores de hiperparâmetros da vez. Esse método de otimização foi aplicado nos modelos: AR, MA, ARMA, ARIMA, SARIMA, ARCH, GARCH, *Exponential Smoothing*, VAR e VECM.

A *Scikit-Optimize*, ou Skopt, é uma biblioteca simples e eficiente para minimizar funções caras e ruidosas em modelos de *Black-Box*. Ela implementa vários métodos para otimização baseados em processos sequenciais (Scikit-Optimize, 2020a). Mais informações de implementações dessas otimizações podem ser encontradas em (Scikit-Optimize, 2020b). A forma de utilizar essa biblioteca também se baseia em definir intervalos ou listas de valores para cada um dos hiperparâmetros de interesse. Os modelos implementados com a Skopt foram: Regressão Linear, Regressão Polinomial, Árvore de Regressão, *Bagging*, Florestas Aleatórias e *Gradient Boosting*.

Por fim, o otimizador Talos foi utilizado nos modelos de Aprendizado Profundo, que são: *Perceptron*, MLP, LSTM e GRU. Ele segue um fluxo de trabalho POD (*Prepare, Optimize, Deploy*) para criar um *pipeline* flexível e eficiente com resultados de previsão próximos ao estado da arte (Xia, 2020). O Talos foi utilizado com o comando de varredura, que testou todas as combinações possíveis para cada experimento. Mais informações sobre as implementações do Talos podem ser encontradas em (Autonomio Talos, 2019).

Na sequência de tabelas abaixo são apresentados os conjuntos de valores utilizados na otimização de cada modelo. Os nomes dos hiperparâmetros e variações de valores foram escolhidos de acordo com os hiperparâmetros das bibliotecas utilizadas nas implementações dos modelos, sendo seus intervalos próximos aos valores padrões recomendados mencionados nas documentações. Mais informações sobre cada hiperparâmetro dos modelos podem ser encontradas nas referências da Tabela 5.2.

Tabela 5.2: Referências detalhadas sobre os hiperparâmetros de cada modelo implementado.

FONTE: elaborado pela autora (2021).

Modelo	Referência detalhada sobre os hiperparâmetros
AR, MA e ARMA	(Statsmodels, 2021a)
ARIMA	(Statsmodels, 2021b)
SARIMA	(Statsmodels, 2021c)
<i>Exponential Smoothing</i>	(Statsmodels, 2021d)
VAR	(Statsmodels, 2021e)
VECM	(Statsmodels, 2021f)
ARCH e GARCH	(Sheppard, 2019)
Regressão Linear	(Scikit-Learn, 2020a)
Regressão Polinomial	(Scikit-Learn, 2020b)
Árvore de Regressão	(Scikit-Learn, 2020c)
<i>Bagging</i>	(Scikit-Learn, 2020d)
Florestas Aleatórias	(Scikit-Learn, 2020e)
<i>Gradient Boosting</i>	(Scikit-Learn, 2020f)
<i>Perceptron</i> e MLP	(Keras, 2021a)
LSTM	(Keras, 2021b)
GRU	(Keras, 2021c)

As tabelas 5.3, 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9 apresentam os hiperparâmetros otimizados de cada modelo implementado. A primeira coluna se refere ao nome do hiperparâmetro, que dependia da biblioteca utilizada na implementação do modelo, a segunda mostra os menores valores e os máximos valores para aquele hiperparâmetro e a terceira coluna indica de quanto em quanto aqueles valores foram incrementados. Por exemplo, no modelo AR, presente na Tabela 5.3 o hiperparâmetro *order* era uma tupla, que iniciava em (0, 0) e seguia até (13, 0), aumentando o primeiro valor da tupla em 1 e o segundo em 0 (constante). Então, as

combinações para esse hiperparâmetro foram: (0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0), (7, 0), (8, 0), (9, 0), (10, 0), (11, 0) e (12, 0). O último valor do intervalo, na segunda coluna, não é considerado, ou seja, são testados todos os valores menores que 13. Como o modelo AR pode ser representado pelo ARMA atribuindo 0 ao valor de q, nesse exemplo, só a ordem de p é alterada.

Tabela 5.3: Intervalos de hiperparâmetros utilizados na otimização dos modelos derivados da Teoria de Séries Temporais do modo Univariado, sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.

FONTE: elaborado pela autora (2021).

AR		
order (p)	(0, 0) ~ (13, 0)	(1, 0)
MA		
order (q)	(0, 1) ~ (0, 13)	(0, 1)
ARMA		
order (p, q)	(1, 1) ~ (13, 13)	(1, 1)
ARIMA		
order (p, d, q)	(0, 1, 0) ~ (10, 10, 10)	(1, 1, 1)
SARIMA		
order (p, d, q)	(0, 0, 0) ~ (2, 2, 2)	(1, 1, 1)
seasonal_order (P, D, Q, S)	(0, 0, 0, 0) ~ (3, 3, 3, 13)	(1, 1, 1, 6)
ARCH / GARCH		
mean	Constant, Zero, LS, AR, ARX, HAR, HARX	-
lags	0 ~ 4	1
order (p, o, q)	(1, 0, 0) ~ (4, 4, 4)	(1, 1, 1)
dist	gaussian, studentst, skewt, ged	-
EXPONENTIAL SMOOTHING (1, 2, 3)		
trend	None, add	-
seasonal	None, add	-
seasonal_periods	2 ~ 13	1

Tabela 5.4: Intervalos de hiperparâmetros utilizados na otimização dos modelos derivados da Teoria de Séries Temporais do modo Multivariado, sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.

FONTE: elaborado pela autora (2021).

VAR		
maxlags	1 ~ 13	1
method	ols	-
ic	None, aic, fpe, hqic	-
trend	c, ct, ctt, nc, n	-

VECM		
k_ar_diff	1 ~ 13	1
coint_rank	1 ~ 13	1
deterministic	nc, co, ci, lo, li	-
seasons	1 ~ 13	1

Tabela 5.5: Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado de Máquina (*Regressão Linear, Regressão Polinomial e Árvore de Regressão*), sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.

FONTE: elaborado pela autora (2021).

REGRESSÃO LINEAR		
fit_intercept	True, False	-

REGRESSÃO POLINOMIAL		
degree	1 ~ 20	1
interaction_only	True, False	-
include_bias	True, False	-
order	C, F	-

ÁRVORE DE REGRESSÃO		
criterion	mse, friedman_mse, mae	-
splitter	best, random	-
max_depth	2 ~ 37	1
min_samples_split	2 ~ 37	1
min_samples_leaf	2 ~ 37	1
max_features	auto, sqrt, log2	-
random_state	None, 3, 12, 21, 28, 36, 42, 60	-
max_leaf_nodes	None, 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	-

Tabela 5.6: Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado de Máquina (*Bagging* e Florestas Aleatórias), sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.

FONTE: elaborado pela autora (2021).

BAGGING		
base_estimator	DecisionTreeRegressor(), SVR(), KNeighborsRegressor(), LinearRegression()	-
n_estimators	2 ~ 100	1
max_features	1 ~ Look Back + 1	1
bootstrap	True, False	1
random_state	0, 1, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60	-

FLORESTAS ALEATÓRIAS		
n_estimators	2 ~ 100	1
criterion	mse, mae	-
max_depth	2 ~ 37	1
min_samples_split	2 ~ 37	1
min_samples_leaf	2 ~ 37	1
max_features	auto, sqrt, log2	-
max_leaf_nodes	None, 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	-
bootstrap	True, False	-
random_state	None, 3, 12, 21, 28, 36, 42, 60	-

Tabela 5.7: Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado de Máquina (*Gradient Boosting*), sendo a primeira coluna o nome do hiperparâmetro, a segunda os limites mínimos e máximos, ou os valores testados, e a terceira o incremento.

FONTE: elaborado pela autora (2021).

GRADIENT BOOSTING		
loss	ls, lad, huber, quantile	-
learning_rate	0.0001, 0.001, 0.05, 0.1, 0.5	-
n_estimators	1 ~ 100	1
subsample	0.05, 0.1, 0.5, 1.0	-
criterion	friedman_mse, mse, mae	-
min_samples_split	2 ~ 37	1
min_samples_leaf	2 ~ 37	1
max_depth	2 ~ 37	1
random_state	None, 3, 12, 21, 28, 36, 42, 60	-
max_features	auto, sqrt, log2, None	-
max_leaf_nodes	None, 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	-

Tabela 5.8: Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado Profundo (*Perceptron* e MLP), sendo a primeira coluna o nome do hiperparâmetro e a segunda os valores testados.

FONTE: elaborado pela autora (2021).

PERCEPTRON	
units	1
activation	relu, sigmoid, softmax, tanh
use_bias	True, False
optimizers	adam, sgd
losses	mean_squared_error, cosine_similarity
batch_size	1
epochs	10, 50, 100, 150
validation_split	0.1

MLP	
units_1	10, 100
activation_1	relu, sigmoid, softmax, tanh
use_bias_1	True, False
units_2	10, 100
activation_2	relu, sigmoid, softmax, tanh
use_bias_2	True, False
optimizers	adam, sgd
losses	mean_squared_error
batch_size	1
epochs	50, 100, 150
validation_split	0.1

Tabela 5.9: Intervalos de hiperparâmetros utilizados na otimização dos modelos de Aprendizado Profundo (LSTM e GRU), sendo a primeira coluna o nome do hiperparâmetro e a segunda os valores testados.

FONTE: elaborado pela autora (2021).

LSTM		GRU	
units_1	50, 100	units_1	50, 100
activation_1	relu, sigmoid, tanh	activation_1	relu, sigmoid, tanh
recurrent_activation_1	relu	recurrent_activation_1	relu
use_bias_1	True	use_bias_1	True
unit_forget_bias_1	True	dropout_1	0.0
dropout_1	0.0, 0.2	recurrent_dropout_1	0.0
recurrent_dropout_1	0.0	go_backwards_1	False
go_backwards_1	False	stateful_1	False
stateful_1	False	unroll_1	False
unroll_1	False	units_2	50, 100
units_2	50, 100	activation_2	relu, sigmoid, tanh
activation_2	relu, sigmoid, tanh	recurrent_activation_2	relu
recurrent_activation_2	relu	use_bias_2	True
use_bias_2	True	dropout_2	0.0
unit_forget_bias_2	True	recurrent_dropout_2	0.0
dropout_2	0.0	go_backwards_2	False
recurrent_dropout_2	0.0	stateful_2	False
go_backwards_2	False	unroll_2	False
stateful_2	False	optimizers	adam
unroll_2	False	losses	mean_squared_error
optimizers	adam	batch_size	1
losses	mean_squared_error	epochs	50, 150
batch_size	1	validation_split	0.1
epochs	50, 150		
validation_split	0.1		

Além da otimização de hiperparâmetros referentes aos modelos, também foi feita uma busca para descobrir qual a melhor combinação de métodos de pré-processamento para cada combinação de valores de hiperparâmetros testados na otimização anterior. Assim, foram feitas todas as otimizações acima para cada combinação de métodos de pré-processamento para todos os modelos. O objetivo dessa etapa foi descobrir a relevância desses métodos em cada modelo e observar como a otimização anterior era afetada por eles. A Tabela 5.10 mostra as combinações de métodos utilizados nos modelos derivados da Teoria de Séries Temporais no modo Univariados e a Tabela 5.11 em todos os outros, incluindo os Multivariados. A diferença entre as duas figuras pode ser explicada pelo fato de que os modelos da Tabela 5.10

são especificamente Univariados, então o *Look Back*, que aumenta o número de características de acordo com o número de meses informado, precisa ser obrigatoriamente igual a 1.

Tabela 5.10: Intervalos de hiperparâmetros do uso das técnicas de pré-processamento no caso dos modelos derivados da Teoria de Séries Temporais do modo Univariado.

FONTE: elaborado pela autora (2021).

HIPERPARÂMETROS DE PRÉ-PROCESSAMENTO		
stationary	True, False	Torna, ou não, a série em uma série estacionária
scale	True, False	Aplica, ou não, escala MinMax nos dados
lookBack	1	Não adiciona Look Back maior do que 1 em nenhum caso
walkForward	True, False	Utiliza, ou não, o método de Walk Forward
representationNet	True, False	Utiliza, ou não, a Rede de Representação
activation	True, False	Utiliza, ou não, a função de ativação na Rede de Representação
lookBackNet	1, 3, 6, 9, 12	Tamanho de Look Back utilizado na Rede de Representação

Tabela 5.11: Intervalos de hiperparâmetros do uso das técnicas de pré-processamento no caso dos modelos de Aprendizado de Máquina e Aprendizado Profundo Univariados e todos os Multivariados. Nesses hiperparâmetros, a Rede de Representação só é utilizada quando não há o uso de *Look Back* maior que 1, pois a rede considera seu próprio *Look Back*.

FONTE: elaborado pela autora (2021).

HIPERPARÂMETROS DE PRÉ-PROCESSAMENTO		
stationary	True, False	Torna, ou não, a série em uma série estacionária
scale	True, False	Aplica, ou não, escala MinMax nos dados
lookBack	1, 3, 6, 9, 12	Aplica Look Back nos dados
walkForward	True, False	Utiliza, ou não, o método de Walk Forward
representationNet	True, False	Utiliza, ou não, a Rede de Representação ("True" só quando lookBack = 1)
activation	True, False	Utiliza, ou não, a função de ativação na Rede de Representação
lookBackNet	1, 3, 6, 9, 12	Tamanho de Look Back utilizado na Rede de Representação

Vale ressaltar que as otimizações foram realizadas nos dados não estacionários, pois o novo conjunto de dados com as condições de estacionariedade respeitadas, mencionado na seção 4.1, ainda não tinha sido disponibilizado para esta pesquisa. Além disso, mesmo com a menção dos gráficos ACF e PACF na seção 2.2.1, e na possibilidade de utilizá-los para encontrar os melhores hiperparâmetros para os modelos derivados da Teoria de Séries Temporais, para que as otimizações mantivessem um padrão para as 3 categorias de modelos,

foi utilizada uma *Grid Search* para encontrar as melhores configurações dentro de um conjunto de valores pré-definidos de hiperparâmetros.

Como considerações finais, nesta seção foram apresentados os experimentos realizados, iniciando com a exposição dos modelos utilizados no modo Univariado e Multivariado, seguindo para a descrição das duas otimizações realizadas. A primeira otimização foi referente aos hiperparâmetros de cada modelo, sendo o otimizador dependente da biblioteca utilizada na implementação de cada modelo. Para os modelos derivados da Teoria de Séries Temporais, foi feita uma *Grid Search*, para os modelos de Aprendizado de Máquina foi utilizado o otimizador Skopt e, por fim, os de Aprendizado Profundo foram otimizados com o Talos. Na segunda otimização, que buscou a melhor combinação de métodos de pré-processamento para cada configuração de hiperparâmetros para todos os modelos, foi realizada uma *Grid Search*. Os resultados dos experimentos serão apresentados na próxima seção.

6 RESULTADOS

Após a otimização realizada no passo anterior, as cinco configurações com os resultados de menor RMSE foram selecionadas para cada modelo. Para cada um desses resultados foram salvos seus sete parâmetros de pré-processamento e os hiperparâmetros do modelo. Com todas essas configurações definidas, a análise dos resultados foi dividida em 3 etapas. A primeira etapa teve como objetivo a observação do melhor resultado de cada modelo, tanto no modo Univariado quanto no Multivariado. A segunda análise visou investigar o impacto de cada passo de pré-processamento no desempenho dos modelos. Por fim, a última parte retreinou os dois melhores modelos, para cada modo, encontrados pela primeira etapa de análise, adicionando mais épocas, aumentando o horizonte de previsão de 12 meses para 60 meses e experimentando os modelos em dados adicionais. As três etapas serão apresentadas com mais detalhes nas próximas seções.

6.1 ANÁLISE DE DESEMPENHO DE TODOS OS MODELOS

Esta etapa teve a intenção de analisar os resultados obtidos no passo de otimização nos dados ainda não vistos que compõem a base de teste. Dessa forma, foram selecionados os sete parâmetros de pré-processamento e os melhores hiperparâmetros dos cinco resultados com RMSEs mais baixos, para cada modelo. Considerando somente os dados de Afrânio, todos os modelos foram treinados com a base de treinamento que consistiu em dados iniciando em janeiro de 1994 e indo até dezembro de 2012, utilizando as configurações selecionadas. Então, após o treinamento, os modelos foram avaliados na base de teste, que foi composta pelos dados de janeiro de 2013 até dezembro de 2013. O melhor resultado de cada modelo, para cada modo, foi escolhido utilizando a métrica do RMSE.

Para que fosse possível decidir se um modelo obteve previsões consideradas boas, foi definido um modelo *baseline* para ser comparado com todos os outros modelos dos dois modos. O modelo escolhido foi o PMF (*Persistence Model Forecast*) que se baseia na ideia de utilizar o valor de velocidade de vento do tempo atual como previsão para o próximo tempo, como foi apresentado na seção 2.3.1.1. O gráfico dos valores esperados e obtidos pelo modelo PMF é apresentado na Figura 6.1 abaixo, juntamente com seu RMSE. Os gráficos, resultados e a análise, tanto do modo Univariado quanto do Multivariado, serão apresentados nas duas próximas subseções e os melhores hiperparâmetros encontrados para todos os modelos estão presentes no Apêndice A.

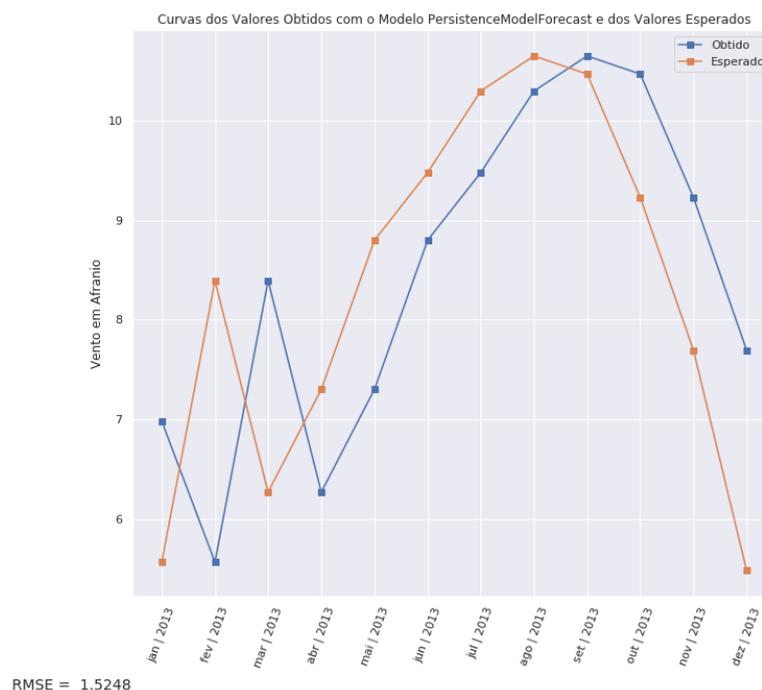


Figura 6.1: Valores esperados e obtidos com o modelo *baseline* PMF para a série temporal de Afrânio.

FONTE: elaborado pela autora (2021).

6.1.1 RESULTADOS DOS MODELOS UNIVARIADOS

As métricas de MSE, RMSE e MAPE para todos os modelos (em m/s), com os respectivos valores de pré-processamento, são apresentadas pela Tabela 6.1. Por questões de espaço, os nomes de alguns modelos foram abreviados, sendo PMF o modelo base *Persistence Model Forecast*; ARC-E o modelo ARCH com adição da transformação de estacionariedade; GAR-E o modelo GARCH com adição da estacionariedade; ES (1) o modelo *Simple Exponential Smoothing*; ES (2) o *Double Exponential Smoothing*; ES (3) o *Triple Exponential Smoothing*; RL o modelo de Regressão Linear; RP o modelo de Regressão Polinomial; ARR o modelo de Árvore de Regressão; BG o modelo *Bagging*; FA o modelo de Florestas Aleatórias; GB o modelo de *Gradient Boosting* e PCT o modelo *Perceptron*.

Tabela 6.1: Parâmetros de pré-processamento e as métricas (MSE, RMSE, MAPE) obtidas por cada modelo do modo Univariado.

FONTE: elaborado pela autora (2021).

Modelo	Estacionariedade	Escala	Look Back	Walk Forw.	Rede de Repr.	Função de Ativ.	Look Back Net	MSE	RMSE	MAPE
PMF	0	0	0	0	0	0	0	2,3251	1,5248	0,1817
AR	0	1	0	1	1	0	6	1,1318	1,0639	0,1169
MA	0	1	0	1	1	0	12	1,5949	1,2629	0,1375
ARMA	0	1	0	1	0	0	0	0,7966	0,8925	0,0879
ARIMA	0	1	0	1	1	0	3	1,7356	1,3174	0,1595
SARIMA	0	0	0	1	0	0	0	0,8530	0,9236	0,1076
ARCH	0	1	0	1	1	0	12	2,6255	1,6203	0,1929
ARC-E	1	1	0	1	1	0	12	2,2784	1,5094	0,1716
GARCH	0	1	0	1	1	0	12	2,4163	1,5544	0,1857
GAR-E	1	1	0	1	1	0	12	2,5875	1,6086	0,1841
ES (1)	0	0	0	1	0	0	0	2,3251	1,5248	0,1817
ES (2)	0	0	0	1	1	1	0	79,8882	8,9380	1,0782
ES (3)	0	1	0	1	0	0	0	$\approx 1,73 \times 10^{29}$	$\approx 1,31 \times 10^{29}$	$\approx 9,82 \times 10^{29}$
RL	0	1	12	1	0	0	0	0,8840	0,9402	0,1005
RP	0	0	6	0	0	0	0	0,7487	0,8653	0,0929
ARR	0	1	6	0	0	0	0	0,5791	0,7610	0,0832
BG	0	1	6	1	0	0	0	0,5846	0,7646	0,0735
FA	0	1	12	0	0	0	0	0,5519	0,7429	0,0704
GB	0	1	12	1	0	0	0	0,4824	0,6946	0,0609
PCT	0	1	12	1	0	0	0	0,8913	0,9441	0,0982
MLP	0	0	12	1	0	0	0	0,5494	0,7412	0,0808
LSTM	0	0	6	1	0	0	0	0,4997	0,7069	0,0714
GRU	0	0	6	1	0	0	0	0,4541	0,6739	0,0718

Analisando os dados, é possível perceber que o RMSE se manteve abaixo de 1 na maioria dos casos, o que é um bom indicativo, considerando que essa métrica penaliza previsões muito distantes do valor real. Como os valores do RMSE dependem muito da relevância que o erro tem no problema que está sendo resolvido, é possível dizer que valores abaixo de 1, no contexto desta pesquisa, são bons resultados, pois o objetivo foi definido como uma previsão mais geral da velocidade de vento, em que não há a necessidade de uma regressão tão exata como em áreas médicas, por exemplo. Além disso, com exceção dos modelos ARCH, GARCH e as variações do *Exponential Smoothing*, todos os modelos atingiram métricas com valores melhores do que o *baseline*, o que indica que os modelos conseguiram capturar mais informações, além do valor último *lag*, para realizar as previsões.

Outra evidência do bom desempenho dos modelos foi apresentada pelo MAPE, que expressa a acurácia do erro em porcentagem. Os valores passaram de 10% em poucos casos, indicando que os modelos, em geral, erram em menos de 10% dos dados. De acordo com (Moreno et. al, 2013), valores de MAPE menores do que 10% indicam previsões altamente precisas, entre 10-20% são previsões boas, entre 20-50% são razoáveis e maiores do que 50% são previsões imprecisas. Essas informações são mais um indicativo de que os modelos alcançaram bons resultados ao realizar as previsões. Tanto o MAPE quanto o RMSE apresentaram resultados altos em alguns modelos, como o ARCH, GARCH e *Exponential Smoothing*. Posteriormente será feita uma análise isolada para investigar estes casos.

Como foram implementados muitos modelos, apenas os gráficos de dois de cada uma das categorias (modelos derivados da Teoria de Séries Temporais, modelos de Aprendizado de Máquina e modelos de Aprendizado Profundo) com menores RMSEs serão apresentados nesta seção, mas todos os gráficos são apresentados no Apêndice B. Abaixo, na Figura 6.2, são mostrados os gráficos dos valores preditos e esperados, juntamente com o RMSE, dos modelos ARMA, SARIMA, Árvore de Regressão, *Gradient Boosting*, LSTM e GRU.

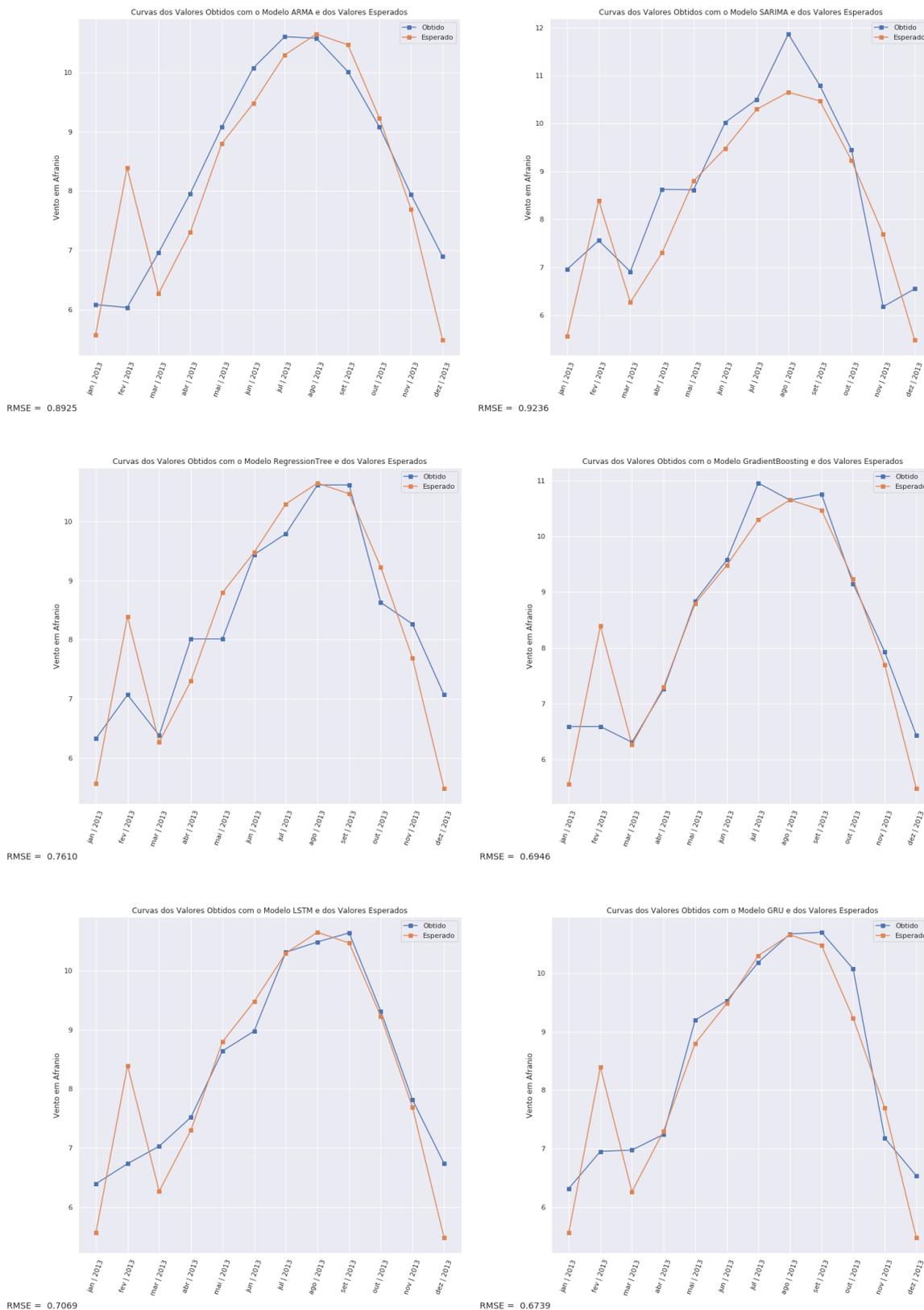


Figura 6.2: Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos modelos ARMA, SARIMA, Árvore de Regressão, *Gradient Boosting*, LSTM e GRU, no modo Univariado, com horizonte de previsão de 12 meses.

FONTE: elaborado pela autora (2021).

Ao observar os gráficos, é possível perceber que todos os modelos conseguem aprender a prever a segunda curva dos dados, que apresenta inclinações mais suaves, indo de março até novembro. Já o primeiro pico, presente nos meses de janeiro e fevereiro, se mostrou um obstáculo para os modelos. Alguns modelos suavizam esse pico, como no caso do ARMA e da LSTM e outros tentam prever o pico, mas erram na altura da velocidade de vento esperada, como o SARIMA, a Árvore de Regressão, o *Gradient Boosting* e a GRU. Em modelos com RMSEs maiores, como AR, Regressão Linear e *Perceptron*, existiram casos em que a previsão do pico foi deslocada para a direita, aparentando ser uma série com atraso. A diversidade de previsões para o primeiro pico dos dados de Afrânio pode ser justificada pela alta dispersão apresentada por essa série temporal, como é possível observar na Figura 6.3.

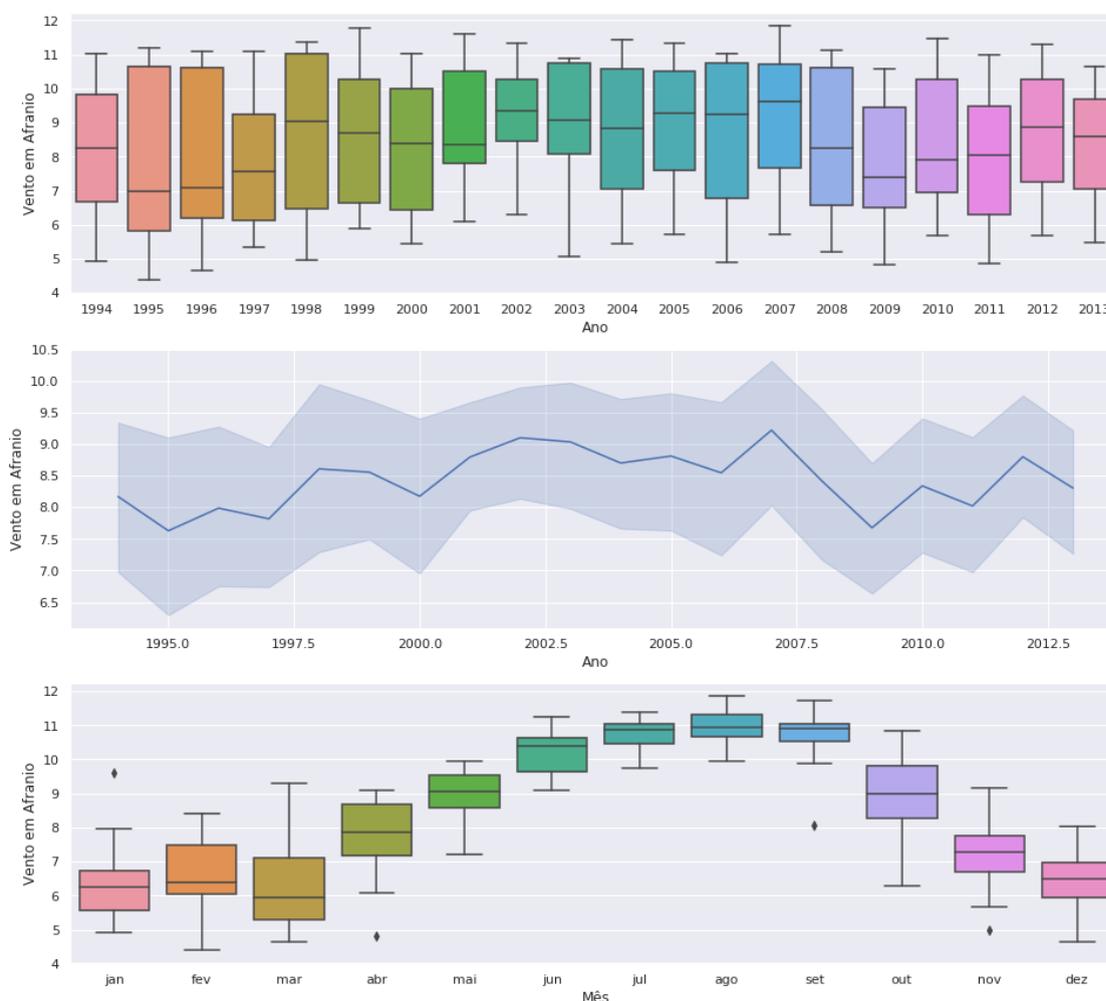


Figura 6.3: *Boxplots* anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e *boxplot* mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Afrânio.

FONTE: elaborado pela autora (2021).

Observando os dois primeiros gráficos, referentes aos *boxplots* anuais, agregando os 12 meses de cada ano, e o espectro de dispersão da série temporal de velocidade de vento de Afrânio ao longo dos anos, é possível ver que a série tem uma grande variação de valores em todos os anos. Considerando as variações dos anos de 1994 até 2012, que compõem a base de treinamento, é possível supor que os modelos tiveram poucos dados para conseguir aprender todas essas disparidades.

Por fim, analisando o último gráfico, dos *boxplots* mensais, agregando todos os anos, há uma grande dispersão nos meses de janeiro a abril, com valores de mínimo e máximo bem distantes da média, além da presença de alguns *outliers*. Esses fatores podem ter influenciado nas previsões do primeiro pico dos dados, ocasionando a suavização próxima à média realizada pelos modelos. Já no período de maio até setembro a série mantém uma baixa dispersão, voltando a variar de outubro a dezembro. Esse período de baixa variação corresponde exatamente à segunda curva com inclinações mais suaves mencionadas anteriormente, onde os modelos conseguiram as melhores previsões. Assim, é possível concluir que as áreas de baixa variação contribuíram para as previsões dos modelos e, de forma semelhante, as áreas com altas dispersões acabaram atrapalhando os modelos, resultando em previsões próximas à média.

A última etapa desta análise se concentrou nos modelos que apresentaram os piores valores de RMSE, que são: o ARCH, o GARCH e as três variações do *Exponential Smoothing*. Considerando os dois primeiros, ao observar os gráficos de valores preditos e obtidos foi possível perceber que as curvas de previsão obtidas se assemelhavam com curvas médias das séries, como é apresentado pelos gráficos à esquerda na Figura 6.4.

