

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

João Henrique Ferreira Flores

**COMPARAÇÃO DE MODELOS MLP/RNA E MODELOS
BOX-JENKINS EM SÉRIES TEMPORAIS NÃO LINEARES**

Porto Alegre
2009

João Henrique Ferreira Flores

**COMPARAÇÃO DE MODELOS MLP/RNA E MODELOS
BOX-JENKINS EM SÉRIES TEMPORAIS NÃO LINEARES**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul como requisito parcial à obtenção do título de Mestre em Engenharia de Produção, modalidade Acadêmica, na área de concentração em Sistemas de Qualidade.

Orientadora: Liane Werner, Dra.

Porto Alegre

2009

João Henrique Ferreira Flores

Comparação de modelos MLP/RNA e modelos Box-Jenkins em séries temporais não lineares

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia de Produção na modalidade Acadêmica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul.

Prof. Liane Werner, Dra.
Orientadora PPGEP/UFRGS

Prof. Flávio Sanson Fogliatto, Ph.D.
Coordenador PPGEP/UFRGS

Banca Examinadora:

Professor José Luis Duarte Ribeiro, Dr.

Professor Paulo Martins Engel, Dr.

Professora Bárbara Patricia Olbermann Pasini, Dra.

Dedicatória

Ao minha orientadora, Prof. Liane Werner, pelo auxílio na confecção deste trabalho, a minha família pelo apoio incondicional e a minha noiva pela paciência.

AGRADECIMENTOS

Quero agradecer primeiramente a minha família. Meu pai João Alcides Flores, minha mãe, Dulcy Ferreira Flores, minha irmã, Manuela Ferreira Flores e meu sobrinho, César Milanez Neto. Seu apoio foi fundamental para a continuidade deste trabalho. Agradeço também a minha noiva, Patrice Zeidler Basso, pela paciência e dedicação.

Desejo agradecer também, em especial, a minha orientadora, Professora Liane Werner, por me apresentar o caminho e ter acreditado em mim durante todo o mestrado. Agradeço também aos integrantes do Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul, seus professores, colaboradores e colegas. Mesmo que indiretamente, muitos foram os que contribuíram para este trabalho.

Por fim, quero agradecer a banca examinadora, seus comentários, críticas e sugestões, que só engrandecem este trabalho. Agradecer também ao restante de meus familiares, por me proporcionar momentos de alegria durante este tempo.

RESUMO

A capacidade de prever resultados futuros, ao se analisar uma série de dados, é uma importante ferramenta para o planejamento de qualquer empresa ou indústria. Porém, a literatura oferece muitas opções de ferramentas e modelos estatísticos que permitem obter estas previsões. Cada qual com suas características e recomendações. Dentre estes modelos, destacam-se os modelos de Box e Jenkins, e os modelos de Redes Neurais Artificiais (RNA) - com destaque aos modelos de *perceptron* de múltiplas camadas (MLP). Estas duas diferentes abordagens são comparadas nesta dissertação com relação a sua capacidade de obter previsões acuradas em séries de dados não lineares quanto a sua média. As abordagens foram comparadas utilizando-se a série mensal do índice de produção física industrial do Estado do Rio Grande do Sul. Bem como a série anual de manchas solares, sendo a segunda utilizada como caso-controle para as comparações, devido ao fato de que as suas propriedades já foram amplamente estudadas. No estudo da série do índice de produção física mensal, os modelos de Box e Jenkins obtiveram melhor rendimento. Na série das manchas solares foram os modelos MLP que se destacaram. Desta forma, não é possível afirmar se alguma das abordagens é superior - tratando-se de séries de dados não lineares quanto a sua média.

Palavras-chave: RNA. MLP. Box-Jenkins. ARIMA. Séries temporais.

ABSTRACT

The capacity to preview future outcomes on the time series analysis is an important tool for any business and industry planning. However, the literature offers many options on statistical tools and models which allow to obtain these forecasts. Each one with their features and recommendations. In these models, the Box and Jenkins and Artificial Neural Networks (ANN) models, with the multilayer perceptron (MLP) highlighted, stand out. These two different approaches are compared in this thesis related to the capacity to obtain accurate forecasts in mean related non-linear time series analysis. These approaches were compared using the monthly physical production index of Rio Grande do Sul time series and the sunspot series, being the second one used as a case-control to the comparisons, due the fact of its properties are already widely studied. In the monthly physical production index series study, the Box and Jenkins models obtained better efficiency. In the sunspot series, the MLP models were highlighted. So, it isn't possible to affirm if any of the approaches is superior, in the case of mean related non-linear time series.

Keywords: ANN. MLP. Box-Jenkins. ARIMA. Time series.

LISTA DE FIGURAS

Figura 1	Representação esquemática do algoritmo Box-Jenkins	32
Figura 2	Exemplo de fac de um modelo AR(2)	35
Figura 3	Exemplo de facp de um modelo AR(2)	35
Figura 4	Exemplo de fac de um modelo MA(2)	36
Figura 5	Exemplo de facp de um modelo MA(2)	36
Figura 6	Exemplo de fac de um modelo ARIMA(1,1,1)	38
Figura 7	Exemplo de facp de um modelo ARIMA(1,1,1)	38
Figura 8	Exemplo de fac de um modelo ARMA(2,2)	39
Figura 9	Exemplo de facp de um modelo ARMA(2,2)	39
Figura 10	Modelo não linear de um neurônio	45
Figura 11	Arquitetura de rede alimentada adiante de camada única	47
Figura 12	Arquitetura de rede alimentada diretamente com múltiplas camadas	48
Figura 13	Arquitetura de rede recorrente	49
Figura 14	Arquitetura de rede recorrente com camada oculta e laços de auto-realimentação	49
Figura 15	Arquitetura de um modelo MLP 3-4-2-1	56
Figura 16	Fluxo de sinais nos neurônios de um modelo MLP	57
Figura 17	Fluxograma da metodologia utilizada	60
Figura 18	Fluxograma da metodologia utilizada na modelagem MLP/RNA	65
Figura 19	Série do índice de produção física industrial mensal do RS - base média/2002 = 100	74
Figura 20	Série do logaritmo do índice de produção física industrial mensal do RS - base média/2002 = 100	75
Figura 21	Série do número de manchas solares - periodicidade anual	76

Figura 22 fac da série do índice de produção física - <i>lag</i> em anos (1 = 12 meses)	79
Figura 23 facp da série do índice de produção física - <i>lag</i> em anos (1 = 12 meses)	79
Figura 24 fac dos resíduos do modelo SARIMA(5, 1, 2)x(2, 1, 1) ₁₂ - <i>lag</i> em anos (1 = 12 meses)	80
Figura 25 facp dos resíduos do modelo SARIMA(5, 1, 2)x(2, 1, 1) ₁₂ - <i>lag</i> em anos (1 = 12 meses)	80
Figura 26 Gráfico da série do índice de produção e do modelo SARIMA(5, 1, 2)x(2, 1, 1) ₁₂ ajustado	81
Figura 27 Gráfico da série do índice de produção e do modelo MLP(12,5,1) ajustado . . .	83
Figura 28 Gráfico da série do índice de produção e do modelo SARIMA(5, 1, 2)x(2, 1, 1) ₁₂ ajustado, com previsões	84
Figura 29 fac da série do log-índice de produção - <i>lag</i> em anos (1 = 12 meses)	85
Figura 30 facp da série do log-índice de produção - <i>lag</i> em anos (1 = 12 meses)	85
Figura 31 fac dos resíduos do modelo SARIMA(6, 1, 2)x(1, 1, 0) ₁₂ - <i>lag</i> em anos (1 = 12 meses)	87
Figura 32 facp dos resíduos do modelo SARIMA(6, 1, 2)x(1, 1, 0) ₁₂ - <i>lag</i> em anos (1 = 12 meses)	87
Figura 33 Gráfico da série do log-índice de produção e do modelo SARIMA(6, 1, 2)x(1, 1, 0) ₁₂ ajustado	88
Figura 34 Gráfico da série do log-índice de produção e do modelo MLP(6,14,1) ajustado	89
Figura 35 Gráfico da série do log-índice de produção e do modelo SARIMA(6, 1, 2)x(1, 1, 0) ₁₂ ajustado, com previsões	91
Figura 36 fac da série das manchas solares - <i>lag</i> em anos	92
Figura 37 facp da série das manchas solares - <i>lag</i> em anos	92
Figura 38 fac dos resíduos do modelo ARMA(3,5) - <i>lag</i> em anos (1 = 12 meses)	93
Figura 39 facp dos resíduos do modelo ARMA(3,5) - <i>lag</i> em anos (1 = 12 meses)	94
Figura 40 Gráfico da série das manchas solares e do modelo ARMA(3,5) ajustado	94
Figura 41 Gráfico da série das manchas solares e do modelo MLP(6,25,1) ajustado	96

Figura 42 Gráfico da série das manchas solares e do modelo MLP(6,25,1) ajustado, com previsões	98
--	----

LISTA DE TABELAS

Tabela 1	Resultados da aplicação do teste de Tukey nas séries de dados para diferentes ordens	77
Tabela 2	Resultados da aplicação dos testes de Teräsvirta e White nas séries de dados para diferentes <i>lags</i>	77
Tabela 3	Resultados dos critérios de modelagem nos modelos Box-Jenkins na série do índice de produção	79
Tabela 4	Resultados dos critérios nos modelos MLP/RNA com 6 neurônios na camada de entrada	82
Tabela 5	Resultados dos critérios nos modelos MLP/RNA com 12 neurônios na camada de entrada	82
Tabela 6	Resultados dos critérios para comparação de modelos clássicos e modelos MLP/RNA	83
Tabela 7	Resultados dos critérios de modelagem nos modelos Box-Jenkins na série do log-índice de produção	86
Tabela 8	Resultados dos critérios nos modelos MLP/RNA com 6 neurônios na camada de entrada	88
Tabela 9	Resultados dos critérios nos modelos MLP/RNA com 12 neurônios na camada de entrada	88
Tabela 10	Resultados dos critérios para comparação de modelos clássicos e modelos MLP/RNA	90
Tabela 11	Resultados dos critérios nos modelos Box-Jenkins	93
Tabela 12	Resultados dos critérios nos modelos MLP/RNA com 6 neurônios na camada de entrada	95
Tabela 13	Resultados dos critérios nos modelos MLP/RNA com 12 neurônios na camada de entrada	95

Tabela 14 Resultados dos critérios para comparação de modelos clássicos e modelos MLP/RNA	97
--	----

LISTA DE SIGLAS

AIC - *Akaike information criteria*, critério de informação de Akaike;

AR - *autoregressive*, autorregressivo;

ARCH - *autoregressive conditional heteroskedastic*, autorregressivo com heteroscedasticidade;

ARCH-M - *autoregressive conditional heteroskedastic in mean*, autorregressivo com heteroscedasticidade na média;

ARIMA - *autoregressive integrated moving average*, autorregressivo integrado de médias móveis;

ARMA - *autoregressive moving average*, autorregressivo de médias móveis;

BIC - *bayesian information criteria*, critério de informação bayesiano;

EGARCH - *exponential generalized autoregressive conditional heteroskedastic*, autorregressivo com heteroscedasticidade generalizado exponencial;

EQM - erro quadrático médio;

ESTAR - *exponential smooth transition autoregressive*, autorregressivo de transição suave exponencial;

fac - função de autocorrelação;

facp - função de autocorrelação parcial;

GAR - *generalized autoregressive*, autorregressivo generalizado;

GARCH - *generalised autoregressive conditional heteroskedastic*, autorregressivo com heteroscedasticidade generalizado;

IGARCH - *integrated generalised autoregressive conditional heteroskedastic*, autorregressivo com heteroscedasticidade generalizado integrado;

MA - *moving average*, médias móveis;

MAPE - *mean absolute percentage error*, erro percentual absoluto médio;

MLP - *Multilayer perceptron*, perceptron de múltiplas camadas;

NLAR - *non-linear autoregressive*, autorregressivo não linear;

NMSE - *normalized mean square error*, erro quadrático médio normalizado;

RBF - *radial basis function*, função de base radial;

RNA - Rede Neural Artificial;

SARIMA - *seasonal autoregressive integrated moving average*, autorregressivo integrado de média móvel sazonal;

SSE - *square sum error*, soma do erro quadrado;

STAR - *smooth transition autoregressive*, autorregressivo de transição suave;

TAR - *treshold autoregressive*, autorregressivo de limiar;

VAR - *vector autoregressive*, autorregressivo vetorial;

SUMÁRIO

1	INTRODUÇÃO	19
1.1	TEMA E OBJETIVOS	21
1.2	JUSTIFICATIVAS	22
1.3	MÉTODOS	23
1.4	DELIMITAÇÕES DO TRABALHO	25
1.5	ESTRUTURA	25
2	ANÁLISE DE SÉRIES TEMPORAIS	27
2.1	SÉRIE TEMPORAL NÃO LINEAR	29
2.2	MODELOS CLÁSSICOS	30
2.2.1	Modelos Box-Jenkins	31
2.2.2	Outros modelos	40
2.2.2.1	Modelos Polinomiais e de Regressão	40
2.2.2.2	Modelos Não Lineares	41
2.2.2.3	Alisamento exponencial	42
2.2.2.4	Modelos vetoriais	42
2.3	REDES NEURAIAS ARTIFICIAIS	43
2.3.1	Definições	45
2.3.2	Arquiteturas de Redes	46
2.3.2.1	Redes alimentadas adiante com camada única	47
2.3.2.2	Redes alimentadas diretamente com múltiplas camadas	47
2.3.2.3	Redes recorrentes	48
2.3.3	Aprendizagem	49

2.3.3.1	Tarefas de aprendizagem	50
2.3.3.2	Métodos de aprendizagem	51
2.3.3.3	Aplicação dos métodos	53
2.3.4	Treinamento	54
2.3.5	<i>Perceptrons</i> de múltiplas camadas	56
3	METODOLOGIA	59
3.1	ESTRUTURA DA METODOLOGIA	59
3.1.1	Séries de dados temporais	59
3.1.2	Testes de linearidade da série de dados	61
3.1.3	Verificação de linearidade da série de dados	62
3.1.4	Modelagem ARIMA/SARIMA	62
3.1.5	Comparação e seleção de modelos ARIMA/SARIMA	63
3.1.6	Ajuste do modelo ARIMA/SARIMA considerado melhor	63
3.1.7	Modelagem MLP/RNA	64
3.1.8	Comparação e seleção de modelos MLP/RNA	67
3.1.9	Ajuste do modelo MLP/RNA considerado melhor	68
3.1.10	Comparação de modelos MLP/RNA e modelos ARIMA/SARIMA	68
3.1.11	Melhor modelo	69
3.2	CRITÉRIOS DE COMPARAÇÃO UTILIZADOS	69
4	RESULTADOS	73
4.1	DESCRIÇÃO DAS SÉRIES	73
4.1.1	Série do índice de produção física industrial mensal do RS	73
4.1.2	Série do número de manchas solares	75
4.2	TESTES DE VERIFICAÇÃO DA LINEARIDADE DAS SÉRIES	76
4.3	APLICAÇÃO DA METODOLOGIA À SÉRIE DO ÍNDICE DE PRODUÇÃO	78
4.3.1	Modelagem ARIMA/SARIMA da série do índice de produção	78

4.3.2	Comparação dos modelos ARIMA/SARIMA na série do índice de produção . . .	80
4.3.3	Ajuste do modelo ARIMA/SARIMA considerado melhor	80
4.3.4	Modelagem MLP/RNA na série do índice de produção	81
4.3.5	Comparação dos modelos MLP/RNA na série do índice de produção	82
4.3.6	Ajuste do modelo MLP/RNA considerado melhor	82
4.3.7	Comparação dos modelos MLP/RNA e ARIMA/SARIMA na série do índice de produção	83
4.4	APLICAÇÃO DA METODOLOGIA À SÉRIE DO LOG-ÍNDICE DE PRODUÇÃO . . .	85
4.4.1	Modelagem ARIMA/SARIMA da série do log-índice	85
4.4.2	Comparação dos modelos ARIMA/SARIMA na série do log-índice	86
4.4.3	Ajuste do modelo ARIMA/SARIMA considerado melhor	86
4.4.4	Modelagem MLP/RNA na série do log-índice	87
4.4.5	Comparação dos modelos MLP/RNA na série do log-índice	87
4.4.6	Ajuste do modelo MLP/RNA considerado melhor	89
4.4.7	Comparação dos modelos MLP/RNA e ARIMA/SARIMA na série do log-índice	89
4.5	APLICAÇÃO DA METODOLOGIA A SÉRIE DAS MANCHAS SOLARES	91
4.5.1	Modelagem ARIMA/SARIMA na série das manchas solares	91
4.5.2	Comparação dos modelos ARIMA/SARIMA na série de manchas solares	92
4.5.3	Ajuste do modelo ARIMA/SARIMA considerado melhor	93
4.5.4	Modelagem MLP/RNA na série das manchas solares	95
4.5.5	Comparação dos modelos MLP/RNA na série de manchas solares	95
4.5.6	Ajuste do modelo MLP/RNA considerado melhor	96
4.5.7	Comparação dos modelos MLP/RNA e ARIMA/SARIMA na série de manchas solares	96
4.6	DISCUSSÃO	97
5	CONSIDERAÇÕES FINAIS	103

5.1	SUGESTÕES PARA TRABALHOS FUTUROS	104
	REFERÊNCIAS	105
	Apêndice A – Código do programa R utilizado nas modelagens	109
	Apêndice B – Critérios dos modelos MLP/RNA	124
	Anexo A – Tabelas das séries	130

1 INTRODUÇÃO

O processo de previsão de resultados futuros é considerado uma importante ferramenta para o planejamento de qualquer organização. Desde a avaliação da qualidade de um produto até previsões de suprimentos e demanda, são várias as áreas que se utilizam de previsões. Entre elas, pode-se citar o planejamento financeiro com previsão de cenários ou mesmo previsões de custos e receitas, e prevendo possíveis falhas do produto, auxiliando o controle de qualidade. Muitas são as técnicas que permitem avaliar e prever estes resultados. Esta avaliação e previsão pode ser realizada por uma área mais ampla chamada de Análise de Séries Temporais. Dentro da análise de séries temporais, existem diversos modelos, com as mais variadas suposições quanto ao comportamento de uma série de dados. Contudo, o fato de existirem diversos modelos pode, muitas vezes, dificultar a análise. Isso se deve ao fato de que muitos modelos foram, e têm sido, desenvolvidos ao longo dos últimos anos. Assim sendo, como saber qual o melhor modelo para cada situação. Efetivamente, muitos dos modelos podem ser adaptados, com menor ou maior grau de sucesso dependendo de cada situação específica.

Um problema, em especial, tem se tornado muito comum, dado o aumento da capacidade dos bancos de dados e a utilização mais ampla de modelagem por séries temporais, é o caso de séries temporais não lineares (PRIESTLEY, 1991). Isso porque, com poucos dados, tem-se uma baixa frequência de observações e a série de dados fica suavizada, ocultando algumas características importantes, como sua possível não linearidade. Com o aumento da capacidade computacional e de armazenamento de dados, as séries puderam ter um número maior de observações registradas, tornando visível estas características até então não observadas.

Contudo, uma série não passa a ser não linear apenas devido ao aumento do número de observações. Uma série de dados é ou não linear quando estas características se apresentam nos dados, independente da frequência e quantidade de observações. A não linearidade de uma série de dados pode ser caracterizada por duas situações. Uma série de dados é dita não linear quando o suposto sistema que gerou aqueles dados possui um comportamento não linear. Outra situação é quando a média é não constante ou a variância é não constante (MORETTIN; TOLOI, 2006).

Séries não lineares quanto a variância também são ditas séries voláteis. Este tipo de não linearidade é mais comum em séries financeiras (ENDERS, 1995; ENGLE, 1982). Muitos dos modelos desenvolvidos para este tipo de não linearidade são aplicados na área financeira de uma maneira geral (ENGLE, 1982; ENDERS, 1995; MORETTIN; TOLOI, 2006; TABAK; LIMA, 2009; GIANNELLIS; PAPADOPOULOS, 2009)

Outra forma de não linearidade é quanto a média. Ao contrário da não linearidade com relação a variância, a média pode aparecer de, pelo menos, duas formas distintas. A primeira, e mais simples de ser resolvida é uma série de dados dita não estacionária. Neste caso a média não é constante, mas varia de forma constante, sempre crescendo ou decrescendo a mesma taxa (ENDERS, 1995; MORETTIN; TOLOI, 2006). A segunda maneira, menos trivial, é quando a média varia, porém não a taxas constantes. Este tipo de não linearidade é mais trabalhosa, do ponto de vista aplicado e são vários os tipos de modelos e técnicas para cada caso específico de não linearidade quanto a média. De qualquer forma, independente do tipo de não linearidade encontrada, seja em relação a média ou a variância, a análise e previsão dos dados continua sendo necessária. Sendo assim, é preciso encontrar formas de se trabalhar com esta falha de atender a suposição de linearidade, necessária para se utilizar os modelos Box-Jenkins.

Os modelos conhecidos como modelos Box-Jenkins têm este nome devido ao algoritmo desenvolvido pelos autores (BOX; JENKINS, 1976). O algoritmo desenvolvido auxilia na busca de modelos adequados com o mínimo de parâmetros possíveis. Os modelos Box-Jenkins, ou modelos ARIMA/SARIMA, são considerados modelos clássicos, por se basearem em suposições para sua correta formulação. Estes modelos são os mais conhecidos modelos para análise de séries temporais no domínio do tempo. Uma de suas suposições é a respeito da linearidade da série de dados, tanto em relação a média quanto em relação a variância. Apesar da construção de suas estimativas em relação aos parâmetros serem razoavelmente robustas, a não linearidade da média continua sendo uma falha de suposição. Porém, uma das possibilidades de se trabalhar com esta falha de suposição é com o uso de modelos de Redes Neurais Artificiais.

A utilização de Redes Neurais Artificiais (RNA, ou *ANN* do original em inglês *Artificial Neural Networks*) para análise de séries temporais, por sua vez, se desenvolveu muito tempo depois dos modelos clássicos (SMITH et al., 2002; WANG; LU, 2006a, 2006b; DELLANA; WEST, 2009). O início das RNA foi em 1943 com McCulloch e Pitts, que sugeriram inicialmente um modelo bem simples se comparado a modelos usados atualmente. As RNA encontraram um vasto campo para desenvolvimento inicialmente junto a Física e seus modelos não-lineares.

A partir da década de 1980, seus usos foram difundidos em outras ciências, assim como a Computação, Matemática e Estatística. Foi então que estes modelos passaram a ser utilizados em maior escala. Apoiados pelo desenvolvimento de algoritmos e pela evolução em termos computacionais, a difusão destas técnicas fica facilitada. Segundo Ripley (1996), os modelos de RNA, assim como outros modelos, fazem parte da Estatística Robusta. Essa classe de modelos, nos quais se inserem as RNA, são modelos que possuem poucas ou nenhuma suposição quanto a natureza dos dados. Apesar de possuir modelos probabilísticos, nas RNAs as suposições não

são tão restritivas quanto na modelagem estatística clássica. Entretanto, mesmo não havendo estas restrições quanto a necessidade de verificação de suposições, estes modelos não podem ser utilizados indiscriminadamente, pois tais modelos, não raro, necessitam de um número elevado de dados e sua construção ainda é baseada em fatores subjetivos. Estas razões, aliada a dificuldade de programação ou obtenção de uma função pré-definida em programas comerciais, faz com que o uso de modelos de RNA para análise de séries temporais ainda sejam, de certa forma, restritos. Além disso, não é, de forma alguma, recomendado o uso indiscriminado de RNA em séries não lineares, dado que muitos modelos estatísticos clássicos podem ser robustos o suficiente. Sendo assim, é necessário comparar estas duas abordagens de métodos de modelagem e verificar o uso prático das RNA, se é ou não compensador utilizar estes modelos. Desta forma, adiciona-se uma ferramenta de análise mais flexível e mais robusta.

Pelo exposto anteriormente, esta dissertação tratará de apresentar resultados a respeito do comportamento de modelos clássicos (modelos Box-Jenkins ou modelos ARIMA/SARIMA) e de modelos RNA frente a séries de dados não lineares quanto a sua média. Contudo, por se tratar de um assunto já bastante analisado pela literatura com relação às suas aplicações, os modelos clássicos terão um tratamento menos aprofundado, ocupando assim, um menor espaço neste trabalho.

1.1 Tema e Objetivos

O tema desta dissertação é a análise de séries temporais, utilizando-se de modelos Box e Jenkins (BOX; JENKINS, 1976; BOX et al., 1994) e de modelos de RNA (HAYKIN, 2001; RIPLEY, 1996). Também é tema desta dissertação a modelagem de séries não lineares, em específico com relação a variação da média não constante.

Quanto aos objetivos, o objetivo principal deste trabalho é comparar o comportamento a estabilidade das previsões de 12 passos adiante dos modelos Box-Jenkins frente aos modelos MLP/RNA, com relação ao cumprimento da suposição de linearidade quanto a média, usando como base a série do índice de produção física industrial mensal para o Estado do Rio Grande do Sul, que se trata de uma série pouco trabalhada e de características desconhecidas.

Como objetivo secundário, têm-se a comparação de estabilidade das previsões entre os modelos Box-Jenkins e os modelos RNA utilizando-se uma série de dados reconhecidamente não linear quanto a média, a série de manchas solares com periodicidade anual. Esta série de dados não possui aplicação prática no meio industrial, mas objetiva-se comparar estes modelos com uma série de dados largamente utilizada e com propriedades já trabalhadas e apresentadas visando apresentar de forma mais clara as vantagens e desvantagens de cada modelo, permitindo

assim subsidiar as comparações estabelecidas com a série da área industrial.

Outro objetivo secundário é apresentar uma metodologia que auxilie na compreensão de como as comparações foram executadas. Desta forma, facilita-se o entendimento dos procedimentos utilizados e de que forma os resultados foram obtidos.

1.2 Justificativas

A análise de séries temporais é uma importante ferramenta nos processos produtivos em geral, conforme vista em trabalhos como Flores (2006) e Chen (2006), por exemplo. As previsões obtidas pelos modelos costuma ser o principal uso da análise de séries temporais, mas não o único. Apesar de ser o foco deste trabalho, muitas análises são construídas de forma a apresentar e detalhar a forma com que certos eventos ocorrem. Outros exemplos de usos podem ser vistos em Makridakis et al. (1998) e Morettin e Tolo (2006). Porém, para que esta análise seja produtiva e informativa, são necessários modelos adequados. Na análise de séries temporais no domínio do tempo e não lineares quanto a média, a quantidade de modelos adequados é menor. Dentre estes, foram escolhidos os modelos Box e Jenkins e modelos de RNA.

A escolha de se utilizar uma série não linear é por esta se tratar um de um problema muito comum em séries temporais. Segundo Enders (1995), muitas séries possuem comportamento não linear. De certa forma, essa não linearidade existente na série não chega a causar maiores problemas, mesmo utilizando-se métodos lineares. Porém isso depende muito do uso que é dado as previsões geradas. Em um contexto mais amplo, desconsiderar fatores não lineares pode acarretar em graves erros, como, por exemplo, estimativas tendenciosas e erro probabilístico falho (ENDERS, 1995). Outra razão recai sobre o fato de que já existem estudos (PINTO; MEDEIROS, 2005) que apresentam um melhor rendimento de modelos Box e Jenkins em comparação a modelos RNA utilizando-se de série lineares.

A opção pela não linearidade em relação a média, se deve ao fato que as comparações feitas entre os modelos Box e Jenkins e os modelos RNA tratam, em grande parte, de uma não linearidade de forma mais ampla, incluída a não linearidade da variância. Sabe-se que quando a série é não linear em relação a sua variância, modelos da família ARCH/GARCH são recomendados (MORETTIN; TOLOI, 2006; ENDERS, 1995) assim como modelos com mudança de regimes (HAMILTON, 1994). Contudo, em séries não lineares quanto a média, são diversos os modelos existentes (ENDERS, 1995; PRIESTLEY, 1991), entre eles o modelo autorregressivo de limiar (TAR). Conforme Enders (1995) e Priestley (1991), a escolha entre qual modelo não linear a ser utilizado costuma ser difícil e, muitas vezes, é melhor ignorar a não linearidade. De forma a verificar se esta abordagem de ignorar os indícios de não linearidade é

adequada ou não, decidiu-se por analisar séries não lineares quanto a sua média.

A escolha dos modelos Box e Jenkins se deve ao fato de apresentar certa robustez quanto a falha de atender a suposição de não linearidade. Conforme Hamilton (1994), os modelos Box e Jenkins são bem robustos quanto a algumas falhas de suposições, entre elas a de não linearidade. Outra razão para esta escolha, se deve ao fato de sua popularização. Modelos Box e Jenkins são de uso comum em análises de séries temporais e muitos aplicativos já contam com rotinas e funções adequadas a sua modelagem. Além disso, trabalhos como Kuo e Reitsch (1995/1996), Smith et al. (2002) e Dellana e West (2009) apresentam comparações semelhantes, utilizando como base os modelos Box e Jenkins.

Porém, no caso da não linearidade da média, existem modelos considerados mais robustos e que têm como principal característica a modelagem de dados não lineares, que são os modelos de RNA (HAYKIN, 2001; CHEN, 2006; TSAI; LI, 2008). A escolha da modelagem e análise por RNA se deve tanto a sua robustez, quanto ao fato de diversos trabalhos reforçarem a idéia de uma modelagem mais propícia a sistemas não lineares (HILL et al., 1996; LIGHTBODY; IRWIN, 1996; HAYKIN, 2001; HAMZAÇEBI et al., 2009). Contudo, os modelos de RNA carecem de métodos menos subjetivos para sua construção e, até o momento, não permitem maiores análises quanto ao comportamento dos dados, fixando-se, exclusivamente, nas previsões geradas (HILL et al., 1996; BALKIN; ORD, 2000).

Apesar da importância da análise de séries temporais, com ênfase nas previsões que podem ser geradas, são diversos os métodos e modelos que podem ser utilizados. Contudo, cada modelo e método de análise é limitado quanto ao seu uso. Os modelos escolhidos para serem utilizados nesta dissertação, assim como as séries de dados, encontram justificativas com base nas afirmações acima, sendo assim, justificam-se as escolhas feitas nesta dissertação.

1.3 Métodos

Segundo Gil (2002), este trabalho trata-se de uma pesquisa aplicada quanto a sua natureza. Desta forma, este trabalho busca obter alternativas frente a falha em atender as suposições de linearidade dos modelos estatísticos clássicos de análise de séries temporais.

Ainda segundo Gil (2002), quanto aos objetivos, a metodologia de pesquisa utilizada é uma pesquisa exploratória. Esta metodologia de pesquisa visa construir hipóteses e tornar mais explícito o problema referente as suposições dos modelos Box e Jenkins. Para tanto, foram selecionadas duas séries de dados, dois estudos de caso distintos.

A primeira série, do índice de produção física industrial mensal para o Estado do Rio

Grande do Sul, é de uso focado no meio industrial. Esta série vai ser usada na comparação de modelos, porém seus resultados serão mais restritos a este trabalho, visto que não é uma série trabalhada de forma extensiva por outros autores, como é o caso das manchas solares. Exatamente por isso, esta série será avaliada de maneira mais completa, verificando-se, inclusive, a sua não linearidade. Além disso, esta série permite avaliações mais focadas na área de produção, por se tratar de um índice importante para o planejamento industrial de uma forma geral.

A segunda série, de manchas solares, é uma série reconhecidamente não linear, utilizada por diversos autores para comparação de modelos, conforme visto em Priestley (1991) e Terui e van Dijk (2002). Esta série tem pouco, ou nenhum, uso prático para a produção, mas apresenta-se como uma série tipicamente não linear. Desta forma, a comparação pode feita entre os modelos apresentados neste trabalho e também frente a outros modelos encontrados na literatura, permitindo uma análise mais ampla desta comparação. Esta série servirá para reforçar ou divergir dos resultados obtidos na série do índice de produção industrial, sendo utilizada como um caso-controle.

Inicialmente são obtidos os dados das duas séries para então a análise ser feita. Para as duas diferentes formas de modelagem (Box-Jenkins e RNA) serão obtidos tantos modelos quanto forem necessários, dadas as limitações impostas por esta dissertação. O processo de obtenção dos modelos será descrito mais adiante. Posteriormente, os modelos considerados de melhor rendimento serão apresentados, assim como suas estatísticas para a comparação.

Conjuntamente será feita uma revisão da literatura, e com base nesta revisão, serão apresentados os modelos de Box e Jenkins para análise de séries temporais. Com base nesta revisão também serão apresentados os modelos de RNA capazes de auxiliar na modelagem dos dados. Tendo sido apresentados os modelos, tanto Box e Jenkins como de RNA, passa-se então a modelagem de ambos.

Para a comparação dos modelos, serão utilizadas as seguintes medidas: erro quadrático médio (EQM), erro quadrático médio de previsão (EQM de previsão), critério de informação de Akaike (AIC) e critério de informação bayesiano (BIC), erro quadrático médio normalizado (NMSE) e o erro absoluto médio percentual (MAPE).

Todas as análises feitas e as modelagens serão realizadas no pacote estatístico R na sua versão 2.8.0. Este pacote se diferencia dos demais por ser gratuito, permitir uma ampla gama de funções, capacidade de programação e funcionamento em diversos sistemas operacionais e arquiteturas de processadores. No apêndice A encontram-se as rotinas utilizadas no programa para a obtenção dos modelos.

1.4 Delimitações do trabalho

Apesar de possuir dois estudos de caso focando a comparação entre modelos RNA e modelos clássicos, este trabalho, de forma alguma, pretende apresentar todos os possíveis modelos para ajuste de uma série de dados, sendo ela linear ou não. Os modelos RNA a serem utilizados utilizarão como critério de ajuste o erro das previsões e serão modelos de alimentação adiante, do original *feedforward*. Além disso, os modelos de RNA serão com apenas uma camada oculta e com um número limitado de neurônios em cada camada. Na camada de entrada serão feitas configurações com 6 e 12 neurônios apenas. Na camada oculta, de 5 a 30 neurônios. A camada de saída será constituída de apenas um neurônio. Em específico, serão utilizados apenas modelos de *perceptron* de múltiplas camadas (MLP).

Porém, as principais restrições se aplicam aos modelos estatísticos clássicos. Não serão vistos modelos desenvolvidos única e exclusivamente para análise de séries não lineares, como os modelos STAR, TAR, por exemplo. Estes modelos, apesar de se basearem em suposições quanto a distribuição de probabilidade dos dados (assim como os modelos clássicos de forma geral) são muito específicos e boa parte dos programas comerciais não possui suas rotinas implementadas. Além disso, este trabalho visa comparações de modelos mais genéricos, mais amplos e não modelos tão específicos. Desta forma, os modelos Box-Jenkins a serem utilizados limitam-se aos modelos da família ARIMA/SARIMA.

Outra delimitação desta dissertação é quanto as séries utilizadas na comparação dos modelos. As séries utilizadas são, ambas, univariadas. Desta forma as previsões são baseadas, única e exclusivamente nos dados anteriores da própria série. Além disso, não foram utilizados modelos simulados, pois para tanto seria necessário optar entre duas alternativas: simulação com base em modelos conhecidos ou simulação de modelos mistos. No caso de simulação com base em modelos conhecidos, o mesmo teria de seguir um modelo não-linear já conhecido, como TAR ou STAR e as comparações perderiam o sentido, já que seria possível, simplesmente, ajustar o modelo simulado. Por outro lado, ao se simular o modelo misto, com diversas características de não-linearidade, isso poderia ocasionar um modelo distante da realidade pretendida por esta dissertação. Sendo assim, decidiu-se por utilizar séries reais de dados.

1.5 Estrutura

O capítulo 1 tratará de apresentar o assunto de forma geral. Também é apresentado neste capítulo: o tema deste trabalho, os objetivos, a metodologia utilizada, as delimitações de pesquisa e a estrutura do trabalho como um todo.

No capítulo 2 está o referencial teórico utilizado neste trabalho. Neste capítulo serão apresentados e descritos os modelos de análise de séries temporais que serão comparados, assim como suas suposições e características. Também são descritos os modelos de RNA mais comuns e seus usos, assim como uma descrição mais detalhada do modelo a ser utilizado.

No capítulo 3 serão apresentados os métodos de análise que serão utilizados em ambos os estudos de caso. Neste capítulo também serão abordados os possíveis testes a serem realizados nas séries e os critérios de comparação entre os diferentes modelos, além das configurações de RNA a serem utilizadas.

O capítulo 4 apresenta a modelagem segundo os métodos apresentados anteriormente. Neste capítulo os modelos são aplicados as séries de dados e seus resultados apresentados. Além disso, os resultados dos critérios de comparação também são apresentados e discutidos frente a resultados obtidos em trabalhos semelhantes.

O quinto capítulo apresenta as conclusões do trabalho quanto aos seus objetivos e resultados apresentados. Aqui também são apresentadas sugestões de trabalhos futuros.

2 ANÁLISE DE SÉRIES TEMPORAIS

A definição mais recorrente de uma série temporal é, basicamente, de dados que foram observados e catalogados ao longo de um certo período de tempo. A provável correlação existente entre os pontos amostrados adiciona uma característica de dependência entre estes pontos (MORETTIN; TOLOI, 2006). Como boa parte das ferramentas estatísticas para análise supõe a independência entre os dados, esta provável correlação provoca a inadequação destas ferramentas. Sendo assim, é necessário incluir estas informações nas ferramentas de análise. Entretanto, uma série temporal nem sempre apresenta apenas dados correlacionados entre si. Uma outra maneira de apresentar uma série temporal é definindo estas observações como uma das prováveis trajetórias que o processo poderia ter sido obtido. Nesta forma de apresentação, os dados da série não são mais apenas dados correlacionados, mas dados advindos de um processo estocástico, no qual observa-se apenas uma das possíveis trajetórias. Estas duas definições não são, de maneira alguma, diferentes entre si. Pelo contrário, estas duas definições se complementam, atribuindo uma correlação temporal e estocástica aos dados observados (ENDERS, 1995).

Existem muitos exemplos de séries temporais, mas apenas dois serão apresentados, na tentativa de esclarecer as definições acima. Se, por exemplo, um dado de seis faces honesto fosse lançado diversas vezes ao longo do tempo e sua face superior anotada a cada lançamento, esta também seria uma série temporal. Aqui fica evidente o componente estocástico: não se supõe que exista alguma correlação entre os lançamentos do dado, sendo o resultado definido apenas por uma função probabilística. O controle de qualidade realizado em uma linha de produção é outro exemplo. Se o produto examinado é mensurado quanto aos seu peso, por exemplo. Estas medidas são realizadas em algumas unidades diariamente e anotadas. Estas anotações também tratam-se de uma série temporal. E estes dados, muito provavelmente, estarão correlacionados entre si. Ainda neste exemplo, não é inadequado supor que anotações realizadas logo após a manutenção de uma máquina estejam mais próximas do esperado, dado que a máquina, a princípio, está em boas condições. Este exemplo, porém, também é útil para apresentar correlações não só entre os pesos anotados, mas com fatores externos também. É de se esperar que a mudança no material de confecção dos produtos possa causar efeitos diversos no peso dos mesmos. Assim como em semanas que, devido a férias, um menor número de produtos é fabricado. Neste caso porém, não é tão trivial visualizar o componente estocástico da série temporal.