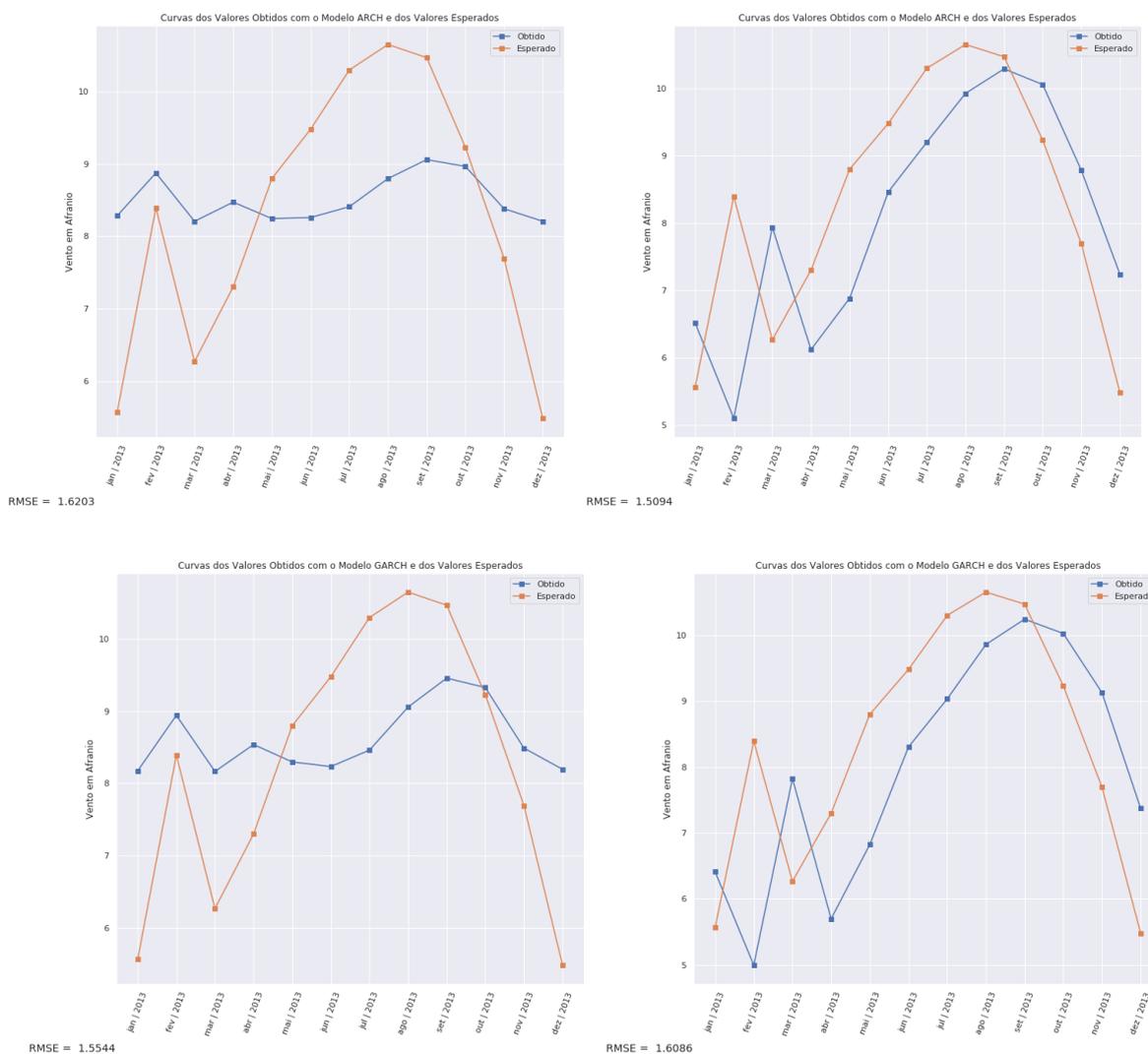


Figura 6.4: Gráficos das curvas obtidas (em azul) e esperadas (em alaranjado) dos modelos ARCH e GARCH sem o uso da estacionariedade (esquerda) e com o uso da estacionariedade (direita).

FONTE: elaborado pela autora (2021).

Como mencionado na seção da base de dados, na metade desta pesquisa foram recebidos dados que respeitavam as condições de estacionariedade e, dessa forma, a aplicação da transformação de estacionariedade foi removida dos experimentos finais. Mas por questões de estudo, a transformação de estacionariedade foi aplicada aos dados de Afrânio, os modelos foram retreinados e as previsões da base de teste obtidas. Como resultado, foi descoberto que, aplicando a transformação nos dados, os dois modelos conseguiram obter resultados visualmente mais próximos à série esperada, mas bem deslocados. Isso pode indicar que os dados utilizados não respeitam todas as condições de estacionariedade, o que dificultou a previsão.

Como a maioria dos modelos conseguiu uma boa previsão, a transformação de estacionariedade continuou a ser desconsiderada para gerar os dados finais deste trabalho, mas

serviu para mostrar que, mesmo respeitando boa parte das condições de estacionariedade, existem alguns casos em que a estacionariedade não é totalmente garantida.

Por fim, as variações do modelo *Exponential Smoothing* apresentaram resultados bem insatisfatórios em relação aos outros modelos. A Figura 6.5 mostra os resultados obtidos com o *Simple*, *Double* e o *Triple Exponential Smoothing*, respectivamente. O primeiro gráfico apresenta resultados deslocados, mas ainda conseguiu capturar o comportamento geral da série. Já os modelos *Double* e *Triple* obtiveram resultados bem longe do esperado. Isso pode ter ocorrido pelo fato da sazonalidade da série ser complexa, pois a dispersão da série é alta, e por ser difícil de definir uma tendência significativa (ascendente ou decrescente), já que os valores sobem e descem com bastante frequência.

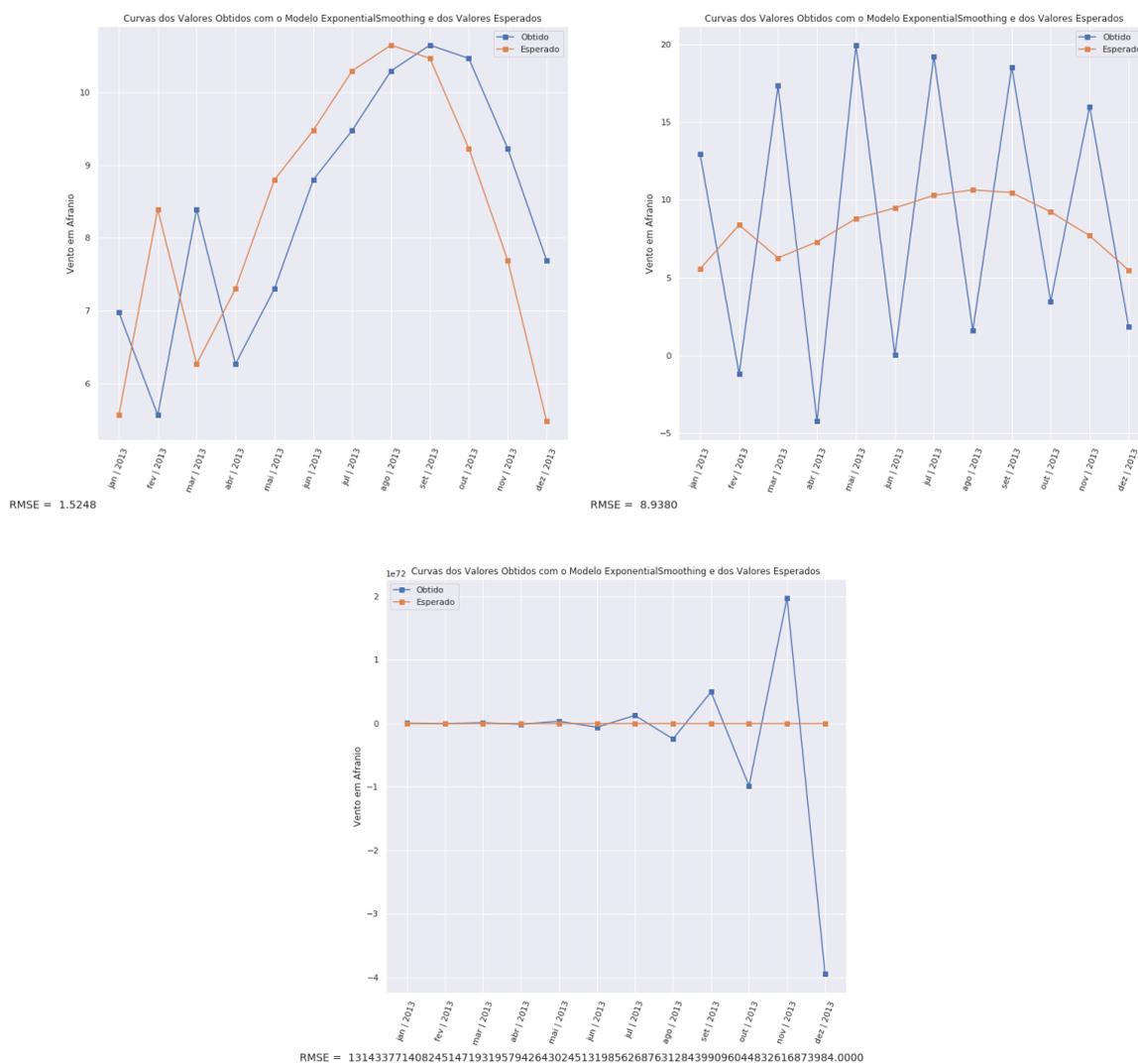


Figura 6.5: Gráficos das curvas esperadas e obtidas pelas três variações do modelo *Exponential Smoothing*, nos dados de Afrânio, no modo Univariado e com horizonte de previsão de 12 meses.

FONTE: elaborado pela autora (2021).

6.1.2 RESULTADOS DOS MODELOS MULTIVARIADOS

As mesmas métricas presentes nos modelos Univariados foram calculadas nos modelos Multivariados e os resultados são apresentados na Tabela 6.2. As abreviações seguem as mesmas da Tabela 6.1.

Tabela 6.2: Parâmetros de pré-processamento e as métricas (MSE, RMSE, MAPE) obtidas por cada modelo do modo Multivariado.

FONTE: elaborada pela autora (2021).

Modelo	Estacionariedade	Escala	Look Back	Walk Forw.	Rede de Repr.	Função de Ativ.	Look Back Net	MSE	RMSE	MAPE
PMF	0	0	0	0	0	0	0	2,3251	1,5248	0,1817
VAR	0	1	0	1	1	0	3	0,6087	0,7802	0,0799
VECM	0	0	0	1	1	0	6	0,7096	0,8424	0,0896
RL	0	0	0	1	0	0	0	0,6718	0,8196	0,0839
RP	-	-	-	-	-	-	-	-	-	-
ARR	0	0	3	0	0	0	0	0,5611	0,7491	0,0730
BG	0	1	3	0	0	0	0	0,6646	0,8152	0,0812
FA	0	1	3	0	0	0	0	0,5561	0,7457	0,0639
GB	0	0	3	0	0	0	0	0,5626	0,7501	0,0720
PCT	0	1	0	0	0	0	0	0,5262	0,7254	0,0844
MLP	0	1	6	0	0	0	0	0,6026	0,7763	0,0741
LSTM	0	1	3	1	0	0	0	0,3766	0,6137	0,0699
GRU	0	1	0	1	0	0	0	0,4740	0,6885	0,0726

Ao observar a tabela acima, é possível notar que faltam os resultados referentes ao modelo de Regressão Polinomial. Este modelo foi implementado para o modo Multivariado, mas ao treiná-lo durante a otimização o limite de memória foi excedido devido ao imenso tamanho do polinômio. Dessa forma, não foi possível trazer os resultados deste modelo.

Nesta modalidade, os RMSEs não passam de 0,85 e o MAPE se mantém próximo a 0,07, informando que os modelos erram somente em 7% dos dados e, de acordo com (Moreno et. al, 2013), isso indica que as predições obtidas são altamente precisas. Como mencionado na análise dos resultados do modo Univariado, estes resultados indicam um bom desempenho dos modelos, considerando o cenário do problema, além de todos os modelos apresentarem

resultados muito melhores do que o *baseline*. Mesmo utilizando séries temporais de vazão para prever outra série temporal, referente à velocidade de vento em Afrânio, todos os modelos conseguiram bons resultados, se destacando a LSTM e GRU como melhores e o VECM e Regressão Linear como piores modelos para solucionar o problema.

Dos 12 modelos implementados, foram escolhidos 2 modelos de cada categoria (modelos derivados da Teoria de Séries Temporais, modelos de Aprendizado de Máquina e modelos de Aprendizado Profundo) para terem seus gráficos (valores obtidos e esperados) apresentados nesta seção, mas todos os gráficos podem ser conferidos no Apêndice B. Os modelos foram selecionados de acordo com o RMSE, resultando no subconjunto dos modelos VAR, VECM, Árvore de Regressão, Florestas Aleatórias, LSTM e GRU. Os gráficos são apresentados na Figura 6.6.

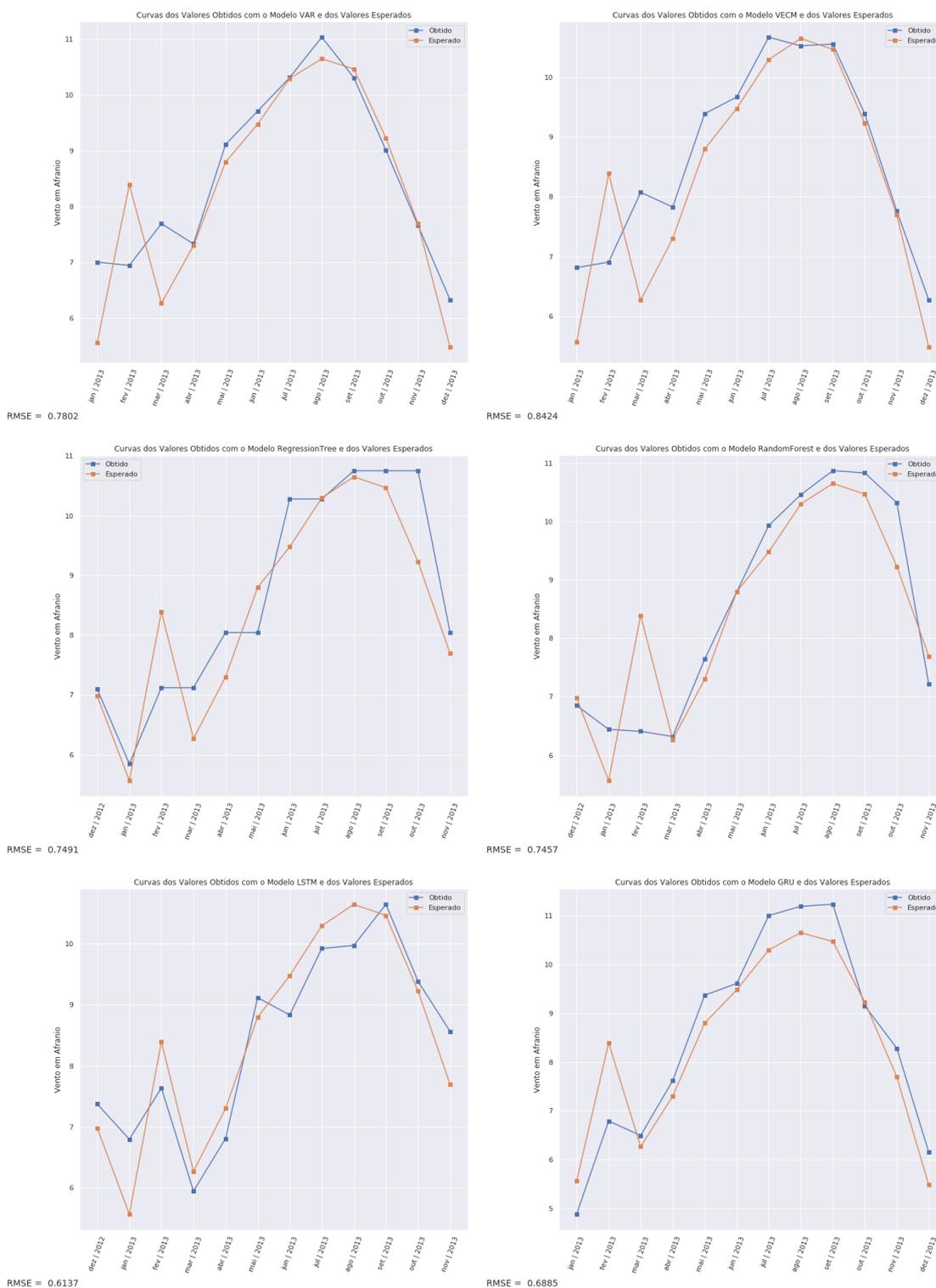


Figura 6.6: Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos modelos VAR, VECM, Árvore de Regressão, Florestas Aleatórias, LSTM e GRU, no modo Multivariado, com horizonte de previsão de 12 meses.

FONTE: elaborado pela autora (2021).

As características das curvas se assemelham com as curvas do modo Univariado, conseguindo boas previsões para a segunda curva, com inclinações mais suaves, e apresentando problemas no primeiro pico de dados. Como os dados utilizados nestes modelos são os dados de Afrânio, a mesma justificativa sobre a alta dispersão dos dados de janeiro a abril pode ser utilizada para explicar as suavizações e erros de altura do pico, e a baixa dispersão entre maio e outubro para esclarecer o bom desempenho dos modelos na segunda curva.

Ao analisar os gráficos dos modelos Multivariados, é possível observar que a série esperada muda o padrão de janeiro a dezembro nos modelos de Árvore de Regressão, Florestas Aleatórias e LSTM. Essa mudança ocorre devido ao uso do *Look Back* com Janela Rolante central com agregação por média. A Figura 6.7 apresenta um exemplo de *Look Back* igual a 3 em uma série temporal simples. Durante o pré-processamento, se o *Look Back* for maior do que 1, tanto as séries utilizadas como características quanto a série *target* precisam ser ajustadas. Por exemplo, na figura, inicialmente há uma série *target* com 6 observações. Quando o *Look Back* de 3 unidades é aplicado aos dados de características, uma janela rolante irá deslizar pelos dados, agrupando os valores dentro da janela pela média. Já na série *target*, a janela também desliza, mas ao invés de fazer uma média dos valores, a observação central é escolhida para representar aquela janela naquele instante (para *Look Back* par, o valor escolhido é aquele mais à direita do meio).

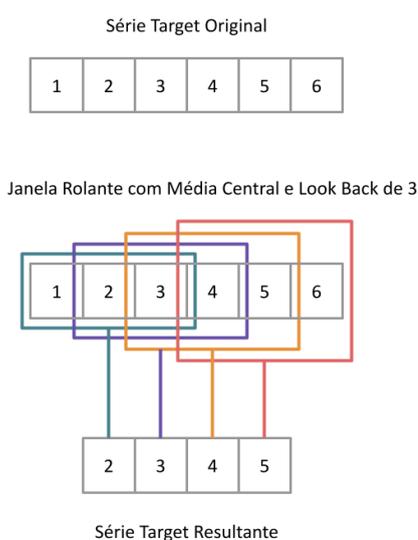


Figura 6.7: Exemplo do impacto da aplicação de *Look Back* utilizando a Janela Rolante de tamanho 3 com agregação por média.

FONTE: elaborado pela autora (2021).

Na Figura 6.7, no primeiro instante, a janela azul passa a ser representada pelo valor central, que é 2. Na janela roxa, o valor 3 é escolhido e assim por diante. Ao chegar na janela vermelha, equivalente ao último instante, o valor escolhido é o 5. Assim, o valor 6 é removido da série *target*. Esse efeito aparece nos 3 modelos mencionados, que utilizam *Look Back* de 3 unidades. Ao compará-los com o gráfico do modelo VAR, que não utiliza *Look Back* maior do que 1, o eixo x muda de janeiro a dezembro de 2013 (no caso do VAR) para dezembro de 2012 a novembro de 2013. Para valores de *Look Back* iguais a 3, 6, 9, e 12, a quantidade de valores perdidos no final da série *target* é de 1, 2, 4 e 5 respectivamente.

O *Look Back* pode impactar nos valores finais da série, mas ajuda a generalizar e captar características presentes em vários meses para predizer um determinado mês, o que pode trazer muito mais significado à predição. Uma solução para não perder os últimos valores da série seria, ao invés de manter o valor central da janela, escolher o valor mais a direita. O efeito de perda de valores da série *target* só acontece no modo Multivariado, pois a Janela Rolante foi uma forma de substituir o *Look Back* original, presente no modo Univariado, que concatena os valores dos meses anteriores.

6.2 ANÁLISE DAS ETAPAS DE PRÉ-PROCESSAMENTO

A segunda análise realizada nos resultados foi a da influência das etapas de pré-processamento dos dados. A ideia por trás desta análise foi mostrar como o uso de alguns pré-processamentos influenciam na obtenção de resultados melhores ou piores. Visando demonstrar o impacto de cada pré-processamento de forma isolada, serão apresentados alguns casos de modelos variados, sem a preocupação de focar somente nos modelos com os RMSEs mais baixos.

Para que as comparações fossem coerentes, os geradores de valores aleatórios de todo o ambiente e dos modelos foram definidos para terem a mesma semente, possibilitando gerar os mesmos resultados em todas as execuções para uma mesma configuração. Dessa forma, foi possível obter comparações justas ao mudar os valores de pré-processamento.

6.2.1 USO DA ESTACIONARIEDADE

O primeiro pré-processamento observado foi a estacionariedade, definida na seção 2.2.5. Mesmo desconsiderando a estacionariedade nos resultados finais, foram realizados alguns experimentos para exemplificar a importância de garantir todas as condições de estacionariedade. Como exemplo, foi escolhido o modelo AR, apresentado na Figura 6.8.

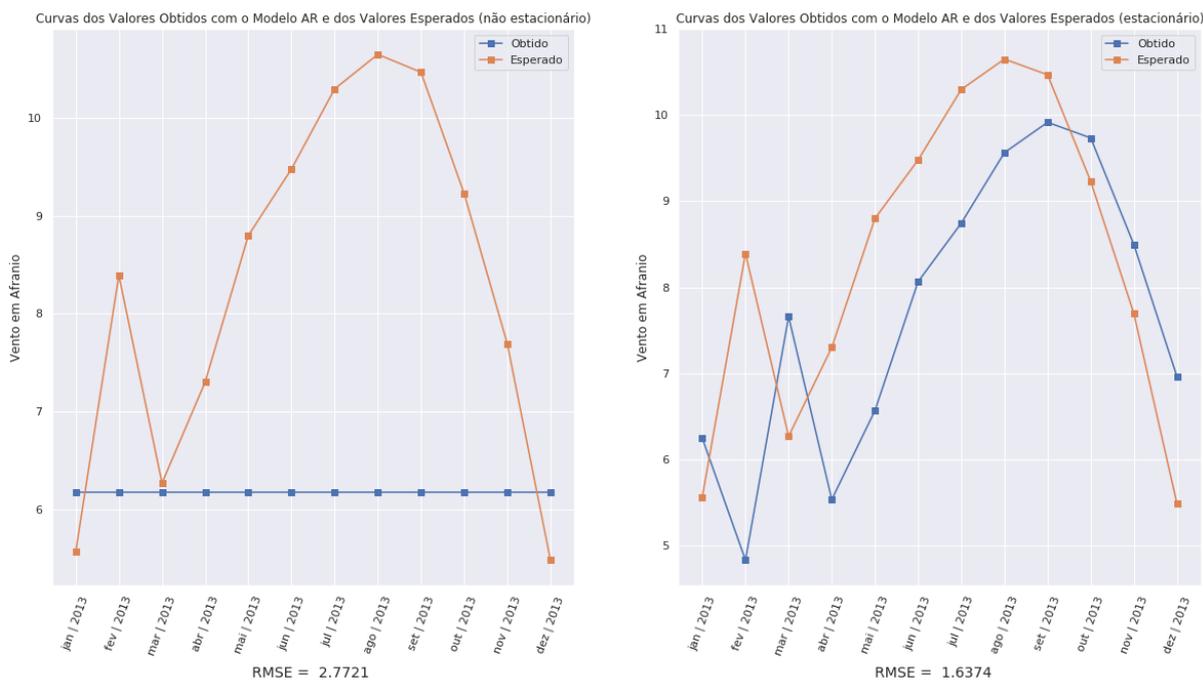


Figura 6.8: Aplicação da estacionariedade no modelo AR (modo Univariado).

FONTE: elaborado pela autora (2021).

O gráfico da esquerda representa os valores obtidos (em azul) e esperados (em alaranjado) do modelo AR nos dados de Afrânio, sem o uso de pré-processamento. Já o gráfico da direita também apresenta os valores esperados e preditos do modelo AR, com a única diferença de que a transformação de estacionariedade foi aplicada aos dados antes do treinamento do modelo. É possível observar que a transformação permitiu uma predição com as características da série temporal esperada, mas com bastante deslocamento. Já os resultados sem a transformação resultaram em uma série constante, sem relação com a série esperada. Mesmo os dois casos apresentando RMSEs altos, só com a aplicação da estacionariedade foi possível reduzir o RMSE de 2,7721 para 1,6374.

Como já foi apresentado, mesmo com os dados não garantindo todas as condições de estacionariedade, com outras configurações, foi possível obter um bom resultado, o que pode indicar que as condições garantidas por esses dados são o suficiente para este modelo fazer as predições.

Nos modelos Multivariados o impacto da transformação de estacionariedade não foi tão extremo quanto no modo Univariado, como é apresentado pela Figura 6.9. O modelo escolhido foi o VAR, mantendo as mesmas configurações, adicionando apenas a transformação de estacionariedade do primeiro para o segundo gráfico. É possível observar

que ocorreu uma aproximação nas laterais da segunda curva no segundo gráfico, principalmente na última observação, reduzindo o RMSE de 0,9864 para 0,9149.

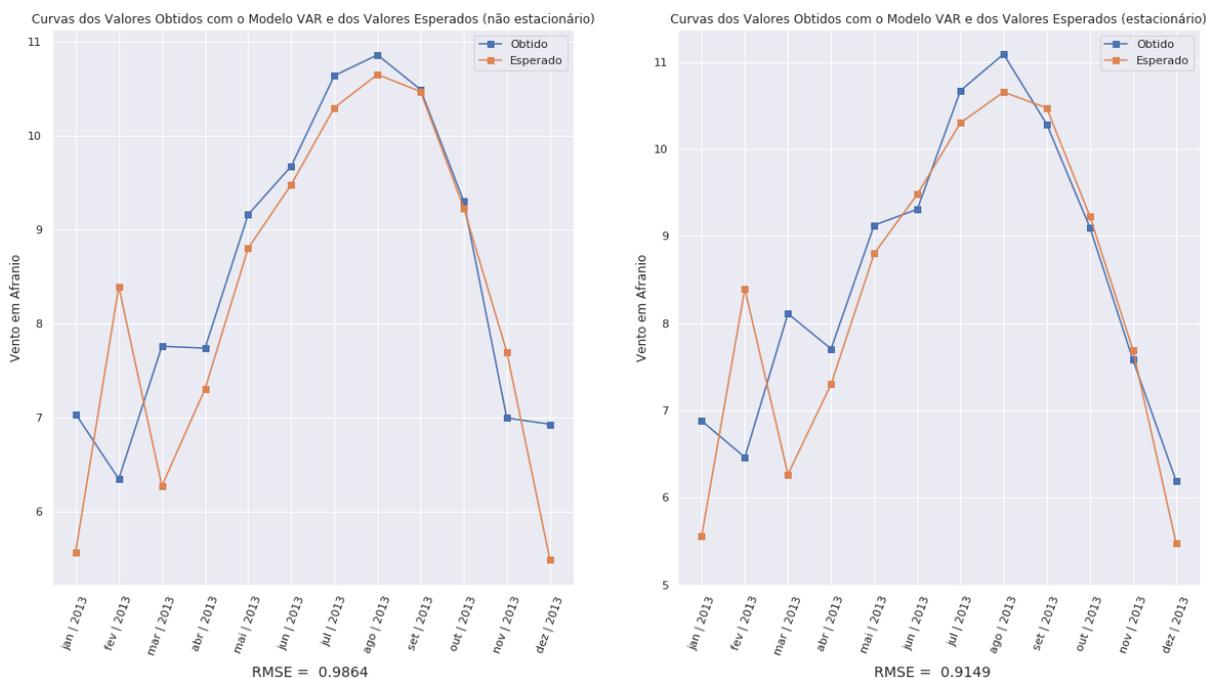


Figura 6.9: Aplicação da estacionariedade no modelo VAR (modo Multivariado).

FONTE: elaborado pela autora (2021).

6.2.2 USO DE ESCALA

O segundo pré-processamento observado foi a escala, apresentada na seção 2.2.10. Dentre os resultados observados, foi possível perceber que a escala fez bastante diferença nos modelos de Aprendizado de Máquina e Aprendizado Profundo. Essa mudança pode ser justificada pela maioria dos modelos de Aprendizado de Máquina implementados depender de distâncias entre os pontos, em que a magnitude tem grande relevância. Nos casos dos algoritmos de Aprendizado Profundo que utilizam gradientes, o uso de escala pode contribuir para uma convergência mais rápida, resultando em melhores métricas.

Abaixo, na Figura 6.10, foi selecionado um exemplo do modelo *Bagging* com SVR, no modo Univariado, em duas situações diferentes: sem escala (gráfico da esquerda) e com escala (gráfico da direita). É possível observar que, no primeiro pico, o uso da escala contribuiu para suavizar as previsões e, na segunda curva, os valores se aproximaram da série esperada, com picos mais baixos no topo da curva. Alterando somente o uso da escala, o RMSE mudou de 0.8079 para 0.7646.

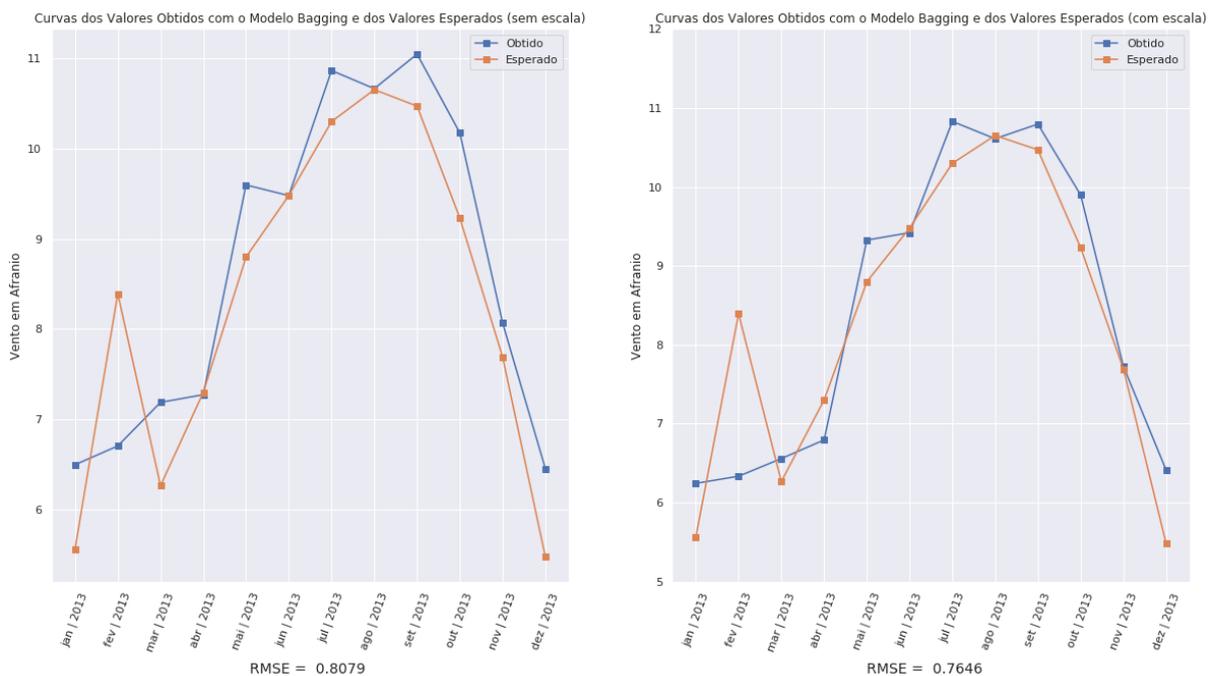


Figura 6.10: Aplicação da escala no modelo *Bagging* (modo Univariado).

FONTE: elaborado pela autora (2021).

No modo Multivariado houve uma mudança mais extrema ao utilizar o modelo GRU. Inicialmente, sem escala, o modelo parece não ter convergido o suficiente, resultando em previsões distantes da série esperada (gráfico da esquerda na Figura 6.11). Ao aplicar a escala (gráfico da direita na Figura 6.11), a curva obtida se aproxima bastante da curva esperada, exceto no primeiro pico, semelhante ao caso dos outros modelos. O impacto da escala pode ser quantificado pelo RMSE, que caiu de 4,0851 para 0,9373.