Ainda com relação ao exemplo dos pesos dos produtos, pode-se supor que exista um componente estocástico de tal forma que poderiam ter sido obtidas outras mensurações devido a seleção aleatória de produto a ser mensurado. Ou seja, poderiam ter sido mensurados produtos com entre 10g e 11g, mas, ao se registrar um peso de 10,2g em um produto, tende-se a ignorar que este resultado poderia ter sido outro.

As conclusões e suposições obtidas nos exemplos aqui apresentados são relativas a mera observação dos dados. A análise de séries temporais não se restringe apenas a isso. Alguns dos principais objetivos da análise de séries temporais estão relacionados a: (i) previsão, (ii) identificação de componentes, (iii) análise dos parâmetros e (iv) análise de frequência.

A análise de séries temporais visando a previsão é um dos objetivos mais comuns. É, também, o objetivo de interesse nesta dissertação. Apesar de alguns modelos terem seu foco na identificação de componentes, a previsão costuma ser o uso mais comum da modelagem de séries temporais (SHUMWAY; STOFFER, 2000; MORETTIN; TOLOI, 2006). A análise de previsão também costuma ser utilizada na seleção de modelos (FLORES, 2006; MORETTIN; TOLOI, 2006). Um modelo pode se adequar muito bem a série de dados apresentada e ter pouco poder de previsão, ou seja, as previsões deste modelo fogem a realidade dos dados. É uma regra geral que, ao se modelar uma série, alguns dados sejam resguardados para que seja feita uma análise de previsão. Muitas vezes é nessa análise de previsão que o modelo final é escolhido (FLORES, 2006; INOUE; KILIAN, 2006; QI; ZHANG, 2001; GRANGER et al., 1995; SIN; WHITE, 1996).

Quanto aos métodos para analisar e modelar estes dados, eles variam desde métodos clássicos (modelos com suposições bem fundamentadas), até métodos bayesianos (informação *a priori* e combinação de distribuições). Os métodos clássicos, em sua maioria, possuem certas características que os diferenciam dos demais métodos, dentre estas pode-se citar: mensuração probabilística do erro de previsão e uma metodologia menos subjetiva para obtenção de modelos. Nesta família de modelos, os modelos de Box-Jenkins (BOX; JENKINS, 1976; BOX et al., 1994), são alguns dos mais conhecidos. Porém, estas características dependem de algumas suposições.

Contudo, existe uma classe de métodos estatísticos que não possuem tantas suposições quanto os modelos clássicos. Os modelos não-paramétricos possuem poucas ou até nenhuma suposição quanto ao comportamento dos dados. Mas, devido a falta de suposições, tais modelos possuem uma metodologia de uso e construção muito subjetiva. Além desta subjetividade, estes modelos costumam ser utilizados apenas com o objetivo de se obter previsões, e são conhecidos por possuírem pouco poder preditivo frente aos modelos clássicos (SMITH et al., 2002). Mas

uma classe de modelos não-paramétricos se diferencia neste ponto. Os modelos de redes neurais artificiais, apesar de serem modelos não-paramétricos, são conhecidos pela sua qualidade preditiva (em muitos casos superior a outros modelos) e sua complexidade.

Feita esta introdução, a seguir serão tratados os modelos clássicos e após, os modelos de RNA. Como citado anteriormente, os modelos clássicos, por se tratar de uma metodologia já muito utilizada e trabalhada, serão apresentados com um foco mais teórico e terão um espaço menor nesta dissertação. Por outro lado, os modelos de RNA serão apresentados tanto a sua base teórica quanto suas aplicações e, portanto, ocuparão um maior espaço. Porém, inicialmente, será abordada série temporal não linear.

2.1 Série Temporal Não Linear

Uma série temporal é dita não linear quando, pelo menos, uma das seguintes situações ocorre: variância não constante ou média não constante. Dentre as suposições feitas para a análise dos dados nos modelos Box-Jenkins, a suposição de linearidade se refere a média e variância constantes ao longo do tempo (ENDERS, 1995; MORETTIN; TOLOI, 2006)

Séries não lineares quanto a sua variância costumam ocorrer em análises financeiras e macroeconômicas. Uma das formas de não linearidade da variância é a variância condicional, que é a base de modelos GARCH/ARCH que serão brevemente abordados posteriormente. Esta forma de não linearidade não é abordada com ênfase neste trabalho, para tanto recomenda-se Engle (1982), Enders (1995) e Morettin e Toloí (2006).

Outra possibilidade de não linearidade é com relação a média. A não linearidade em relação a média costuma ser mais ampla, pois é comum uma série de dados que apresente uma média inconstante apresentar, também, uma variância inconstante. A não linearidade da média pode ser dividida em dois tipos de situações: a média que varia a uma taxa constante e a média que varia a taxas inconstantes (ENDERS, 1995; MORETTIN; TOLOI, 2006). Quando a média varia com uma taxa constante, a mesma pode facilmente ser transformada em uma série linear. O mesmo não pode ser dito quando a variação ocorre com taxas inconstantes.

Quando a média possui uma variação com uma taxa constante, diz-se que a série é não estacionária. Esta variação pode ser com uma taxa constante em ciclos, chamada de não estacionariedade sazonal. Posteriormente serão apresentadas medidas como as funções de autocorrelação e autocorrelação parcial que podem identificar os comportamentos relativos a não estacionariedade da série (ENDERS, 1995; MORETTIN; TOLOI, 2006). Este tipo de não linearidade porém, é facilmente resolvido ao se tomar diferenças da série de dados. Este processo, também,

será explanado posteriormente nesta dissertação.

A não linearidade da média de interesse nesta dissertação é quando a variação da média não ocorre a uma taxa constante. Este tipo de não linearidade pode se caracterizar por apresentar médias distintas ao longo da série, médias que são modificadas por alguma variável externa a série, ou ainda por processos probabilísticos. Como são vários as causas que levam a não linearidade em relação a média, muitas vezes torna-se complexa a escolha de um modelo que seja adequado. Modelos de mudança de regime probabilística, por exemplo, como vistos em Hamilton (1994) e Kim e Nelson (1999) são adequados quando a variação na média ocorre devido a um processo probabilístico, oculto ou não. Quando esta variação ocorre de uma maneira clara, de forma determinística ou ainda conhecida pelo pesquisador, têm-se modelos de intervenção ou modelos de limiar, como vistos em Enders (1995), Priestley (1991) e Morettin e Tolo (2006). Descrições mais detalhadas e casos específicos, de interesse desta dissertação, serão tratados na explanação dos modelos clássicos, na seção 2.2.2.

Em muitos casos é difícil obter o tipo exato de efeito que está incidindo sobre a média da série. Por isso o uso de modelos RNA têm se difundido. Os modelos de RNA são abrangentes e com nenhuma suposição quanto a linearidade da série. Isso pode ser visto em diversos trabalhos, como Dellana e West (2009), Cheng e Titterington (1994), Balkin e Ord (2000), Hill et al. (1996), Hoo et al. (2002), Kuo e Reitsch (1995/1996), Lai e Wong (2001), Lightbody e Irwin (1996) e Terui e van Dijk (2002). As definições teóricas e o uso dos modelos RNA de interesse desta dissertação serão discutidos e apresentados na seção 2.3.

2.2 Modelos Clássicos

Uma das mais conhecidas maneiras de se tratar séries temporais é com a utilização de modelos clássicos. Segundo Morettin e Tolo (2006), os modelos clássicos com ênfase na análise no domínio do tempo mais comuns são: (i) modelos de regressão, de erro ou polinomiais, (ii) modelos não-lineares, como modelos GARCH/ARCH e modelos com mudança de estados markoviana e (iii) modelos autorregressivos integrados de médias móveis (ARIMA).

A classe de modelos clássicos é nomeada desta maneira pois deriva da estatística clássica e suas suposições com relação aos dados. E, assim como na estatística clássica, os modelos clássicos partem da suposição de que os dados seguem uma distribuição de probabilidade conhecida. Todos os modelos vistos aqui partem da suposição que os resíduos são normalmente distribuídos ao longo do tempo, com exceção dos modelos da família GARCH/ARCH, que costumam supor uma distribuição *t* de *Student* para seus resíduos.

Para que se possa apresentar os modelos, algumas definições são necessárias, principalmente quanto ao tipo de modelagem que se está interessado em utilizar. Com relação as suposições iniciais, muitas destas são demasiadas importantes para o uso das técnicas e conclusões. Outros modelos, porém, são mais robustos e podem relaxar em algumas delas. Contudo, as suposições são praticamente as mesmas para os modelos que serão vistos aqui. Qualquer suposição diferente ou adicional será tratada conjuntamente com a apresentação do modelo.

Uma definição necessária é dada supondo T como sendo um conjunto arbitrário composto das observações ao longo do tempo. Um processo estocástico é uma família $X = x(t), t \in T$, tal que, para cada $t \in T$, $x(t)$ é uma variável aleatória, também denotada por x_t . Desta forma, t é uma amostra de T . Para cada instante de tempo t tem-se que x_t é uma variável aleatória definida sobre Ω , o conjunto universo de probabilidade. Esta definição permite que, para cada instante de t , x_t siga uma distribuição de probabilidade diferente. Entretanto, neste trabalho x_t vai ser assumida igualmente distribuída para todo o t , exceto quando dito de outra forma.

Para exemplificar esta definição, retorna-se ao exemplo do número de acidentes em uma rodovia. Neste exemplo, T representaria todo o tempo de existência da rodovia, desde sua construção até uma possível demolição. Como não é possível analisar todo este período de tempo, então toma-se um período menor de tempo, t , que pode ser dias, meses ou até anos. Para cada dia, mês ou ano, têm-se um resultado, um número de acidentes registrados x_t . Pode-se supor que em cada instante, seja ele dia, mês ou ano, a distribuição probabilística dos dados é diferente. Porém, neste trabalho, exceto quando citado de outra forma, será assumida a distribuição de probabilidade Normal para todo e qualquer instante.

2.2.1 Modelos Box-Jenkins

Os modelos ARIMA (do original em inglês *autoregressive integrated moving average*) são alguns dos mais utilizados modelos para análise de séries temporais (MORETTIN; TOLOI, 2006). Estes modelos têm esta denominação devido a metodologia estruturada por Box e Jenkins (BOX; JENKINS, 1976; BOX et al., 1994), onde um algoritmo é utilizado para auxiliar o usuário na busca por um modelo acurado e que apresente um menor número de parâmetros, um modelo parsimonioso. É este algoritmo que é apresentado na figura (1).

Observando-se a figura (1), têm-se três fases distintas para obtenção de modelos: Fase de Identificação, Fase de Estimação e Testes e a Fase da Aplicação. A separação utilizada na figura (1) contribui para o entendimento do processo utilizado nesta dissertação para obtenção dos modelos da família Box-Jenkins a serem comparados.

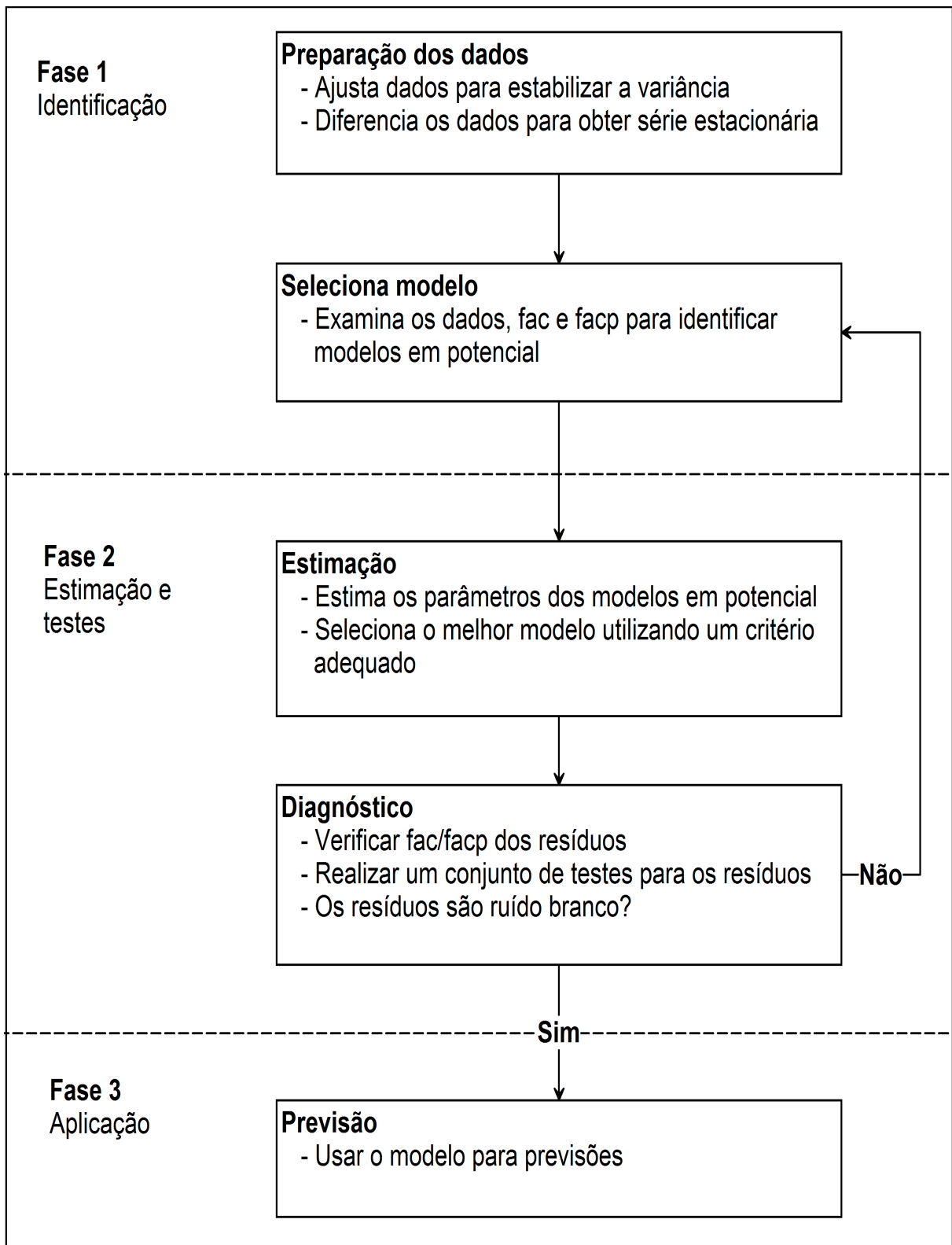


Figura 1: Representação esquemática do algoritmo Box-Jenkins
 Fonte: Adaptado de Makridakis et al. (1998)

Dentre a família dos modelos, destacam-se os autorregressivos (AR) e os de médias móveis (MA). A partir destes modelos, muitos outros foram desenvolvidos, entre eles: autorregressivo integrado de médias móveis (ARIMA), no qual são tomadas d diferenças de uma série não estacionária de forma a torná-la estacionária. Há ainda o autorregressivo de médias móveis (ARMA, do original em inglês *autoregressive moving average*) quando não há a necessidade de se tomar diferenças da série para torná-la estacionária e o modelo sazonal, mais completo, que capta a sazonalidade da série, chamado de ARIMA sazonal, ou SARIMA (do original em inglês *seasonal autoregressive integrated moving average*).

O algoritmo de Box-Jenkins auxilia o usuário a escolher o melhor modelo baseado nos gráficos da função de autocorrelação (fac) e da função de autocorrelação parcial (facp). Uma das maiores críticas quanto a este algoritmo é relativo à subjetividade na identificação do modelo (ENDERS, 1995; SHUMWAY; STOFFER, 2000). Não raro, diferentes usuários chegam a diferentes modelos para uma mesma série de dados. Entretanto, comparada com outras técnicas, a metodologia Box-Jenkins ainda se sobressai quanto a escolha de modelos devido, entre outras razões, a sua facilidade de uso, principalmente em relação a modelos não paramétricos, como, por exemplo, os modelos de RNA.

Os modelos Box-Jenkins além de permitirem a previsão, também permite que sejam analisados os seus parâmetros, ao contrário do que ocorre com modelos RNA. Para tanto, é necessário conhecer algumas formulações dos modelos. Os 4 modelos mais importantes são: AR, MA, ARIMA e SARIMA. Uma maior ênfase será dada aos modelos AR e MA pois são a base dos modelos da metodologia de Box-Jenkins (HAMILTON, 1994; MORETTIN; TOLOI, 2006).

Contudo, para que os modelos possam ser identificados e selecionados, é necessário definir duas importantes funções: a função de autocorrelação (fac) e a função de autocorrelação parcial (facp). A fac é obtida ao se obter a correlação linear de cada valor y_t da série de dados com outros valores em *lags* distintos, como y_{t-1}, y_{t-2} , e assim por diante. A facp, por sua vez, tem um conceito semelhante. A facp é a correlação entre, por exemplo, y_t e y_{t-3} , mas sem levar em consideração os efeitos causados por y_{t-1} e y_{t-2} . No caso da fac, estes efeitos são considerados. Estas duas funções tem por objetivo auxiliar na obtenção de modelos, cada uma a sua forma.

Sendo assim, tendo definido as funções fac e facp, passa-se então ao algoritmo de Box-Jenkins. Como a metodologia apresentada por Box-Jenkins (BOX et al., 1994) é baseada nas propriedades das funções fac e facp. Os passos descritos aqui, na maioria das vezes, ocorrem conjuntamente, tornando mais complexa a identificação dos modelos. Box et al. (1994) sugere a busca por modelos com menor número de parâmetros e então agregar novos termos, novos

parâmetros.

Primeiramente, quando a série é não estacionária, nestes casos, a *fac* decai muito vagarosamente e a *facp* tem um grande pico no *lag* 1. Possivelmente pode-se chegar a uma série estacionária através de d diferenças, onde, normalmente, d é igual a 1. Quando a não-estacionaridade é sazonal a *fac* é zero exceto nos *lags* $s, 2s, 3s, \dots$ e decai muito vagarosamente. Pode-se obter uma série estacionária tomando-se D diferenças sazonais. Normalmente, D é igual a 1.

O comportamento autorregressivo pode ser observado quando a *fac* é diferente de zero para os *lags* $m = 1, 2, \dots, p$ e zero após, nesses casos um $AR(p)$ é sugerido. O comportamento autorregressivo sazonal possui uma *facp* igual a zero, exceto nos *lags* $m = s, 2s, \dots, Ps$, nesse caso sugere-se um modelo $AR_s(P)$. O comportamento autorregressivo se refere a correlação existente entre os valores atuais e entre os valores anteriores.

No caso de médias móveis, este comportamento pode ser identificado quando possui uma *facp* diferente de zero para os *lags* $m = 1, 2, \dots, q$ e zero após, nesse caso, sugere-se um $MA(q)$. O comportamento de médias móveis sazonais possui uma *fac* igual a zero exceto nos *lags* $m = s, 2s, \dots, Qs$, nesse caso a sugestão é de um $MA_s(Q)$. O comportamento de médias móveis se refere a correlação existente entre os valores atuais e os resíduos obtidos de valores anteriores.

Apresentado o algoritmo de Box e Jenkins utilizado na etapa de seleção do modelo, passa-se a definição dos modelos em si. Um modelo $AR(p)$ é construído como função da variável em tempos passados e o resíduo no período, sendo expresso pela equação (1).

$$x_t = c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_i x_{t-i} + \varepsilon_t; \quad (1)$$

onde c é uma constante qualquer, também chamada de constante de nível, p é a ordem do modelo, x_t é o valor da variável mensurada no instante de tempo t , ϕ_i os parâmetros do modelo, onde $i = 1, 2, \dots, p$ e ε o resíduo, que deve seguir uma distribuição de probabilidade Normal, contudo, os modelos Box-Jenkins são bem robustos e conseguem boas estimativas mesmo com falha nesta suposição (HAMILTON, 1994).