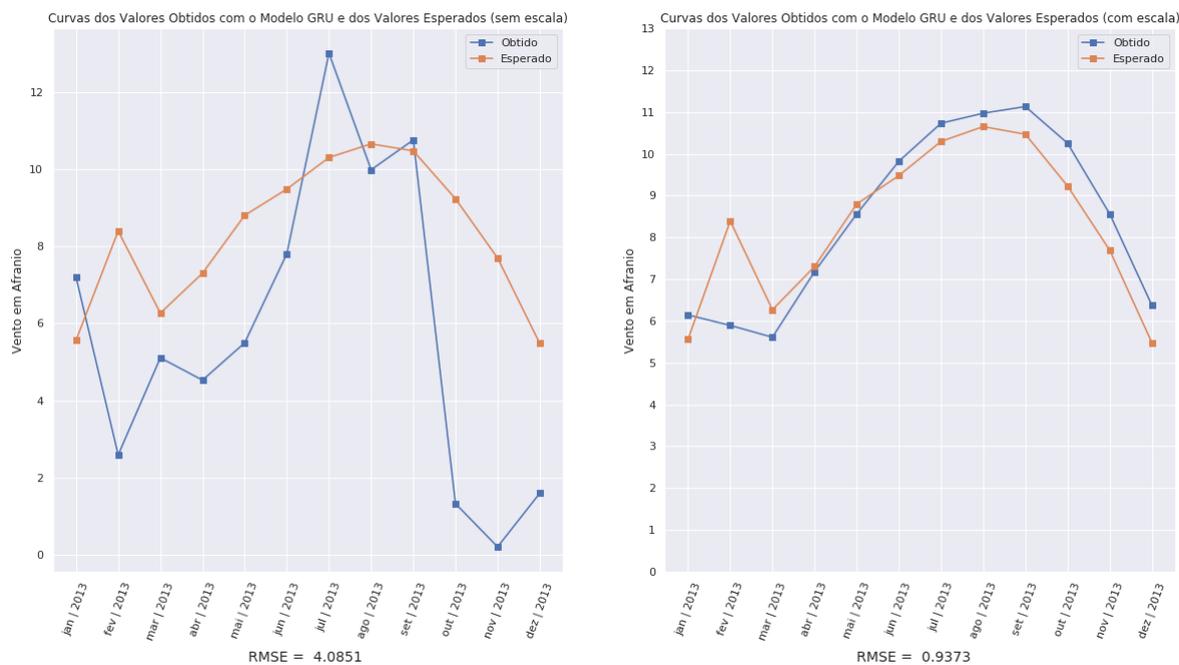


Figura 6.11: Aplicação da escala no modelo GRU (modo Multivariado).

FONTE: elaborado pela autora (2021).

6.2.3 USO DE LOOK BACK

A terceira etapa foi analisar os diferentes valores de *Look Back* utilizados, conceito explicado na seção 2.2.9. O modelo de exemplo escolhido para o modo Univariado foi o de Florestas Aleatórias com escala. Os resultados para os valores de *Look Back* de 0, 3, 6, 9 e 12 são apresentados pela Figura 6.12. Conforme o valor de *Look Back* aumenta, é possível perceber que o pico inicial nos dados é suavizado, mas a segunda curva apresenta cada vez mais o comportamento da série esperada. A melhora pode ser reparada pela mudança no RMSE, que cai um pouco a cada aumento de *Look Back*, iniciando em 1,5087 e chegando em 0,7429. Esses resultados podem indicar que considerar mais de um mês como característica pode trazer mais informação para o modelo, facilitando suas predições.

No modo Multivariado, o modelo escolhido foi o *Bagging* com escala. A Figura 6.13 apresenta os resultados com valores de *Look Back* de 0, 3, 6, 9 e 12. Diferentemente do modo Univariado, nesses gráficos é possível perceber que aumentar o valor de *Look Back* não implica em uma melhora do RMSE, mas utilizar um *Look Back* de 3 unidades resultou em um RMSE melhor do que não utilizar nenhum *Look Back*. A conclusão do modo Univariado se mantém, mas com a ressalva de que não há um valor fixo ideal de *Look Back*, é necessário experimentar diferentes valores para descobrir qual se adequa melhor ao problema em questão.

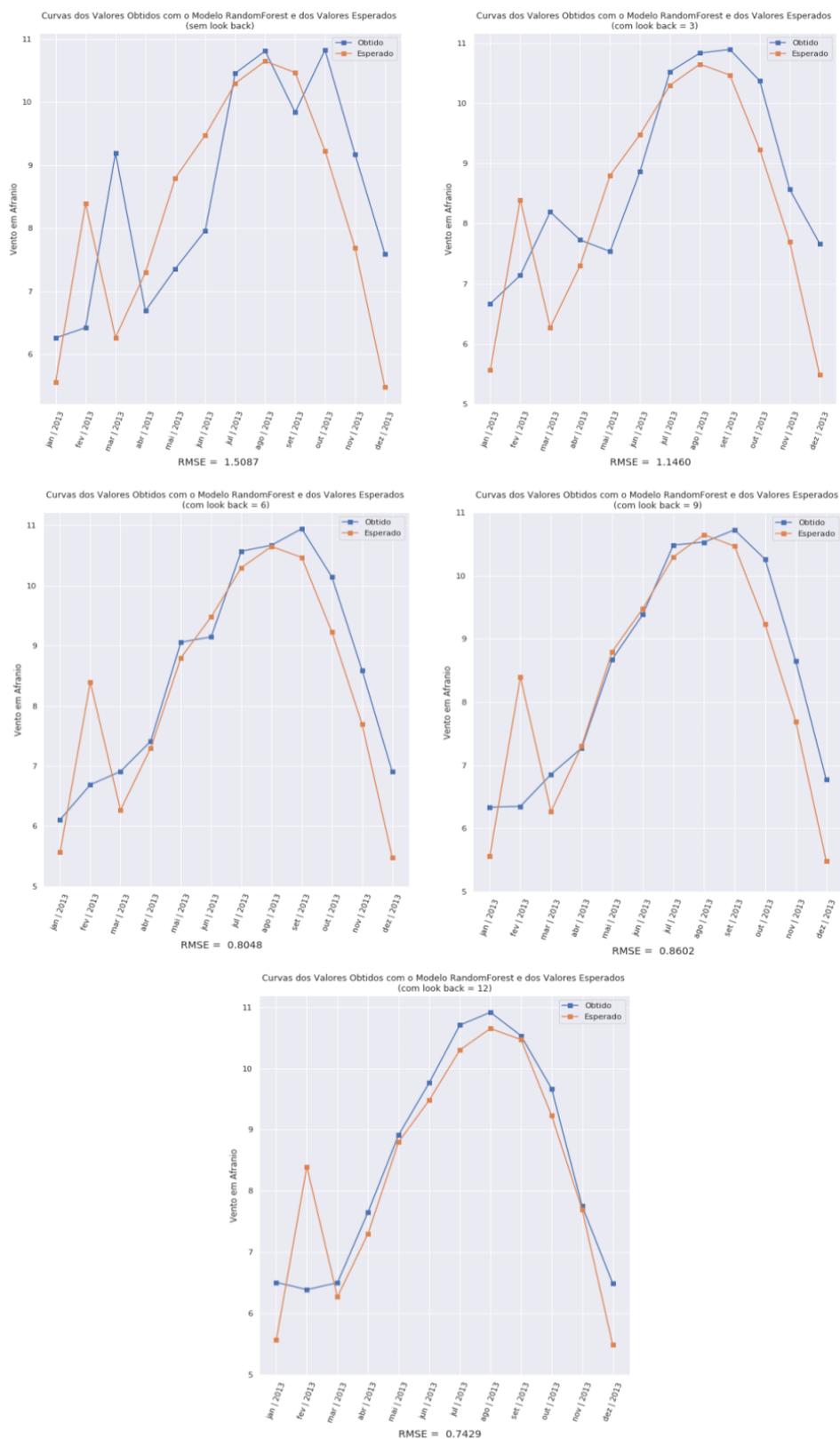


Figura 6.12: Aplicação de diferentes valores de *Look Back* no modelo de Florestas Aleatórias (modo Univariado).

FONTE: elaborado pela autora (2021).

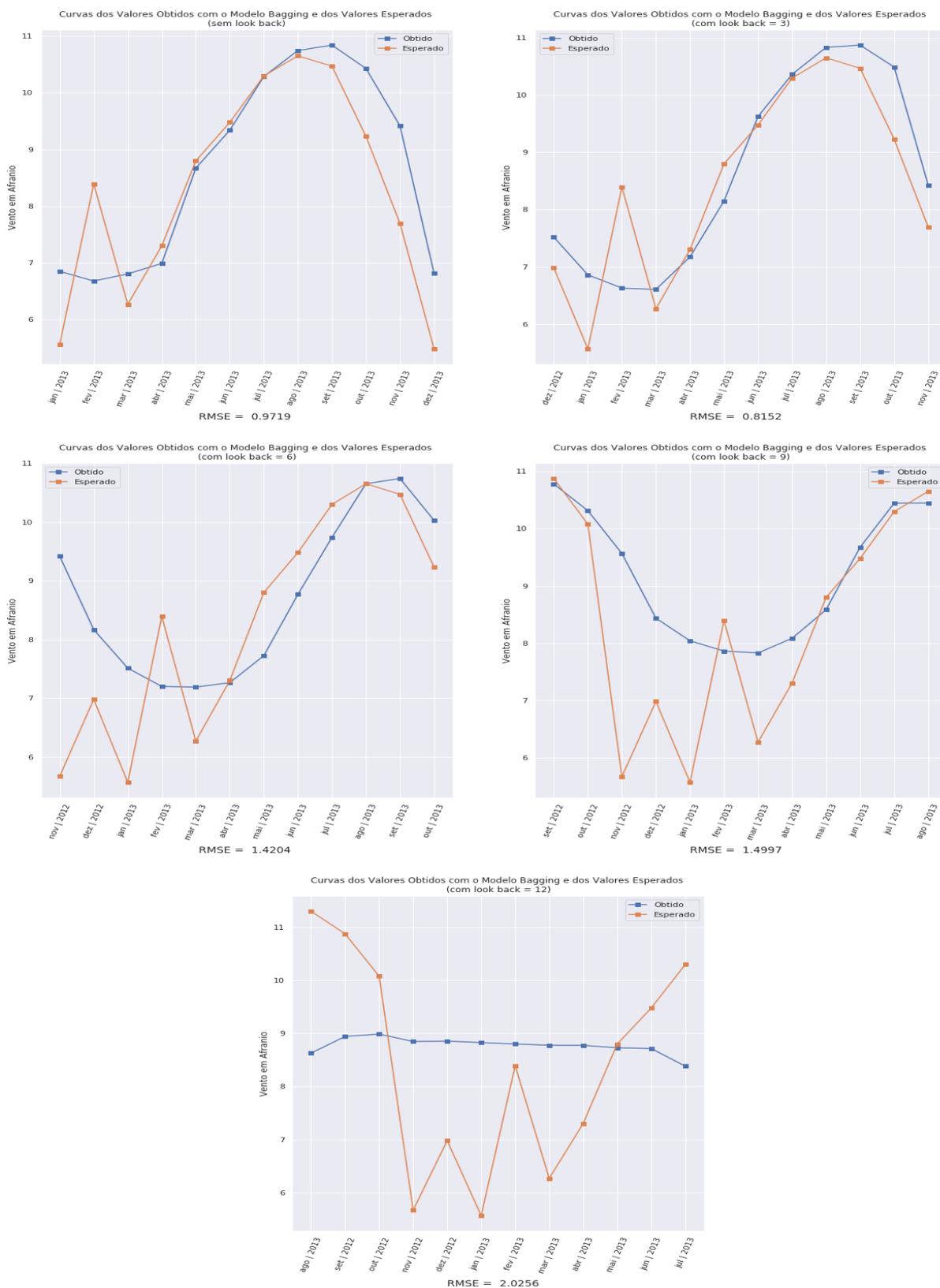


Figura 6.13: Aplicação de diferentes valores de *Look Back* no modelo *Bagging* (modo Multivariado).
 FONTE: elaborado pela autora (2021).

6.2.4 USO DE WALK FORWARD

O quarto pré-processamento analisado foi o *Walk Forward*, definido na seção 2.2.11. Observando os resultados, foi possível perceber que esta etapa teve bastante importância para melhorar as previsões. O impacto desta técnica pode ser explicado pelo fato dos dados de Afrânio conterem apenas dados mensais de 19 anos, o que resultou em somente 216 exemplos de treinamento. Com uma série pequena como os dados de Afrânio, esta técnica não impactou muito no tempo de execução, mas para séries maiores, o *Walk Forward* pode custar muito tempo e não implicar em uma melhora proporcional.

A Figura 6.14 apresenta o modelo AR (Univariado), inicialmente sem pré-processamento (esquerda) e, ao seu lado, está o gráfico resultante do modelo AR com *Walk Forward*. Sem o *Walk Forward* o modelo obteve uma série constante, sem representar as características da curva esperada. Com a aplicação do *Walk Forward*, o processo de retrainar o modelo utilizando os dados já preditos ajudou na predição, conseguindo uma curva próxima à série esperada. Esse resultado pode indicar que, em alguns casos, quanto mais próxima a observação utilizada para o treinamento, melhor o resultado. Essa análise se relaciona com o caso da *Random Walk*, explicada dentro da seção 2.2.11 na fundamentação teórica.

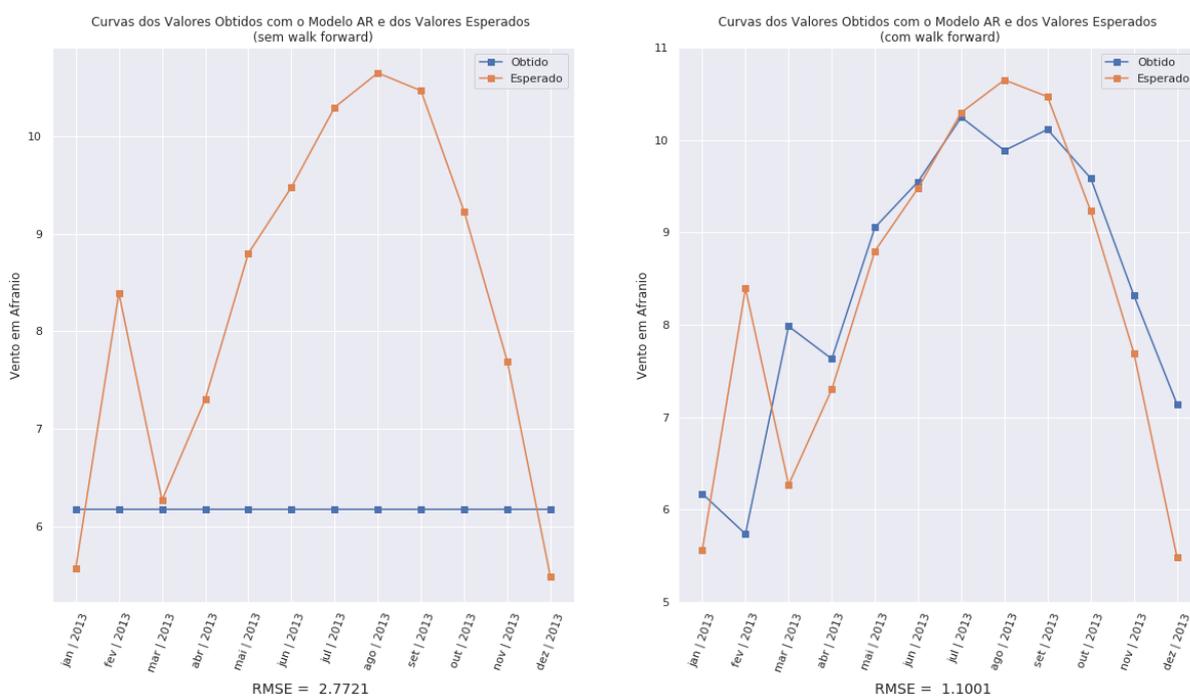


Figura 6.14: Aplicação do *Walk Forward* no modelo AR (modo Univariado).

FONTE: elaborado pela autora (2021).

No modo Multivariado o resultado foi semelhante. Ao aplicar o *Walk Forward* no modelo VECM (Figura 6.15), as previsões passaram de um valor constante para uma série com várias características em comum com a série esperada.

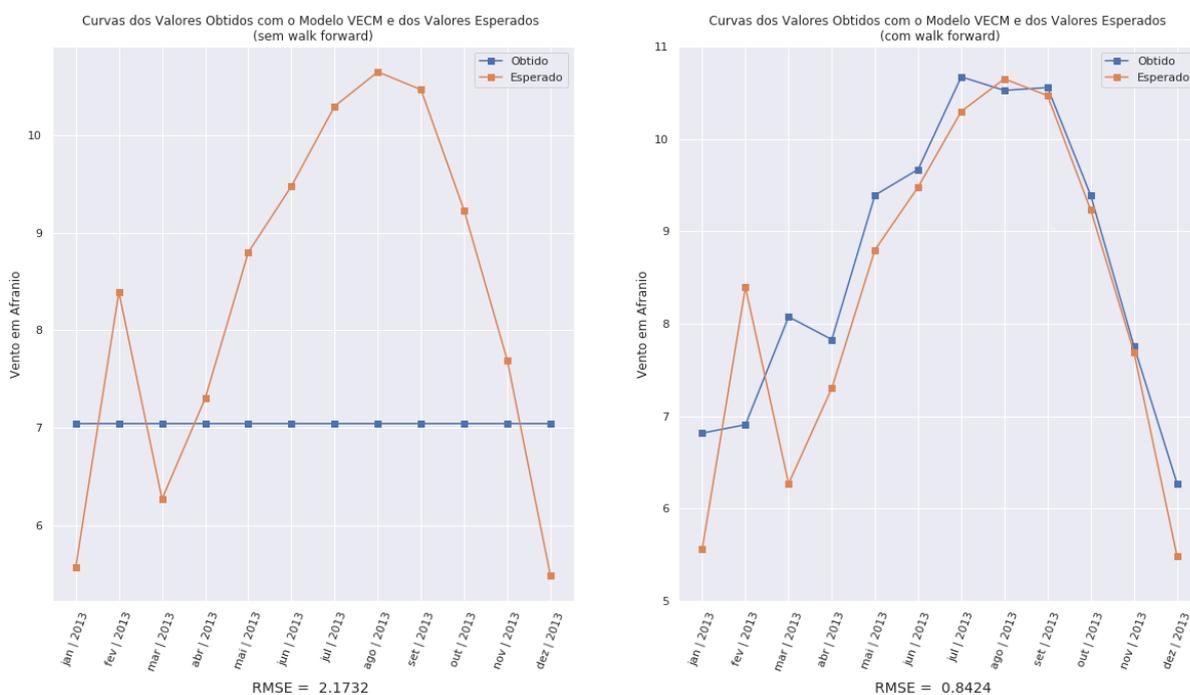


Figura 6.15: Aplicação do *Walk Forward* no modelo VECM (modo Multivariado).

FONTE: elaborado pela autora (2021).

6.2.5 USO DA REDE DE REPRESENTAÇÃO

A quinta etapa de pré-processamento analisada foi o uso da Rede de Representação, proposta e explicada na seção 4.4. A ideia dessa rede surgiu da dificuldade de implementar um *Look Back* Multivariado que seguisse a mesma lógica do *Look Back* Univariado, devido ao aumento excessivo do número de características concatenadas. A Janela Rolante acabou se mostrando uma boa opção para considerar os dados de vários meses, mas tinha desvantagem de perder algumas observações da série temporal.

Para fins de comparação, a Rede de Representação também foi implementada para o modo Univariado, e os resultados obtidos pelo modelo AR, com (direita) e sem a rede (esquerda), com as mesmas configurações restantes, podem ser observados na Figura 6.16. É possível notar que, ao utilizar a rede, o RMSE diminuiu um pouco e as curvas obtidas mantiveram características parecidas com o caso em que a rede não foi utilizada. Esse resultado é um bom indicativo, pois como não houve um esforço para otimizar a Rede de

Representação, o esperado era que, ao utilizá-la, o resultado se aproximasse das execuções com os dados brutos ou com a Janela Rolante.

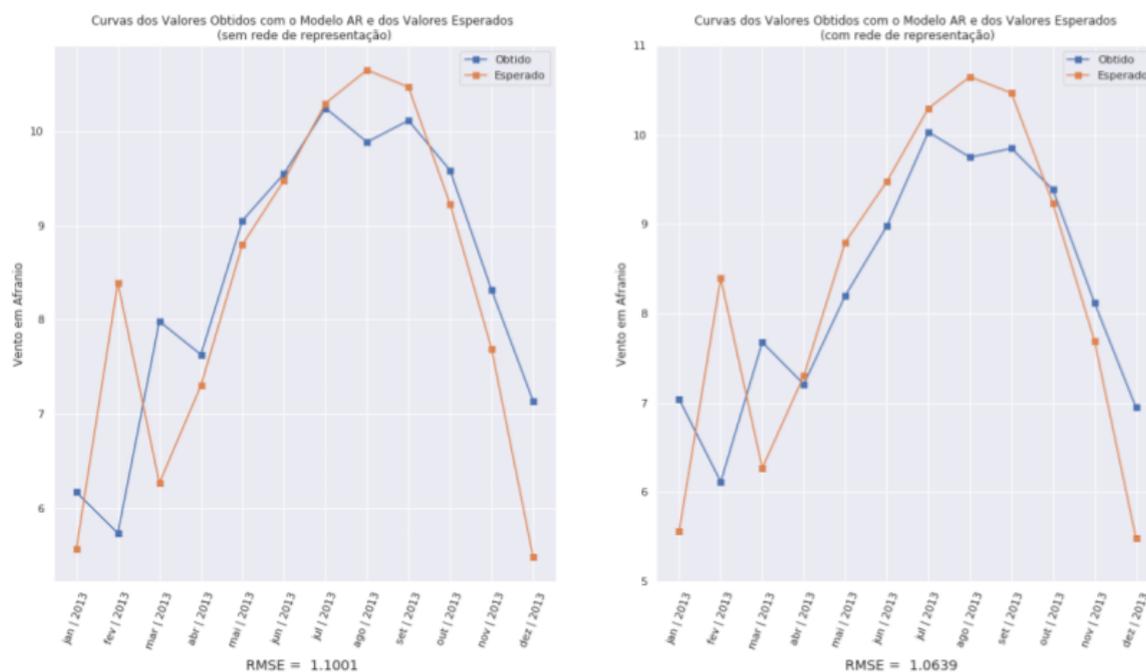


Figura 6.16: Aplicação da Rede de Representação no modelo AR (modo Univariado).

FONTE: elaborado pela autora (2021).

No modo Multivariado, o impacto do uso da rede foi mais intenso, indo de previsões constantes, sem muito significado, a curvas com RMSEs menores do que 0,8, como pode ser observado na Figura 6.17 com o modelo VAR sem a rede (esquerda) e após aplicar a rede (direita), mantendo as mesmas configurações restantes. Esse resultado pode ser justificado pelo fato da rede utilizar informações de vários meses ao invés de somente um e não considerar apenas a média deles, mas valores que armazenam significado ao serem treinados. Assim, para problemas com séries temporais Multivariadas, principalmente ao utilizar modelos derivados da Teoria de Séries Temporais, o uso deste método pode ser interessante, pois consegue captar as características gerais da série sem suavizá-la em excesso.

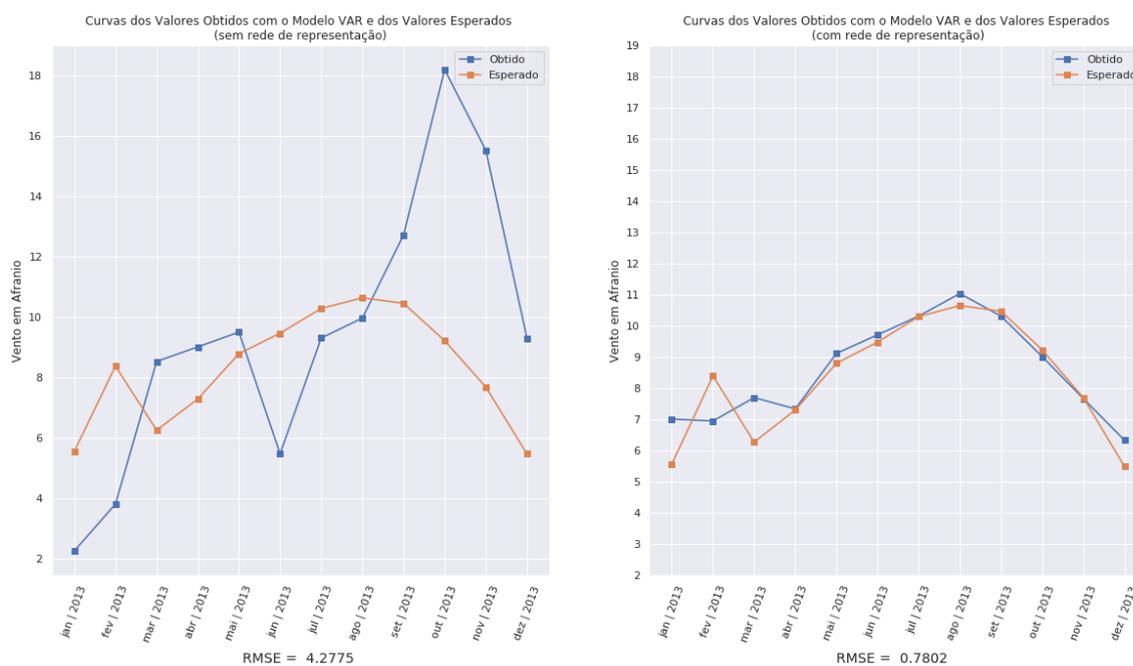


Figura 6.17: Aplicação da Rede de Representação no modelo VAR (modo Multivariado).
 FONTE: elaborado pela autora (2021).

6.2.6 USO DE FUNÇÃO DE ATIVAÇÃO

O penúltimo pré-processamento analisado foi relacionado com a questão da Rede de Representação, referente à parte de utilizar, ou não, a função de ativação nas características obtidas pela Rede, técnica discutida na seção 4.4. O desempenho esperado deste método era uma incógnita, pois a função de ativação aplicaria uma normalização nas características obtidas pela Rede de Representação, podendo aproximar valores que deveriam ser similares mas foram afastados pela dispersão ou acabaria aproximando características que deveriam ser significativamente diferentes.

Ao analisar os resultados, foi possível perceber uma piora nos resultados ao utilizar a função de ativação. A Figura 6.18 apresenta um exemplo do modo Univariado, com o modelo ARIMA, e a Figura 6.19 mostra o modelo VAR, do modo Multivariado. Em ambos casos houve um aumento do RMSE, que foi mais acentuado no modo Univariado, o que pode indicar a segunda suposição de que os valores normalizados pela *Softmax* acabaram se aproximando, o que dificultou a predição.

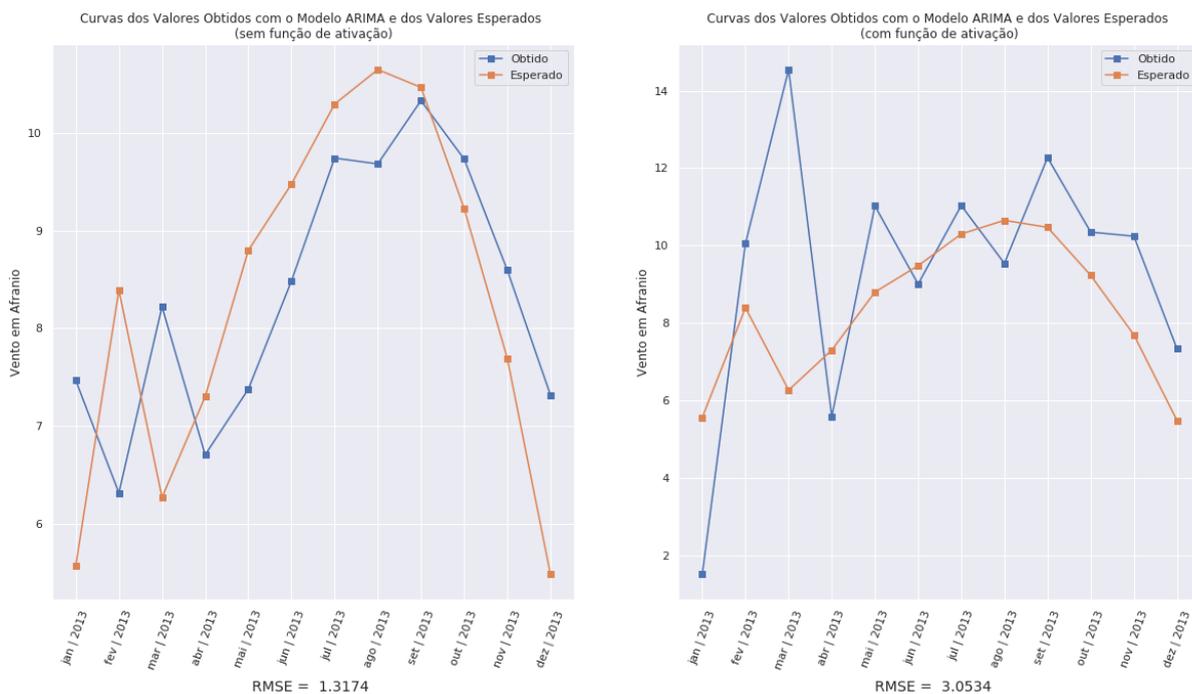


Figura 6.18: Aplicação da função de ativação na Rede de Representação com o modelo ARIMA (modo Univariado).

FONTE: elaborado pela autora (2021).

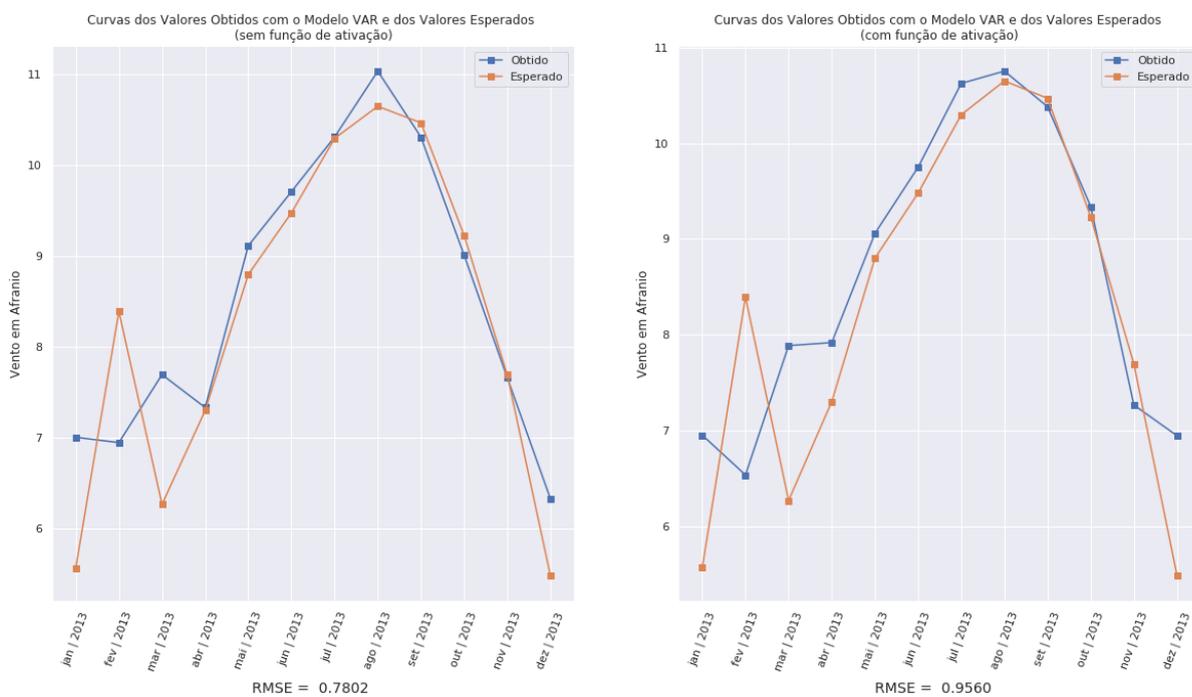


Figura 6.19: Aplicação da função de ativação na Rede de Representação com o modelo VAR (modo Multivariado).

FONTE: elaborado pela autora (2021).

6.2.7 USO DE LOOK BACK NA REDE DE REPRESENTAÇÃO

Por fim, o último pré-processamento analisado foi a aplicação de *Look Back* na Rede de Representação, explicado dentro da seção 4.4. Essa informação indicava para a Rede de Representação quantos meses deveriam ser considerados para gerar as características. Como exemplo, a Figura 6.20 apresenta o modelo MA, do modo Univariado, com diferentes valores de *Look Back* (0, 3, 6, 9 e 12). Neste exemplo, o aumento do *Look Back* contribuiu para uma melhor predição, ajudando a aproximar os valores deslocados para a direita da segunda curva dos dados.

No modo Multivariado, o uso do *Look Back* na Rede de Representação também apresentou melhoras, mas, diferente do modo Univariado, o *Look Back* de 3 obteve resultados melhores do que os outros valores maiores, como é possível observar na Figura 6.21, em que foram aplicados diferentes valores de *Look Back* (0, 3, 6, 9 e 12) nos dados com o modelo VAR. Com essa informação é possível dizer que tanto esse *Look Back* quanto o *Look Back* da Janela Rolante conseguem capturar as informações dos meses anteriores e, portanto, apresentam resultados parecidos. Dessa forma, é possível concluir que, nos resultados obtidos neste problema, em grande parte dos casos, ao utilizar valores de *Look Back* na Rede de Representação maiores do que 1 é possível obter melhores resultados, mas ainda com a ressalva de que vários valores de *Look Back* precisam ser testados para definir qual deles se adapta melhor ao problema em questão.