Os modelos $AR(p)$ podem ser representados por um processo de ordem finita, sendo esta propriedade conhecida como invertibilidade, e são conhecidos modelos invertíveis (MORETTIN; TOLOI, 2006; ENDERS, 1995). Como exemplo de como se apresenta a *fac* e *facp* de um modelo autorregressivo, têm-se as figuras (2) e (3) que apresentam a *fac* e a *facp* de um modelo $AR(2)$, respectivamente.

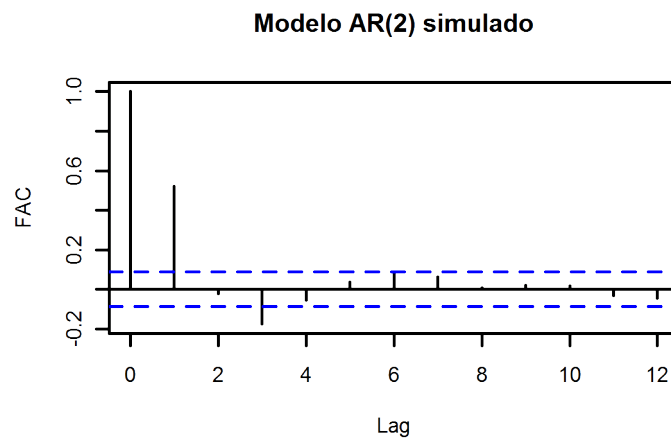


Figura 2: Exemplo de fac de um modelo AR(2)
Fonte: Elaborado pelo autor

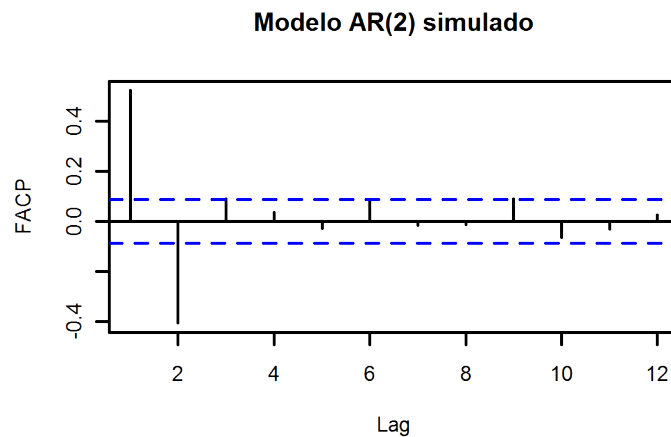


Figura 3: Exemplo de facp de um modelo AR(2)
Fonte: Elaborado pelo autor

Observando-se as figuras (2) e (3), verifica-se os dois termos significativos¹ na figura (3), indicando um modelo autorregressivo de 2ª ordem, um AR(2). Nota-se também o decaimento rápido da fac, apresentada na figura (2). O gráfico da figura (2) indica o primeiro *lag* significativo e igual a 1(um). Isso ocorre porque o gráfico da fac apresenta os *lags* a partir de 0(zero) e, no *lag* 0(zero) a correlação é sempre perfeita e positiva, pois representa a correlação entre dois valores iguais, o valor no instante zero correlacionado consigo mesmo.

O modelo MA(q), por sua vez é definido pelo comportamento dos erros no passado e é representado pela equação (2). Segundo Morettin e Tolo (2006), os modelos MA(q) são sempre estacionários.

$$x_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_i \varepsilon_{t-i}; \quad (2)$$

¹Por significativo entende-se um valor que exceda os limites considerados aceitáveis.

onde c é uma constante qualquer, q é a ordem do modelo, θ_i são os parâmetros do modelo, sendo $i = 1, 2, \dots, q$ e ε_t é o resíduo no tempo t .

Como exemplo de como se apresenta a fac e facp de um modelo de médias móveis, têm-se as figuras (4) e (5) que apresentam a fac e a facp de um modelo MA(2), respectivamente. Observando-se a figura (4), pode-se perceber os *lags* 1 e 2 significativos, indicando um modelo de médias móveis de 2ª ordem, um MA(2). Aqui também nota-se o rápido decaimento da fac.

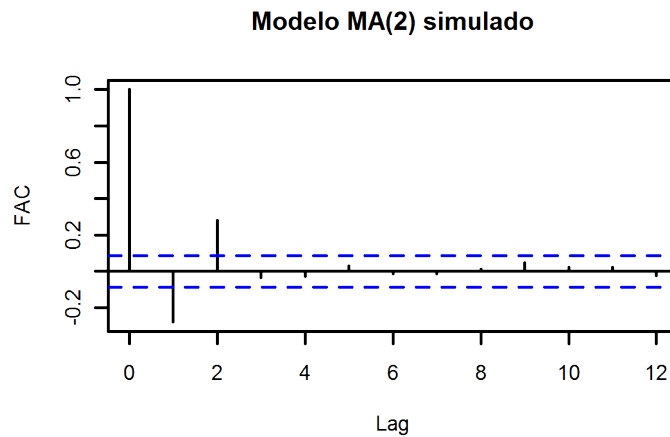


Figura 4: Exemplo de fac de um modelo MA(2)
Fonte: Elaborado pelo autor

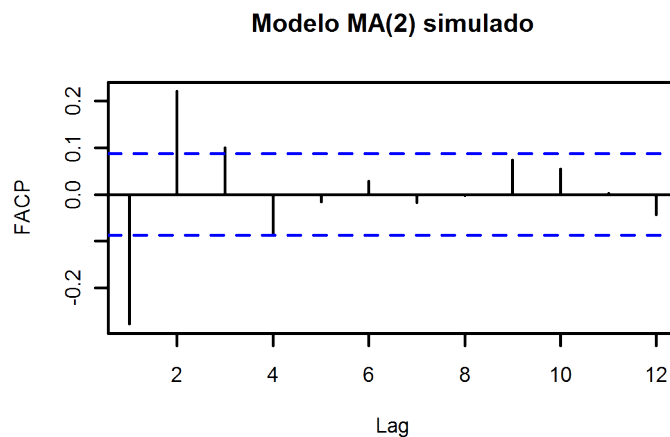


Figura 5: Exemplo de facp de um modelo MA(2)
Fonte: Elaborado pelo autor

Outro ponto importante a se destacar é que o gráfico da facp (figura (5)) apresenta valores significativos até o *lag* 3, que poderia indicar um modelo AR(3). Porém, pelo algoritmo de Box e Jenkins, deve-se sempre optar pelo modelo com menor número de parâmetros, no caso, um MA(2). Isto porque o MA(2) possui apenas dois parâmetros, enquanto um modelo AR(3) possui três parâmetros.

Um modelo ARIMA é uma conjunção de modelos AR(p) e modelos MA(q), onde, tanto a parte autorregressiva como a parte de médias móveis são modeladas. Além da série ser integrada com d diferenças, para que a série torne-se estacionária. Porém, para que os modelos ARIMA e SARIMA possam ser apresentados corretamente, é necessário introduzir a notação $\Phi(L)$ e $\Psi(L)$. Esta notação visa simplificar a apresentação dos modelos.

Seja o modelo AR(p) definido na equação (1) e $L^k x_t = x_{t-k}$. Este modelo também pode ser escrito como na equação (3): $(1 - \phi_1 L + \phi_2 L^2 + \dots + \phi_p L^p)x_t = c + \varepsilon_t$; ou ainda:

$$\Phi(L)x_t = c + \varepsilon_t. \quad (3)$$

A notação $\Psi(L)$ é semelhante, porém utilizada nos modelos MA(q). Conforme a equação (2) e a equação (3), o modelo MA(q) pode ser descrito como:

$$x_t = c + \Psi(L)\varepsilon_t. \quad (4)$$

Segundo Box et al. (1994), se são tomadas d diferenças em uma série estacionária, então a d -ésima diferença definida por $\Delta^d = (1 - L)^d$, então um processo ARIMA(p,d,q) é definido por:

$$\Phi(L)\Delta^d x_t = \Psi(L)\varepsilon_t; \quad (5)$$

Porém, frequentemente há componentes periódicos em uma série temporal que tendem a se repetir em s unidades de tempo. Esse componente é chamado de componente sazonal. Ainda segundo Box et al. (1994), o modelo ARIMA pode ser multiplicativo. Um modelo ARIMA multiplicativo que possua uma parte sazonal é definido pela equação (6):

$$\phi_P(L^s)\Phi_p(L)\Delta_s^D\Delta^d x_t = \psi_Q(L^s)\Psi_q(L)\varepsilon_t; \quad (6)$$

onde s é o período sazonal e $\Delta_s^D = (1 - L^s)^D$.

A equação (6) define um modelo SARIMA(p, d, q) \times (P, D, Q) $_s$, com (P, D, Q) sendo os componentes autorregressivos, integrados e de médias móveis, respectivamente, em períodos sazonais.

Como exemplo, foi simulada uma série não-estacionária com um termo autorregressivo de 1ª ordem e um termo de médias móveis de 1ª ordem, um ARIMA(1,1,1), já que a estacionariedade pode ser removida na primeira integração ou diferenciação. As figura (6) e (7) apresentam os gráficos da fac e facp, respectivamente, de um modelo ARIMA(1,1,1).

Observando-se a figura (6) percebe-se o lento decaimento proveniente de uma série de

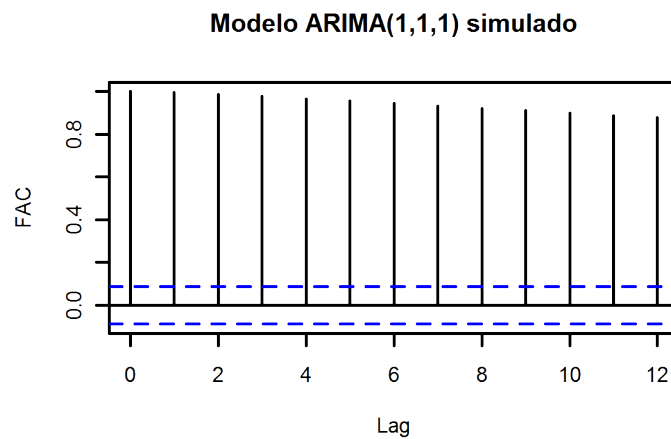


Figura 6: Exemplo de fac de um modelo ARIMA(1,1,1)
Fonte: Elaborado pelo autor

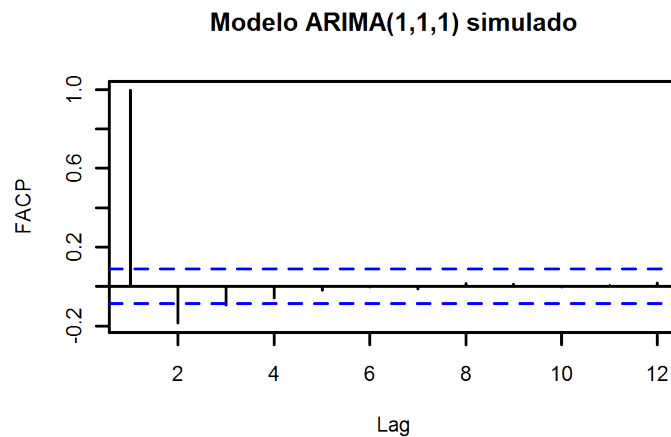


Figura 7: Exemplo de facp de um modelo ARIMA(1,1,1)
Fonte: Elaborado pelo autor

dados não estacionária. O gráfico da facp, na figura (7), indica um componente de 2^a ou até 3^a ordem em um modelo autorregressivo, porém, devido a não estacionaridade, nenhuma análise pode ser feita antes de tornar a série estacionária.

Outro exemplo, a respeito dos modelos e do algoritmo de Box e Jenkins se refere aos modelos ARMA que não possuem a parte integrada, partindo de uma série já estacionária. Estes modelos costumam representar melhor o cuidado entre que modelo deve ser seguido. Os gráficos da fac e facp apresentados respectivamente nas figuras (8) e (9), se referem a um modelo ARMA(2,2) simulado.

Na figura (8) percebe-se o rápido decaimento, indicando uma série estacionária e um *lag* significativo de 2^a ordem, indicando um modelo MA(2). A facp, na figura (9), por sua vez possui um *lag* significativo de 3^a ordem, indicando um modelo AR(3). Como o modelo MA(2)

possui menor número de parâmetros do que um AR(3), então primeiro modela-se um MA(2) e após é feita a análise dos resíduos do modelo novamente para então definir o próximo passo.

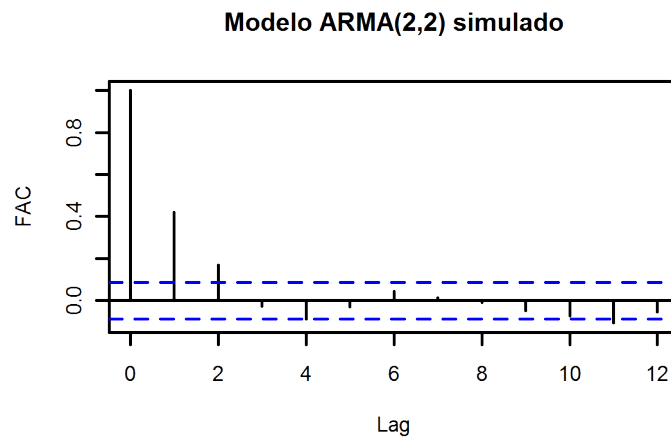


Figura 8: Exemplo de fac de um modelo ARMA(2,2)
Fonte: Elaborado pelo autor

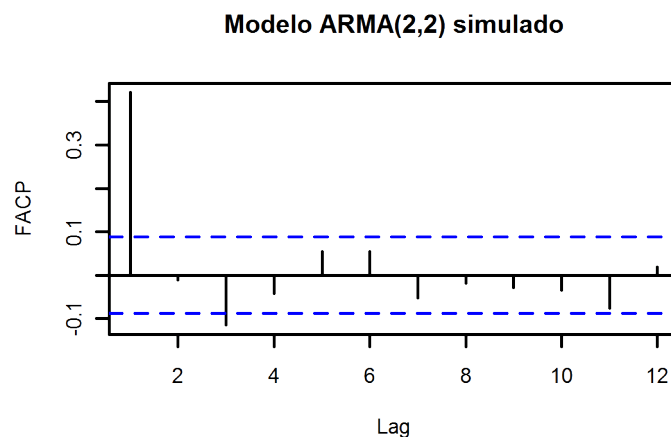


Figura 9: Exemplo de facp de um modelo ARMA(2,2)
Fonte: Elaborado pelo autor

Desta forma, observando-se a figura (1), têm-se a preparação dos dados e a seleção do modelo, que se resume a Fase 1. Tendo sido selecionado um dos modelos citados, a próxima etapa é a de estimar os parâmetros do modelo escolhido. São vários os métodos utilizados para estimação de parâmetros. O filtro de Kalmann é recomendado por Hamilton (1994). Mas pode ser utilizado, também, métodos de mínimos quadrados. Feito isso, critérios como AIC e BIC são utilizados de forma a comparar modelos, caso mais de um seja obtido. Estes critérios devem ser minimizados, de forma a se obter um modelo com menor erro de ajuste e menor número de parâmetros (parcimônia).

Selecionado o modelo e obtidos os seus parâmetros, a próxima etapa é diagnosticar o modelo, verificando seu ajuste, se o modelo ainda está incompleto e detectar um possível sobre

ajuste. Para se diagnosticar um modelo, inicialmente são construídos os gráficos da *fac* e *facp* dos resíduos do modelo. Estes gráficos não devem apresentar nenhum *lag* significativo. Muitas vezes, um ou outro *lag* aparecem como sendo significativos. Nestes casos recomenda-se o uso de testes estatísticos, de forma a verificar se estes *lags* podem prejudicar o modelo. Os testes de Box-Pierce e Ljung-Box são recomendados (ENDERS, 1995; MORETTIN; TOLOI, 2006). O objetivo da análise da *fac* e *facp*, assim como os testes, é verificar se o modelo está bem ajustado, restando apenas o ruído branco, o erro aleatório da série de dados. Com relação ao ajuste, ele pode ser detectado ao se verificar a qualidade preditiva do modelo. Modelos sobre ajustados tendem a possuir um EQM de ajuste pequeno e um EQM de previsão alto, o que significa que o modelo se ajustou bem aos dados, mas devido ao excesso de parâmetros, perdeu o seu poder preditivo. Neste trabalho, além dos testes e da análise gráfica da *fac* e *facp*, também serão utilizados os critérios do EQM de previsão, NMSE e MAPE para comparação de modelos. Estes critérios serão apresentados no capítulo 3.

Tendo sido realizadas as Fases 1 e 2, conforme figura (1), a última Fase se traduz na aplicação do modelo para obtenção das previsões. Normalmente, em modelos aplicados, a cada período de tempo, repete-se os passos descritos nas Fases 1 e 2, de forma a atualizar o modelo, caso seja necessário (MORETTIN; TOLOI, 2006). Este trabalho se resume a um período definido de tempo e, portanto, essa atualização não é necessária.

2.2.2 Outros modelos

A seguir são apresentados outros modelos de séries temporais. Os modelos a seguir são apresentados de maneira resumida pois nenhum deles é utilizado neste trabalho. Alguns destes são de uso bem específico para certas características de séries temporais.

2.2.2.1 Modelos Polinomiais e de Regressão

Quando os dados apresentam uma linearidade em relação a média, estes modelos são adequados. Os modelos polinomiais são, de uma forma geral, modelos basicamente descritivos com baixo poder inferencial. Existem algumas variações dos modelos polinomiais, como os modelos com mudança de regime, que serão apresentados em um tópico em separado.

Quanto aos Modelos de Regressão, estes raramente são utilizados na análise de séries temporais, devido, principalmente, a algumas suposições necessárias ao seu uso. Uma destas suposições é que a capacidade inferencial de uma análise de regressão é, com algumas exceções, exclusiva para o período estudado (NETER et al., 1996; BUSSAB; MORETTIN, 2006). O que deixa claro que não é adequado fazer inferências sobre o comportamento de variáveis além dos

limites utilizados na construção do modelo. Por exemplo, supondo que alguém deseje estudar a temperatura média diária e utilize os meses de dezembro a março, nesta ordem. Caso seja feita uma inferência para a temperatura em abril, esta poderá ser muito mais alta do que de fato ocorre, dado o comportamento sazonal da temperatura e o período modelado. Quando se utiliza a análise de regressão para estudos de séries temporais, uma das variáveis, obrigatoriamente, deve ser o tempo. Além disso, quando se conhece a sazonalidade dos dados, a utilização de variáveis *dummy* podem ser úteis.

2.2.2.2 Modelos Não Lineares

Como citado anteriormente, uma série de dados pode ser não linear de, pelo menos, duas formas: com relação a sua variância e/ou com relação a sua média. Quando a série é não linear com relação a variância, se utiliza a técnica de modelos autorregressivos com heteroscedasticidade condicional (ARCH, da sigla em inglês *autoregressive conditional heteroskedastic*), introduzidos por Engle (1982). Entretanto, os modelos ARCH contêm apenas a parte autorregressiva. Em 1986, Bollerslev generalizou, adicionando uma parte de médias móveis. Estes são os modelos ARCH generalizados, ou GARCH, do original em inglês. Nesta classe de modelos existem ainda os ARCH-M (onde a média depende da variância condicional), IGARCH (modelo GARCH integrado), EGARCH (um modelo GARCH exponencial) entre outros (MORRETTIN; TOLOI, 2006; ENDERS, 1995).

Quanto a não linearidade em relação a média, uma das maneiras de ser abordada é quando se supõe que a média é constante apenas em alguns períodos, também chamados de regimes ou estados, assim como definido anteriormente. Os modelos são utilizados quando o período de tempo t estudado contém mais de um regime ou estado. Desta forma, para o tempo t a série é vista como não linear, porém, dentro de cada regime, a série é vista como linear. Existem de dois tipos de modelos: determinísticos e probabilísticos.

A mudança de regime determinística ocorre quando o pesquisador conhece o comportamento dos cenários. Os modelos com mudança de regime determinística ou modelos de intervenção tratam de séries nas quais houve algum evento, uma intervenção, que a modificasse. Nestes modelos são utilizadas variáveis indicadoras, que recebem o nome de função de transferência, já que, não obrigatoriamente, sejam binárias.