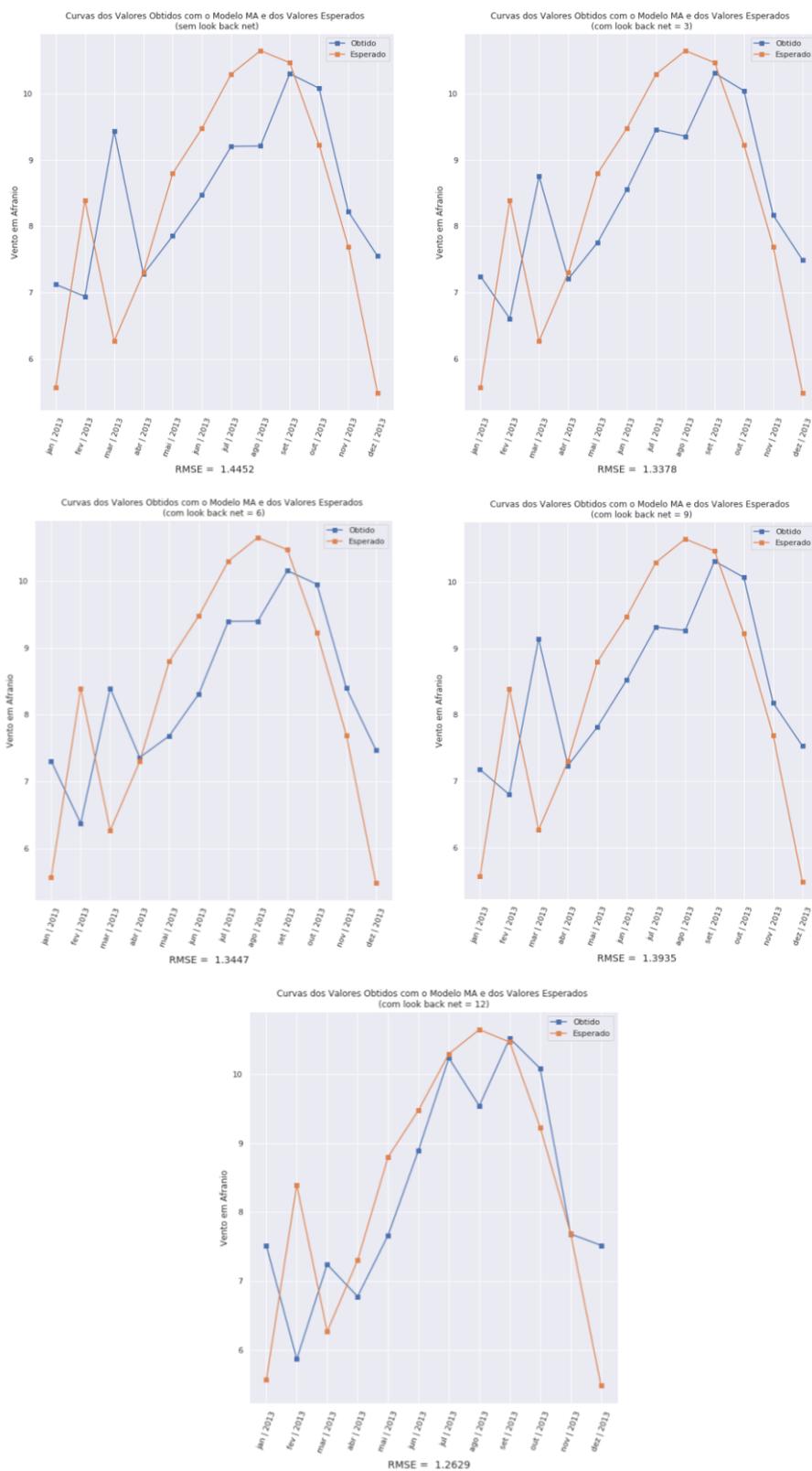


Figura 6.20: Aplicação de diferentes valores de *Look Back* na Rede de Representação com o modelo MA (modo Univariado).

FONTE: elaborado pela autora (2021).

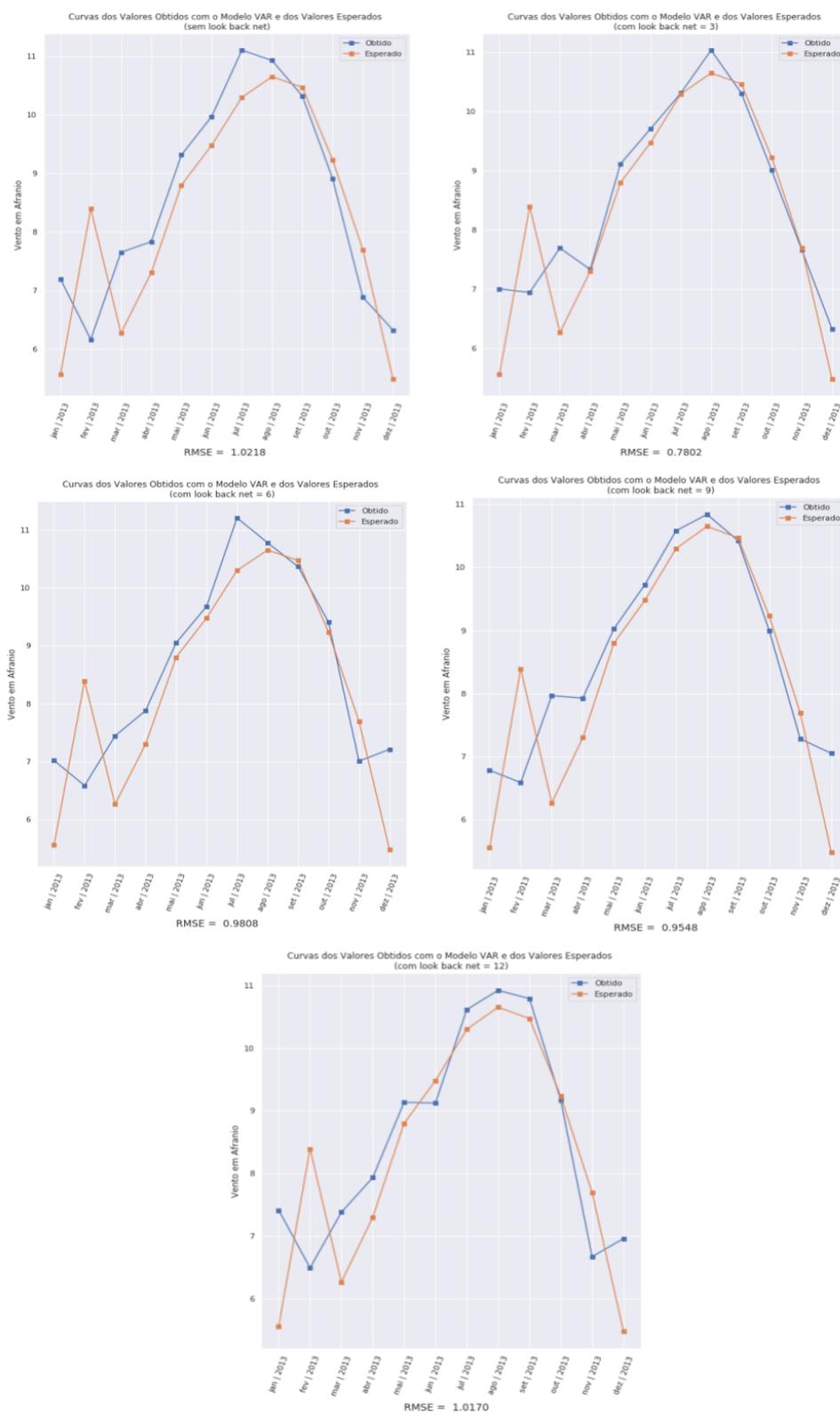


Figura 6.21: Aplicação de diferentes valores de *Look Back* na Rede de Representação com o modelo VAR (modo Multivariado).

FONTE: elaborado pela autora (2021).

6.2.8 ANÁLISE DOS PRÉ-PROCESSAMENTOS UTILIZADOS NOS RESULTADOS

Ao analisar os parâmetros de pré-processamento utilizados nos melhores modelos do modo Univariado (Tabela 6.1) é possível perceber que quase todos os modelos obtiveram melhores resultados utilizando a escala *MinMax* e a técnica de *Walk Forward*. Dessa forma, as conclusões demonstradas nas seções referentes a esses pré-processamentos se mantiveram também nos melhores resultados.

Tanto os valores de *Look Back* quanto os valores de *Look Back* da Rede de Representação variaram entre 6 e 12, o que pode indicar uma sazonalidade de 6 ou 12 meses na série. Já a Rede de Representação foi pouco utilizada, aparecendo somente nos modelos derivados de séries temporais. Isso pode ser justificado pelo fato da Rede de Representação extrair informações de vários meses para dentro de 1 exemplo. Como os modelos derivados da Teoria de Séries Temporais geralmente utilizam as informações dos últimos *lags* da série, cada *lag* pode ter apresentado características mais preditivas com a ajuda das informações dos últimos meses, o que ajudou esses modelos a obterem melhores resultados.

Por fim, a função de ativação só foi utilizada em um modelo, que foi o *Double Exponential Smoothing*, que obteve resultados ruins, com RMSE maior do que 8. Isso confirma a afirmação apresentada na análise da função de ativação na Rede de Representação que enuncia que o uso deste pré-processamento piora o resultado em todos os casos observados.

Nos resultados do modo Multivariado, a escala continua sendo muito utilizada, mas o *Walk Forward* se mantém equilibrado, sendo utilizado em apenas metade dos modelos. Isso pode indicar que as séries de vazão, utilizadas como características, apresentam informação suficiente para os modelos conseguirem bons resultados, mesmo com poucos dados.

No modo Multivariado, tanto os valores de *Look Back* quanto os valores de *Look Back* da Rede de Representação variaram entre 3 e 6, o que pode indicar uma sazonalidade na série ou que os 3 ou 6 meses anteriores contêm informações relevantes para prever o próximo mês. E, por fim, a Rede de Representação foi pouco utilizada, aparecendo somente nos dois modelos derivados da Teoria de Séries Temporais. Este fato reforça a conclusão sobre este passo de pré-processamento no modo Univariado.

6.3 ANÁLISE DOS MELHORES MODELOS

A última etapa da análise foi o estudo dos dois melhores modelos para horizontes de predição de 12 e 60 meses, com o aumento do número de épocas para 200 iterações. Além deste estudo, como uma análise adicional, os melhores modelos foram também executados nas outras seis bases de dados além de Afrânio (1), que são: Estreito (2), *Offshore* CE (3), *Offshore* SC (4), Palmas (5), Quilombo (6) e São João (7). Vale ressaltar que o objetivo deste trabalho se concentrou nos dados de Afrânio, sendo a execução dos melhores modelos nos demais dados uma avaliação preliminar que pode ser estendida posteriormente.

6.3.1 RESULTADOS DOS MODELOS UNIVARIADOS

Os resultados de MSE, RMSE e MAPE, além dos melhores parâmetros de pré-processamento obtidos pelos melhores modelos são apresentados pela Tabela 6.3, mas todos os resultados e gráficos podem ser conferidos no Apêndice C. Vale lembrar que os parâmetros e hiperparâmetros utilizados para escolher e treinar os dois melhores modelos desta etapa foram selecionados dos 5 melhores resultados obtidos na primeira etapa de análise.

Tabela 6.3: Resultados dos melhores modelos, com o aumento de épocas, para todas as bases de dados no modo Univariado, para o horizonte de predição de 12 meses.

FONTE: elaborado pela autora (2021).

Bases de Dados	Modelo	Estac.	Escala	Look Back	Walk Forw.	Rede de Repr.	Função de Ativ.	Look Back Net	MSE	RMSE	MAPE
1	GRU	0	0	6	1	0	0	0	0,4541	0,6739	0,0718
2	GRU	0	0	6	0	0	0	0	0,4535	0,6734	0,0890
3	GB	0	1	12	1	0	0	0	0,8981	0,9477	0,0753
4	GB	0	0	12	0	0	0	0	0,9820	0,9910	0,0980
5	GB	0	0	0	1	0	0	0	0,3247	0,5698	0,0602
6	GRU	0	0	6	1	0	0	0	0,2559	0,5059	0,0956
7	GB	0	1	12	1	0	0	0	0,4177	0,6463	0,0601

Observando a tabela acima, é possível concluir que os modelos conseguiram alcançar bons resultados em todas as bases de dados, mesmo com seus hiperparâmetros e parâmetros de pré-processamento sendo otimizados somente nos dados de Afrânio. Os piores casos de RMSE e MAPE ocorreram nos dados de Estreito e *Offshore* SC, e os melhores em Palmas e

São João. Para obter os resultados acima foram executados os dois modelos, *Gradient Boosting* e GRU, e o modelo que apresentou melhor resultado, em cada base de dados, foi selecionado para ser inserido na tabela. Pelo fato dos modelos apresentarem resultados semelhantes, em média, como é apresentado pela Tabela 6.4, os resultados do segundo melhor modelo, em cada base, foi omitido.

Tabela 6.4: Valores médios de MSE, RMSE e MAPE dos modelos Univariados, considerando os resultados em todos os sete conjuntos de dados, com horizonte de 12 meses.

FONTE: elaborado pela autora (2021).

Modelo	MSE Médio	RMSE Médio	MAPE Médio
GB	0,5546	0,7257	0,0760
GRU	0,5703	0,7328	0,0796

Analisando os parâmetros de pré-processamento utilizados pelos modelos (Tabela 6.3), os mesmos valores, presentes nas análises anteriores, se mantêm. Houve uso de escala algumas vezes, o *Walk Forward* foi utilizado em quase todos os casos, os valores de *Look Back* variaram entre 6 e 12 meses e, como não havia nenhum modelo derivado da Teoria de Séries Temporais nesses experimentos, a Rede de Representação não foi utilizada.

Ao observar os gráficos de valores esperados e obtidos, apresentados pelas Figuras 6.22 e 6.23, é possível dizer que, em Afrânio, visualmente, os modelos parecem ter alcançado melhores resultados do que em Palmas, que obteve o menor RMSE. Além disso, mesmo com RMSEs pequenos, em algumas bases de dados, o modelo não conseguiu generalizar completamente o comportamento da série. Essas observações podem indicar que, devido à dispersão das séries temporais, apresentadas no Apêndice D, e também por influência da quantidade de picos presentes nas partes das bases utilizadas para teste, os modelos podem ter encontrado dificuldades para realizar previsões muito boas, mas como as previsões se mantiveram na média da dispersão da série, os valores de RMSE se mantiveram baixos.

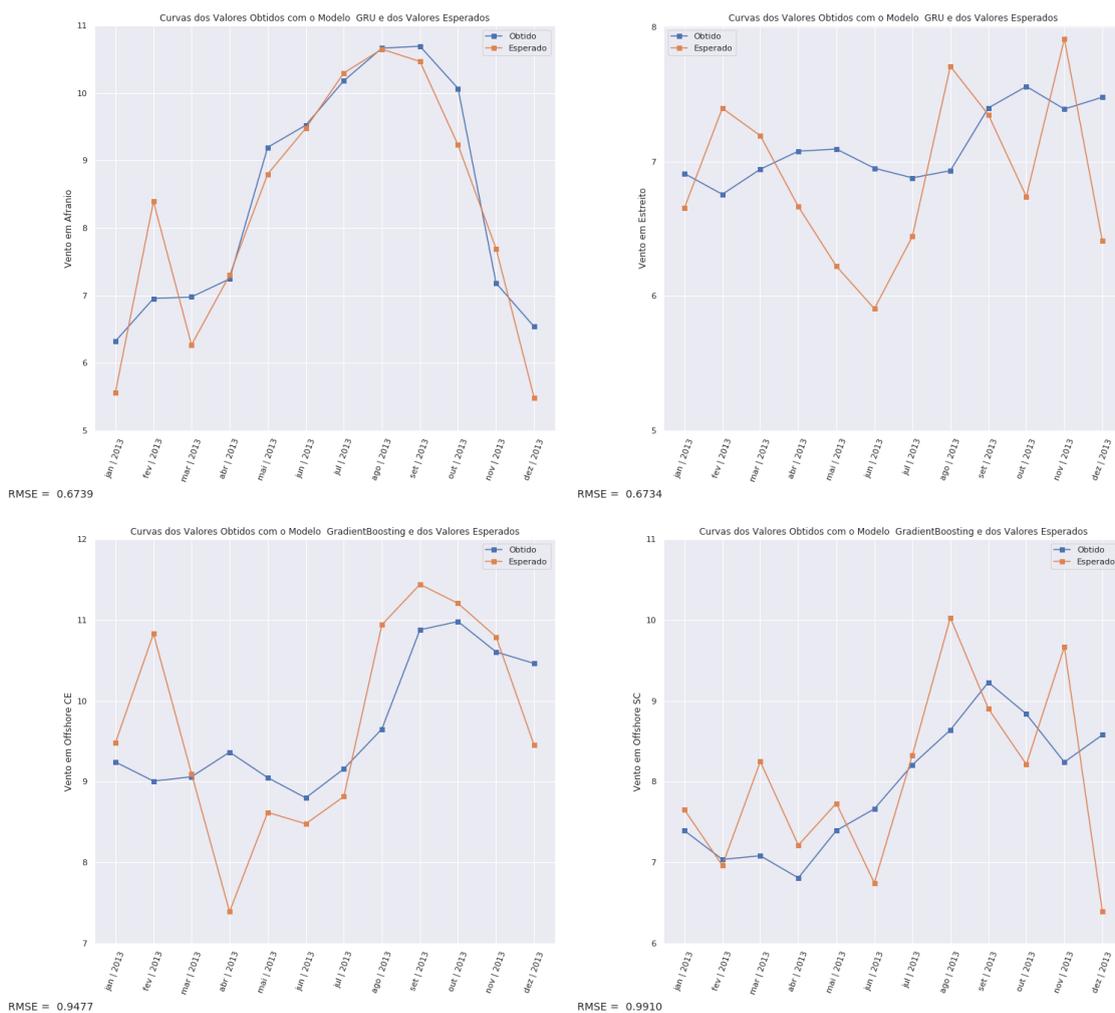


Figura 6.22: Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos melhores modelos, entre *Gradient Boosting* e GRU, no modo Univariado, com horizonte de previsão de 12 meses, para os dados de Afrânio, Estreito, *Offshore CE* e *SC*.
 FONTE: elaborado pela autora (2021).

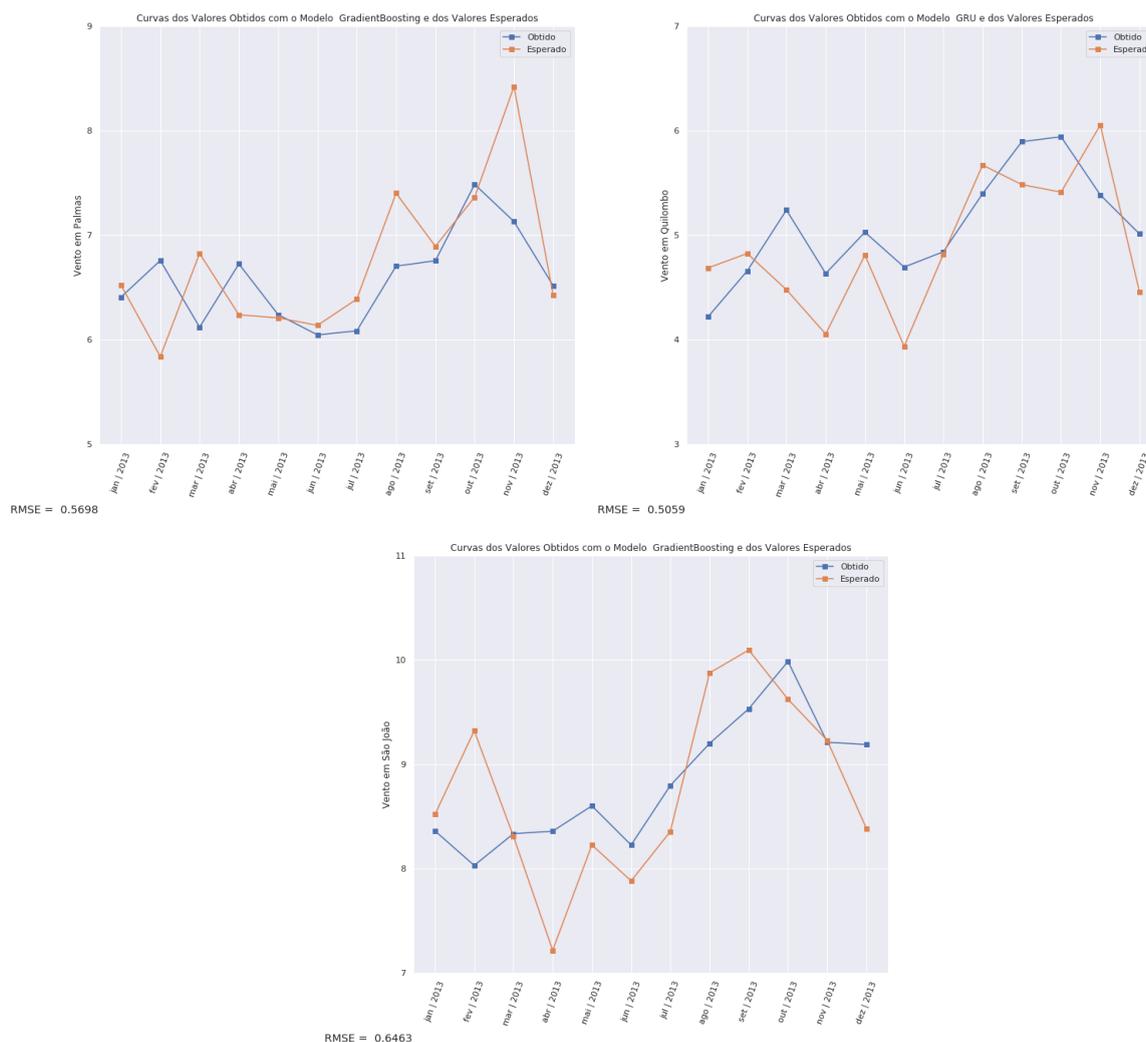


Figura 6.23: Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos melhores modelos, entre *Gradient Boosting* e GRU, no modo Univariado, com horizonte de previsão de 12 meses, para os dados de Palmas, Quilombo e São João.
 FONTE: elaborado pela autora (2021).

As Figuras 6.24 e 6.25 apresentam os gráficos de valores esperados e obtidos pelos melhores modelos ao alterar o tamanho da base de teste de 12 meses para 60 meses. O objetivo desses gráficos não foi apresentar melhores resultados, pois a base de treinamento passou por uma redução para que a base de teste fosse aumentada, mas mostrar, de uma forma mais abrangente, o comportamento das previsões em um horizonte de predição maior. Os gráficos são referentes aos menores RMSEs obtidos pelos modelos *Gradient Boosting* e GRU.

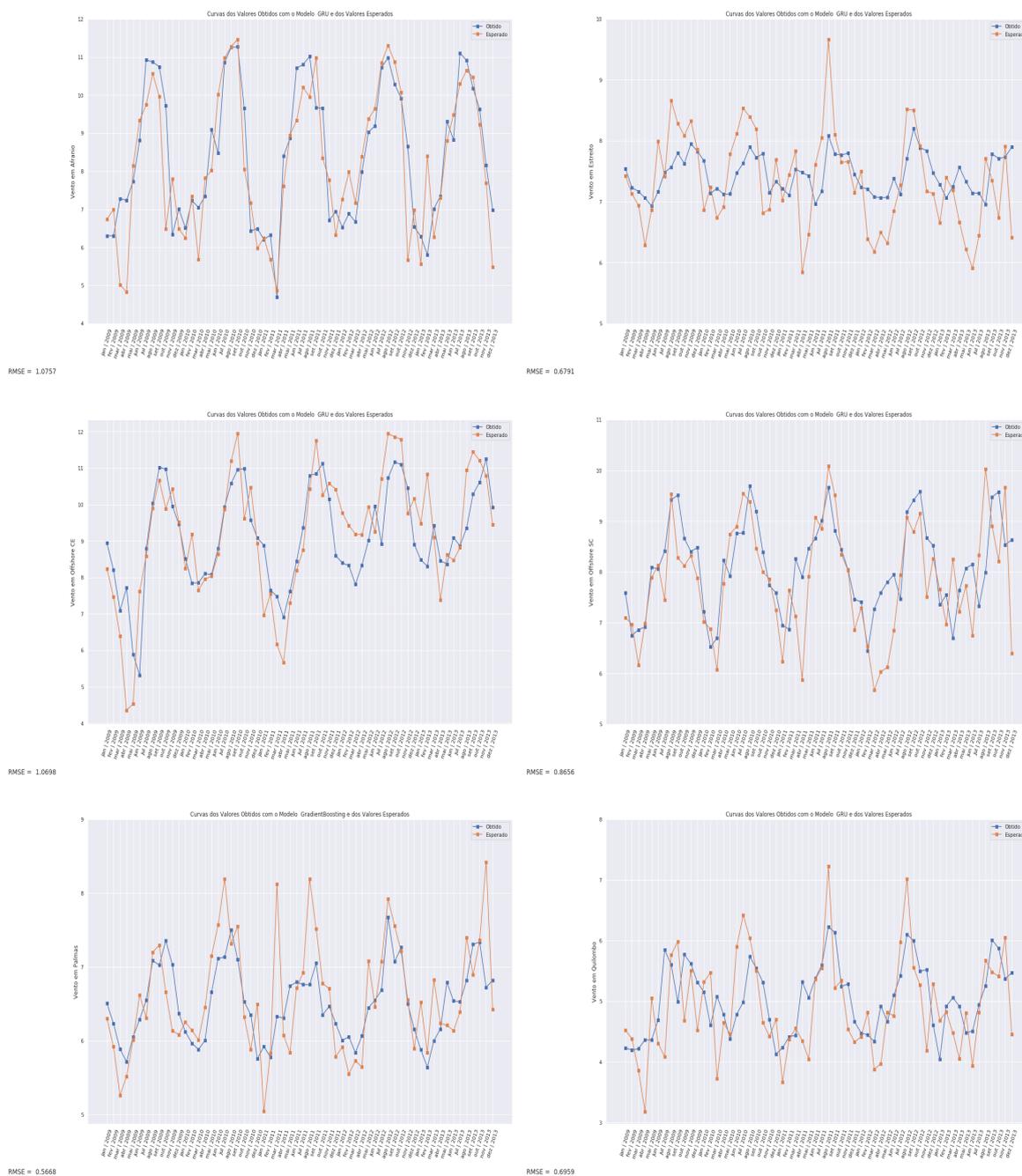


Figura 6.24: Gráficos das curvas obtidas e esperadas, no modo Univariado, dos melhores modelos, entre *Gradient Boosting* e GRU, em todas as bases de dados, com horizonte de previsão de 60 meses, dos dados de Afriano, Estreito, *Offshore CE*, *Offshore SC*, Palmas e Quilombo.

FONTE: elaborado pela autora (2021).

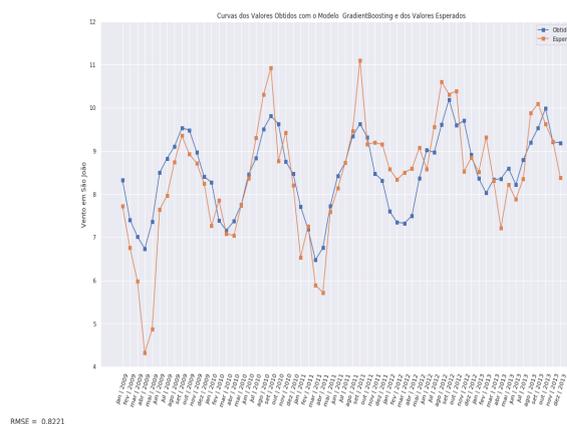


Figura 6.25: Gráficos das curvas obtidas e esperadas, no modo Univariado, dos melhores modelos, entre *Gradient Boosting* e GRU, em todas as bases de dados, com horizonte de previsão de 60 meses, dos dados de São João.

FONTE: elaborado pela autora (2021).

6.3.2 RESULTADOS DOS MODELOS MULTIVARIADOS

A mesma análise foi realizada para os modelos LSTM e GRU no modo Multivariado. As métricas podem ser observadas na Tabela 6.5 e os gráficos estão presentes nas Figuras 6.26 e 6.27. Dentre os resultados, os piores RMSEs e MAPEs ocorreram nos dados de *Offshore* CE, *Offshore* SC e Quilombo, e os resultados restantes apresentaram desempenhos semelhantes. Já nos gráficos, é possível dizer que em todas as bases de dados o comportamento geral da série foi capturado pelos modelos, mas, exceto em Afrânio, esse comportamento se aproxima da média, também devido às dispersões das séries e do fato dos modelos terem sido otimizados utilizando somente os dados de Afrânio. Por fim, a Tabela 6.6 mostra os valores médios obtidos pelos dois modelos, possibilitando concluir que o modelo da LSTM se saiu melhor em questão de RMSE e MAPE no cenário de 12 meses.

Tabela 6.5: Resultados dos melhores modelos, com o aumento de épocas, para todas as bases de dados no modo Multivariado, para o horizonte de predição de 12 meses.

FONTE: elaborado pela autora (2021).

Planilha	Modelo	Estac.	Escala	Look Back	Walk Forw.	Rede de Repr.	Função de Ativ.	Look Back Net	MSE	RMSE	MAPE
1	LSTM	0	1	3	1	0	0	0	0,3766	0,6137	0,0699
2	GRU	0	1	9	0	0	0	0	0,2699	0,5196	0,0671
3	GRU	0	1	6	0	0	0	0	0,7207	0,8489	0,0651
4	LSTM	0	1	6	0	0	0	0	0,4968	0,7048	0,0690
5	LSTM	0	1	6	0	0	0	0	0,2090	0,4571	0,0579
6	LSTM	0	1	6	0	0	0	0	0,2519	0,5019	0,0861
7	GRU	0	1	6	0	0	0	0	0,4246	0,6516	0,0676

Tabela 6.6: Valores médios de MSE, RMSE e MAPE dos modelos Multivariados, considerando os resultados em todos os sete conjuntos de dados, com horizonte de 12 meses.

FONTE: elaborado pela autora (2021).

Modelo	MSE Médio	RMSE Médio	MAPE Médio
LSTM	0,4406	0,6499	0,0714
GRU	0,4816	0,6761	0,0765

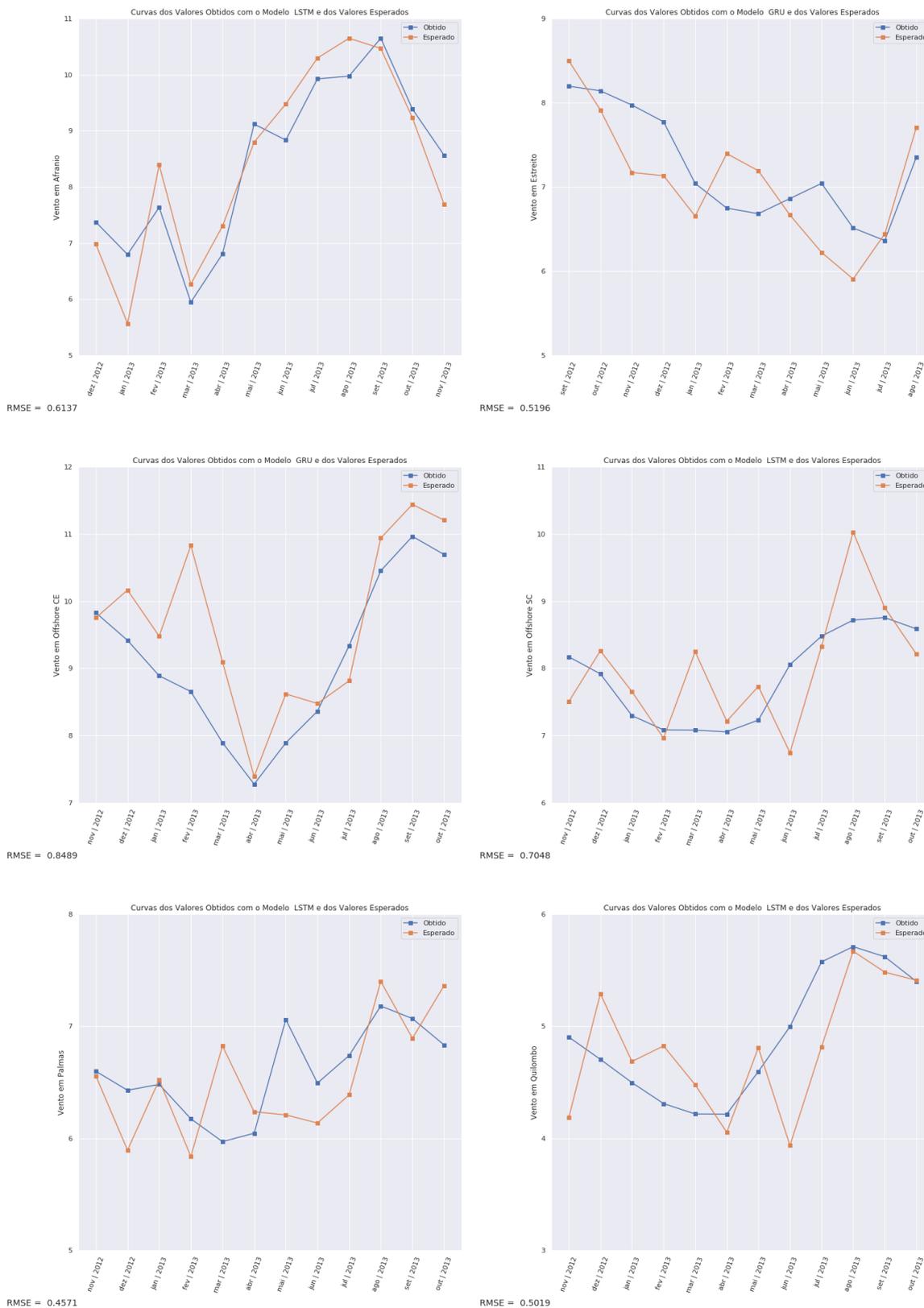


Figura 6.26: Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos melhores modelos, entre LSTM e GRU, no modo Multivariado, com horizonte de previsão de 12 meses, nos dados de Afrânio, Estreito, Offshore CE, Offshore SC, Palmas e Quilombo.

FONTE: elaborado pela autora (2021).

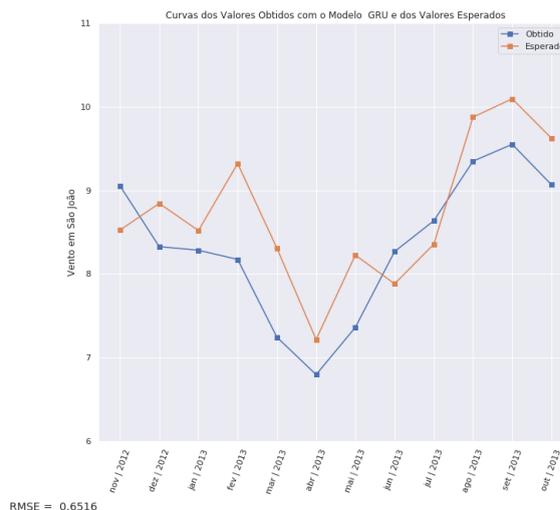


Figura 6.27: Gráficos dos valores obtidos (em azul) e esperados (em alaranjado) dos melhores modelos, entre LSTM e GRU, no modo Multivariado, com horizonte de previsão de 12 meses, dos dados de São João. FONTE: elaborado pela autora (2021).