A mudança de regime probabilística, por outro lado, é mais complexa e pode ser de dois tipos: mudança de regime markoviana (não observável) e a mudança de regime observável. Estes modelos costumam ser utilizados quando as mudanças de regime ocorrem segundo algum princípio probabilístico, como as Cadeias de Markov. No caso não observável, a Cadeia de

Markov é estimada enquanto que no caso observável a mesma é informada pelo pesquisador, semelhante ao que ocorre em modelos de intervenção.

Outros modelos não lineares, como o autorregressivo não linear (do original em inglês, *non-linear autoregressive*, NLAR), autorregressivo generalizado (do original em inglês, *generalized autoregressive*, GAR) e modelos de limiar como o autorregressivo de limiar (do original em inglês, *threshold autoregressive*, TAR) e os autorregressivos de transição suave e autorregressivos de transição suave exponencial, respectivamente STAR (do original em inglês, *smooth transition autoregressive*) e ESTAR (do original em inglês, *exponential smooth transition autoregressive*) tem suas características descritas por Priestley (1991) e Enders (1995).

2.2.2.3 Alisamento exponencial

Os modelos de alisamento exponencial, assim como os modelos de regressão, são considerados como alguns dos mais simples para análise de séries temporais (MORETTIN; TOLOI, 2006; ABRAHAM; LEDOLTER, 2005). Os modelos de alisamento exponencial se baseiam na suposição de que os valores extremos da série são devidos a certo tipo de ruído. Sendo assim, uma suavização na série permite que o padrão da série seja reconhecido, utilizando-o para previsão de dados futuros. Existem três classes de modelos de alisamento exponencial: (i) séries localmente constantes analisadas pelos modelos de médias móveis simples (MMS) ou a suavização exponencial simples (SES); (ii) séries com tendência existe a suavização exponencial de Holt (SEH); (iii) nas séries com comportamento sazonal utiliza-se a suavização exponencial sazonal de Holt-Winters (HW). A sazonalidade a ser modelada pode ser tanto multiplicativa quanto aditiva.

2.2.2.4 Modelos vetoriais

Os modelos vetoriais possuem este nome devido a característica da variável. Nesta situação, ao invés de uma variável x_t , é utilizado um vetor de variáveis x_t , de tal forma que tanto a modelagem quanto as previsões são feitas de forma vetorial e não mais de forma univariada. Muitas das questões sobre os modelos vetoriais se refere a inclusão de variáveis no vetor x_t . Assim como nas outras técnicas apresentadas, a experiência do usuário é fundamental para a escolha adequada de variáveis a serem utilizadas na modelagem. Nestes casos, costuma se utilizar a análise de regressão de forma a obter variáveis que sejam correlacionadas. O mais conhecido modelo vetorial é o autorregressivo vetorial (VAR, do original em inglês *vector autoregressive*), parametrizado como VAR(p), onde p é o *lag* autorregressivo (WANG; BESSLER, 2002; CHAO; PHILLIPS, 1999).

2.3 Redes Neurais Artificiais

Uma das maneiras de se analisar uma série temporal é com o uso de modelos de redes neurais artificiais (RNA). Apesar de alguns modelos terem suposições quanto a distribuição de probabilidade dos dados, os modelos de RNA são considerados não-paramétricos ou de estatística robusta (RIPLEY, 1996). O termo robusto se deve ao fato dos modelos de RNA serem capazes de lidar melhor com certas características dos dados, como a não linearidade, do que outros modelos. Outra característica importante é que modelos de RNA têm uma certa facilidade de lidar com sistemas dinâmicos, como séries temporais (DE WILDE, 1997; HAYKIN, 2001; MÜLLER et al., 1995). Por sistema dinâmico entende-se um sistema que não possui, ou não se encontra, em equilíbrio. Uma série temporal não estacionária é um exemplo de sistema dinâmico.

Entretanto, os modelos de RNA não se restringem apenas a modelagem de séries temporais. Uma RNA é um modelo matemático(ou estatístico) que, em sua essência, tenta reproduzir o modelo de funcionamento do cérebro, de redes chamadas de redes neurais naturais (RNN). Todos os cérebros, independentes de sua natureza, são formados por neurônios, os quais comunicam-se entre si por sinapses. Uma RNA segue esta seqüência. A definição de uma RNA está ligada a algoritmos para tarefas cognitivas, como aprendizagem, otimização e modelagem, assim como as RNN (MÜLLER et al., 1995; DE WILDE, 1997). Contudo Haykin (2001) vai mais além. O autor define uma RNA, também chamada simplesmente de rede neural, como uma forma de obter junto aos computadores a forma de raciocinar utilizada pelo cérebro humano. Haykin (2001) cita ainda, algumas características do cérebro que a RNA tenta igualar: um sistema de informações complexo, não-linear e de funcionamento paralelo. Como pode-se perceber, tanto Haykin (2001) quanto Müller et al. (1995) e De Wilde (1997), apesar das definições dadas, concordam que a RNA é uma estrutura complexa para analisar os dados.

O primeiro modelo de RNA data de 1943. Neste ano, Warren McCulloch e Walter Pitts desenvolveram aquele que é considerado o primeiro modelo de RNA. Este modelo tem neurônios bem simples, que só podiam assumir valores 1 ou 0, ligado e desligado, respectivamente. Também são chamados de neurônios de decisão, ou de escolha, devido a seus dois estados bem definidos. No modelo de McCulloch-Pitts, os neurônios são ativados, ou seja, passando estado 0 para o estado 1, caso seus pesos w_i ultrapassem um certo parâmetro ϑ_i . O estado de um certo neurônio é definido pela combinação linear de seus valores de saída, conforme a equação (7).

$$h_i(t) = \sum_j w_{ij} n_j(t), \quad (7)$$

onde $n_i = 0, 1$, w_{ij} representa os pesos sinápticos (ou força sináptica) entre os neurônios j e i e $h_i(t)$ modela o que na época foi chamado de polarização potencial total pós-sináptica no neurônio i , ou a entrada (*input*).

Com esse modelo, a rede neural é completamente determinada pela relação entre $h_i(t)$ e $n_i(t+1)$. No caso mais simples, o neurônio muda de estado se o valor de entrada excede o parâmetro ϑ_i . Neste modelo, a evolução da rede segue a regra apresentada na relação (8).

$$n_i(t+1) = \theta(h_i(t) - \vartheta_i), \quad (8)$$

onde $\theta(x)$ é a função de degrau unitário, de tal forma que $\theta(x < 1) = 0$ e $\theta(x \geq 1) = 1$.

Para McCulloch e Pitts esta rede neural poderia resolver qualquer tipo de computação, similar a um computador atual. Entretanto, para se utilizar este modelo, era necessário estabelecer os pesos da matriz w_{ij} para cada uma das tarefas que a rede iria modelar. Para resolver este problema, segundo Haykin (2001), Eduardo Caianiello implantou um algoritmo capaz de determinar os pesos sinápticos. O algoritmo, chamado de equação mnemônica, seria o princípio da regra de aprendizagem de Hebb (MÜLLER et al., 1995).

Em 1949, com a publicação do livro de Hebb, é apresentada, pela primeira vez, uma regra explícita de aprendizagem para a modificação sináptica. O que Hebb propôs é que a conectividade encontrada no cérebro é modificada conforme o aprendizado de novas tarefas e que isso cria agrupamentos neurais. O postulado de aprendizagem de Hebb afirma que o peso sináptico, agora também chamado de eficiência sináptica, é aumentado pela ativação repetida entre dois neurônios, via sinapse. Com esta afirmação, o peso do neurônio tende a um valor ótimo quando este é ativado repetidas vezes por outro neurônio, ligado a ele pela mesma sinapse. Apesar do livro de Hebb ter tido uma boa aceitação na área da psicologia, o mesmo não ocorreu na área da engenharia (HAYKIN, 2001).

Na década de 70, houve um esmorecimento quanto a pesquisa em redes neurais na área da engenharia, interesse que só foi retomado em 1982 com o artigo de Hopfield. Ele desenvolveu uma classe especial de redes neurais com realimentação, que passou a ser chamada de redes de Hopfield. Estas redes, apesar de não serem realísticas do ponto de vista neurobiológico, incorporaram o armazenamento de informação em redes dinamicamente estáveis (HAYKIN, 2001). Uma das críticas feitas as redes de Hopfield é que elas fazem a atualização dos pesos de maneira seqüencial, o que difere da maneira pela qual o cérebro funciona (processamento paralelo)

e a qual se atribui a razão de o cérebro ser tão superior em tarefas complexas a praticamente qualquer máquina existente (MÜLLER et al., 1995).

Outros autores importantes também merecem destaque, como Rosenblatt, em 1958, que descreveu seu trabalho sobre o *perceptron*, Little e Gordon Shaw, em 1978, pelo desenvolvimento de redes com atribuição de peso sincronizada e a tese de doutoramento de Minski em 1954, voltado a inteligência artificial e pela primeira denominação de rede neural. Ver Haykin (2001), Müller et al. (1995), Ripley (1996) para um maior aprofundamento nas questões históricas das RNA.

2.3.1 Definições

O principal elemento em uma RNA, assim como em redes neurais naturais é o neurônio (HAYKIN, 2001; MÜLLER et al., 1995; DE WILDE, 1997; RIPLEY, 1996). É ele que executa todos os passos. O funcionamento descrito a seguir serve para qualquer modelo de RNA. As mudanças entre os modelos costumam ser a respeito do funcionamento exato das partes apresentadas na Figura (10), que apresenta o modelo geral de um neurônio e onde são identificados seus elementos: conexões, junção aditiva ou somador, função de ativação e o viés ou *bias*.

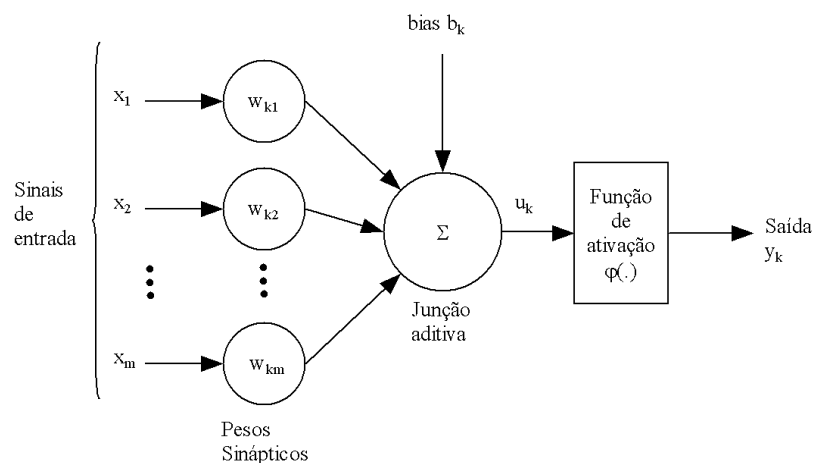


Figura 10: Modelo não linear de um neurônio
Fonte: Adaptado de Haykin (2001)

O primeiro elemento importante são as conexões, os elos de ligação entre os neurônios, chamados basicamente de sinapses. Cada sinapse possui um peso w_{kj} , chamado de peso sináptico. Um sinal x_j é disparado na sinapse j que está ligada ao neurônio k . Os pesos sinápticos w_{kj} podem possuir tanto valores positivos como negativos.

O segundo elemento é um somador ou junção aditiva, também chamado de combinador linear. Sua função é basicamente combinar os sinais x_j com seus respectivos pesos sinápticos w_{kj} de forma a dar seguimento as operações.

Outro elemento é a função de ativação. A função de ativação do neurônio que tem como objetivo restringir ou limitar o intervalo permitido de amplitude do sinal de saída. Boa parte dos modelos são diferenciados pela maneira e pelo formato das funções de ativação dos neurônios. Ressalta-se que cada neurônio pode ter uma função de ativação própria, apesar de isso não ser comum (HAYKIN, 2001; MÜLLER et al., 1995; HAMZAÇEBI et al., 2009). A família de funções de ativação mais utilizada são as funções sigmóides (HAYKIN, 2001; MÜLLER et al., 1995). Dentre estas uma das mais conhecidas é a função logística, definida pela equação (9):

$$\varphi(v) = \frac{1}{1 + \exp -av}; \quad (9)$$

onde a é o parâmetro de inclinação da função.

As vantagens da função logística são a sua capacidade não linear e a limitação dos valores entre 0 e 1, sendo ainda diferenciável. Outra importante função de ativação é a função de ativação de limiar, onde o resultado é binário sendo 0 um neurônio desligado e 1 um neurônio funcional. A função de ativação de limiar é mais comum em redes onde se pretende uma aprendizagem competitiva.

O viés ou *bias* de uma rede pode ser incluído tanto como consta na Figura (10), como uma entrada constante do neurônio, com um dos sinais x_j . O principal objetivo do *bias* é aumentar ou diminuir a entrada líquida da função de ativação. De forma geral, pode-se definir um neurônio segundo as equações (10) e (11).

$$u_k = \sum_{j=1}^m w_{kj}x_j \quad (10)$$

$$y_k = \varphi(u_k + b_k) \quad (11)$$

onde x_1, x_2, \dots, x_m são os sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos do neurônio k ; u_k é a saída do combinador linear devido aos sinais de entrada; b_k é o *bias*; $\varphi(\cdot)$ a função de ativação e y_k é o sinal de saída do neurônio.

2.3.2 Arquiteturas de Redes

Os modelos de RNA podem ser divididos em muitas classes especiais, como por exemplo, redes de função de base radial, *perceptron* e *perceptron* de múltiplas camadas. Entretanto, quanto ao formato, a topologia de uma rede, este pode ser dividido em 3 classes gerais, diferentes no seu funcionamento (HAYKIN, 2001; MÜLLER et al., 1995; DE WILDE, 1997): (i)redes alimentadas adiante com camada única, (ii)redes alimentadas adiante com múltiplas camadas e

(iii) redes recorrentes.

2.3.2.1 Redes alimentadas adiante com camada única

É considerado o mais simples modelo de rede (HAYKIN, 2001; MÜLLER et al., 1995; DE WILDE, 1997). Redes com esta arquitetura têm a camada de entrada e a camada de saída, não possuindo nenhuma camada oculta. É alimentada adiante pois o fluxo de informação é unidirecional, sendo projetado apenas da camada de entrada para a camada de saída. A Figura (11) apresenta um exemplo de rede alimentada adiante com uma camada única.

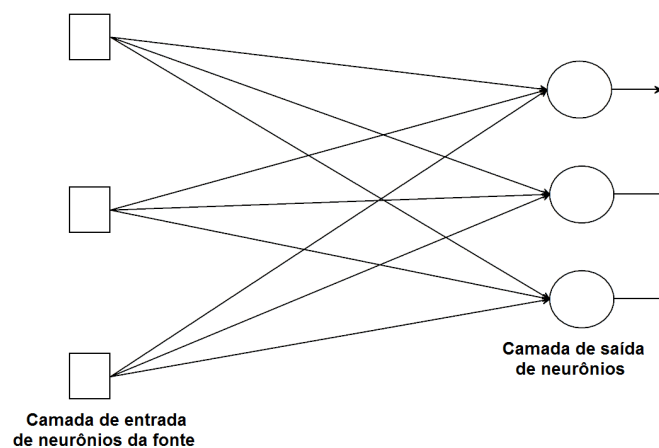


Figura 11: Arquitetura de rede alimentada adiante de camada única
Fonte: Adaptado de Haykin (2001)

2.3.2.2 Redes alimentadas diretamente com múltiplas camadas

Este é o modelo mais conhecido de RNA (HAYKIN, 2001; ZHANG; BERARDI, 2001; QI; ZHANG, 2001; AUER et al., 2008). Segundo Qi e Zhang (2001), mais utilizado também. Nesse modelo, a rede conta com uma ou mais camadas ocultas de neurônios. São chamados de neurônios ocultos porque eles não são visíveis como os neurônios da camada de entrada e nem como os neurônios da camada de saída (HAYKIN, 2001; AUER et al., 2008). Modelos baseados em redes de múltiplas camadas seguem a nomenclatura $X - Y - Z$, onde X é o número de neurônios na camada de entrada, Y o número de neurônios na camada oculta e Z o número de neurônios na camada de saída. No caso de redes com mais de uma camada oculta, coloca-se a quantidade de neurônios na seqüência. Por exemplo, uma rede com 5 neurônios na camada de entrada, seguido por uma camada oculta de 4 neurônios e finalmente uma camada de saída com 2 neurônios. Este modelo é descrito como 5-4-2. A Figura (12) apresenta uma rede 5-4-2, pois é uma rede com 5 neurônios na camada de entrada, uma única camada oculta com 4 neurônios e uma camada de saída com 2 neurônios.

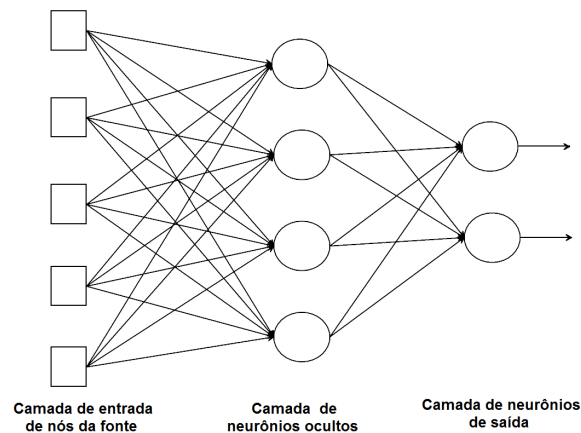


Figura 12: Arquitetura de rede alimentada diretamente com múltiplas camadas
 Fonte: Adaptado de Haykin (2001)

Redes de múltiplas camadas podem ser ainda totalmente conectadas ou parcialmente conectadas. Quando são totalmente conectadas, cada neurônio é conectado a todos os neurônios da camada seguinte e assim sucessivamente. Quando isso não ocorre, chama-se um rede parcialmente conectada. A grande vantagem de redes de camadas múltiplas é a complexidade dos sistemas que elas são capazes de lidar. Isso se deve ao aumento de capacidade de sinapse oferecido pela camada oculta e a camada de saída. Como o sinal de cada camada oculta depende da camada imediatamente anterior, isso aumenta a capacidade da rede, finalizando na camada de saída. Isso também aumenta a quantidade de parâmetros do modelo.

2.3.2.3 Redes recorrentes

Esta arquitetura de rede se diferencia das anteriores por possuir, pelo menos, um laço de realimentação (HAYKIN, 2001; MÜLLER et al., 1995). Este laço pode ser de realimentação ou auto-realimentação. Realimentação é quando a saída da rede alimenta a entrada de mais de um neurônio. Auto-realimentação é quando a saída de um neurônio é realimentada para sua própria entrada, conforme Haykin (2001), Müller et al. (1995). A Figura (13) apresenta uma rede recorrente sem laços de auto-realimentação.

Uma rede recorrente pode ou não possuir neurônios ocultos. Redes recorrentes, devido a realimentação, possuem a capacidade de um comportamento dinâmico e não-linear, desde que a rede contenha elementos não-lineares, ainda conforme Haykin (2001), Müller et al. (1995). A figura (14) apresenta uma rede recorrente com laços de auto-realimentação.

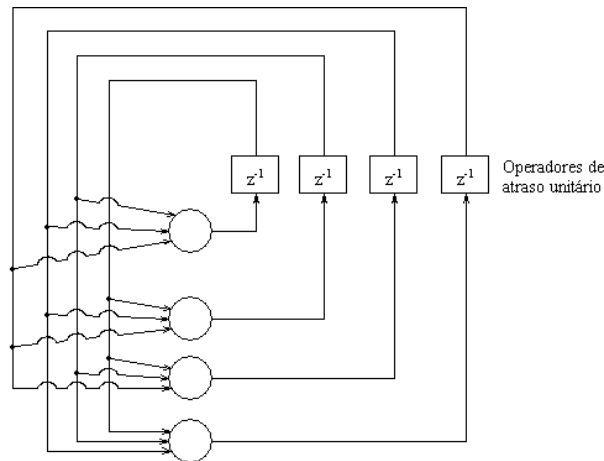


Figura 13: Arquitetura de rede recorrente
Fonte: Adaptado de Haykin (2001)

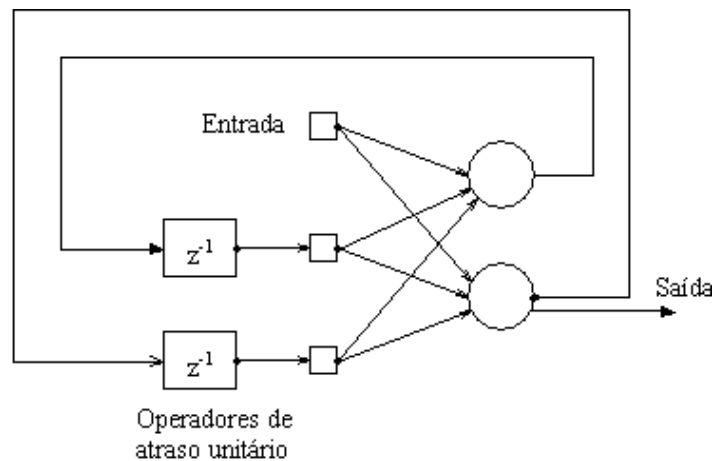


Figura 14: Arquitetura de rede recorrente com camada oculta e laços de auto-realimentação
Fonte: Adaptado de Haykin (2001)

2.3.3 Aprendizagem

Os modelos de RNA são, em sua totalidade, formatados a partir de dois pontos: aprendizagem e treinamento (HAYKIN, 2001; RIPLEY, 1996). Aprendizagem é o processo de ajuste dos pesos sinápticos. Este processo está diretamente ligado tanto ao uso que será dado ao modelo, chamado de tarefa de aprendizagem, quanto ao método de aprendizagem que será utilizado. Este ajuste é feito utilizando-se um treinamento. Desta forma, a aprendizagem é o objetivo ao se treinar uma RNA.

Após determinada a tarefa de aprendizagem, deve ser definido o método a ser utilizado e como ele é aplicado. Existem métodos de aprendizagem específicos para modelos classificatórios, específicos para análise de séries temporais ou mais abrangentes. Definido o método de aprendizagem é necessário se optar entre uma das duas maneiras de aplicá-lo: com ou sem supervisão, que também é chamada de aplicação com ou sem professor, respectivamente. Exis-

tem modelos mais complexos que têm sua aprendizagem dividida em etapas e, em cada etapa, um método de aprendizagem diferente é utilizado, variando, inclusive, como são aplicados. Entretanto, isso não altera a forma com que estes métodos trabalham (CHENG; TITTERINGTON, 1994; MÜLLER et al., 1995; HAYKIN, 2001). A aprendizagem de um modelo de RNA é dividida em três etapas: (i) tarefa de aprendizagem, (ii) método de aprendizagem e (iii) aplicação do método de aprendizagem. A seguir serão apresentadas algumas das tarefas, métodos e suas aplicações.

2.3.3.1 Tarefas de aprendizagem

Quanto as tarefas de aprendizagem, pode-se entender como o objetivo da aprendizagem, ou seja, o objetivo do modelo propriamente dito. Segundo Haykin (2001), estas tarefas são divididas em (i) associação de padrões, (ii) reconhecimento de padrões, (iii) controle, (iv) aproximação de funções, (v) formação de feixe e (vi) filtragem. Outros autores, como Cheng e Titterington (1994) e Müller et al. (1995) não especificam claramente estas divisões, apenas citam diversos usos de modelos RNA. Devido a isso, será utilizada a divisão feita por Haykin (2001).

Quanto a associação de padrões, Haykin (2001) cita que podem ser de dois tipos distintos: auto-associação e heteroassociação. Na auto-associação a rede decide onde alocar os elementos. Neste caso dados completos são apresentados a rede que os classifica em um igual número de classes (padrões). Na heteroassociação, os exemplos apresentados a rede já possuem suas classes definidas, provavelmente em quantidade inferior aos dados. Em ambas, o objetivo do modelo é associar uma classe a um dado incompleto (corrompido). Pode-se perceber que a auto-associação têm seu aprendizado de uma maneira não-supervisionada. Na heteroassociação a aprendizagem é realizada de forma supervisionada.

No reconhecimento de padrões, ao contrário da associação de padrões, os dados apresentados à rede para reconhecimento são sempre completos. Neste caso o treinamento deve ser realizado de maneira supervisionada. Entretanto, apenas uma amostra da população de dados deve ser apresentada, mas todas as classes devem estar presentes. Esta tarefa de aprendizagem segue um método semelhante ao da heteroassociação: os dados devem ser apresentados conjuntamente com a classe (padrão) a qual cada um pertence durante o treinamento, contudo, com um objetivo diferente. O reconhecimento de padrões visa reconhecer um padrão utilizando não um dado corrompido (como na associação de padrões), mas um dado completo. Para que a rede seja capaz de reconhecer um padrão, todos os possíveis padrões devem ter sido apresentados à rede. Para uma população de N elementos, classificada por k padrões existentes, seleciona-se n elementos, onde $k < n$, a, após, treina-se a rede. Quando um elemento da população de

N elementos que ainda não tenha sido classificado é apresentado à rede, ele é então classificado entre um dos k padrões existentes. Este reconhecimento é feito baseado nas informações intrínsecas que a rede captou durante o treinamento. Segundo Haykin (2001), este processo pode ser dividido em duas redes ou uma única rede. No caso de duas redes, uma é feita de maneira não supervisionada (provavelmente com aprendizagem competitiva) para qualificar as informações e uma com aprendizagem supervisionada para finalmente classificar. No caso de uma rede única, os dois processos são realizados pela camada oculta.

A aproximação de funções é um outro uso principal de RNA, ao lado do reconhecimento e associação de padrões (HAYKIN, 2001; DE WILDE, 1997; CHENG; TITTERINGTON, 1994; AUER et al., 2008). Neste caso, um vetor contendo as entradas de uma função desconhecida e um vetor contendo os resultados desta mesma função que são apresentados à rede. A aprendizagem, nesse caso, é costumeiramente supervisionada. A rede, após ter sido treinada adequadamente é então capaz de reproduzir a função. Segundo Haykin (2001), este processo é ideal para ajuste por correção de erro. Isso se deve a construção do algoritmo que verifica o ajuste da função com base nos erros entre os valores gerados pela rede e os valores corretos da função (CHENG; TITTERINGTON, 1994; HAYKIN, 2001; NIKOLAEV; IBA, 2003).

As outras tarefas de aprendizagem são de uso mais restrito. Tarefas de controle estão ligadas a sistemas ou processos que devem ser mantidos em condições estáveis. Pode ser feito tanto com uma aprendizagem supervisionada quanto sem professor. Exemplos e descrições mais detalhadas podem ser vistas em Haykin (2001), Gauri e Chakraborty (2006). A tarefa de filtragem, por sua vez, está associada a extração de informação a partir de dados com ruído ou poluídos. Algumas de suas aplicações são, por exemplo, *data-mining*, suavização e até previsão segundo Haykin (2001) e Bigus (1996). Aqui também a aprendizagem pode ser feita tanto se maneira supervisionada quanto não supervisionada. A formação de feixe é um caso especial da filtragem. Neste caso o tipo de filtro utilizado é espacial (HAYKIN, 2001).

2.3.3.2 Métodos de aprendizagem

O método de aprendizagem é a forma pela qual a rede será habilitada a cumprir com a sua tarefa previamente definida. Cada método de aprendizagem tem sua própria lógica de como será aplicada, efetivamente, a rotina de treinamento. Existem vários métodos, porém os mais utilizados são: aprendizado baseado em memória, correção de erro e de Hebb (HAYKIN, 2001).

O método de aprendizagem baseado em memória assemelha-se a um banco de dados relacional. Um vetor de entradas e saídas é apresentado à rede que memoriza estes resultados. Após esta parte, a rede está pronta para classificar novas entradas. As diferenças entre os algoritmos

desta aprendizagem são relacionados aos critérios utilizados para determinar o quão diferentes são as entradas e como, exatamente, estes exemplos são oferecidos à rede para treinamento. Conforme Hair et al. (2005), talvez o correspondente mais próximo da aprendizagem baseada em memória sejam os métodos estatísticos de classificação como a comparação de médias e análise de agrupamentos (*cluster*). Esta metodologia de aprendizagem está associada a modelos de RNA focados na classificação e análise de agrupamentos dos dados, como modelos de função de base radial (RBF, do original em inglês *Radial Basis Function*).

Outro método de aprendizagem é o por correção de erro. O princípio deste método é minimizar o erro da RNA quando comparado a um valor verdadeiro. Assemelha-se a o método dos mínimos quadrados, entretanto, neste caso, existe um fator de aprendizagem, também chamado de delta de aprendizagem. Segundo Haykin (2001), esta também é chamada de regra de Widrow-Hoff em homenagem a seus criadores. A definição de erro segue a equação (12).

$$e_k(n) = d_k(n) - y_k(n) \quad (12)$$

onde n é o passo iterativo no processo de ajustes dos pesos sinápticos, k é o neurônio de saída, $y_k(n)$ é a resposta estimada (sinal de saída) do neurônio k no passo n , $d_k(n)$ é a resposta desejada (saída-alvo) do neurônio k no passo n e $e_k(n)$ é o erro (sinal de erro) do neurônio k no passo n .

O erro $e_k(n)$ aciona um mecanismo de controle, que aplica uma série de correções ao peso sináptico do neurônio k de forma a reduzir $e_k(n)$, aproximando a resposta estimada $y_k(n)$ ao valor alvo $d_k(n)$. Para isso, utiliza-se a função de custo ou índice de desempenho $E(n)$, definido segundo a equação (13):

$$E(n) = \frac{1}{2} e_k^2(n) \quad (13)$$

Desta forma o sistema continua até que os pesos sinápticos estejam estabilizados. Minimizando esta função custo resulta na regra delta, definida pela equação (14).

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \quad (14)$$

onde $w_{kj}(n)$ é o peso sináptico do neurônio k excitado por um elemento $x_j(n)$ do vetor $x(n)$ no tempo n e η é uma constante positiva que determina a taxa de aprendizado para cada novo passo. Na aprendizagem por correção de erro, é muito importante selecionar uma taxa de aprendizado η que possa assegurar que a rede se torne estável, já que esta taxa determina o desempenho da aprendizagem (HAYKIN, 2001).

O postulado de aprendizado de Hebb, ou aprendizagem de Hebb, ou aprendizagem hebbiana, é o mais antigo método de aprendizagem. O método de Hebb baseia-se no fato de que, se existe uma interação entre um ou mais neurônios, a força destes é modificada. Redes que utilizam a aprendizagem Hebbiana dependem da correlação existente entre os sinais pré-sinápticos e pós-sinápticos de cada neurônio. Se esta correlação é positiva, ou seja, há excitação pré e pós-sináptica, a força do neurônio aumenta. Caso contrário, ela é diminuída. Esta força do neurônio a qual a aprendizagem Hebbiana faz menção é equivalente ao peso sináptico (HAYKIN, 2001; MÜLLER et al., 1995; DE WILDE, 1997; KRÖSE; SMAGT, 1996).

Alguns outros métodos de aprendizagem são a aprendizagem competitiva e a aprendizagem de Boltzmann. A aprendizagem competitiva baseia-se numa competição entre os neurônios, chamada de o vencedor leva tudo, onde um único neurônio se torna funcional ao competir com os demais e facilita detectar eventos estatisticamente significantes (HAYKIN, 2001). Algumas tarefas de aprendizagem utilizam este método. Normalmente este método é utilizado em redes que são treinadas em etapas, como, por exemplo, o reconhecimento de padrões. A aprendizagem de Boltzmann é outro método bem conhecido. Ele se baseia em métodos estocásticos derivados da mecânica estatística.

2.3.3.3 Aplicação dos métodos

Com relação a aplicação dos métodos de aprendizagem, têm-se duas possibilidades: com ou sem supervisão. No método com supervisão, um vetor de treinamento é oferecido a rede, juntamente com as respostas ótimas desejadas. Este procedimento é repetido até que a rede se estabilize, passando então a trabalhar de maneira não supervisionada. Isso é possível na medida que uma certa quantidade de exemplos sejam apresentados, de forma a permitir o uso futuro da rede (previsão, estimação). Normalmente este tipo de aplicação está ligado a aprendizagem por correção de erro (HAYKIN, 2001; CHENG; TITTERINGTON, 1994).

Por outro lado, na aprendizagem sem supervisão, não há exemplos prontos no qual a rede será ensinada, desta forma a rede não pode realizar a aprendizagem baseada em resposta desejadas. De acordo com Haykin (2001), são encontradas duas subdivisões: a aprendizagem por reforço ou programação neurodinâmica e a aprendizagem não supervisionada comum. Um exemplo de aprendizagem não supervisionada é quando se deseja separar os dados em k grupos distintos quando os dados de treinamento não apresentam esta classificação explícita. Neste caso a rede poderia ser instruída a minimizar, ou maximizar, um indicador de eficiência. Este indicador de eficiência costuma ser, em geral, a distância euclidiana (CHENG; TITTERINGTON, 1994; JAKUBEK; STRASSER, 2004). A programação neurodinâmica tem uma aborda-

gem mais temporal, onde se busca minimizar um índice de desempenho baseado nas decisões tomadas pela rede (HAYKIN, 2001).

2.3.4 Treinamento

Para que os métodos e tarefas de aprendizagem abordados anteriormente funcionem adequadamente, é necessário que a rede seja submetida a um treinamento. A diferença entre treinamento e aprendizagem é que o treinamento é a parte aplicada da aprendizagem. A aprendizagem é mais ampla. O treinamento é uma das fases da aprendizagem. O treino de um modelo de RNA é a fase na qual serão ajustados os pesos sinápticos, dados os exemplos a serem apresentados. Tanto a aprendizagem quanto o treinamento são fundamentais para qualquer modelo de RNA (MÜLLER et al., 1995; DE WILDE, 1997; HAYKIN, 2001).

Os principais conceitos relacionados ao treinamento dizem respeito a natureza e a quantidade de dados (KAASTRA; BOYD, 1996; HILL et al., 1996; HAYKIN, 2001; TSAI; LI, 2008). Ambos podem prejudicar a generalização e aplicação do modelo. Assim como ocorre na inferência estatística, o tamanho e a qualidade de uma amostra selecionada tem de estar de acordo com a necessidade do pesquisador. Caso contrário o processo inferencial pode perder seu poder de generalização ou até ser invalidado.

De acordo com Haykin (2001), um conceito fundamental para o treinamento é a dimensão de Vapnik-Chervonenkis, ou dimensão V-C (VAPNIK; CHERVONENKIS, 1971). A dimensão V-C é um parâmetro utilizado para determinar um tamanho de amostra ideal para treinamento. Ainda de acordo com Haykin (2001), na maioria dos casos práticos a dimensão V-C não pode ser obtida analiticamente, mas seus limites sim. Mas, mesmo a dimensão V-C não apresenta um tamanho ideal para treinamento. O problema relacionado a quantidade ideal de dados a serem utilizados é semelhante ao problema da parametrização de modelos estatísticos. O super-treinamento e o sub-treinamento causam, ambos, uma rede sem poder de generalização. O efeito do super-treinamento é pior pois ele só é perceptível quando a generalização do modelo é feita. No caso de séries temporais, quando as previsões são testadas, isso ocorre porque a rede se adapta muito bem a série de treino, mas devido a super-parametrização perde poder de previsão (KAASTRA; BOYD, 1996; HILL et al., 1996; FLORES, 2006).

Ainda segundo Haykin (2001), o treinamento, que pode ser realizado de diversas maneiras, pode ser dividido em dois interesses fundamentais: ou a rede já está definida e é necessário descobrir qual o tamanho ideal da amostra de treinamento para a generalização da rede ou a amostra de treinamento já está definida e é necessário descobrir qual a melhor rede para generalização. Para ambos os casos pode ser utilizada a ferramenta estatística de validação

cruzada (HAIR et al., 2005; MINGOTI, 2005), adaptada ao uso com RNA.

Para o uso da validação cruzada a série de dados é separada aleatoriamente em dois conjuntos principais: conjunto de teste e conjunto de treinamento. O conjunto de treinamento é dividido ainda em dois subconjuntos: subconjunto de estimação e subconjunto de validação. O modelo é então treinado utilizando-se o subconjunto de estimação e testado frente ao subconjunto de validação.

Caso o interesse seja a escolha do melhor modelo, esse é o passo mais importante. Tendo sido selecionado um ou mais modelos, estes então seriam testados no conjunto de teste. Esse passo é importante para evitar séries super-treinadas, que possuem um ótimo ajuste aos dados apresentados mas pouco ou nenhum poder de generalização. Porém, estes passos da validação cruzada, em nenhum momento, indica quais os tamanhos ideais de cada conjunto ou subconjunto.

É importante, porém, que, durante a fase de treinamento, o máximo de informação seja passada à rede (HAYKIN, 2001; DE WILDE, 1997; HILL et al., 1996; QI; ZHANG, 2001). Isso inclui *outliers*, valores que fogem ao comportamento da série. A inclusão dos *outliers* é realizada para que a rede possa obter o máximo de informação implícita aos dados. Sendo assim, em alguns casos é preferível que a seleção dos dados não seja feita de maneira inteiramente aleatória e sim realizada utilizando o conhecimento prévio do pesquisador.

De forma a evitar que o modelo treinado seja adequado apenas a um certo período da série, é recomendável evitar de utilizar apenas os dados mais recentes para verificação de generalização, isto é, os dados devem ser selecionados de forma aleatória tanto para inclusão no conjunto de treinamento quanto para inclusão no conjunto de teste (HAYKIN, 2001; QI; ZHANG, 2001; HILL et al., 1996). Isso evita que modificações recentes na série de dados sejam excluídas da modelagem.

Tsai e Li (2008) deixa claro que isso é de difícil aplicação em séries consideradas pequenas. Nesses casos a aleatorização deve ser utilizada com cuidado ou então a série é treinada com técnicas de *bootstrap* ou reamostragem (CONGDON, 2003). Em estudos de séries temporais é importante que os valores mais recentes da série sejam excluídos destes dois conjuntos, sendo utilizados posteriormente para verificar a capacidade preditiva do modelo.

Outra metodologia que pode ser utilizada é, de acordo com Qi e Zhang (2001), basear o ajuste do modelo com os dados da própria amostra, assim como feito na análise de séries temporais pelos modelos clássicos. Nessa metodologia um modelo é ajustado utilizando-se todos os dados da amostra, exceto por aqueles que serão utilizados para testes de previsão. A escolha dos modelos e da maneira de treino envolve paradas estabelecidas com base em

uma medida de erro aceitável conforme estabelecido pelo pesquisador. Os modelos ajustados são então selecionados com base em medidas de desempenho como AIC (*Akaike Information Criteria*), BIC (*Bayesian Information Criteria*) e o EQM (Erro Quadrático Médio, ou MSE do original em inglês *Mean Square Error*). A técnica a ser usada é semelhante a que foi usada em Flores (2006). Desta forma, uma amostra de treino selecionada aleatoriamente será utilizada e a qualidade dos modelos será mensurada com os critérios AIC, BIC e EQM de ajuste e previsão.