Analisando os parâmetros de pré-processamento, das cinco configurações testadas para cada modelo em cada base de dados, é possível notar o uso de escala e *Look Back* em todos os casos. A escala, como já mencionado, pode estar relacionada com a convergência dos modelos que utilizam gradientes, e o *Look Back* indica que existe uma sazonalidade nas série que ajuda a definir a predição, geralmente assumindo o valor de 6 meses, mas existindo casos em que uma sazonalidade de 3 e 9 meses foi considerada. O *Walk Forward* não se mostrou muito necessário, aparecendo em um dos sete experimentos, o que pode indicar que esses modelos não tiveram uma melhoria ao retreinar com os dados já preditos, o que impacta positivamente no desempenho de tempo para gerar as predições. Por fim, a Rede de Representação não foi utilizada em nenhum dos sete experimentos, o que reforça as análises anteriores sobre esse método não ajudar nos modelos de Aprendizado de Máquina e Aprendizado Profundo, somente nos modelos derivados da Teoria de Séries Temporais.

Assim como no modo Univariado, para permitir uma melhor visualização do comportamento geral das predições obtidas, foram gerados gráficos para os modelos com melhores RMSEs em um horizonte de predição de 60 meses. Esses gráficos são apresentados nas Figuras 6.28 e 6.29 e mostram que, em todas as bases de dados, os modelos conseguiram captar o comportamento geral da série temporal esperada.

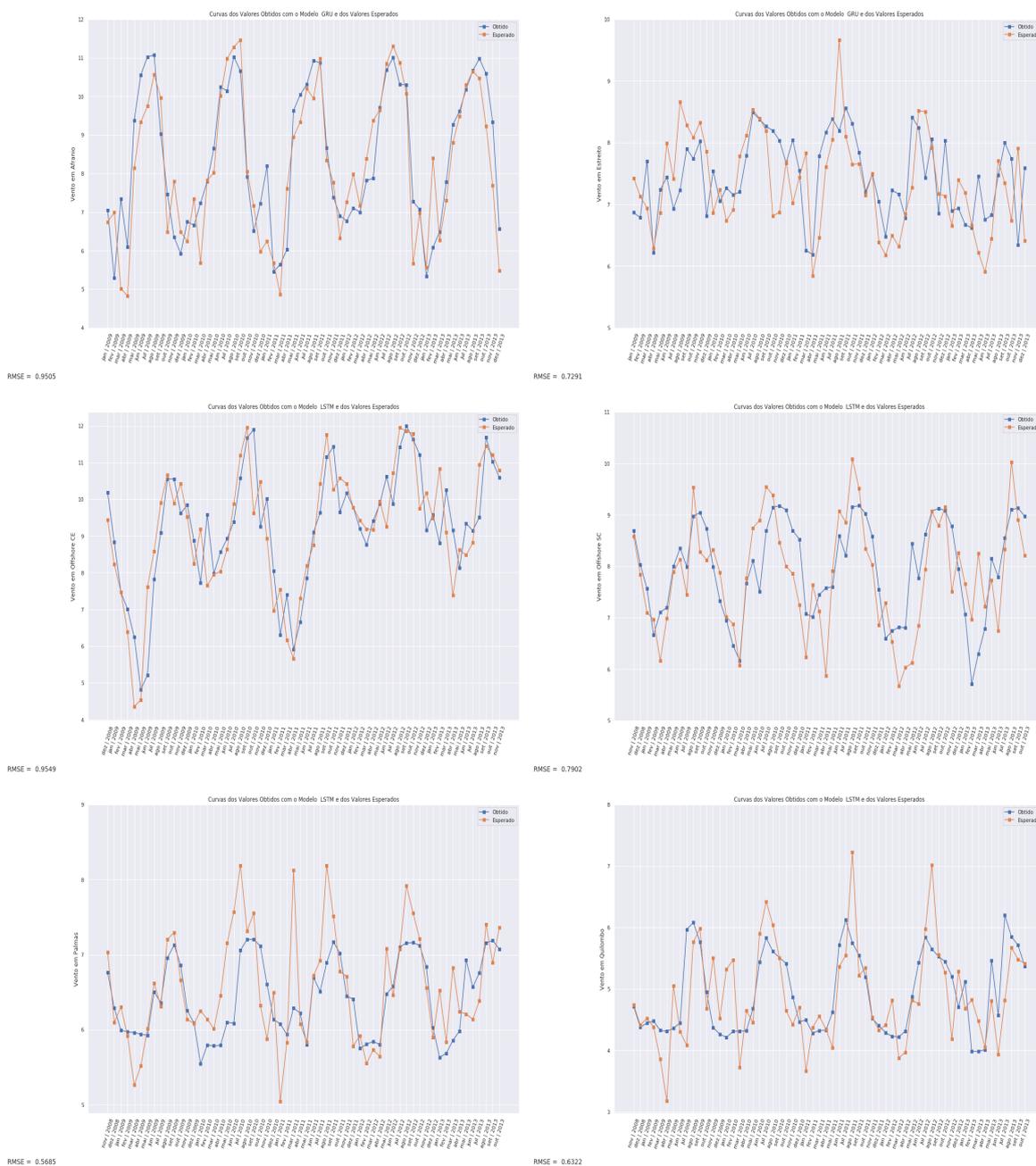


Figura 6.28: Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos melhores modelos, entre LSTM e GRU, em todas as bases de dados, com horizonte de previsão de 60 meses, dos dados de Afrânio, Estreito, *Offshore* CE, *Offshore* SC, Palmas e Quilombo.

FONTE: elaborado pela autora (2021).

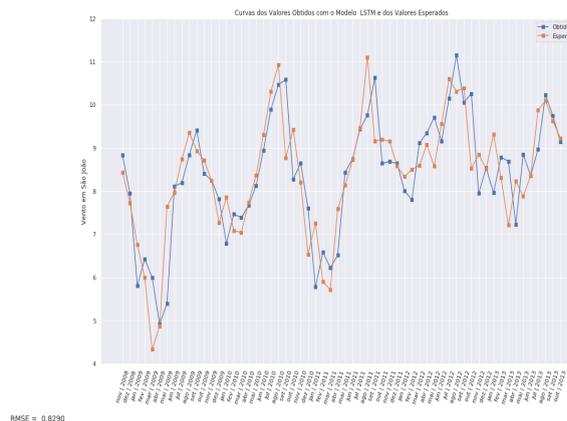


Figura 6.29: Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos melhores modelos, entre LSTM e GRU, em todas as bases de dados, com horizonte de previsão de 60 meses, dos dados de São João.

FONTE: elaborado pela autora (2021).

6.4 CONSIDERAÇÕES FINAIS

Nesta seção foi feita uma análise de 3 etapas, sendo a primeira uma avaliação de todos os modelos implementados, a segunda um estudo do impacto de cada método pré-processamento isoladamente e, por fim, na última etapa, os melhores resultados encontrados, com o treinamento e avaliação do modelos em 6 bases de dados adicionais, além da visualização das previsões em horizontes de 12 e 60 meses. Na primeira etapa, com exceção de três modelos, todos os resultados superaram as métricas do modelo *baseline* e a maioria dos modelos apresentou MAPEs abaixo de 10%, indicando que as previsões foram altamente precisas. Ainda nesta etapa, foi possível perceber o impacto da dispersão das séries temporais nos resultados obtidos pelos modelos, contribuindo quando eram baixas e atrapalhando quando eram altas.

Na segunda etapa, a estacionariedade impactou em alguns modelos e demonstrou precisar ser compensada por outros pré-processamentos, como o *Walk Forward*. A escala se mostrou indispensável e, tanto o *Look Back* da Rede de Representação, quanto o *Look Back* das características mostraram indícios de que valores maiores do que 1 geram melhores resultados, mas o ideal se resume a testar vários valores para definir qual deles se adequa melhor ao problema em questão. A Rede de Representação proposta contribuiu para resultados melhores nos modelos derivados da Teoria de Séries Temporais, ressaltando a validade desta nova técnica.

Por fim, na última etapa, os melhores resultados encontrados para os hiperparâmetros dos modelos e parâmetros de pré-processamento mostraram uma robustez, pois apresentaram bons resultados em outras 6 bases de dados, sendo otimizados somente nos dados de Afrânio.

No modo Univariado, o melhor modelo para prever a série de Afrânio foi a GRU, com RMSE de 0,6739 e MAPE de 0,0718. Nas métricas médias de todas as bases de dados, o *Gradient Boosting* se sobressaiu, com RMSE médio de 0,7257 e MAPE médio de 0,0760. No modo Multivariado, a LSTM apresentou os melhores resultados tanto em Afrânio quanto na média das métricas, com RMSE de 0,6137 e MAPE de 0,0699 em Afrânio e RMSE médio de 0,6499 e MAPE médio de 0,0765 no conjunto com todas as bases de dados.

7 CONCLUSÃO

A elaboração deste trabalho proporcionou várias metodologias diferentes para realizar a predição de velocidade de vento, incluindo unicamente o uso de dados eólicos, o que foi chamado de modo Univariado, e também a utilização de 22 séries temporais hidrológicas distintas, que recebeu o nome de modo Multivariado. Nos dois modos foram explorados e otimizados vários pré-processamentos em séries temporais, como Estacionariedade, escala, *Look Back* e *Walk Forward*, além da proposta da Rede de Representação, que foi uma técnica criada para extrair informações dos meses passados da série e construir uma nova característica que representasse esses meses.

Diferentes modelos foram implementados e comparados, sendo o AR, MA, ARMA, ARIMA, SARIMA, ARCH, GARCH, *Exponential Smoothing*, VAR e VECM pertencentes à categoria dos modelos derivados da Teoria de Séries Temporais, a Regressão Linear, Regressão Polinomial, Árvore de Regressão, *Bagging*, Florestas Aleatórias e *Gradient Boosting* os modelos da área de Aprendizado de Máquina e o *Perceptron*, MLP, LSTM e GRU pertencentes aos modelos do campo de Aprendizado Profundo. Esses modelos foram melhorados com diferentes otimizadores, como a *Grid Search*, o Skopt e o Talos. Em todas as otimizações o objetivo foi treinar os modelos com diferentes valores de hiperparâmetros e avaliar os resultados em uma base de validação utilizando o critério de redução da métrica RMSE.

Com experimentos que buscavam não só encontrar o melhor modelo para realizar as predições, mas também analisar a influência de cada etapa de pré-processamento e modelo utilizado, foi realizada uma análise de três etapas para investigar os resultados obtidos. A primeira etapa apresentou os valores de MSE, RMSE e MAPE alcançados por todos os modelos executados com os melhores valores encontrados nas otimizações. Nesta etapa, quase todos os modelos atingiram métricas melhores que o modelo PMF definido como *baseline*, sendo somente os modelos ARCH, GARCH e *Exponential Smoothing* que obtiveram resultados piores. Tanto o ARCH quanto o GARCH apresentaram problemas com a estacionariedade dos dados, sendo que os dados utilizados já respeitavam as condições de estacionariedade. No caso do *Exponential Smoothing*, devido às dispersões apresentadas pelas séries temporais da base de dados utilizada, as condições de tendência e sazonalidade definidas pelo modelo não contribuíram para a realização das predições.

O segundo passo da análise se concentrou em estudar os casos em que as técnicas de pré-processamento tiveram maior impacto nos resultados. Mesmo não sendo considerada nos resultados finais, a estacionariedade foi investigada em modelos mais simples, sem nenhum outro pré-processamento. Como resultado, um modelo que antes previa todas as observações de forma constante, com a aplicação da estacionariedade, passa a prever uma série temporal próxima aos valores reais, o que indicou que, sem outros pré-processamentos para compensar, é necessário que todas as condições de estacionariedade sejam respeitadas para que seja possível realizar boas previsões.

A escala impactou na melhora da maioria dos resultados, principalmente nos modelos das áreas de Aprendizado de Máquina e Aprendizado Profundo, devido à importância da magnitude em algoritmos fundamentados em distâncias entre pontos e dos métodos baseados em gradientes. Os *Look Backs* maiores, que consideram mais de 3 meses, também ajudaram os modelos a encontrar melhores valores para as métricas, devido à sazonalidade existente na série, apesar da dispersão. A necessidade do *Walk Forward* foi equilibrada, reiterando a ideia de treinar os modelos sem este pré-processamento e, se os resultados apresentarem *underfitting*, investir no uso deste método.

Por fim, a Rede de Representação proposta alcançou os melhores resultados ao utilizar *Look Backs* maiores do que 3 meses e ao desativar a função de ativação em modelos derivados da Teoria de Séries Temporais. Não houve um investimento na otimização da Rede de Representação, mas ela superou as expectativas ao contribuir para vários modelos, mostrando não só a importância de utilizar informações dos meses anteriores, mas também em como ainda há espaço para a criação de novas técnicas de pré-processamento que consigam contribuir para melhores resultados. A conclusão do uso desta técnica se concentrou no significado contido nas características geradas por ela durante o treinamento dos pesos. Essas características representaram as informações de vários meses anteriores, criando uma nova informação. Como os modelos derivados da Teoria de Séries Temporais se baseiam nos *lags* anteriores, supõe-se que as características tenham gerado *lags* mais significativos, impactando positivamente nas previsões.

A última etapa foi responsável por apresentar as melhores metodologias, por retratar o comportamento dos modelos em horizontes de previsão de 12 e 60 meses e por treinar e avaliar os dois melhores modelos de cada modo em outras 6 bases de dados diferentes, mantendo os valores de hiperparâmetros e parâmetros de pré-processamento encontrados nas

otimizações com a base de dados principal de Afrânio. Como resultado, em Afrânio, o melhor modelo Univariado foi a GRU, com RMSE de 0,6739 e MAPE de 7,18% e, no modo Multivariado, a LSTM de sobressaiu, com RMSE de 0,6137 e MAPE de 6,99%. Considerando todas as 7 bases de dados, no modo Univariado, o *Gradient Boosting* se destacou, com RMSE médio de 0,7257 e MAPE médio de 7,60%. Dentre os modelos Multivariados, a LSTM continuou apresentando os melhores resultados, com RMSE médio de 0,6499 e MAPE médio de 7,65%. A conclusão desta etapa se resume em resultados muito melhores do que as métricas do modelo *baseline* e com valores de MAPE que indicam que as previsões são altamente precisas. Além dos melhores resultados, nas visualizações das previsões no horizonte de 60 meses, foi possível notar que todos os modelos capturaram o comportamento geral das séries, mesmo com uma redução no conjunto de treinamento, o que ressalta a afirmação de que os modelos estavam ajustados adequadamente.

Em trabalhos futuros a serem desenvolvidos, recomenda-se um estudo mais aprofundado nos dados de Estreito, *Offshore CE*, *Offshore SC*, Palmas, Quilombo e São João, visando descobrir se é possível melhorar as previsões ao alterar os hiperparâmetros dos modelos e os parâmetros de pré-processamento. Além disso, poderia ser feito um investimento na Rede de Representação, alterando sua topologia, adicionando e otimizando seus hiperparâmetros e tentando desvendar se esta técnica consegue compensar a falta de algum dos pré-processamentos observados neste trabalho.

REFERÊNCIAS

- ABEEólica (2021). Usinas Instaladas no Brasil. <http://abeeolica.org.br/>. Acessado em 04/05/2021.
- Abhigyan (2020). Understanding Polynomial Regression!!!. <https://medium.com/analytics-vidhya/understanding-polynomial-regression-5ac25b970e18>. Acessado em 16/10/2020.
- Alam, M. (2020). Six essential plots in time series data analysis. <https://towardsdatascience.com/six-essential-plots-for-time-series-data-analysis-477bd7e778ac>. Acessado em 23/09/2020.
- Alvarez, L. F. J., González, S. R., López, A. D., Delgado, D. A. H., Espinosa, R. e Gutiérrez, S. (2020). Renewable Energy Prediction through Machine Learning Algorithms. Em *2020 IEEE ANDESCON*, páginas 1-6.
- ANA (2014). Recursos Hídricos no Brasil: Usinas Hidrelétricas Existentes e Principais Planejadas. <https://arquivos.ana.gov.br/institucional/sge/CEDOC/Catalogo/2014/RecursosHidricosNoBrasilUsinasHidreletricasExistentesEPP.pdf>. Acessado em 18/12/2020.
- ANEEL (2019). Hidrelétricas superam 100 mil MW de potência fiscalizada. https://www.aneel.gov.br/sala-de-imprensa-exibicao-2/-/asset_publisher/zXQREz8EVlZ6/content/hidreletricas-superam-100-mil-mw-de-potencia-fiscalizada/656877?inheritRedirect=false. Acessado em 04/05/2021.
- Autonomio Talos (2020). Talos: Hyperparameter Optimization for Keras. <https://github.com/autonomio/talos>. Acessado em 03/05/2021.
- Bilgili, M., Sahin, B. e Yasar, A. (2007). Application of artificial neural networks for the wind speed prediction of target stations using reference stations data, Em *Renewable Energy*, Volume 32, número 14.
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C. e Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control*, 5ª edição, John Wiley & Sons, Inc., Hoboken, New Jersey.
- Brownlee, J. (2016a). 7 Time Series Datasets for Machine Learning. <https://machinelearningmastery.com/time-series-datasets-for-machine-learning/>. Acessado em 15/01/2021.

Brownlee, J. (2016b). How To Backtest Machine Learning Models for Time Series Forecasting.

<https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/>. Acessado em 10/11/2020.

Brownlee, J. (2016c). Bagging and Random Forest Ensemble Algorithms for Machine Learning.

<https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/>. Acessado em 17/10/2020.

Brownlee, J. (2017a). How to Remove Trends and Seasonality with a Difference Transform in Python.

<https://machinelearningmastery.com/remove-trends-seasonality-difference-transform-python/>. Acessado em 10/11/2020.

Brownlee, J. (2017b). Time Series Forecasting with the Long Short-Term Memory Network in Python.

<https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/>. Acessado em 10/11/2020.

Brownlee, J. (2018a). How to Develop LSTM Models for Time Series Forecasting.

<https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/>. Acessado em 10/11/2020.

Brownlee, J. (2018b). 11 Classical Time Series Forecasting Methods in Python.

<https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/>. Acessado em 21/09/2020.

Brownlee, J. (2018c). How to Model Volatility with ARCH and GARCH for Time Series Forecasting in Python.

<https://machinelearningmastery.com/develop-arch-and-garch-models-for-time-series-forecasting-in-python/>. Acessado em 05/10/2020.

Brownlee, J. (2020a). How to Create an ARIMA Model for Time Series Forecasting in Python.

<https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>. Acessado em 28/09/2020.

Brownlee, J. (2020b). A Gentle Introduction to SARIMA for Time Series Forecasting in Python.

<https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/>. Acessado em 30/09/2020.

- Campos, R. (2017). Árvores de Decisão. <https://medium.com/machine-learning-beyond-deep-learning/%C3%A1rvores-de-decis%C3%A3o-3f52f6420b69>. Acessado em: 16/10/2020.
- Capital (2020). Moving Average. <https://capital.com/moving-average-definition>. Acessado em 21/09/2020.
- CCEE (2017). Deck de preços do modelo NEWAVE, ref. 05/2017. <https://www.ccee.org.br/ccee/documentos/NW201705>. Acessado em 18/12/2020.
- COPEL (2018). Localização da Usina Hidrelétrica Colider. <https://www.copel.com/uhecolider/noticia.jsp?not=%2Fuhecolider%2Fpagcopel2.nsf%2Fdocs%2F41ADCC7347A798A6032578B1004B9B89>. Acessado em 18/12/2020.
- Data Science Academy (2019). Deep Learning Book. <http://www.deeplearningbook.com.br/>. Acessado em 20/10/2020.
- Del Ser, J., Casillas-Perez, D., Cornejo-Bueno, L., Prieto-Godino, L., Sanz-Justo, J., Casanova-Mateo, C. e Salcedo-Sanz, S. (2021). Randomization-based Machine Learning in Renewable Energy Prediction Problems: Critical Literature Review, New Results and Perspectives Em *CoRR*, vol. abs/2103.14624.
- Detzel, D. H. M., Bessa, M. R., Vallejos, C. A. V., Santos, A. B., Thomsen, L. S., Mine, M. R. M., Bloot, M. L. e Estrócio, J. P. (2011). Estacionariedade das Afluências às Usinas Hidrelétricas Brasileiras. Em *RBRH - Revista Brasileira de Recursos Hídricos*, vol. 16, n. 3.
- Dubey, B. (2020). Decision Tree Regression. <https://medium.com/@bhartendudubey/decision-tree-regression-e202008c2df>. Acessado em: 16/10/2020.
- Feijó, C. A. (2015). Conceitos Básicos de Séries Temporais para Modelagem Macroeconômica. http://www.icad.puc-rio.br/cfeijoo/pdf/revis%C3%A3o%20b%C3%A1sica%20s%C3%A9ries%20temporais_material%20de%20apoio_curso%20teoria%20macroeconomica_PPGE%20UFF.pdf. Acessado em 22/09/2020.
- Glen, S. (2016). Unit Root: Simple Definition, Unit Root Tests. <https://www.statisticshowto.com/unit-root/>. Acessado em 10/11/2020.
- Goldkamp, J. (2020). Cointegration for Time Series Analysis. <https://medium.com/analytics-vidhya/cointegration-for-time-series-analysis-1d0a758a20f1>. Acessado em 28/10/2020.

- Grover, P. (2017). Gradient Boosting from scratch. <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d>. Acessado em 19/10/2020.
- Gupta, M. (2019). Why Time Series has to be STATIONARY???. <https://medium.com/data-science-in-your-pocket/why-time-series-has-to-be-stationary-37ca8800ddf>. Acessado em 10/11/2020.
- Hill, T. (2018). *Part 1: A neural network from scratch - Foundation*. <https://towardsdatascience.com/part-1-a-neural-network-from-scratch-foundation-e2d119df0f40>. Acessado em 20/10/2020.
- Hyndman, R. J. e Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*, 2ª edição, OTexts: Melbourne, Australia.
- IRENA (2020). Wind energy. <https://www.irena.org/wind>. Acessado em 04/05/2021.
- Jiang, P., Qin, S., Wu, J. e Sun, B. (2015). Time Series Analysis and Forecasting for Wind Speeds Using Support Vector Regression Coupled with Artificial Intelligent Algorithms. Em *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, vol. 2015, n. 939305, 14 páginas.
- Karbhari, V. (2018). How to evaluate regression models? <https://medium.com/acing-ai/how-to-evaluate-regression-models-d183b4f5853d>. Acessado em 30/04/2021.
- Karim, R. (2018). Animated RNN, LSTM and GRU. <https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45>. Acessado em 23/10/2020.
- Keras API Reference (2020). Optimizers. <https://keras.io/api/optimizers/>. Acessado em 21/10/2020.
- Keras (2021a). Dense layer. https://keras.io/api/layers/core_layers/dense/. Acessado em 03/05/2021.
- Keras (2021b). LSTM layer. https://keras.io/api/layers/recurrent_layers/lstm/. Acessado em 03/05/2021.
- Keras (2021c). GRU layer. https://keras.io/api/layers/recurrent_layers/gru/. Acessado em 03/05/2021.

- Kostadinov, S. (2017). Understanding GRU Networks. <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>. Acessado em 26/10/2020.
- Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis*, 1ª edição, Springer: Berlin, Alemanha.
- Maitra, S. (2019). Time-series Analysis with VAR & VECM: Statistical approach. <https://towardsdatascience.com/vector-autoregressions-vector-error-correction-multivariate-model-a69daf6ab618>. Acessado em 28/10/2020.
- Malik, F. (2020). What Is Grid Search? <https://medium.com/fintechexplained/what-is-grid-search-c01fe886ef0a>. Acessado em 03/05/2021.
- Masum, M. (2020). Time Series Analysis: Identifying AR and MA using ACF and PACF Plots. <https://towardsdatascience.com/identifying-ar-and-ma-terms-using-acf-and-pacf-plots-in-time-series-forecasting-ccb9fd073db8>. Acessado em 24/09/2020.
- Miller, J. (2018). Predicting Success in Online Education. <https://towardsdatascience.com/predicting-success-in-online-education-2b5979fa7016>. Acessado em 26/04/2021.
- Mohr, F. X. (2019). An Introduction to Vector Error Correction Models (VECMs). <https://www.r-econometrics.com/timeseries/vecintro/>. Acessado em 28/10/2020.
- Moreira, S. (2018). Rede Neural Perceptron Multicamadas. <https://medium.com/ensina-ai/rede-neural-perceptron-multicamadas-f9de8471f1a9>. Acessado em 21/10/2020.
- Moreno, J. J. M., Pol, A. P., Abad, A. S. e Blasco, B. C. (2013). Using the R-MAPE index as a resistant measure of forecast accuracy. Em *Psicothema 2013*, vol. 25 n. 4, páginas 500-506.
- Norte Energia (2011). Relatório das Usinas Hidrelétricas de Belo Monte. <http://www2.defensoria.pa.def.br/portal/anexos/File/BeloMonte/PBA/Volume%20I%20-%20Item%2001%20a%2003/VOL%20I%20-%201%20-%20caracterizacao.pdf>. Acessado em 18/12/2020.

- Oguntayo, S. (2020). Preprocessing Time Series Data for Supervised Machine Learning. <https://towardsdatascience.com/preprocessing-time-series-data-for-supervised-learning-2e27493f44ae>. Acessado em 04/05/2021.
- Olah, C. (2015). Understanding LSTM Networks. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acessado em 23/10/2020.
- Oliveira, B. (2019). Características das séries temporais. <https://operdata.com.br/blog/caracteristicas-das-series-temporais/>. Acessado em 10/11/2020.
- Papacharalampous, G. e Tyrallis, H. (2020). Hydrological time series forecasting using simple combinations: Big data testing and investigations on one-year ahead river flow predictability. Em *Journal of Hydrology*, vol. 590, n. 125205.
- Pardoe, I. (2020). Regression Methods: Autoregressive Models. <https://online.stat.psu.edu/stat501/lesson/14/14.1>. Acessado em 21/09/2020.
- Peixeiro, M. (2019a). The Complete Guide to Time Series Analysis and Forecasting. <https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775>. Acessado em 10/11/2020.
- Peixeiro, M. (2019b). Almost Everything You Need to Know About Time Series. <https://towardsdatascience.com/almost-everything-you-need-to-know-about-time-series-860241bdc578>. Acessado em: 05/10/2020.
- Peixeiro, M. (2020). Advanced Time Series Analysis with ARMA and ARIMA. <https://towardsdatascience.com/advanced-time-series-analysis-with-arma-and-arima-a7d9b589ed6d>. Acessado em: 22/09/2020.
- PennState University. (2020). ARCH/GARCH Models. <https://online.stat.psu.edu/stat501/lesson/14/14.1>. Acessado em 05/10/2020.
- Phi, M. (2018). Illustrated Guide to LSTM's and GRU's: A step by step explanation. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>. Acessado em 23/10/2020.
- Ramkumar, R. (2019a). Time Series Talk : Autocorrelation and Partial Autocorrelation. https://youtu.be/DeORzP0go5I?list=PLvcbYUQ5t0UHOLnBz146_Q6QKtFgfMGc3. Acessado em 10/11/2020.

- Ramkumar, R. (2019b). Time Series Talk: Autoregressive Model. https://youtu.be/5-2C4eO4cPQ?list=PLvcbYUQ5t0UHOLnBz146_Q6QKtFgfMGc3. Acessado em 21/09/2020.
- Ramkumar, R. (2019c). Time Series Talk: Moving Average Model. https://youtu.be/voryLhxiPzE?list=PLvcbYUQ5t0UHOLnBz146_Q6QKtFgfMGc3. Acessado em 21/09/2020.
- Ramkumar, R. (2019d). Time Series Talk : ARMA Model. https://youtu.be/HhvTlan06AM?list=PLvcbYUQ5t0UHOLnBz146_Q6QKtFgfMGc3. Acessado em 22/09/2020.
- Ramkumar, R. (2019e). Time Series Talk : ARIMA Model. https://youtu.be/3UmyHed0iYE?list=PLvcbYUQ5t0UHOLnBz146_Q6QKtFgfMGc3. Acessado em 28/09/2020.
- Ramkumar, R. (2019f). Time Series Talk : Seasonal ARIMA Model. https://youtu.be/WjeGUs6mzXg?list=PLvcbYUQ5t0UHOLnBz146_Q6QKtFgfMGc3. Acessado em 30/09/2020.
- Ramkumar, R. (2019g). Time Series Talk : ARCH Model. https://youtu.be/Li95a2biFCU?list=PLvcbYUQ5t0UHOLnBz146_Q6QKtFgfMGc3. Acessado em 05/10/2020.
- Ramkumar, R. (2019h). GARCH Model : Time Series Talk. https://youtu.be/inoBpq1UE4?list=PLvcbYUQ5t0UHOLnBz146_Q6QKtFgfMGc3. Acessado em 05/10/2020.
- Ramkumar, R. (2019i). Vector Auto Regression : Time Series Talk. https://www.youtube.com/watch?v=UQQHSbeIaB0&list=RDCMUcUcpVo i5KkJmnE3bvEhHR0Q&start_radio=1. Acessado em 26/10/2020.
- Raschka, S. (2020). MinMax Scaling. http://rasbt.github.io/mlxtend/user_guide/preprocessing/minmax_scaling/. Acessado em 14/01/2021.
- Ritchie, H. e Roser, M. (2020). Renewable Energy. <https://ourworldindata.org/renewable-energy>. Acessado em 04/05/2021.
- Rodrigues, L. H. (2015). Regressão Linear Simples. <https://www.ime.usp.br/~fmachado/MAE229/AULA10.pdf>. Acessado em: 16/10/2020.

- Salvi, J. (2019). Significance of ACF and PACF Plots In Time Series Analysis. <https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8>. Acessado em 10/11/2020.
- São Manoel Energia (2018). São Manoel: a Empresa. <https://www.saomanoelenergia.com.br/pt-br/institucional>. Acessado em 18/12/2020.
- Scikit-Learn (2020a). `sklearn.linear_model.LinearRegression`. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. Acessado em 03/05/2021.
- Scikit-Learn (2020b). `sklearn.preprocessing.PolynomialFeatures`. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html>. Acessado em 03/05/2021.
- Scikit-Learn (2020c). `sklearn.tree.DecisionTreeRegressor`. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html>. Acessado em 03/05/2021.
- Scikit-Learn (2020d). `sklearn.ensemble.BaggingRegressor`. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html>. Acessado em 03/05/2021.
- Scikit-Learn (2020e). `sklearn.ensemble.RandomForestRegressor`. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>. Acessado em 03/05/2021.
- Scikit-Learn (2020f). `sklearn.ensemble.GradientBoostingRegressor`. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html>. Acessado em 03/05/2021.
- Scikit-Learn (2021). `sklearn.preprocessing.MinMaxScaler`. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>. Acessado em 14/01/2021.
- Scikit-Optimize (2020a). Getting started. https://scikit-optimize.github.io/stable/getting_started.html#. Acessado em 03/05/2021.
- Scikit-Optimize (2020b). 5. skopt's top level minimization functions. https://scikit-optimize.github.io/stable/modules/minimize_functions.html#. Acessado em 03/05/2021.

- Seif, G. (2018). Selecting the best Machine Learning algorithm for your regression problem. <https://towardsdatascience.com/selecting-the-best-machine-learning-algorithm-for-your-regression-problem-20c330bad4ef>. Acessado em: 16/10/2020.
- Sen, B. (2019). An Introduction to Time Series Analysis. <https://medium.com/swlh/an-introduction-to-time-series-analysis-ef1a9200717a>. Acessado em 10/11/2020.
- Sharma, S. (2017). What the Hell is Perceptron? The Fundamentals of Neural Networks. <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>. Acessado em 20/10/2020.
- Sheppard, K. (2019). Introduction to ARCH Models. <https://arch.readthedocs.io/en/latest/univariate/introduction.html>. Acessado em 03/05/2021.
- Singh, H. (2018). Understanding Gradient Boosting Machines. <https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab>. Acessado em 19/10/2020.
- Sinop Energia (2020). Localização da Usina Hidrelétrica Sinop. <https://www.sinopenergia.com.br/localizacao/>. Acessado em 18/12/2020.
- Smith, T. (2020). Autocorrelation. <https://www.investopedia.com/terms/a/autocorrelation.asp>. Acessado em 10/11/2020.
- Srivastava, T. (2015). A Complete Tutorial on Time Series Modeling in R. <https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>. Acessado em 10/11/2020.
- Statsmodels (2021a). statsmodels.tsa.arima_model.ARMA. https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima_model.ARMA.html. Acessado em 03/05/2021.
- Statsmodels (2021b). statsmodels.tsa.arima.model.ARIMA. <https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html>. Acessado em 03/05/2021.
- Statsmodels (2021c). statsmodels.tsa.statespace.sarimax.SARIMAX. <https://www.statsmodels.org/stable/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html>. Acessado em 03/05/2021.

- Statsmodels (2021d). statsmodels.tsa.holtwinters.ExponentialSmoothing. <https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.html>. Acessado em 03/05/2021.
- Statsmodels (2021e). statsmodels.tsa.vector_ar.var_model.VAR. https://www.statsmodels.org/stable/generated/statsmodels.tsa.vector_ar.var_model.VAR.html. Acessado em 03/05/2021.
- Statsmodels (2021f). statsmodels.tsa.vector_ar.vecm.VECM. https://www.statsmodels.org/stable/generated/statsmodels.tsa.vector_ar.vecm.VECM.html. Acessado em 03/05/2021.
- Swaroop, M. V. e Reddy, P. L. (2018). Solar and Wind Power Forecasting with Optimal ARIMA Parameters. Em *International Journal of Engineering & Technology* 7(1.8):201.
- Tyagi, N. (2020a). Introduction to Time Series Analysis: Time-Series Forecasting Machine learning Methods & Models. <https://medium.com/analytics-steps/introduction-to-time-series-analysis-time-series-forecasting-machine-learning-methods-models-ecaa76a7b0e3>. Acessado em 10/11/2020.
- Tyagi, N. (2020b). 7 types of regression techniques you should know in Machine Learning. <https://www.analyticssteps.com/blogs/7-types-regression-technique-you-should-know-machine-learning>. Acessado em: 16/10/2020.
- UHE Teles Pires (2020). Sobre a UHE Teles Pires. <https://www.uhetelespires.com.br/site/sobre-a-uhe-teles-pires>. Acessado em 18/12/2020.
- Venkatachalam, M. (2019). Recurrent Neural Networks. <https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce>. Acessado em 23/10/2020.
- Vickery, R. (2020). The Art of Finding the Best Features for Machine Learning. <https://towardsdatascience.com/the-art-of-finding-the-best-features-for-machine-learning-a9074e2ca60d>. Acessado em 09/02/2021.
- Wilson, A. (2019). A Brief Introduction to Supervised Learning. <https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590>. Acessado em 10/11/2020.

Xia, Y. (2020). Tune the hyperparameters of your deep learning networks in Python using Keras and Talos. <https://towardsdatascience.com/tune-the-hyperparameters-of-your-deep-learning-networks-in-python-using-keras-and-talos-2a2a38c5ac31>. Acessado em 03/05/2021.

APÊNDICE A - HIPERPARÂMETROS DE TODOS OS MELHORES MODELOS

Tabela A.1: Valores dos melhores hiperparâmetros encontrados para todos os modelos.

FONTE: elaborado pela autora (2021).

Modelo	Nome Hiperparâmetro	Univariado	Multivariado
		Valor Hiperparâmetro	Valor Hiperparâmetro
AR	p	7	-
MA	q	10	-
ARMA	p	2	-
	q	2	-
ARIMA	p	0	-
	d	0	-
	q	5	-
SARIMA	p	0	-
	d	0	-
	q	0	-
	P	2	-
	D	1	-
	Q	1	-
	S	6	-
ARCH	$mean$	<i>Constant</i>	-
	$lags$	1	-
	p	1	-
	o	0	-
	q	2	-
	$dist$	<i>Gaussian</i>	-

GARCH	<i>mean</i>	AR	-
	<i>lags</i>	1	-
	<i>p</i>	1	-
	<i>o</i>	1	-
	<i>q</i>	1	-
	<i>dist</i>	<i>Gaussian</i>	-
<i>Exponential Smoothing (Simple, Double e Triple)</i>	<i>trend</i>	<i>None</i>	-
	<i>seasonal</i>	<i>None</i>	-
	<i>seasonal_periods</i>	0	-
VAR	<i>maxlags</i>	-	12
	<i>method</i>	-	<i>ols</i>
	<i>ic</i>	-	<i>None</i>
	<i>trend</i>	-	<i>nc</i>
VECM	<i>k_ar_diff</i>	-	12
	<i>coint_rank</i>	-	2
	<i>deterministic</i>	-	<i>nc</i>
	<i>seasons</i>	-	0
Regressão Linear	<i>fit_intercept</i>	<i>False</i>	<i>False</i>
Regressão Polinomial	<i>degree</i>	3	-
	<i>interaction_only</i>	<i>True</i>	-
	<i>include_bias</i>	<i>True</i>	-
	<i>order</i>	C	-
Árvore de Regressão	<i>criterion</i>	<i>mse</i>	<i>mae</i>
	<i>splitter</i>	<i>random</i>	<i>best</i>
	<i>max_depth</i>	37	30
	<i>min_samples_split</i>	27	19
	<i>min_samples_leaf</i>	2	16

	<i>max_features</i>	<i>sqrt</i>	<i>None</i>
	<i>random_state</i>	36	12
	<i>max_leaf_nodes</i>	60	35
<i>Bagging</i>	<i>base_estimator</i>	<i>LinearRegression</i>	SVR
	<i>n_estimators</i>	54	14
	<i>max_features</i>	7	1
	<i>bootstrap</i>	<i>True</i>	<i>True</i>
	<i>random_state</i>	21	42
Florestas Aleatórias	<i>n_estimators</i>	100	45
	<i>criterion</i>	<i>mae</i>	<i>mae</i>
	<i>max_depth</i>	21	25
	<i>min_samples_split</i>	2	5
	<i>min_samples_leaf</i>	2	3
	<i>max_features</i>	<i>log2</i>	<i>log2</i>
	<i>max_leaf_nodes</i>	<i>None</i>	30
	<i>bootstrap</i>	<i>True</i>	<i>False</i>
	<i>random_state</i>	12	36
<i>Gradient Boosting</i>	<i>loss</i>	<i>lad</i>	<i>ls</i>
	<i>learning_rate</i>	0,1	0,1
	<i>n_estimators</i>	75	13
	<i>subsample</i>	1,0	1,0
	<i>criterion</i>	<i>friedman_mse</i>	<i>friedman_mse</i>
	<i>min_samples_split</i>	37	2
	<i>min_samples_leaf</i>	5	19
	<i>max_depth</i>	12	37
	<i>random_state</i>	28	28
	<i>max_features</i>	<i>sqrt</i>	<i>None</i>
	<i>max_leaf_nodes</i>	<i>None</i>	40

<i>Perceptron</i>	<i>units</i>	1	1
	<i>activation</i>	<i>tanh</i>	<i>tanh</i>
	<i>use_bias</i>	<i>False</i>	<i>False</i>
	<i>optimizers</i>	<i>sgd</i>	<i>adam</i>
	<i>losses</i>	<i>mean_squared_error</i>	<i>mean_squared_error</i>
	<i>batch_size</i>	1	1
	<i>epochs</i>	100	10
	<i>validation_split</i>	0,1	0,1
MLP	<i>units_1</i>	10	50
	<i>activation_1</i>	<i>relu</i>	<i>relu</i>
	<i>use_bias_1</i>	<i>True</i>	<i>True</i>
	<i>units_2</i>	10	100
	<i>activation_2</i>	<i>relu</i>	<i>tanh</i>
	<i>use_bias_2</i>	<i>True</i>	<i>True</i>
	<i>optimizers</i>	<i>adam</i>	<i>adam</i>
	<i>losses</i>	<i>mean_squared_error</i>	<i>mean_squared_error</i>
	<i>batch_size</i>	1	1
	<i>epochs</i>	100	50
	<i>validation_split</i>	0,1	0,1
LSTM	<i>units_1</i>	50	100
	<i>activation_1</i>	<i>sigmoid</i>	<i>relu</i>
	<i>recurrent_activation_1</i>	<i>relu</i>	<i>relu</i>
	<i>use_bias_1</i>	<i>True</i>	<i>True</i>
	<i>unit_forget_bias_1</i>	<i>True</i>	<i>True</i>
	<i>dropout_1</i>	0,0	0,2
	<i>recurrent_dropout_1</i>	0,0	0,0
	<i>go_backwards_1</i>	<i>False</i>	<i>False</i>
	<i>stateful_1</i>	<i>False</i>	<i>False</i>
	<i>unroll_1</i>	<i>False</i>	<i>False</i>

	<i>units_2</i>	50	50
	<i>activation_2</i>	<i>sigmoid</i>	<i>relu</i>
	<i>recurrent_activation_2</i>	<i>relu</i>	<i>relu</i>
	<i>use_bias_2</i>	<i>True</i>	<i>True</i>
	<i>unit_forget_bias_2</i>	<i>True</i>	<i>True</i>
	<i>dropout_2</i>	0,0	0,0
	<i>recurrent_dropout_2</i>	0,0	0,0
	<i>go_backwards_2</i>	<i>False</i>	<i>False</i>
	<i>stateful_2</i>	<i>False</i>	<i>False</i>
	<i>unroll_2</i>	<i>False</i>	<i>False</i>
	<i>optimizers</i>	<i>adam</i>	<i>adam</i>
	<i>losses</i>	<i>mean_squared_error</i>	<i>mean_squared_error</i>
	<i>batch_size</i>	1	1
	<i>epochs</i>	150	50
	<i>validation_split</i>	0,1	0,1
GRU	<i>units_1</i>	100	50
	<i>activation_1</i>	<i>tanh</i>	<i>relu</i>
	<i>recurrent_activation_1</i>	<i>relu</i>	<i>relu</i>
	<i>use_bias_1</i>	<i>True</i>	<i>True</i>
	<i>dropout_1</i>	0,0	0,0
	<i>recurrent_dropout_1</i>	0,0	0,0
	<i>go_backwards_1</i>	<i>False</i>	<i>False</i>
	<i>stateful_1</i>	<i>False</i>	<i>False</i>
	<i>unroll_1</i>	<i>False</i>	<i>False</i>
	<i>units_2</i>	50	50
	<i>activation_2</i>	<i>tanh</i>	<i>tanh</i>
	<i>recurrent_activation_2</i>	<i>relu</i>	<i>relu</i>
	<i>use_bias_2</i>	<i>True</i>	<i>True</i>
	<i>dropout_2</i>	0,0	0,0

	<i>recurrent_dropout_2</i>	0,0	0,0
	<i>go_backwards_2</i>	<i>False</i>	<i>False</i>
	<i>stateful_2</i>	<i>False</i>	<i>False</i>
	<i>unroll_2</i>	<i>False</i>	<i>False</i>
	<i>optimizers</i>	<i>adam</i>	<i>adam</i>
	<i>losses</i>	<i>mean_squared_error</i>	<i>mean_squared_error</i>
	<i>batch_size</i>	1	1
	<i>epochs</i>	200	200
	<i>validation_split</i>	0,1	0,1

APÊNDICE B - MELHORES RESULTADOS EM TODOS OS MODELOS

B.1 MODELOS UNIVARIADOS

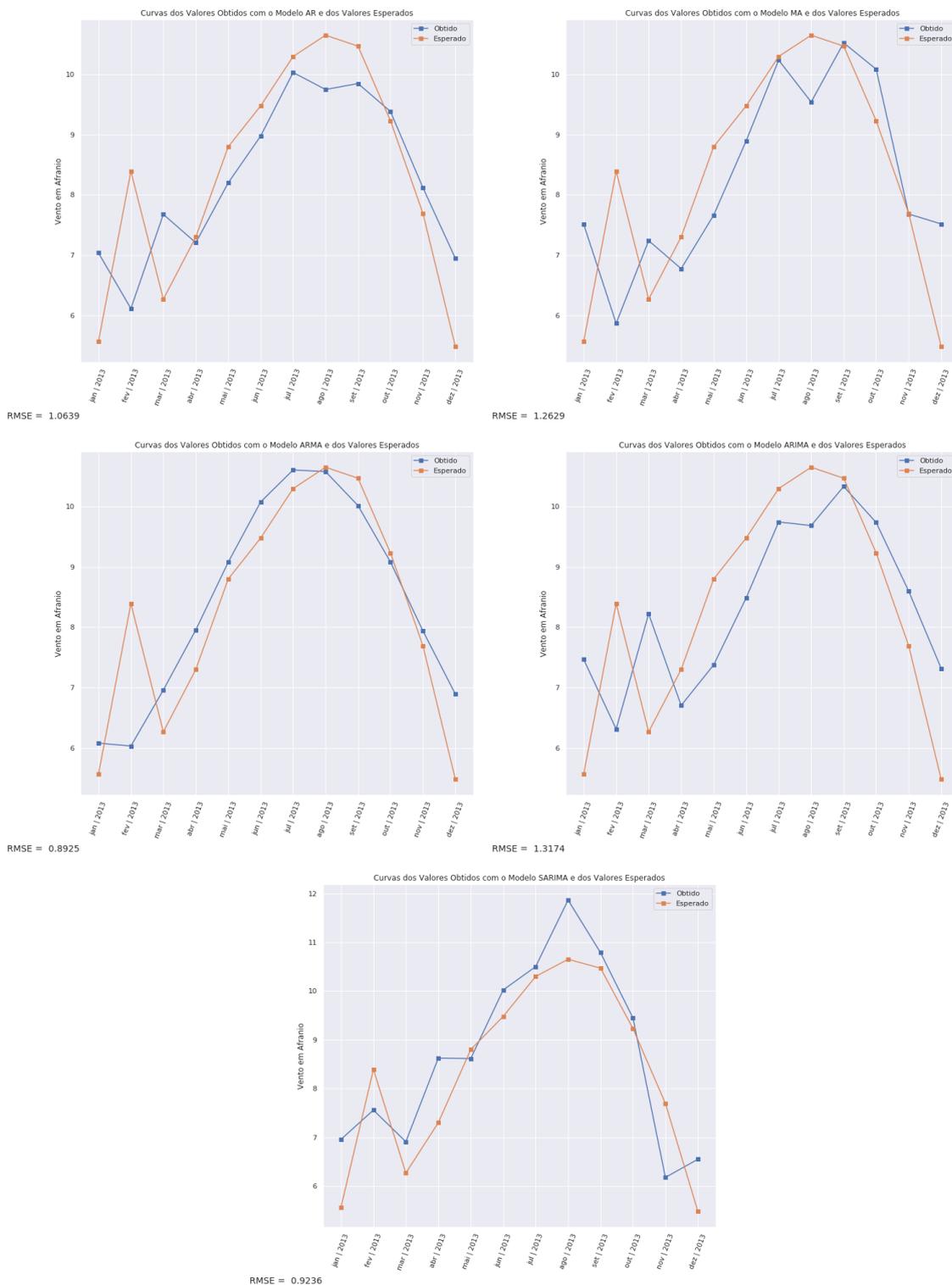


Figura B.1: Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos AR, MA, ARMA, ARIMA e SARIMA.

FONTE: elaborado pela autora (2021).

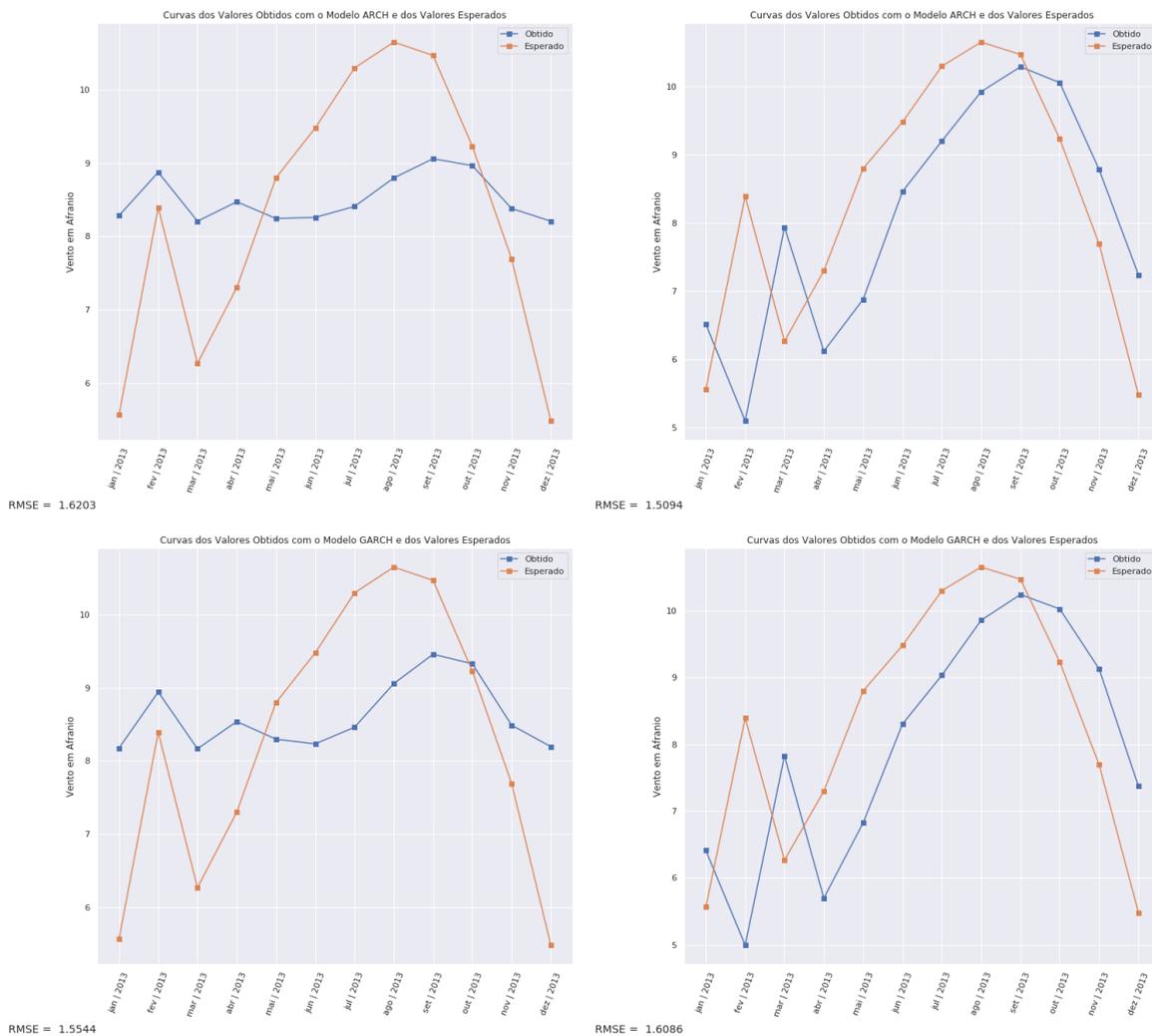


Figura B.2: Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos ARCH sem transformação de estacionariedade, ARCH estacionário, GARCH sem transformação de estacionariedade e GARCH estacionário.

FONTE: elaborado pela autora (2021).

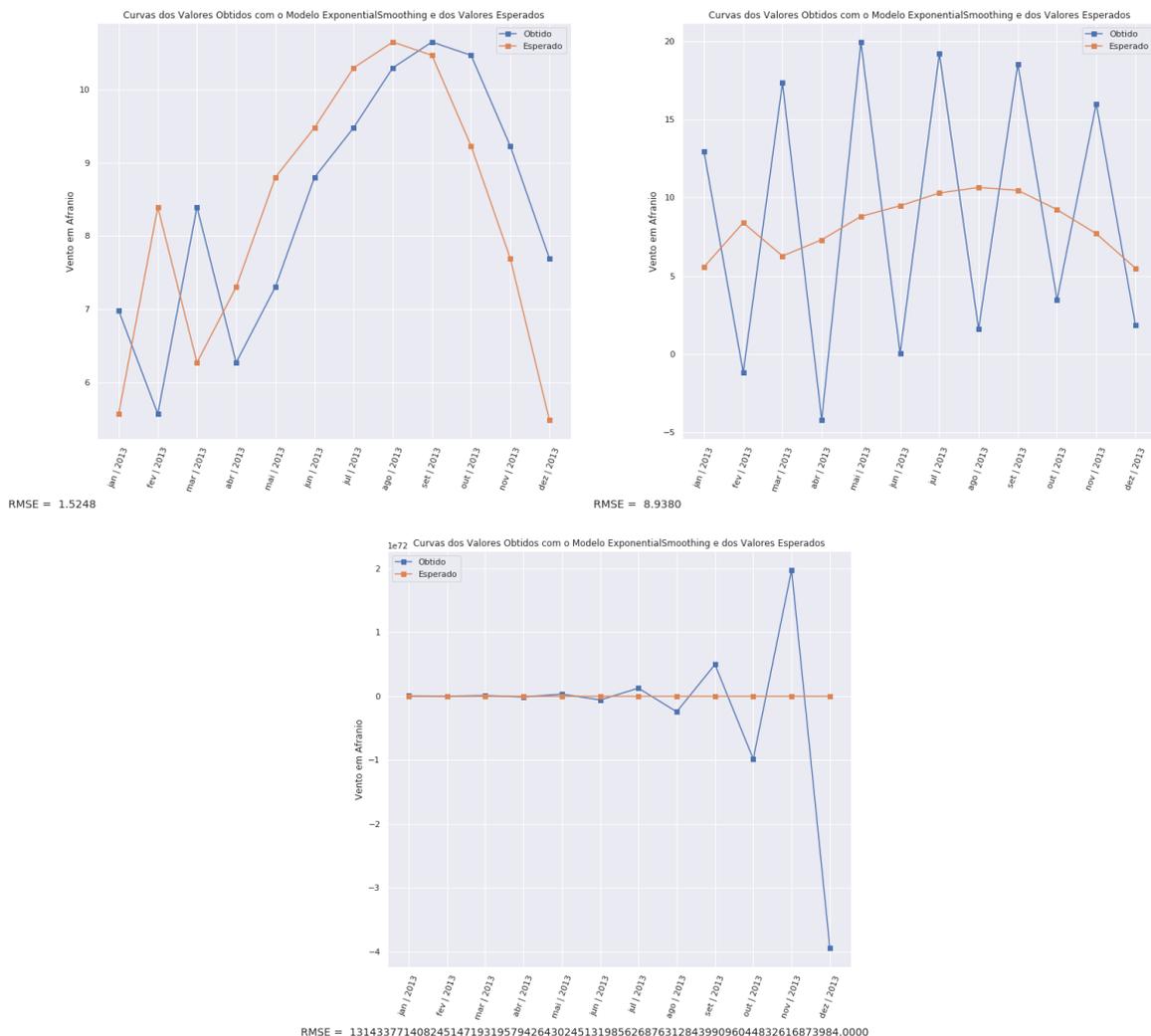


Figura B.3: Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos *Simple Exponential Smoothing*, *Double Exponential Smoothing* e *Triple Exponential Smoothing*.
 FONTE: elaborado pela autora (2021).

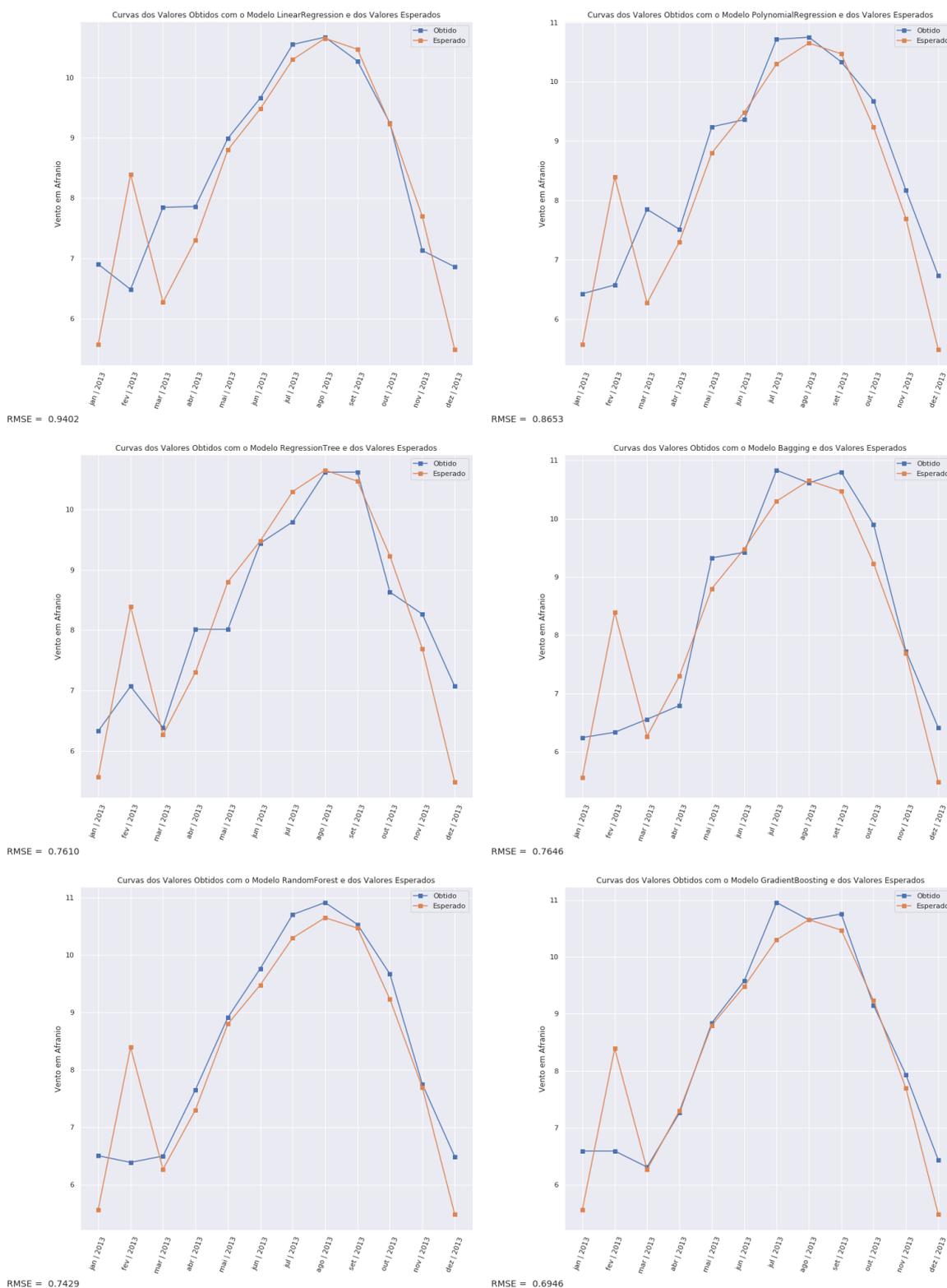


Figura B.4: Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos Regressão Linear, Regressão Polinomial, Árvore de Regressão, *Bagging*, Florestas Aleatórias e *Gradient Boosting*.

FONTE: elaborado pela autora (2021).

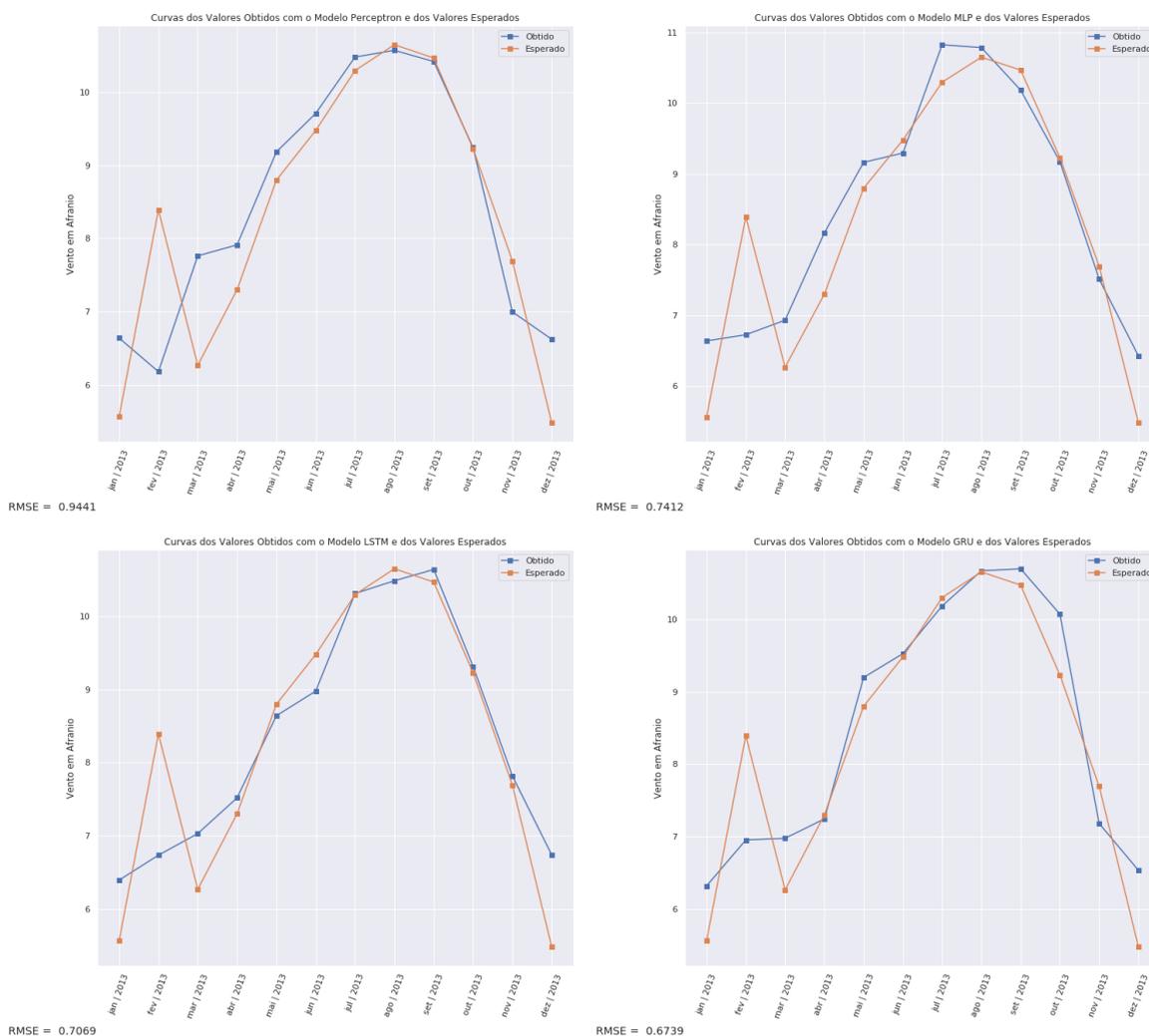


Figura B.5: Gráficos das curvas obtidas e esperadas, no modo Univariado, pelos modelos *Perceptron*, *MLP*, *LSTM* e *GRU*.

FONTE: elaborado pela autora (2021).

B.2 MODELOS MULTIVARIADOS

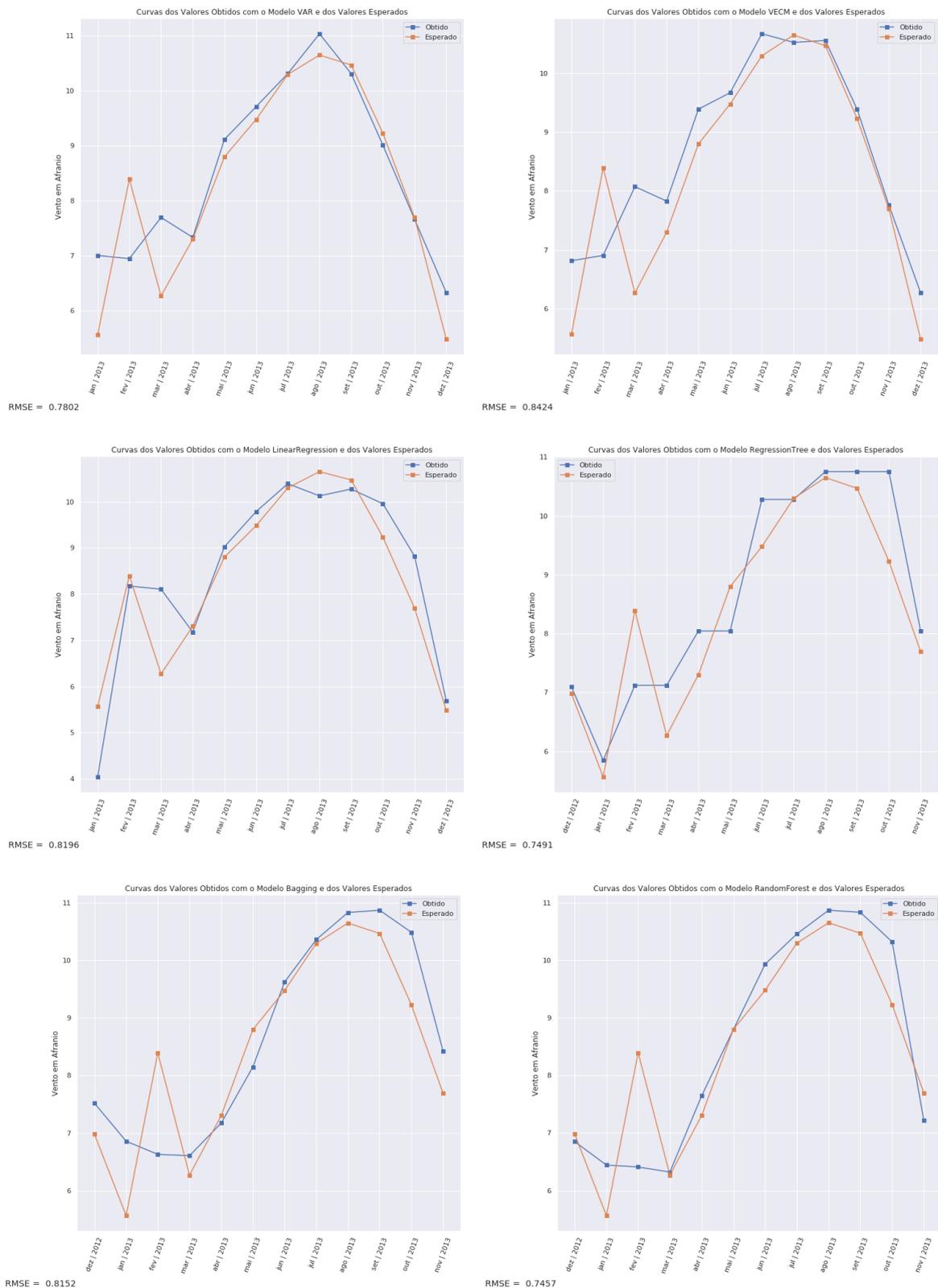


Figura B.6: Gráficos das curvas obtidas e esperadas, no modo Multivariado, pelos modelos VAR, VECM, Regressão Linear, Árvore de Regressão, *Bagging* e Florestas Aleatórias.
 FONTE: elaborado pela autora (2021).