2.3.5 *Perceptrons* de múltiplas camadas

Após ter sido definido como vai ser o treinamento, é necessário optar por algum modelo de RNA. O modelo de RNA utilizado nesta dissertação é o modelo de *perceptrons* de múltiplas camadas. O modelo de *perceptrons* de múltiplas camadas (MLP, do original em inglês, *Multi-layer Perceptron*) é uma evolução do *perceptron* desenvolvido por Rosenblatt. Inicialmente o *perceptron* era capacitado, apenas, para sua forma mais simples de classificação: a de objetos linearmente separáveis. Com o aumento do número de *perceptrons* e as camadas ocultas, estes modelos atualmente são capacitados para as mais diversas funções, desde aproximação de funções (caso de séries temporais) até classificações mais complexas (classificação com poucas informações). Este aumento do número de *perceptrons* também aumentou seu poder computacional e sua capacidade de utilização (MÜLLER et al., 1995; HAYKIN, 2001). A Figura (15) apresenta a arquitetura básica de um MLP: um modelo com três neurônios na camada de entrada, duas camadas ocultas, sendo uma de quatro neurônios e uma de dois neurônios, tendo apenas um neurônio na camada de saída.

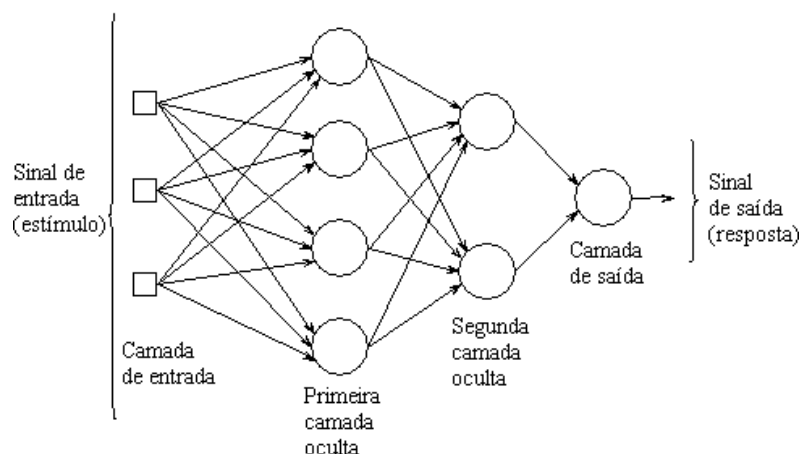


Figura 15: Arquitetura de um modelo MLP 3-4-2-1

Fonte: Adaptado de Haykin (2001)

A rede apresentada na Figura (15) trata de uma rede totalmente conectada, conforme visto anteriormente. As setas representam apenas os sinais de entrada, ou sinais funcionais da rede. Entretanto, existem ainda os fluxos de sinais de erro, que transmitem o valor do erro encontrado, ajustando os pesos sinápticos. A Figura (16) apresenta ambos os fluxos em uma rede.

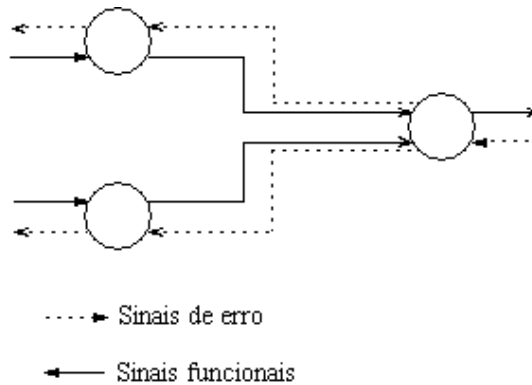


Figura 16: Fluxo de sinais nos neurônios de um modelo MLP
 Fonte: Adaptado de Haykin (2001)

O sinal de entrada parte da camada de entrada, ou camada externa, e vai de camada em camada até, finalmente, a camada de saída, onde o sinal se transforma no resultado da rede, também chamado de sinal de saída. Neste momento a resposta da rede é comparada a uma resposta verdadeira, proveniente dos dados de treinamento. Esta comparação gera um erro, chamado aqui de sinal de erro. Este sinal então segue o caminho inverso do sinal de entrada. Durante este percurso é que os pesos sinápticos são ajustados a fim de minimizar este sinal de erro. Este algoritmo descrito é o algoritmo de retropropagação do erro (do original em inglês, *error back-propagation*). Este algoritmo é semelhante a um método de mínimos quadrados médios (LMS, do original em inglês, *least mean square*). Segundo Haykin (2001), a retropropagação é tida como uma generalização do LMS. Estes modelos são utilizados para os mais diversos fins, dadas as suas capacidades (HILL et al., 1996; FERNANDES et al., 1996; OZCELIK; ERZURUMLU, 2006; WANG; LU, 2006a).

Os modelos de MLP, não raro, costumam ser confundidos como os únicos modelos de RNA, dadas as suas características e propriedades. Quanto as suas limitações, estas são caracterizadas, principalmente, quanto a não interpretação dos modelos (efeito caixa-preta) e também quanto ao seu uso em amostras pequenas. O nome do efeito caixa-preta se deve a relação entre as caixas-pretas usadas na aviação e o suposto desconhecimento dos seus componentes. Assim como as caixas-pretas, as camadas ocultas de um modelo MLP são conhecidas, mas o seu exato funcionamento, supostamente, não. De tal maneira que os modelos não são interpretáveis quanto aos seus parâmetros, os pesos sinápticos. Contudo, o problema da caixa-preta permanece. O foco principal em uma modelagem por MLP é obter previsões e não o conhecimento da

função geradora ou de seus parâmetros. Outro problema associado é a falta de um algoritmo para selecionar o melhor modelo de MLP a ser usado. Alguns artigos tratam do assunto como Qi e Zhang (2001), Rathbun et al. (1997), Wang e Lu (2006a), Balkin e Ord (2000), mas nenhum destes autores trata de um algoritmo geral. Existem algoritmos genéticos que se utilizam de informações obtidas pelos próprios modelos para os evoluir para um modelo tido como ideal. Algumas idéias podem ser vistas Ozcelik e Erzurumlu (2006), Oh e Pedrycz (2006).

3 METODOLOGIA

Este capítulo visa apresentar a metodologia utilizada para comparar os modelos clássicos e os modelos MLP/RNA na previsão de dados em séries temporais não lineares quanto a média. Os resultados foram obtidos seguindo os passos aqui descritos. Em muitas partes, como na modelagem por RNA, são feitos sorteios aleatórios, os quais podem causar problemas na sua reprodução. Contudo, isso será descrito no texto. Um fluxograma será apresentado e descrito detalhadamente, assim como os testes a serem utilizados e suas respectivas formulações e interpretações.

3.1 Estrutura da Metodologia

A metodologia está estruturada conforme o fluxograma apresentado na figura (17). Este fluxograma compreende todos os passos que foram executados para realizar a modelagem e a comparação. É importante citar que, apesar de muitas passos ocorrerem em paralelo, eles serão descritos aqui seqüencialmente, em uma ordem que não representa sua importância, apenas para melhor fluidez do texto.

Para cada um dos passos apresentados no fluxograma será apresentada uma descrição detalhada. O fluxograma é o mesmo para as duas séries de dados utilizadas, a da produção industrial e a série das manchas solares. Com relação aos testes utilizados, a descrição deste passo visa mostrar como eles foram utilizados e para que fim. Quanto a descrição e formulação dos testes, essa será dada no momento oportuno.

3.1.1 Séries de dados temporais

O primeiro passo é obter as séries de dados. Como todos os modelos foram gerados utilizando o pacote estatístico R (R Development Core Team, 2007), é necessária a inclusão ou obtenção dos dados no pacote. Além disso, é necessário apresentar os períodos nos quais foram obtidos os dados.

Com relação a série das manchas solares, esta já consta na biblioteca de banco de dados do programa, além de estar disponível em diversos outros programas devido ao seu extenso uso. Ela funciona como uma espécie de caso-controle, já que muitas de suas características são conhecidas, como sua não linearidade, enquanto o mesmo não pode ser dito da série do índice

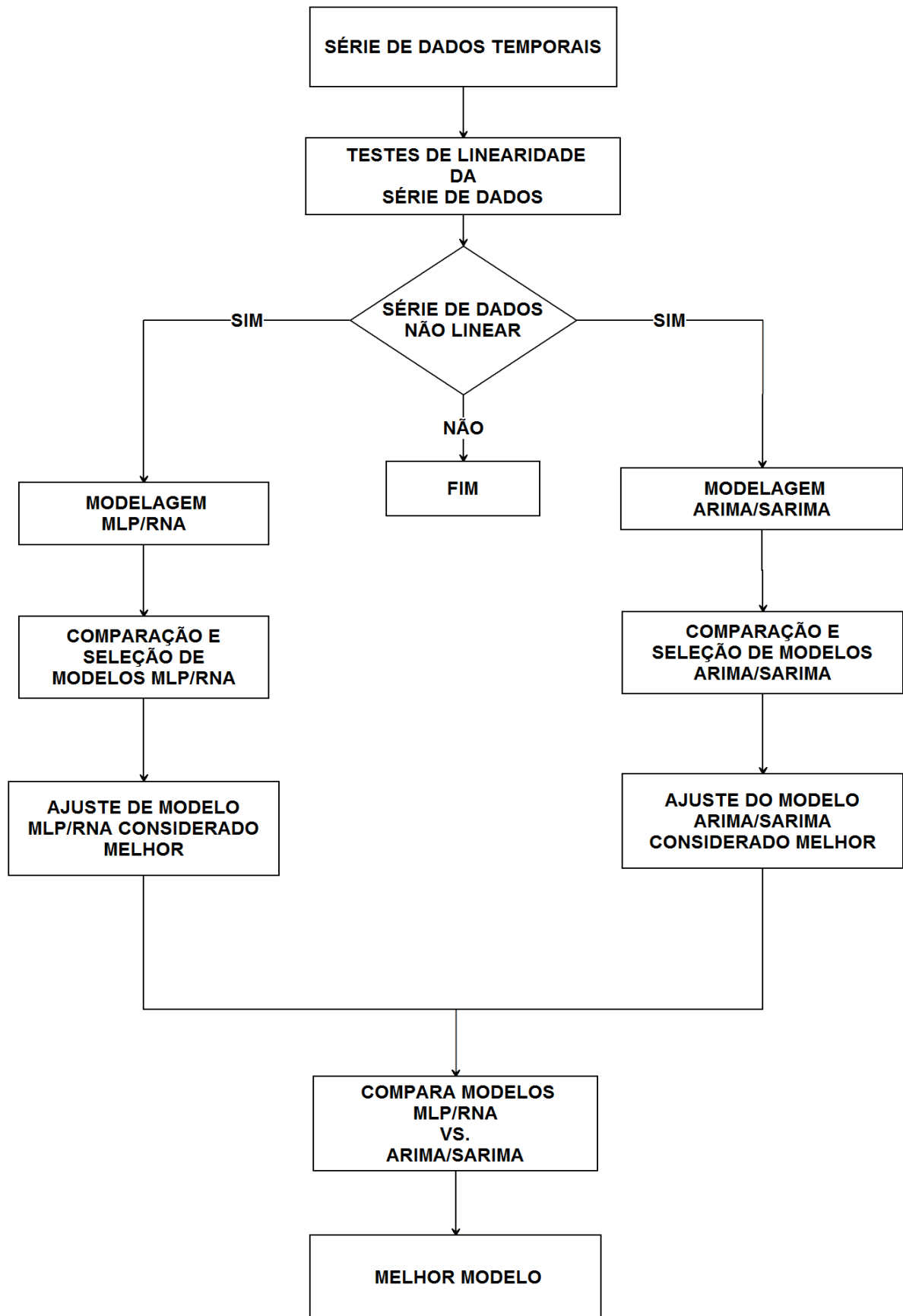


Figura 17: Fluxograma da metodologia utilizada
Fonte: Elaborado pelo autor

de produção industrial.

Quanto a série da produção industrial, ela foi obtida junto ao Instituto Brasileiro de Geografia e Estatística (IBGE) e posteriormente foi introduzida no programa. Esta série será tratada tanto com seus valores absolutos, quanto com o logaritmo de seus resultados, de forma a identificar problemas relativos a uma possível alta variabilidade.

3.1.2 Testes de linearidade da série de dados

Selecionada a série e obtidos os dados, passa-se aos testes de linearidade. Nesta dissertação foram utilizados três diferentes testes de linearidade: o teste de Teräsvirta (TERÄSVIRTA et al., 1993), o teste de White (LEE et al., 1993) e o teste de Tukey (TSAY, 1986).

Cada um destes testes possui diferentes características, mas possuem um ítem em comum: as hipóteses. Para todos os testes, a construção das hipóteses é a mesma, a hipótese nula afirma que a série de dados é linear, enquanto que a hipótese alternativa afirma que a série de dados é não linear. Além disso, os testes de Teräsvirta e White são específicos com relação a não linearidade da média. Desta forma, o modelo é dito não linear quanto a sua média quando a hipótese nula é rejeitada. Quanto ao teste de Tukey, este consiste de um teste mais amplo, captando não só a não linearidade da média.

Tanto o teste de Teräsvirta como o teste de White são semelhantes, também, quanto aos seus parâmetros. Em ambos os testes, além da série de dados, é necessário informar o atraso de tempo, *lag*, no qual se deseja verificar a não linearidade. Por exemplo, quando os testes verificam a não linearidade para um *lag* igual a 1, é porque a série deve ser não linear com relação ao período anterior e o período atual. Nesta dissertação, ambos os testes foram realizados nas duas séries de dados utilizando-se *lags* de 2 até 12.

No teste de Tukey, além da série de dados, é necessário informar uma ordem de não linearidade. Isso porque o teste verifica a não linearidade utilizando um método semelhante a interpolação polinomial. Ele verifica o ajuste de uma interpolação polinomial de primeira ordem com uma interpolação da ordem informada. Caso a interpolação da ordem informada seja melhor, rejeita-se a hipótese de linearidade. Os testes de Tukey foram realizados verificando-se ordens de 2 a 5, que representam séries polinomiais de segunda a quinta ordem. Os resultados dos testes serão apresentados na forma de tabelas, apresentando os resultados obtidos com relação aos p-valores obtidos em cada teste realizado e para cada série testada.

3.1.3 Verificação de linearidade da série de dados

O próximo passo é averiguar os resultados apresentados pelos testes do passo anterior e verificar se a série deve ser considerada não linear ou se pode ser considerada linear. No caso da série apresentar resultados que levem a decisão de tratá-la como uma série não linear, o processo continua. Caso contrário, o processo é encerrado e a série descartada.

Porém, como serão realizados três diferentes testes, além de suas parametrizações, a escolha entre continuar com as análises e encerrar o processo deve ser tomada com certa precaução. O teste de Tukey, que é mais amplo, tem um peso menor na avaliação. Os testes de White e Teräsvirta ficam mais em primeiro plano. Por outro lado, não basta apenas uma parametrização de um dos testes apresentar p-valor significativo, quando todos os outros resultados indicam o contrário. Essa avaliação será discutida juntamente com os resultados a serem apresentados e uma conclusão será tomada. Caso se conclua que a série deve ser considerada não linear, passa-se ao próximo passo: as modelagens.

3.1.4 Modelagem ARIMA/SARIMA

O algoritmo Box-Jenkins vai ser utilizado para realizar as modelagens ARIMA/SARIMA. Desta forma, serão utilizadas a fac e facp das séries e os modelos ajustados de acordo com o algoritmo de Box e Jenkins, apresentado na figura (1). No caso da série de produção, que é uma série pouco analisada, serão obtidos, no mínimo, dois diferentes modelos ARIMA/SARIMA. Conforme apresentado anteriormente, o algoritmo Box-Jenkins permite a construção de mais de um modelo adequado. Por exemplo, é possível obter um modelo $\text{ARIMA}(1,1,3)$ e um $\text{ARIMA}(3,1,1)$, ambos com o mesmo número de parâmetros mas diferentes entre si, sempre priorizando o modelo que seja mais parcimonioso.

Já a série das manchas solares, por se tratar de uma série mais conhecida e analisada, alguns modelos ARIMA/SARIMA já foram apresentados pela literatura. Sendo assim, além dos modelos que serão obtidos por esta dissertação, serão utilizados e apresentados alguns dos modelos presentes na literatura.

Para a obtenção dos modelos, será utilizada apenas uma parte das séries, sendo que os últimos valores serão separados para posterior verificação do poder preditivo dos modelos. Em ambas as modelagens serão utilizados os últimos 12 valores para a verificação das previsões e posterior cálculo do EQM de previsão.

Como o algoritmo já é bem conhecido e por questões de espaço, não serão apresentados todos os passos para cada modelo ajustado. Serão apresentados os gráficos das funções de

autocorrelação e autocorrelação parcial da série antes da modelagem e os mesmos após a escolha do melhor modelo. Durante a etapa de modelagem ARIMA/SARIMA, os modelos serão apenas obtidos e, até aqui, nenhum teste ou estatística é utilizado.

3.1.5 Comparação e seleção de modelos ARIMA/SARIMA

Após a construção dos modelos na etapa anterior, ou obtenção de modelos usados em outros trabalhos, no caso da série das manchas solares, é necessário verificar quais são os que possuem melhores resultados no que tange ao ajustamento do modelo. Nesta etapa são calculados os critérios de comparação dos modelos candidatos a serem possíveis de representar o comportamento da série temporal. Os critérios utilizados nesta etapa são: AIC, BIC e EQM de ajuste. A descrição detalhada e as fórmulas utilizadas serão apresentadas na seção específica dos critérios.

Não serão atribuídos pesos ou alguma ordem de importância para os critérios, pois, dado que o objetivo da modelagem é quanto as previsões do modelo, não há necessidade de priorizar um único critério. Estes critérios foram escolhidos por serem alguns dos mais utilizados.

Como existe a possibilidade de serem muitos os modelos ajustados, serão apresentados, no máximo, os três modelos considerados melhores, segundo os critérios citados. Tendo sido realizada esta parte, passa-se então a seleção de um dos modelos para ajuste. Esta parte, no caso da modelagem ARIMA/SARIMA, se resume apenas a apresentar os resultados da comparação feita anteriormente. As estatísticas incluem, não só os critérios AIC, BIC, EQM de ajuste, como os gráficos da fac e $facp$ do modelo considerado melhor, dados estes critérios.

Nesta etapa também é feita uma explanação dos resultados obtidos, apresentando falhas e vantagens de cada modelo. A comparação feita na etapa anterior serve para filtrar o possível excesso de modelos. Nesta etapa são apresentados e comentados apenas os resultados dos modelos que se destacaram.

3.1.6 Ajuste do modelo ARIMA/SARIMA considerado melhor

Esta etapa serve para uma análise gráfica do modelo selecionado na etapa anterior. Esta análise gráfica é discutida de forma a apresentar tanto os pontos positivos como negativos em relação ao modelo. É nesta etapa que pode-se verificar se o modelo possui algum problema de ajuste que não fora detectado na análise dos resultados dos critérios de comparação.

Feito isso, a próxima etapa é a comparação entre os modelos clássicos da modelagem ARIMA/SARIMA frente aos modelos MLP/RNA. Apesar de serem feitas de maneira para-

lela, as etapas referentes aos modelos MLP/RNA serão descritas a seguir e, posteriormente, será apresentada a etapa de comparação.

3.1.7 Modelagem MLP/RNA

Ao contrário da modelagem ARIMA/SARIMA, a modelagem MLP/RNA não possui um único algoritmo para a construção de modelos. Existem algoritmos como o Algoritmo Genético citado no Referencial Teórico, mas nenhum algoritmo genérico e amplo, como o algoritmo de Box-Jenkins. Devido a essas características, também será apresentado um fluxograma apresentando as etapas realizadas na obtenção dos modelos MLP/RNA. Este fluxograma pode ser visto na figura (18) e foi utilizado da mesma maneira tanto na série da produção industrial quanto na série de manchas solares.

A primeira etapa é separar a série de dados em dois conjuntos de dados. São retirados os últimos 12 valores para verificação do poder preditivo e cálculo do EQM de previsão. O restante é utilizado na modelagem. Têm-se, assim, dois conjuntos de dados: série de previsão, contendo os últimos 12 valores, e a série de modelagem, contendo o restante dos dados, ou padrões de treinamento.