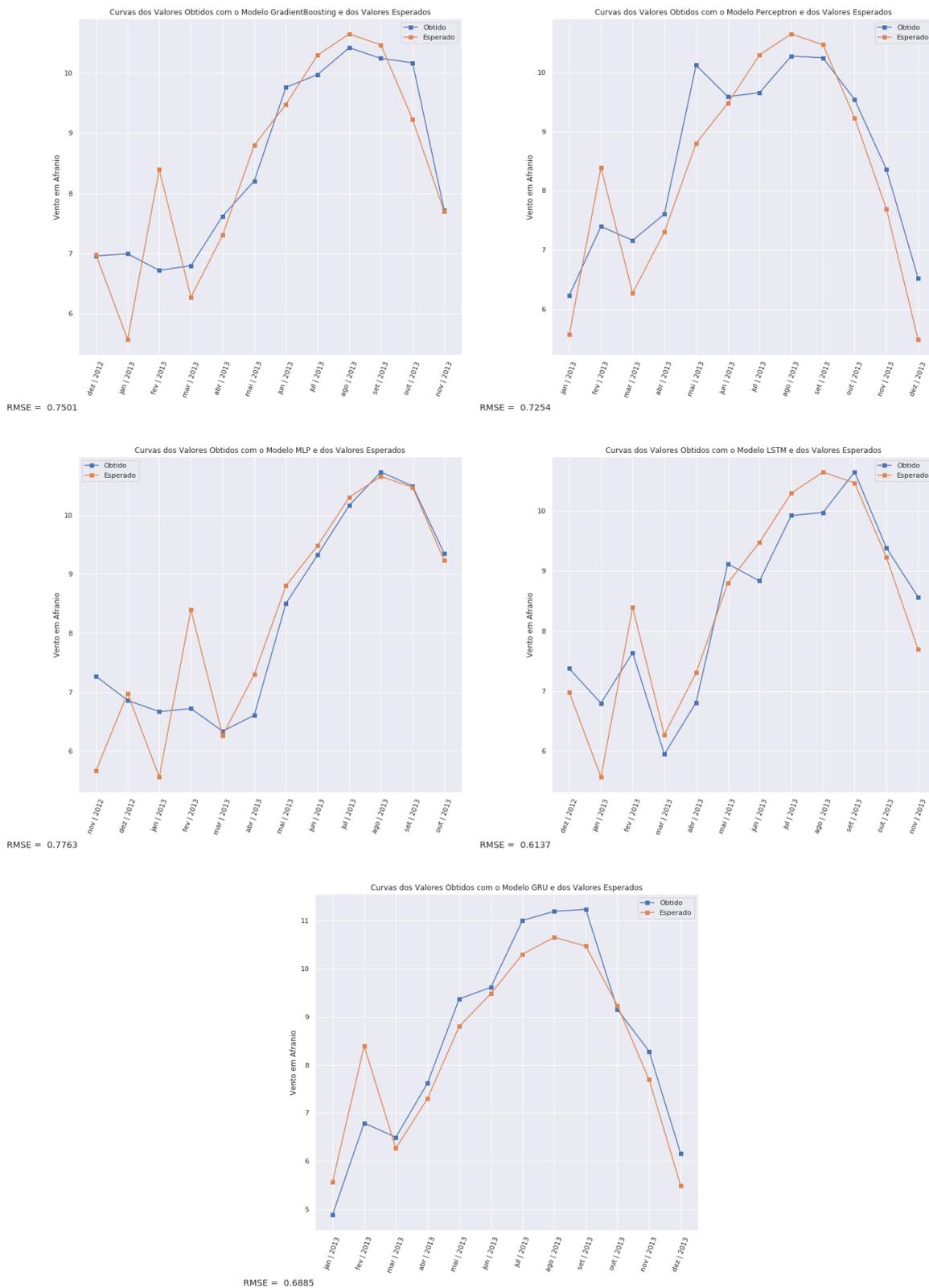


Figura B.7: Gráficos das curvas obtidas e esperadas, no modo Multivariado, pelos modelos *Gradient Boosting*, *Perceptron*, *MLP*, *LSTM* e *GRU*.
 FONTE: elaborado pela autora (2021).

APÊNDICE C - MÉTRICAS E GRÁFICOS DOS DOIS MELHORES MODELOS DE CADA MODO

C.1 MODELOS UNIVARIADOS COM HORIZONTE DE PREVISÃO DE 12 MESES

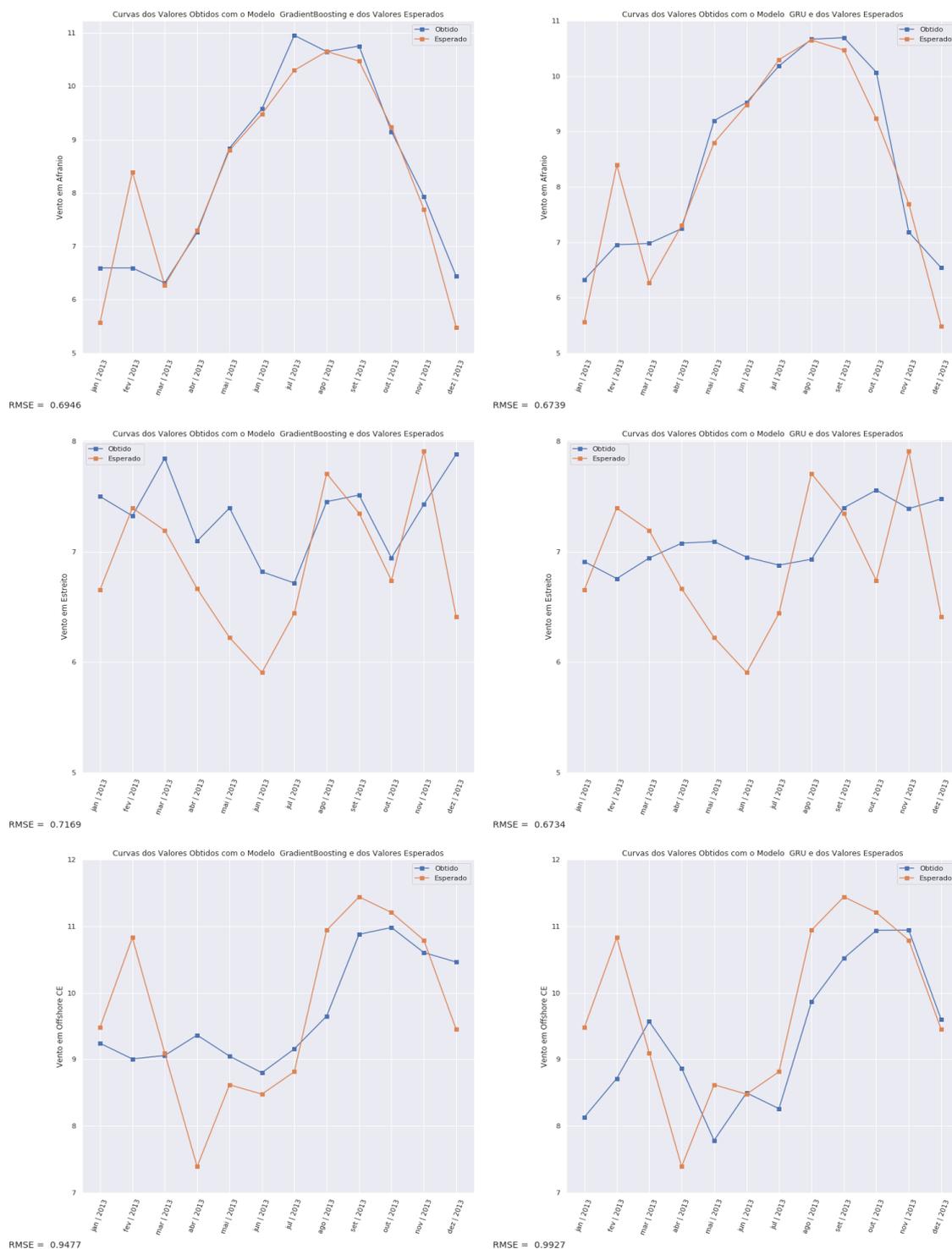


Figura C.1: Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos *Gradient Boosting* e *GRU* nos dados de Afrânio, Estreito e *Offshore CE*, com horizonte de previsão de 12 meses.

FONTE: elaborado pela autora (2021).

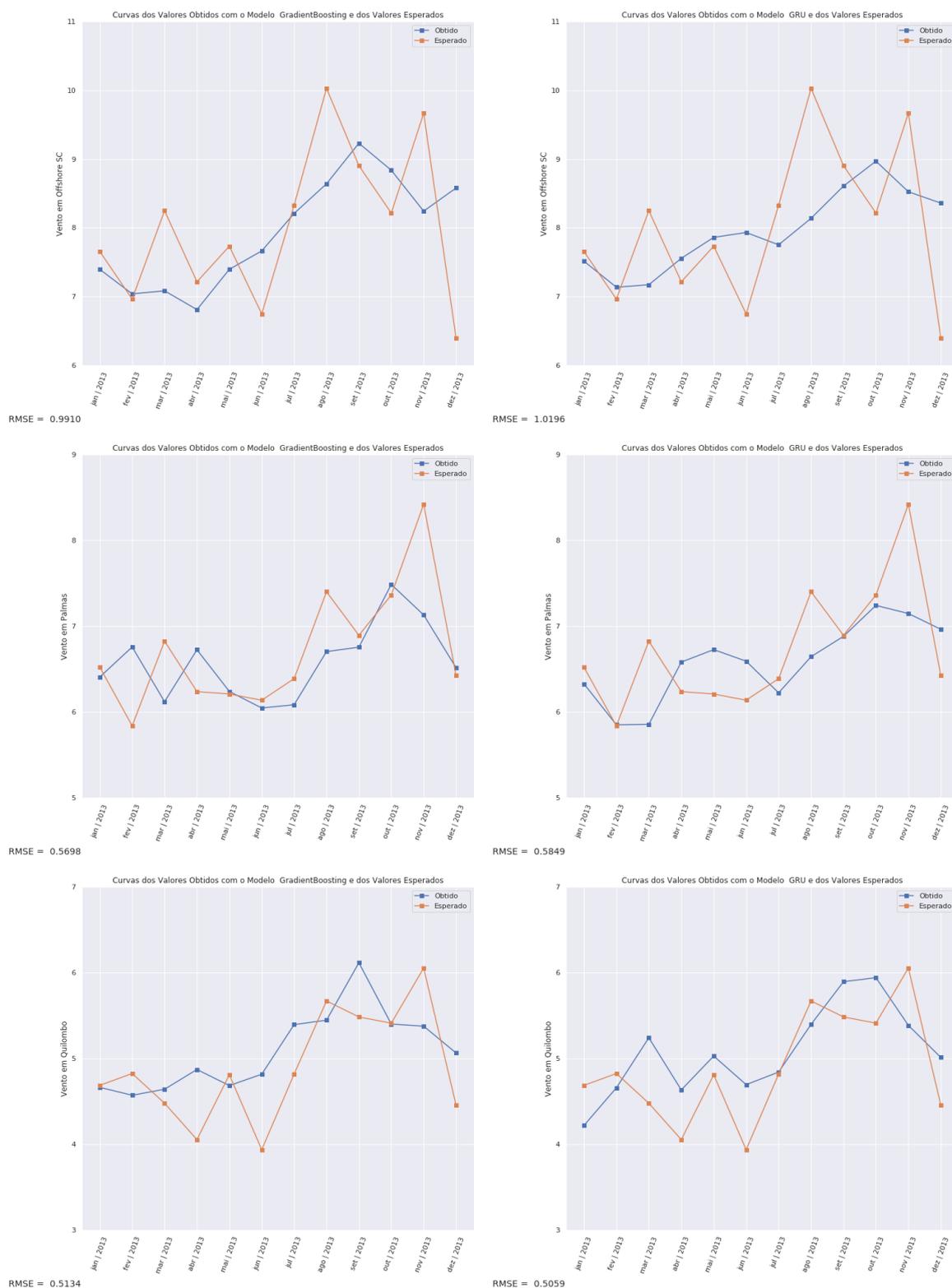


Figura C.2: Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos *Gradient Boosting* e *GRU* nos dados de *Offshore SC*, *Palmas* e *Quilombo*, com horizonte de previsão de 12 meses.

FONTE: elaborado pela autora (2021).

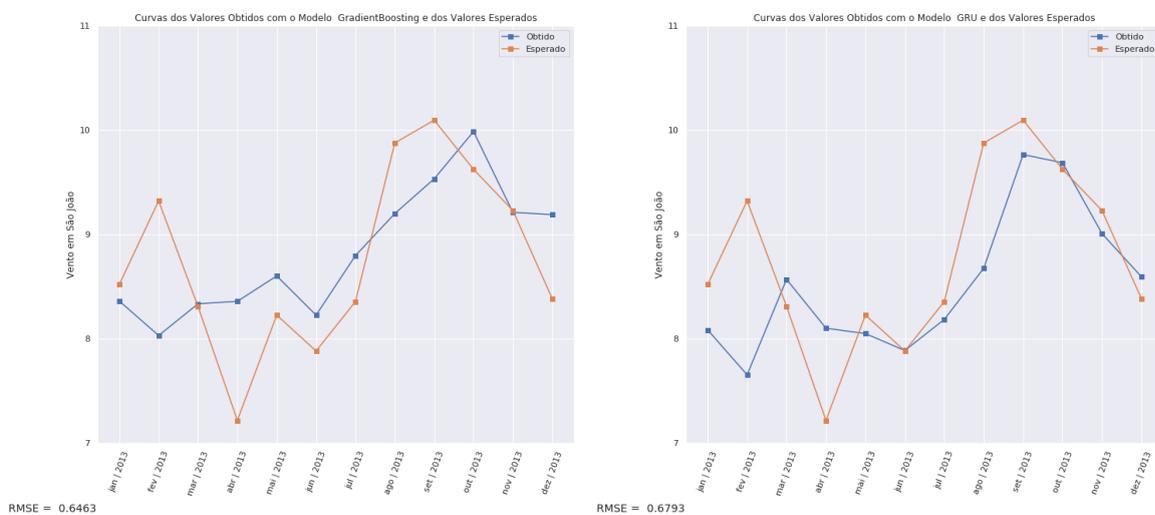


Figura C.3: Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos *Gradient Boosting* e GRU nos dados de São João, com horizonte de previsão de 12 meses.
 FONTE: elaborado pela autora (2021).

Tabela C.1: Melhores métricas e parâmetros de pré-processamento dos melhores modelos (*Gradient Boosting* e GRU) Univariados, com horizonte de previsão de 12 meses.

FONTE: elaborado pela autora (2021).

Planilha	Modelo	Estac.	Escala	Look Back	Walk Forw.	Rede de Repr.	Função de Ativ.	Look Back Net	MSE	RMSE	MAPE
1	GB	0	1	12	1	0	0	0	0,4824	0,6946	0,0609
	GRU	0	0	6	1	0	0	0	0,4541	0,6739	0,0718
2	GB	0	0	0	1	0	0	0	0,5139	0,7169	0,0880
	GRU	0	0	6	0	0	0	0	0,4535	0,6734	0,0890
3	GB	0	1	12	1	0	0	0	0,8981	0,9477	0,0753
	GRU	0	0	6	1	0	0	0	0,9855	0,9927	0,0820
4	GB	0	0	12	0	0	0	0	0,9820	0,9910	0,0980
	GRU	0	0	6	0	0	0	0	1,0396	1,0196	0,1018
5	GB	0	0	0	1	0	0	0	0,3247	0,5698	0,0602
	GRU	0	0	6	1	0	0	0	0,3421	0,5849	0,0641
6	GB	0	0	12	0	0	0	0	0,2636	0,5134	0,0892
	GRU	0	0	6	1	0	0	0	0,2559	0,5059	0,0956
7	GB	0	1	12	1	0	0	0	0,4177	0,6463	0,0601
	GRU	0	0	6	1	0	0	0	0,4614	0,6793	0,0531

C.2 MODELOS UNIVARIADOS COM HORIZONTE DE PREVISÃO DE 60 MESES

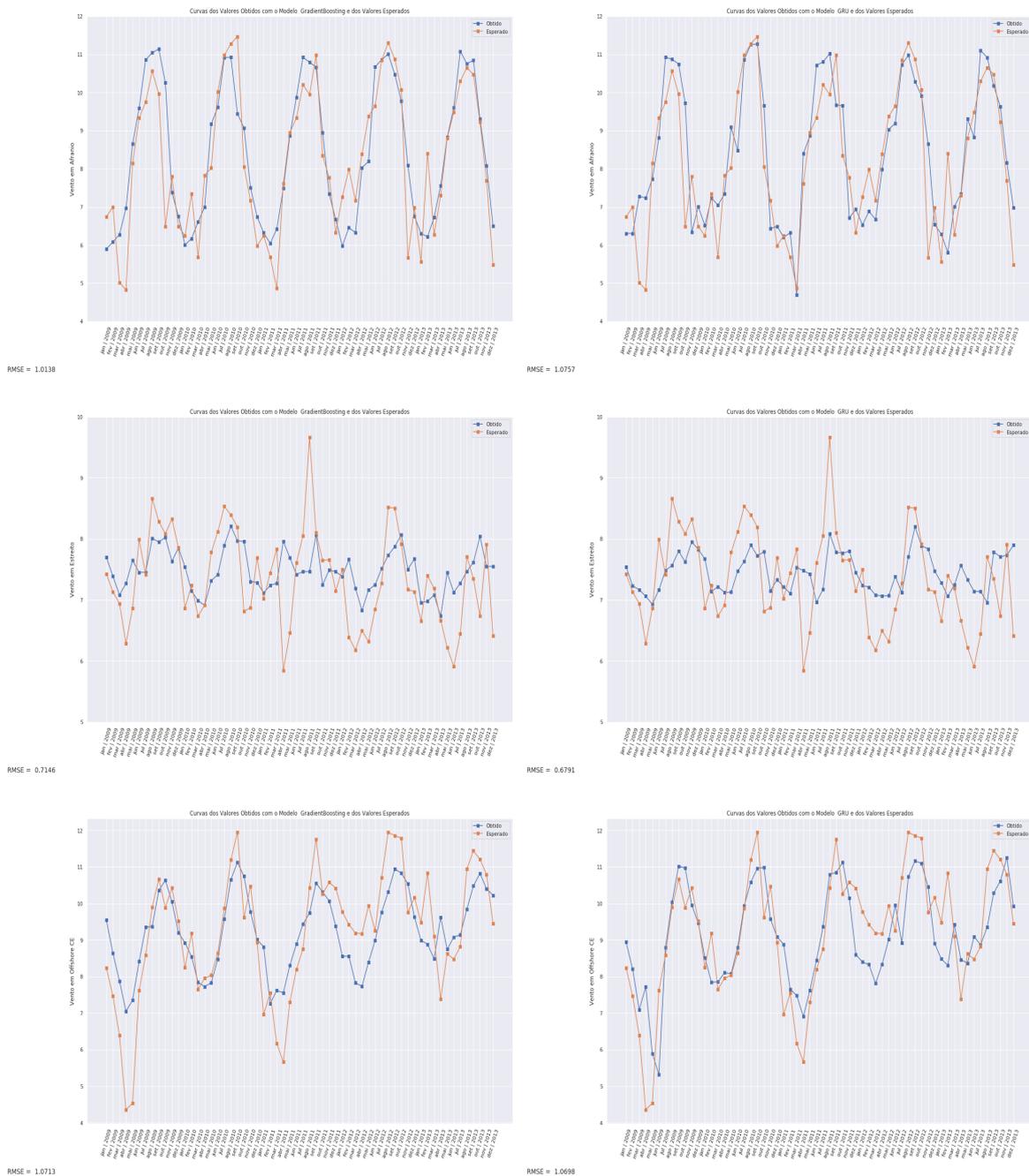


Figura C.4: Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos *Gradient Boosting* e *GRU* nos dados de Afrânio, Estreito e *Offshore CE*, com horizonte de previsão de 60 meses.

FONTE: elaborado pela autora (2021).

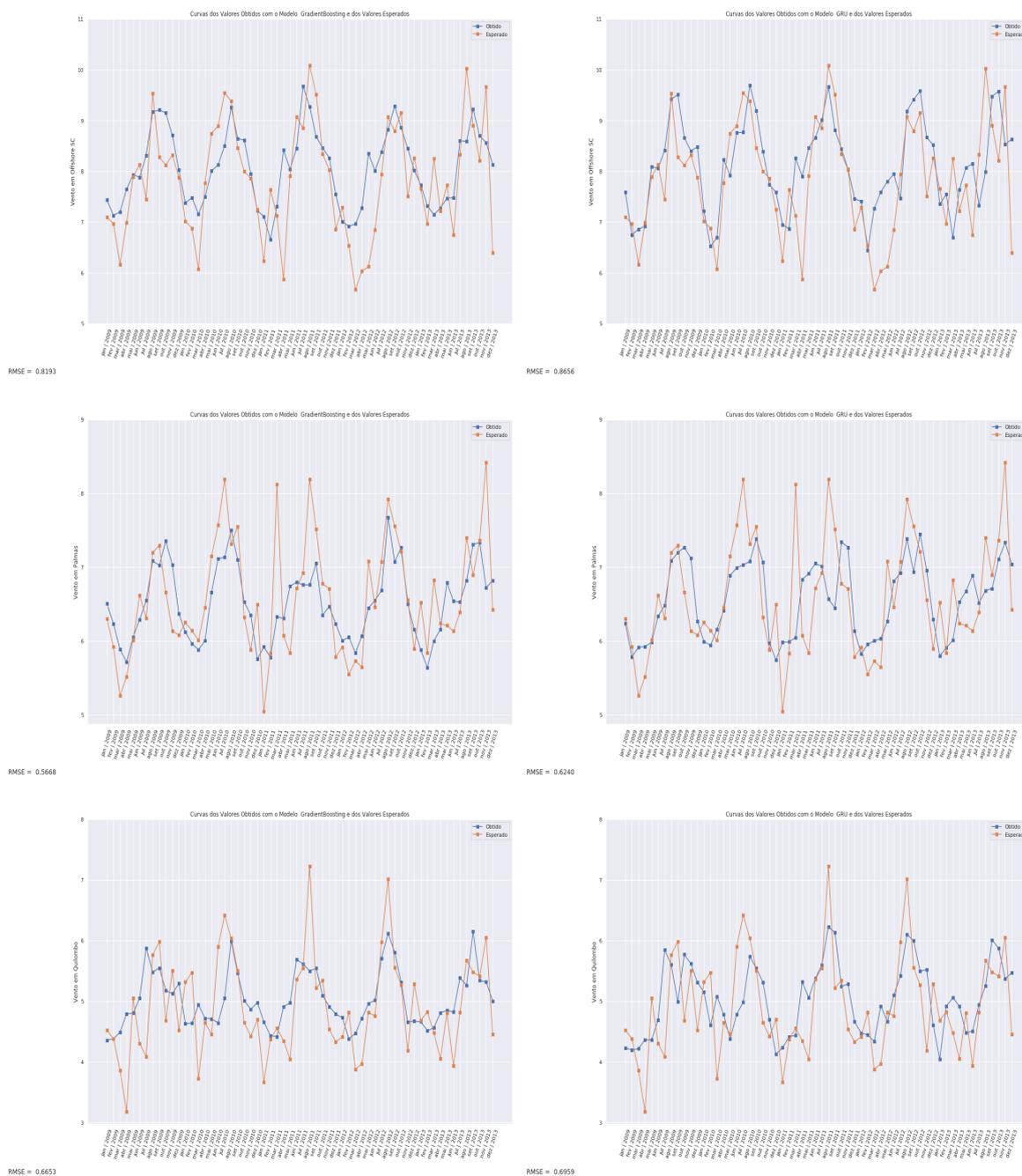


Figura C.5: Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos *Gradient Boosting* e GRU nos dados de *Offshore SC*, *Palmas* e *Quilombo*, com horizonte de previsão de 60 meses.

FONTE: elaborado pela autora (2021).

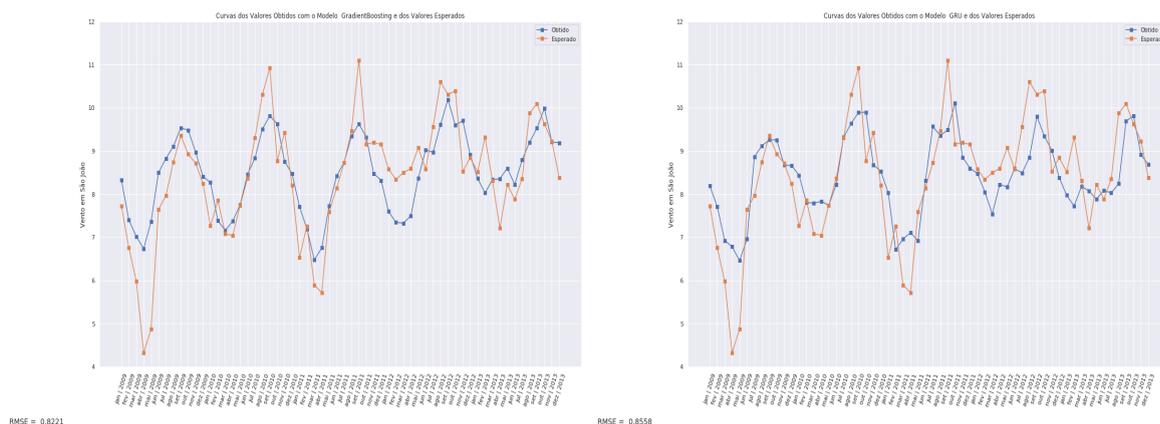


Figura C.6: Gráficos das curvas obtidas e esperadas, no modo Univariado, dos modelos *Gradient Boosting* e GRU nos dados de São João, com horizonte de previsão de 60 meses.

FONTE: elaborado pela autora (2021).

Tabela C.2: Melhores métricas e parâmetros de pré-processamento dos modelos *Gradient Boosting* e GRU univariados, com horizonte de previsão de 60 meses.

FONTE: elaborado pela autora (2021).

Planilha	Modelo	Estac.	Escala	Look Back	Walk Forw.	Rede de Repr.	Função de Ativ.	Look Back Net	MSE	RMSE	MAPE
1	GB	0	0	12	0	0	0	0	1,0278	1,0138	0,1032
	GRU	0	0	6	0	0	0	0	1,1571	1,0757	0,1117
2	GB	0	1	12	1	0	0	0	0,5106	0,7146	0,0761
	GRU	0	0	6	1	0	0	0	0,4612	0,6791	0,0780
3	GB	0	0	12	0	0	0	0	1,1477	1,0713	0,1095
	GRU	0	0	9	1	0	0	0	1,1446	1,0698	0,1013
4	GB	0	0	12	0	0	0	0	0,6713	0,8193	0,0867
	GRU	0	0	9	1	0	0	0	0,7493	0,8656	0,0910
5	GB	0	1	12	1	0	0	0	0,3212	0,5668	0,0635
	GRU	0	0	6	0	0	0	0	0,3894	0,6240	0,0716
6	GB	0	1	12	1	0	0	0	0,4426	0,6653	0,1099
	GRU	0	0	6	0	0	0	0	0,4842	0,6959	0,1196
7	GB	0	1	12	1	0	0	0	0,6759	0,8221	0,0851
	GRU	0	0	6	0	0	0	0	0,7324	0,8558	0,0885

C.3 MODELOS MULTIVARIADOS COM HORIZONTE DE PREVISÃO DE 12 MESES

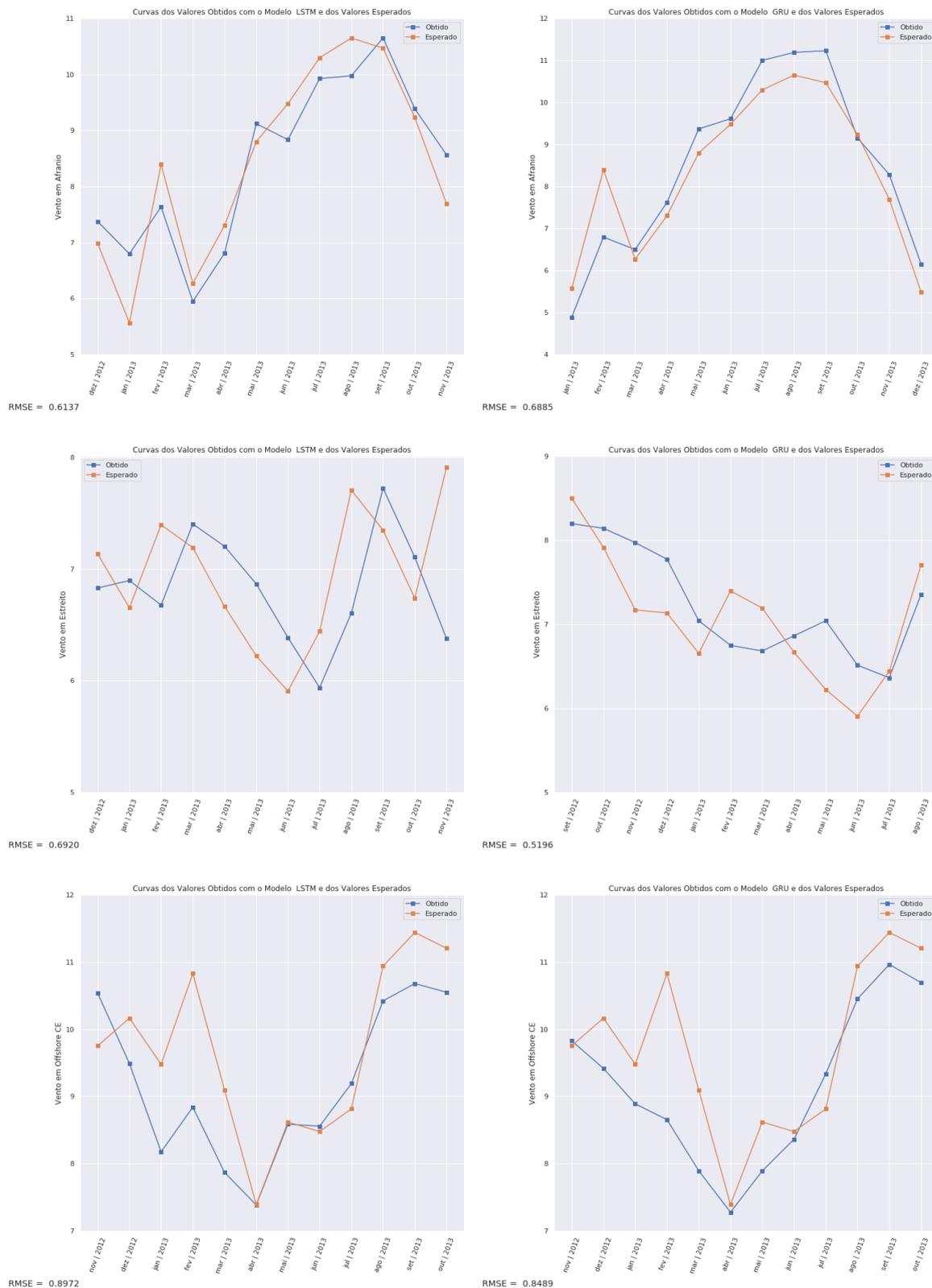


Figura C.7: Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de Afrânio, Estreito e Offshore CE, com horizonte de previsão de 12 meses.

FONTE: elaborado pela autora (2021).

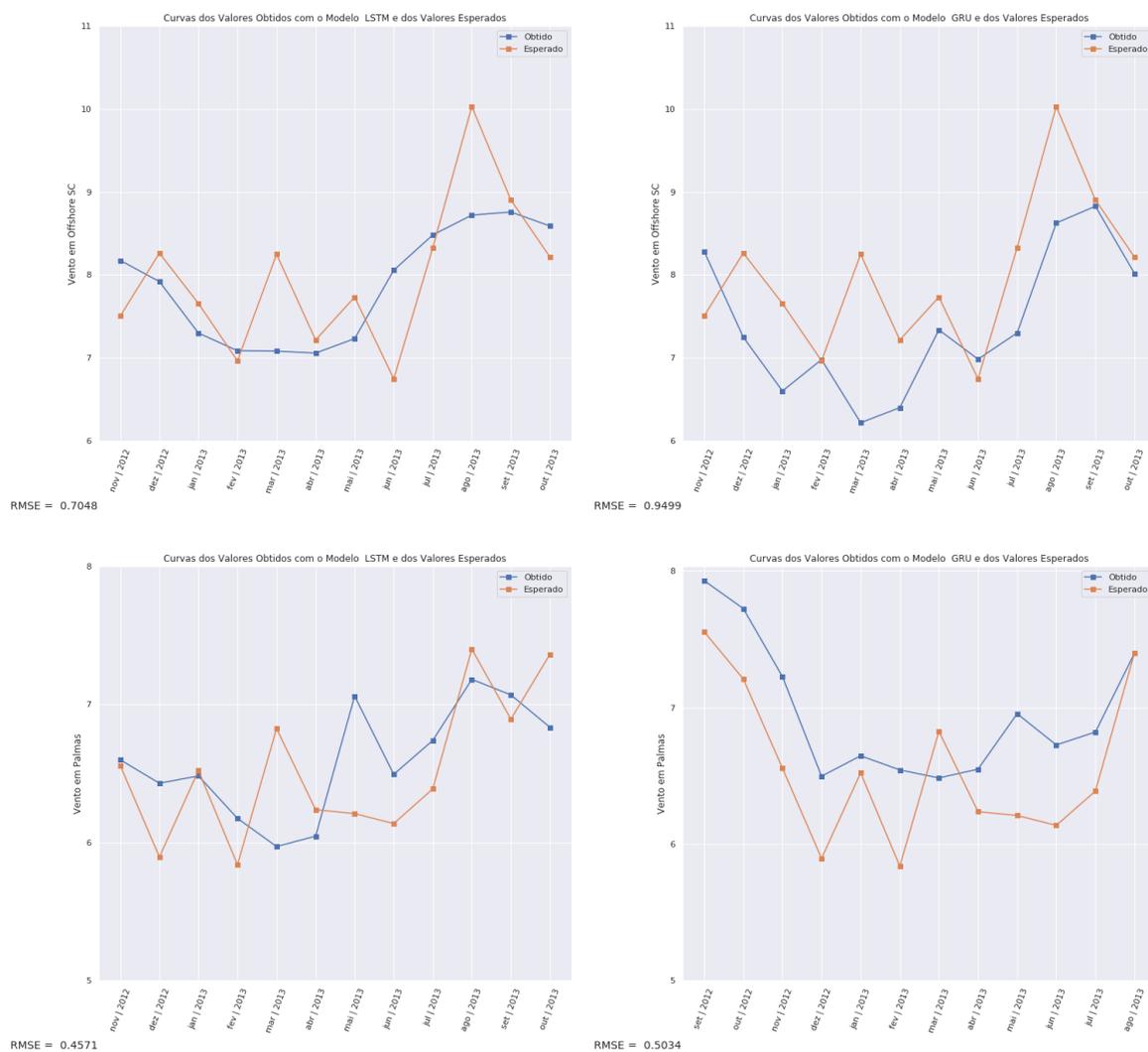


Figura C.8: Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de *Offshore SC* e *Palmas*, com horizonte de previsão de 12 meses.
 FONTE: elaborado pela autora (2021).

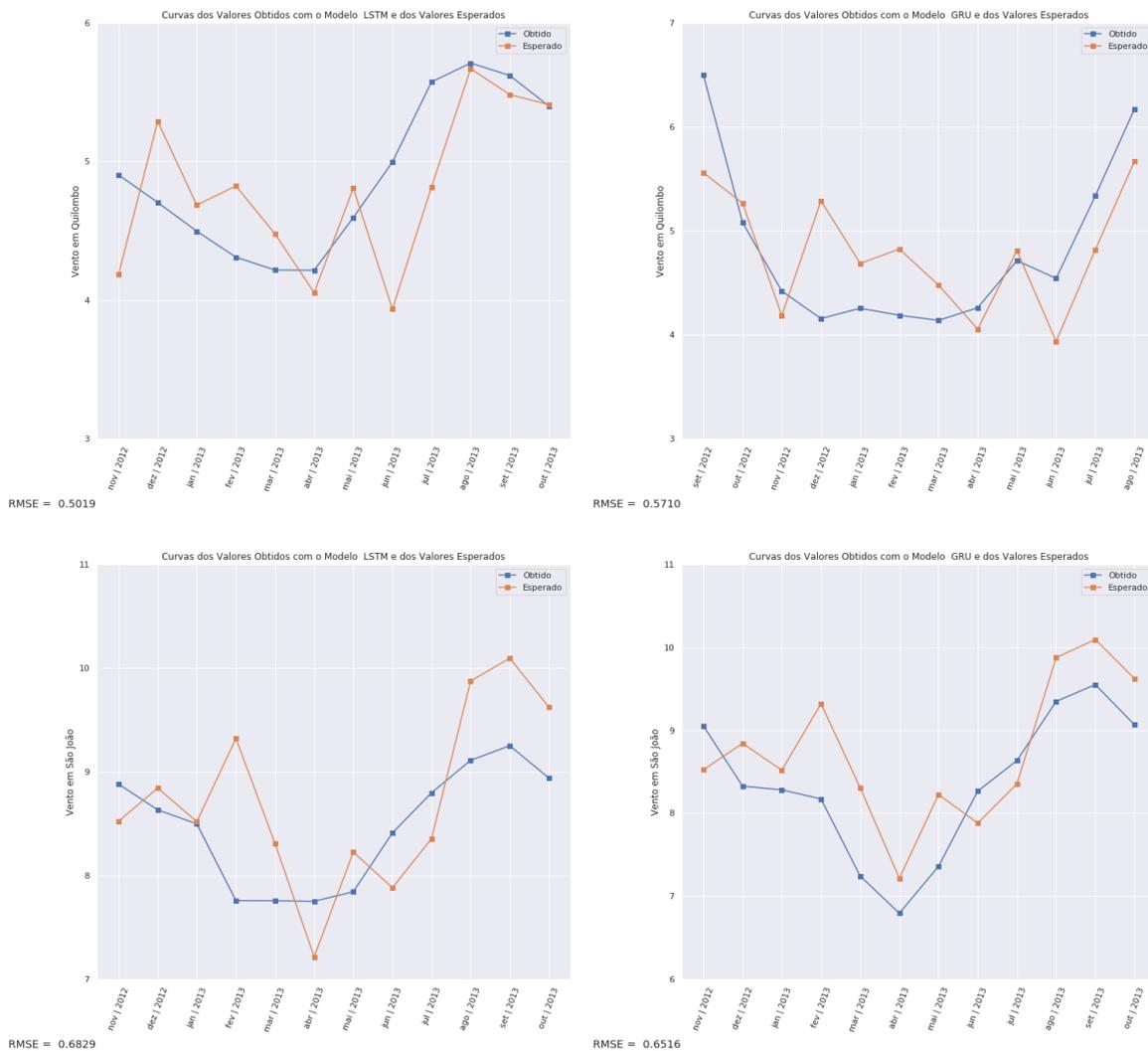


Figura C.9: Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de Quilombo e São João, com horizonte de previsão de 12 meses.

FONTE: elaborado pela autora (2021).

Tabela C.3: Melhores métricas e parâmetros de pré-processamento dos melhores modelos (LSTM e GRU) Multivariados, com horizonte de previsão de 12 meses.
FONTE: elaborado pela autora (2021).

Planilha	Modelo	Estac.	Escala	Look Back	Walk Forw.	Rede de Repr.	Função de Ativ.	Look Back Net	MSE	RMSE	MAPE
1	LSTM	0	1	3	1	0	0	0	0,3766	0,6137	0,0699
	GRU	0	1	0	1	0	0	0	0,4740	0,6885	0,0726
2	LSTM	1	1	3	1	0	0	0	0,2699	0,5196	0,0671
	GRU	0	1	9	0	0	0	0	0,2699	0,5196	0,0671
3	LSTM	0	1	6	0	0	0	0	0,8050	0,8972	0,0693
	GRU	0	1	6	0	0	0	0	0,7207	0,8489	0,0651
4	LSTM	0	1	6	0	0	0	0	0,4968	0,7048	0,0690
	GRU	0	1	6	0	0	0	0	0,9022	0,9499	0,0923
5	LSTM	0	1	6	0	0	0	0	0,2090	0,4571	0,0579
	GRU	0	1	9	0	0	0	0	0,2534	0,5034	0,0709
6	LSTM	0	1	6	0	0	0	0	0,2519	0,5019	0,0861
	GRU	0	1	9	0	0	0	0	0,3261	0,5710	0,0997
7	LSTM	0	1	6	0	0	0	0	0,4664	0,6829	0,0646
	GRU	0	1	6	0	0	0	0	0,4246	0,6516	0,0676

C.4 MODELOS MULTIVARIADOS COM HORIZONTE DE PREVISÃO DE 60 MESES

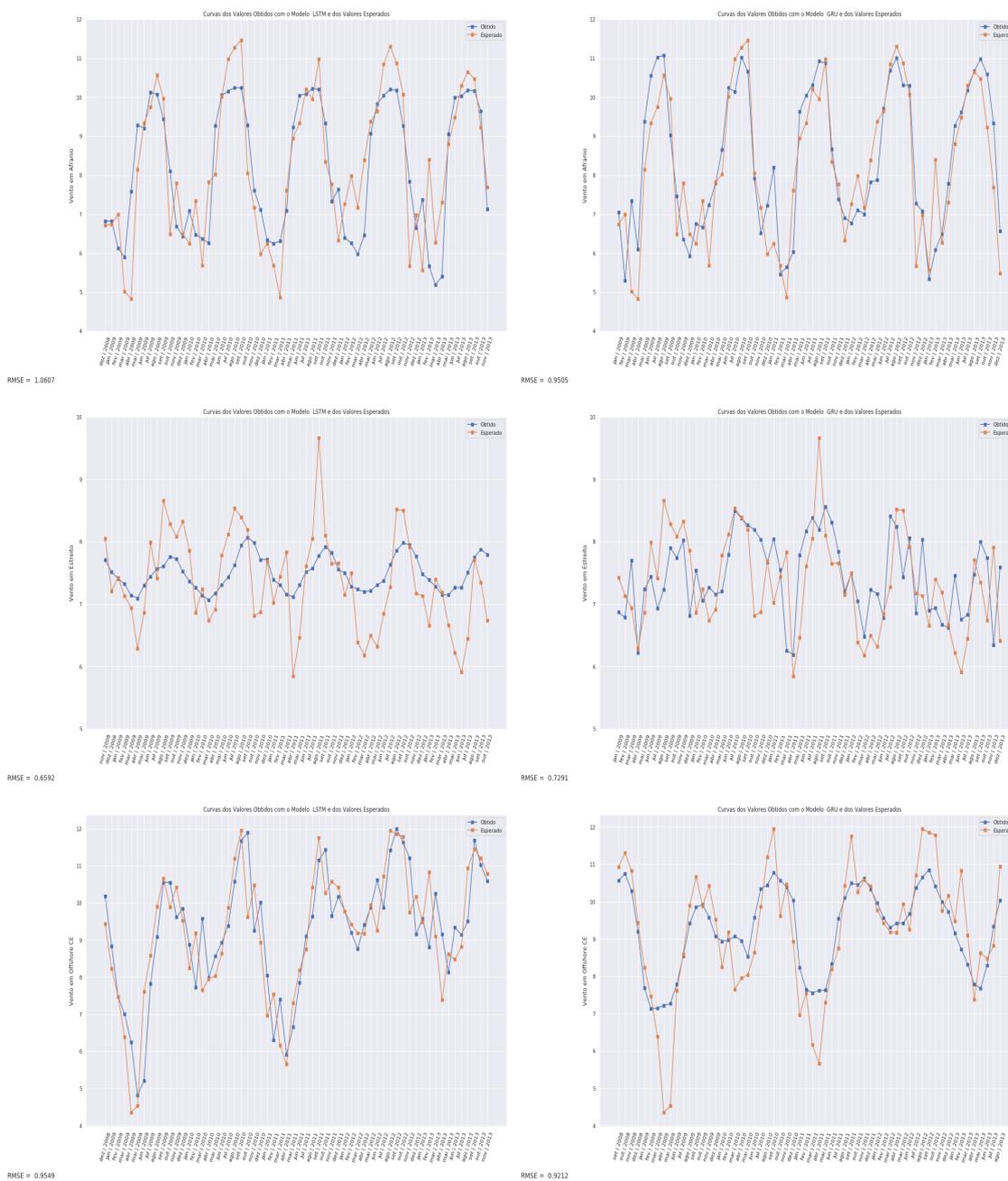


Figura C.10: Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de Afrânio, Estreito e *Offshore* CE, com horizonte de previsão de 60 meses.

FONTE: elaborado pela autora (2021).

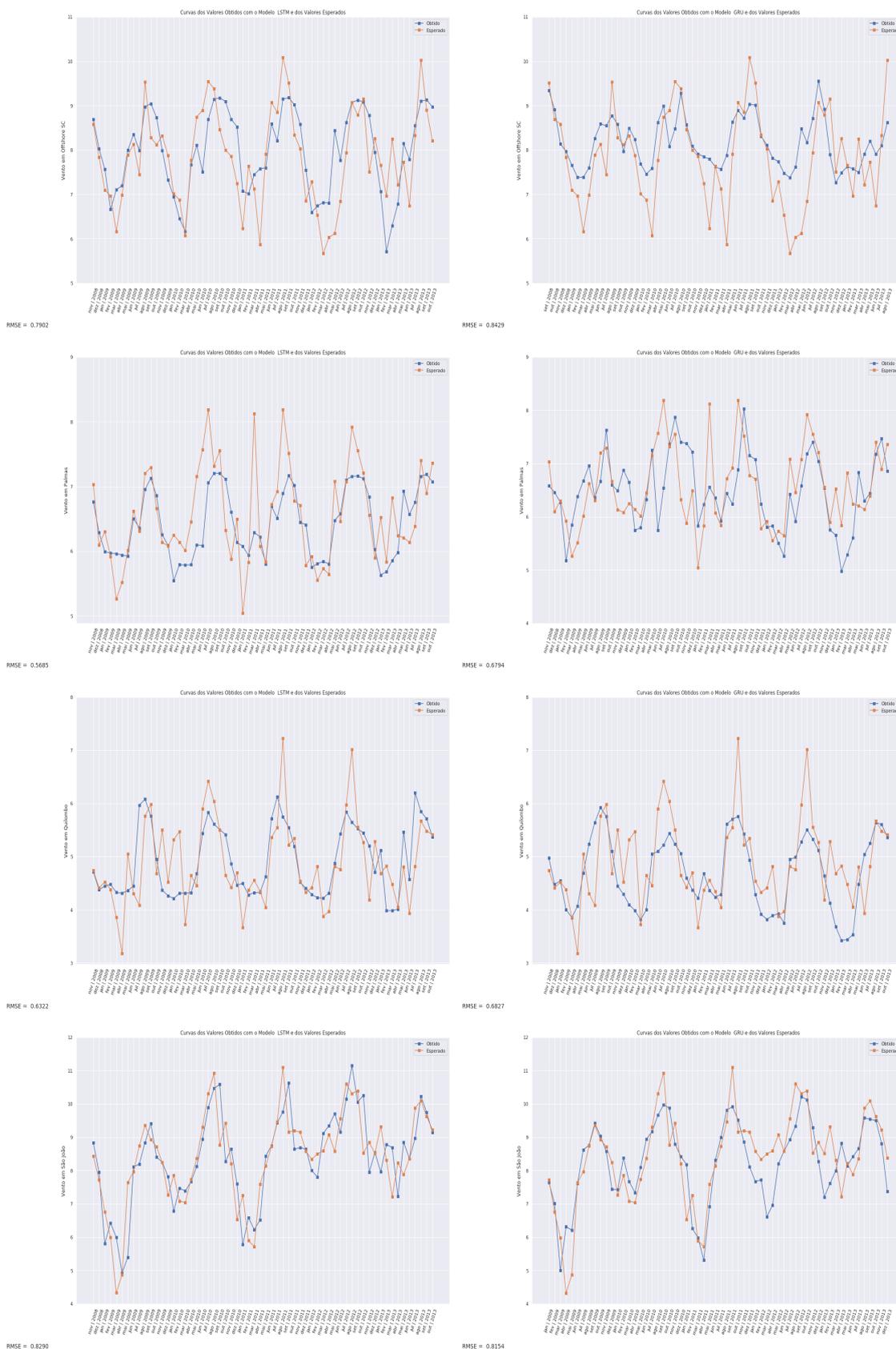


Figura C.11: Gráficos das curvas obtidas e esperadas, no modo Multivariado, dos modelos LSTM e GRU nos dados de *Offshore* SC, Palmas, Quilombo e São João, com horizonte de previsão de 60 meses.

FONTE: elaborado pela autora (2021).

Tabela C.4: Melhores métricas e parâmetros de pré-processamento dos melhores modelos (LSTM e GRU) Multivariados, com horizonte de previsão de 60 meses.
FONTE: elaborado pela autora (2021).

Planilha	Modelo	Estac.	Escala	Look Back	Walk Forw.	Rede de Repr.	Função de Ativ.	Look Back Net	MSE	RMSE	MAPE
1	LSTM	0	1	3	0	0	0	0	1,1250	1,0607	0,1165
	GRU	0	1	0	1	0	0	0	0,9035	0,9505	0,1026
2	LSTM	0	1	6	0	0	0	0	0,4346	0,6592	0,0744
	GRU	0	1	0	1	0	0	0	0,5316	0,7291	0,0801
3	LSTM	1	1	3	1	0	0	0	0,9119	0,9549	0,0882
	GRU	0	1	9	0	0	0	0	0,8486	0,9212	0,0878
4	LSTM	0	1	6	0	0	0	0	0,6245	0,7902	0,0856
	GRU	0	1	9	0	0	0	0	0,7104	0,8429	0,0906
5	LSTM	0	1	6	0	0	0	0	0,3232	0,5685	0,0622
	GRU	0	1	6	0	0	0	0	0,4615	0,6794	0,0798
6	LSTM	0	1	6	0	0	0	0	0,3997	0,6322	0,0983
	GRU	0	1	6	0	0	0	0	0,4661	0,6827	0,1068
7	LSTM	1	1	3	1	0	0	0	0,6873	0,8290	0,0809
	GRU	0	1	0	1	0	0	0	0,6648	0,8154	0,0812

APÊNDICE D - GRÁFICOS DE DISPERSÃO DE TODAS AS BASES DE DADOS

D.1 AFRÂNIO

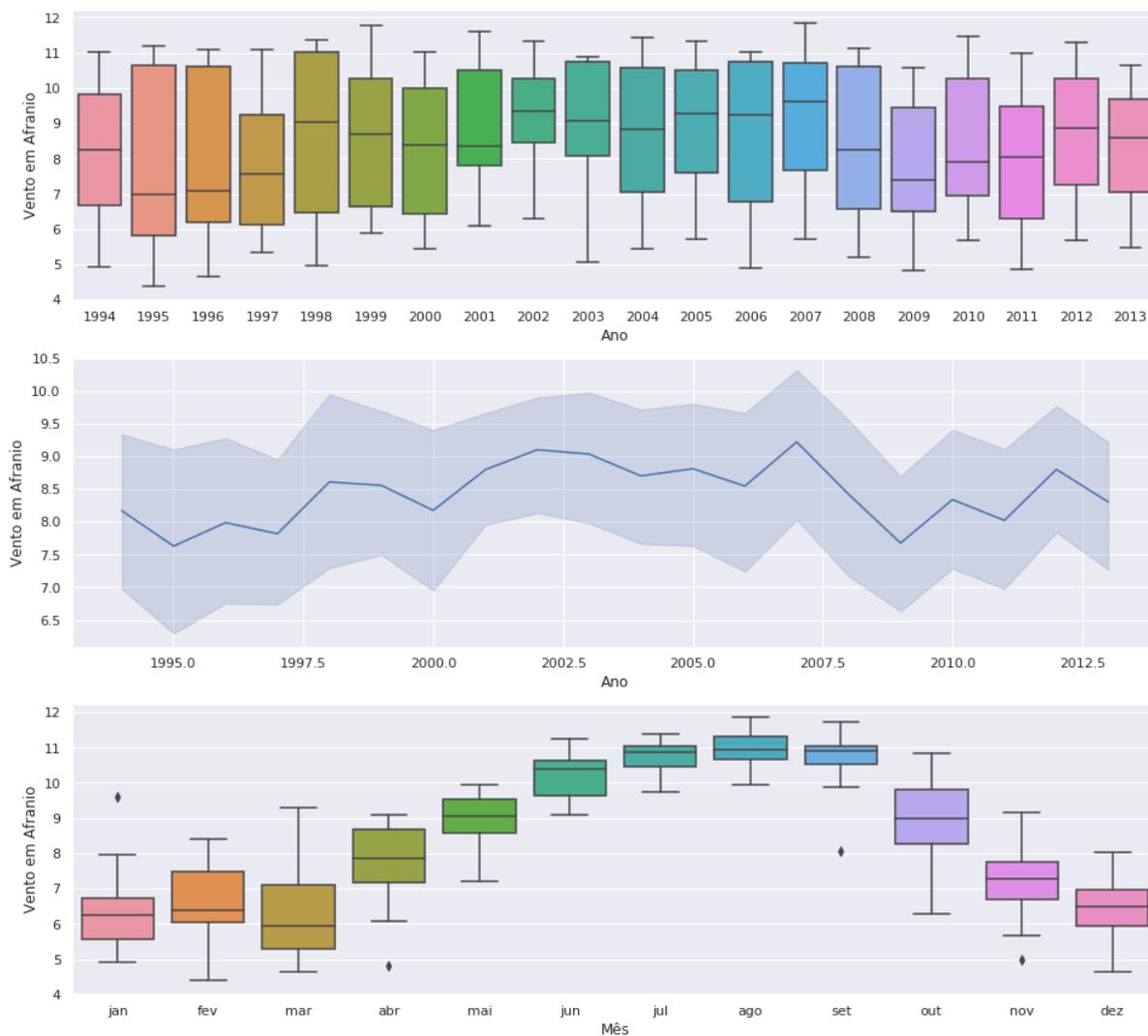


Figura D.1: *Boxplots* anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e *boxplot* mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Afrânio.

FONTE: elaborado pela autora (2021).

D.2 ESTREITO

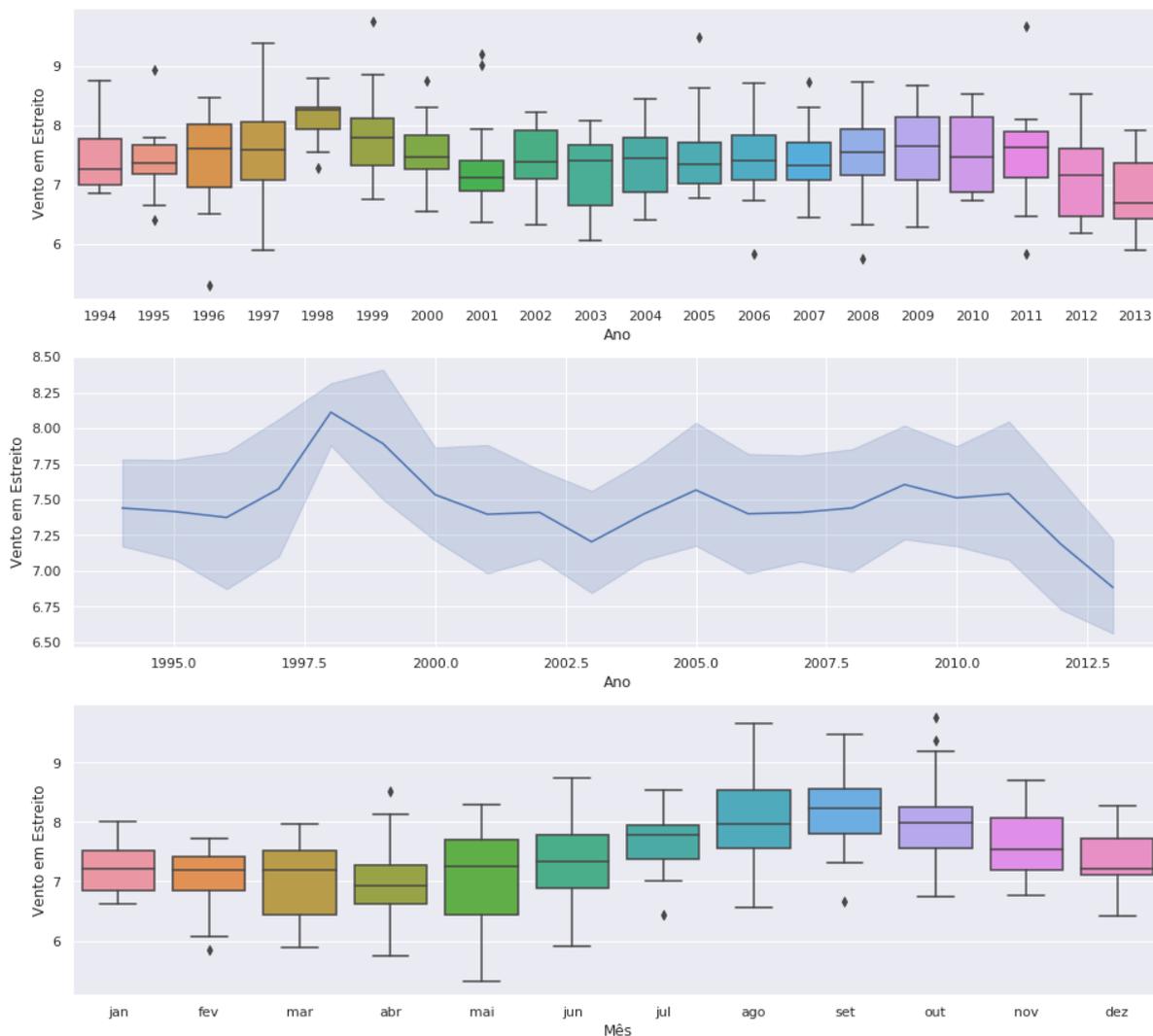


Figura D.2: *Boxplots* anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e *boxplot* mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Estreito.

FONTE: elaborado pela autora (2021).

D.3 OFFSHORE CE



Figura D.3: *Boxplots* anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e *boxplot* mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de *Offshore CE*.

FONTE: elaborado pela autora (2021).

D.4 OFFSHORE SC

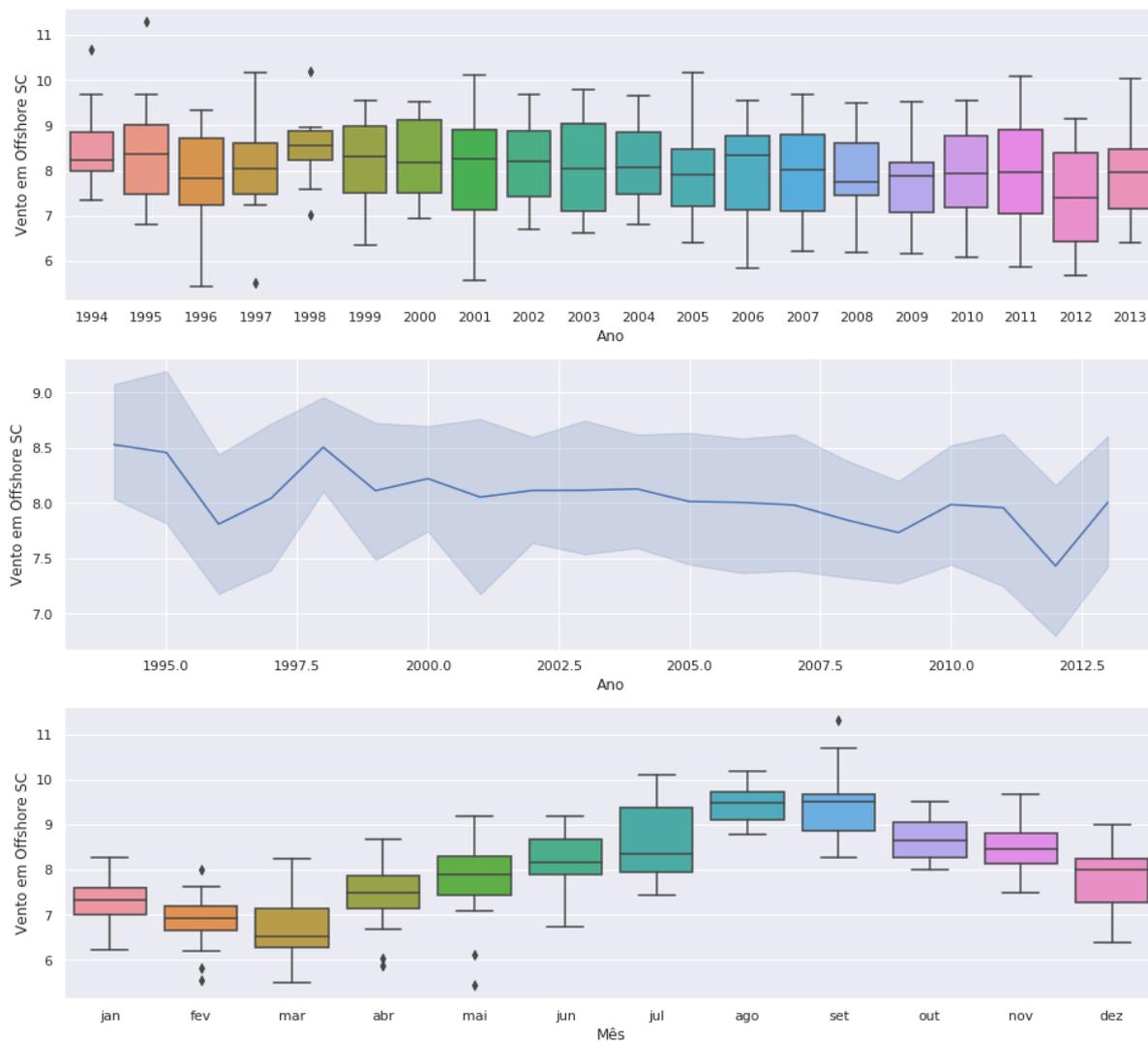


Figura D.4: *Boxplots* anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e *boxplot* mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de *Offshore SC*.

FONTE: elaborado pela autora (2021).

D.5 PALMAS

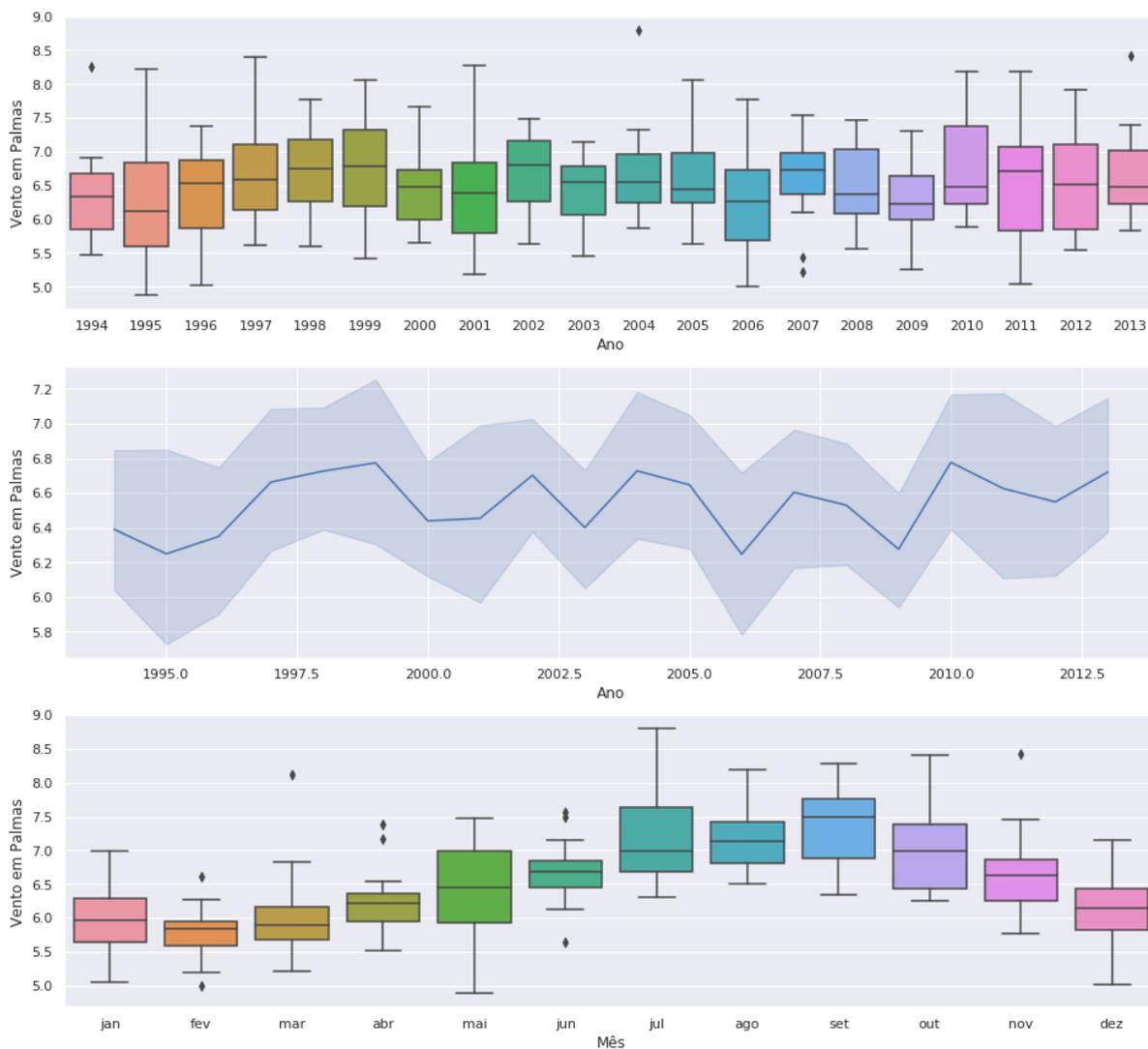


Figura D.5: *Boxplots* anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e *boxplot* mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Palmas.

FONTE: elaborado pela autora (2021).

D.6 QUILOMBO

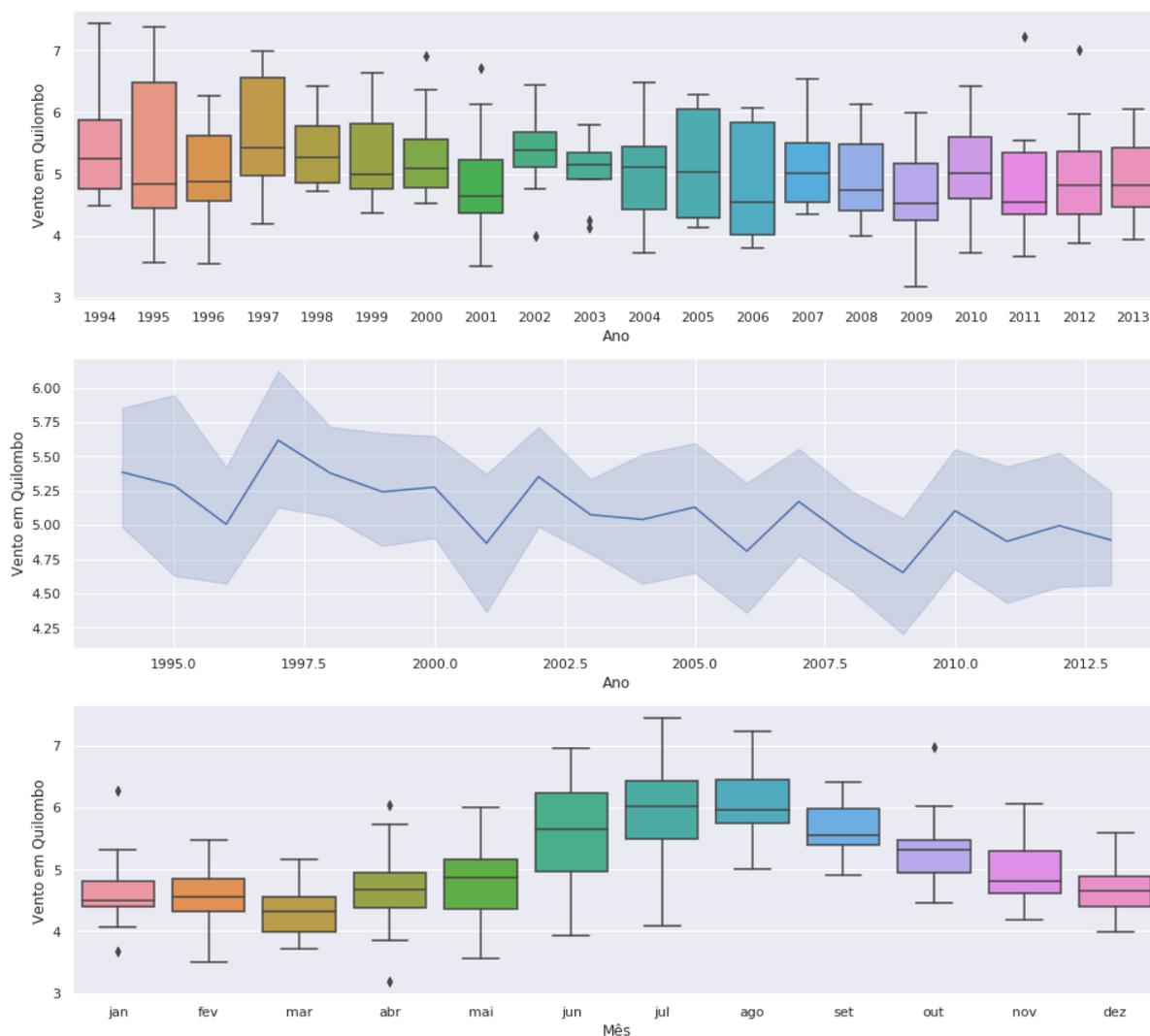


Figura D.6: *Boxplots* anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e *boxplot* mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de Quilombo.

FONTE: elaborado pela autora (2021).

D.7 SÃO JOÃO



Figura D.7: *Boxplots* anuais dos dados de velocidade de vento, gráfico da dispersão da série ao longo dos anos e *boxplot* mensal, considerando todos os anos, dos valores de velocidade de vento do conjunto de dados de São João.

FONTE: elaborado pela autora (2021).