A próxima etapa necessita de aleatorização. Um modelo MLP necessita de dois conjuntos distintos de padrões para ser modelado: conjunto de padrões de treinamento e aprendizagem e conjunto de padrões de ajuste, ou simplesmente conjunto de treinamento e conjunto de ajuste. Para obter estes dois conjuntos, foi utilizada a série de modelagem, obtida na etapa anterior. O percentual utilizado nesta dissertação é de 70% dos padrões da série de modelagem foram selecionados aleatoriamente e constituem-se no conjunto de treino e aprendizagem. Posteriormente a série de modelagem completa será utilizada para verificar o ajuste do modelo.

Feita a divisão da série de modelagem, é necessário treinar o modelo MLP. Aqui é definido o modelo a ser utilizado dentre as limitações impostas nesta dissertação. As configurações utilizadas nesta dissertação quanto a camada de entrada são duas: modelos com 6 neurônios na camada de entrada e modelos com 12 neurônios na camada de entrada. Para cada configuração da camada de entrada, foram modeladas 26 diferentes configurações da camada oculta: desde uma camada oculta com 5 neurônios até uma camada oculta com 30 neurônios. A camada de saída possui apenas um neurônio em todos os modelos. É nesta etapa que os pesos sinápticos, os parâmetros dos modelos MLP, são ajustados. Esta também é uma etapa que necessita de aleatorização, pois os pesos sinápticos iniciais são aleatorizados.

Quanto ao procedimento de sorteio dos padrões de treinamento, o mesmo é feito da seguinte maneira: inicialmente os dados são dispostos em uma matriz com um número de colunas igual

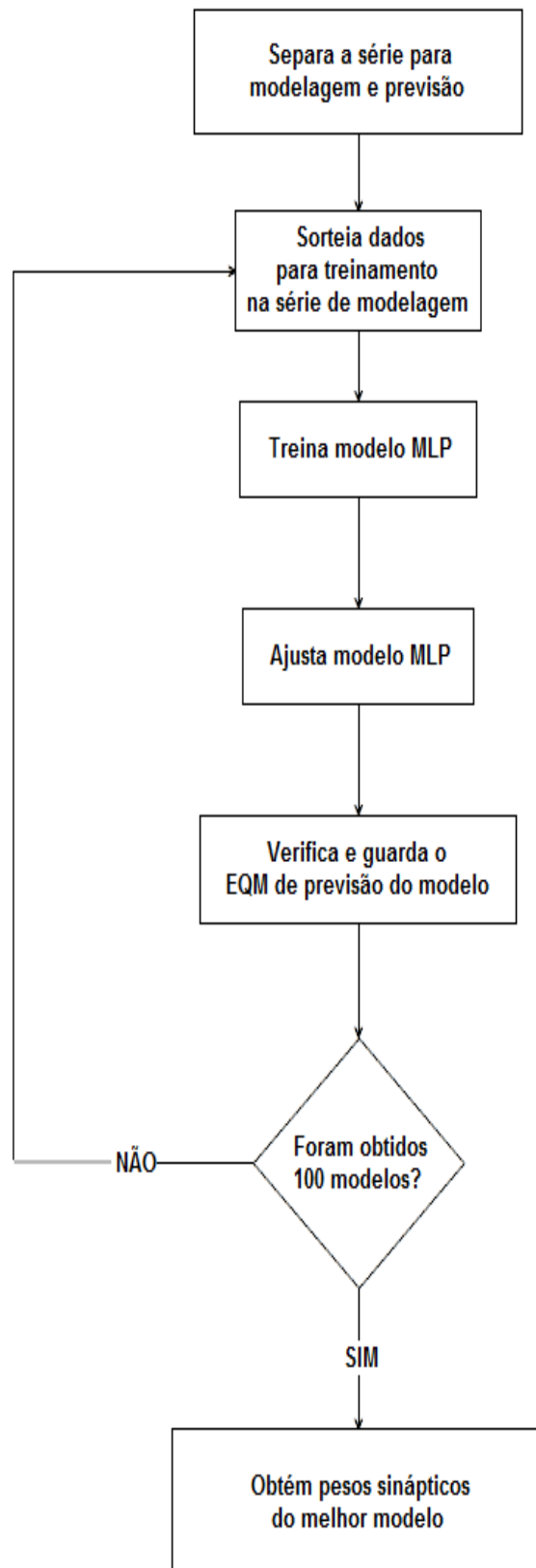


Figura 18: Fluxograma da metodologia utilizada na modelagem MLP/RNA
Fonte: Elaborado pelo autor

ao número de neurônios na camada de entrada mais uma coluna para os valores desejados. Cada linha da matriz representa um padrão de treinamento. E cada linha tem um atraso no tempo, de forma a capturar todos os padrões existentes no conjunto de treino e ajuste, conforme pode ser vista na equação (15).

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{t-6} & x_{t-5} & x_{t-4} & x_{t-3} & x_{t-2} & x_{t-1} & x_t \end{bmatrix} \quad (15)$$

onde x_t representa o último valor da série de dados.

Na matriz da equação (15) os padrões de treinamento estão adequados a um modelo MLP/RNA com 6 neurônios na camada de entrada e apenas 1 na camada de saída, iguais a alguns dos modelos utilizados neste trabalho. O primeiro padrão submete ao modelo os valores de x_1 a x_6 na camada de entrada que tem como valor desejado o valor x_7 . De maneira semelhante ocorre nas demais linhas da matriz. Desta forma, devido a aleatorização dos pesos sinápticos iniciais, os padrões de treinamento também são sorteados, na tentativa de otimizar os resultados modelo.

Com relação a escolha dos modelos, isso se deve a implicações práticas. O número de neurônios na camada de entrada indica o tamanho máximo de ciclo ou sazonalidade que o modelo pode detectar. O modelo de seis neurônios na camada de entrada pode captar uma sazonalidade de período igual ou inferior a seis, enquanto que o modelo com doze neurônios pode captar sazonalidade com período igual ou inferior a doze. A camada oculta é a capacidade de processamento do modelo e foi limitada tanto na sua capacidade inferior quanto superior, respectivamente, em cinco e trinta neurônios.

Feito o treinamento e aprendizagem do modelo, o modelo é então ajustado. Esta etapa serve para verificar o ajuste do modelo a série de modelagem. É a partir destas informações que são obtidos os critérios AIC, BIC e EQM de ajuste. É nesta etapa, também, que pode-se verificar se o modelo treinado está sobre ajustado aos dados ou não. EQM de ajuste muito baixo pode significar um modelo com sobre ajuste, pois indica que o modelo está muito próximo dos dados de treino e aprendizagem, o que costuma prejudicar as previsões. O EQM de ajuste muito alto pode significar um modelo mal treinado, ou com problemas de inicialização nos pesos sinápticos, dependendo da sua magnitude em relação aos dados.

Independentemente do EQM de ajuste obtido, são então verificadas as previsões do modelo. Nesta etapa são previstos 12 valores futuros, com base no modelo treinado. Com os resultados das previsões, é obtido o EQM de previsão. Este EQM de previsão também é apresentado

conjuntamente com os critérios AIC, BIC e EQM de ajuste. Isso deve tanto para auxiliar a escolha de um modelo considerado adequado e também para ressaltar problemas de sobre ajuste. Contudo, cabe uma ressalva. Os modelos MLP/RNA costumam obter EQM de previsão muito baixos em horizontes de previsão menores, como 1 ou 2 passos adiante. As previsões deste trabalho se referem a um horizonte de previsão de 12 passos adiante.

A etapa seguinte trata de um ciclo de repetições. Cada modelo apresentado na etapa de treino e aprendizagem é modelado 100 vezes. Desta forma, desde a seleção da amostra de treino até a verificação e obtenção do EQM de previsão e pesos sinápticos são repetidos 100 vezes. Com isso, espera-se minimizar os efeitos da aleatorização necessária do processo de treino, ajuste e previsão de cada modelo.

Após as etapas necessárias terem sido repetidas 100 vezes, têm-se uma matriz contendo 100 EQM de previsão e 100 configurações de pesos sinápticos. Estes valores serão utilizados para comparação. Será utilizada a configuração de pesos sinápticos do modelo que possuir o menor EQM de previsão, conforme Haykin (2001). Com essa medida, são selecionados os melhores modelos quanto as previsões, seguindo, desta forma, o algoritmo apresentado na figura (17).

3.1.8 Comparação e seleção de modelos MLP/RNA

Concluída a etapa de modelagem, têm-se agora 52 modelos MLP/RNA. São eles: 26 modelos com 6 neurônios na camada de entrada e 26 modelos com 12 neurônios na camada de entrada. Estes 26 modelos obtidos para cada configuração da camada de entrada se referem aos modelos com 5 até 30 neurônios na camada oculta. Desta forma, têm-se os modelos MLP(6,5,1), MLP(6,6,1), MLP(6,7,1), ..., MLP(6,30,1), MLP(12,5,1), MLP(12,6,1), ..., MLP(12,29,1) e MLP(12,30,1).

Para cada um destes modelos tem-se calculados os critérios AIC, BIC e EQM de ajuste e resgatado o EQM de previsão, obtido na etapa de modelagem, além do EQM de previsão no horizonte de 1 passo. Os modelos são então comparados, e os que se destacarem, positiva ou negativamente são filtrados. Esta comparação visa limitar o número de modelos a serem apresentados a um máximo de 6, sendo 3 modelos com 6 neurônios na camada de entrada e 3 modelos com 12 neurônios na camada de entrada. Isso será feito de forma a melhorar a apresentação dos modelos, os quais poderão ser analisados de forma mais ampla. Os resultados dos 52 modelos obtidos para cada série de dados são apresentados no Apêndice B. Aqui também é apresentado o resultado da comparação entre os seis modelos considerados superiores.

O critério do EQM de previsão é priorizado sobre os demais pois, ao contrário dos modelos clássicos, não há suposições iniciais e, portanto, qualquer configuração pode ser tida como

adequada. Feita a comparação dos modelos, os critérios são apresentados para os 6 modelos considerados melhores, sendo, obrigatoriamente, 3 modelos com 6 neurônios na camada de entrada e 3 modelos com 12 neurônios na camada de entrada. Nesta etapa, além dos valores dos critérios utilizados, também é feita uma análise a respeito dos valores dos critérios obtidos pelos modelos. Estes resultados são apresentados por meio de uma tabela contendo o número de neurônios na camada oculta, o AIC, BIC, EQM de ajuste, o EQM de previsão dos 12 passos e o EQM de previsão de 1 passo. Com base na análise destes critérios, um modelo é selecionado e, na próxima etapa, ajustado a série de dados.

3.1.9 Ajuste do modelo MLP/RNA considerado melhor

Realizada a etapa anterior, um modelo é então ajustado a série de dados. Assim como na metodologia utilizada nos modelos ARIMA/SARIMA, aqui também é feita uma análise gráfica a respeito do ajuste. Serão destacados tanto pontos positivos como negativos do modelo tido como melhor com base nos critérios apresentados anteriormente. Nesta etapa é possível verificar possíveis falhas do modelo, as quais não foram detectadas nas análises dos critérios de comparação como, por exemplo, um sobre ajuste.

3.1.10 Comparação de modelos MLP/RNA e modelos ARIMA/SARIMA

Tendo obtido os critérios de comparação de ambas metodologias, os modelos são então comparados. Nesta etapa do algoritmo são apresentados os EQM de previsão, MAPE e NMSE dos modelos pré-selecionados e então comparados entre si. Este passo é bem simples, se resumindo a conclusão baseada nos melhores critérios de previsão, de acordo com o objetivo principal desta dissertação. Este passo é complementado pela apresentação final do modelo escolhido.

No caso específico de modelos MLP/RNA apresentarem menor EQM de previsão, poderá ser feita uma comparação gráfica também. Isso para evitar que um modelo que tenha uma previsão constante, e com EQM de previsão inferior, possa ser escolhido como melhor modelo. O problema da previsão constante pode ocorrer em modelos sobre ajustados e, dependendo dos valores da série, o EQM de previsão pode ser relativamente pequeno quando comparado a outros modelos MLP/RNA.

3.1.11 Melhor modelo

Assim como a etapa anterior, esta também é sintética. Nesta etapa são apresentados os gráficos do melhor modelo ajustado à série de dados utilizada, inclusive com os valores preditos, de acordo com o EQM de previsão. No caso do modelo escolhido ser um modelo clássico, ARIMA/SARIMA, além do gráfico do modelo ajustado à série de dados, também são apresentados os gráficos da função de autocorrelação e função de autocorrelação parcial para os resíduos do modelo.

3.2 Critérios de comparação utilizados

Tendo sido apresentado o fluxograma da metodologia utilizada, passa-se a descrição e formulação dos critérios utilizados nas comparações. Estes foram escolhidos por terem sido utilizados em outros trabalhos e por se tratarem de critérios de construção relativamente simples e de fácil comparação. Porém, antes de serem apresentados os critérios, é necessário definir a formulação da Soma de Erros Quadrados (do original em inglês, *sum square error*, SSE), conforme equação (16).

$$SSE = \sum_{i=1}^t (x_i - \hat{x}_i)^2 \quad (16)$$

onde x_i é o valor verdadeiro, real da série, \hat{x}_i o valor estimado pelo modelo e t é o tamanho da série de dados utilizada para modelagem. Esta estatística é utilizada em todos os critérios utilizados.

O critério de informação de Akaike (AIC) é um dos mais comuns utilizados em análises de séries temporais, principalmente em modelos clássicos, como visto em Qi e Zhang (2001). A formulação utilizada é apresentada na equação (17).

$$AIC = \log\left(\frac{SSE}{t}\right) + \frac{2\log(m)}{t} \quad (17)$$

onde t é o tamanho da série de dados utilizada para modelagem, SSE é a soma dos erros quadráticos, e m o número de parâmetros ajustáveis do modelo.

Assim como o AIC, o critério de informação bayesiano (BIC) é muito utilizado e também pode ser visto em Qi e Zhang (2001). O BIC atribui uma penalização maior ao número de parâmetros do modelo do que o AIC, e deve ser utilizado quando o foco principal é a obtenção de modelos mais enxutos. A formulação utilizada é apresentada na equação (18).

$$BIC = \log\left(\frac{SSE}{t}\right) + \frac{m\log(t)}{t} \quad (18)$$

onde t é o tamanho da série de dados utilizada para modelagem, SSE é a soma dos erros quadráticos, e m o número de parâmetros ajustáveis do modelo.

O outro critério utilizado trata-se do erro quadrático médio (EQM). Nesta dissertação foram obtidos dois tipos diferentes de EQM: o EQM de ajuste, ou simplesmente EQM, e o EQM de previsão. O EQM é um dos mais conhecidos métodos para avaliação de modelos e leva em conta apenas o ajuste aos dados, sem levar em consideração o número de parâmetros. A formulação utilizada é apresentada pela equação (19).

$$EQM = \frac{SSE}{n} \quad (19)$$

onde SSE é a soma dos erros quadráticos, e n o número de elementos utilizados, no caso, a quantidade de valores da série de ajuste.

Ao contrário dos critérios apresentados até o momento, o EQM de previsão não é tão comum. Este critério foi escolhido por possuir as mesmas qualidades do EQM. Isso se deve ao fato de que modelos MLP/RNA costumam possuir um número muito grande parâmetros, o que prejudica os critérios AIC e BIC, conforme visto em (17) e (18), respectivamente. O EQM de previsão segue a mesma formulação apresentada pela equação (19), porém utiliza-se do SSE de previsão e o n passa a ser 12, que é o número de elementos utilizados para a previsão. O EQM de previsão de 1 passo segue a mesma formulação, porém, a cada previsão é utilizado o último valor verdadeiro da série. O SSE de previsão foi calculado conforme a equação (20).

$$SSE_{prev} = \sum_{i=1}^{12} (x_i - \hat{x}_i)^2 \quad (20)$$

onde \hat{x}_i é o valor estimado pelo modelo e x_i o valor verdadeiro da série.

Os critérios MAPE (do original em inglês, *mean absolute percentage error*) e NMSE (do original em inglês, *normalized mean square error*) também são utilizados, porém apenas na etapa de comparação dos modelos ARIMA/SARIMA com os modelos MLP/RNA. Estes critérios foram obtidos tanto de forma a permitir comparações frente a outros trabalhos e também devido ao seu uso freqüente em trabalhos como Hill et al. (1996), Chen (2006). A equação (21) apresenta o MAPE, enquanto a equação (22) apresenta o NMSE.

$$MAPE = \frac{\sum_{i=1}^{12} \frac{|x_i - \hat{x}_i|}{x_i}}{12} 100\% \quad (21)$$

onde \hat{x}_i é o valor estimado pelo modelo e x_i o valor verdadeiro da série.

$$NMSE = \frac{1}{\delta^2} \sum_{i=1}^{12} (x_i - \hat{x}_i)^2 \quad (22)$$

onde \hat{x}_i é o valor estimado pelo modelo, x_i o valor verdadeiro da série e δ^2 é descrito pela equação (23).

$$\delta^2 = \frac{1}{11} \sum_{i=1}^{12} (x_i - \bar{x})^2 \quad (23)$$

onde x_i é o valor verdadeiro da série e \bar{x} a média dos valores verdadeiros.

Para se obter as previsões, tanto dos modelos Box-Jenkins quanto dos modelos MLP/RNA, o procedimento é obter previsões em um horizonte de 12 passos adiante. Porém isso pode ser realizado de, pelo menos, duas maneiras distintas. Por exemplo, supondo um modelo MLP com 6 neurônios na camada de entrada e um na camada de saída, como visto anteriormente. A equação (24) representa uma destas maneiras, enquanto a equação (25) representa a maneira que foi utilizada neste trabalho.

$$\begin{bmatrix} x_{t-5} & x_{t-4} & x_{t-3} & x_{t-2} & x_{t-1} & x_t & \hat{x}_{t+1} \\ x_{t-4} & x_{t-3} & x_{t-2} & x_{t-1} & x_t & x_{t+1} & \hat{x}_{t+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{t+6} & x_{t+7} & x_{t+8} & x_{t+9} & x_{t+10} & x_{t+11} & \hat{x}_{t+12} \end{bmatrix} \quad (24)$$

onde x_t representa o último valor da série de dados e \hat{x}_{t+i} , para $i = 1, 2, \dots, 12$ os valores preditos pelo modelo.

$$\begin{bmatrix} x_{t-5} & x_{t-4} & x_{t-3} & x_{t-2} & x_{t-1} & x_t & \hat{x}_{t+1} \\ x_{t-4} & x_{t-3} & x_{t-2} & x_{t-1} & x_t & \hat{x}_{t+1} & \hat{x}_{t+2} \\ x_{t-3} & x_{t-2} & x_{t-1} & x_t & \hat{x}_{t+1} & \hat{x}_{t+2} & \hat{x}_{t+3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{x}_{t+6} & \hat{x}_{t+7} & \hat{x}_{t+8} & \hat{x}_{t+9} & \hat{x}_{t+10} & \hat{x}_{t+11} & \hat{x}_{t+12} \end{bmatrix} \quad (25)$$

onde x_t representa o último valor da série de dados e \hat{x}_{t+i} , para $i = 1, 2, \dots, 12$ os valores preditos pelo modelo.

Tanto na equação (24) quanto na equação (25) são obtidos os 12 valores preditos, representados pela última coluna de ambas as matrizes. Porém, na equação (24) os valores são obtidos apenas 1 passo a frente, de tal forma que, para prever o valor de x_{t+2} , por exemplo, são utilizados os valores verdadeiros da série até o momento imediatamente anterior, no caso, os valores de x_{t-4} até x_{t+1} . As previsões geradas pelo exemplo da equação (24) não são de interesse

deste trabalho, apesar de serem apresentados os EQM de previsão a 1 passo para os modelos MLP/RNA para posterior discussão.

A forma de obter as previsões utilizada neste trabalho segue o exemplo apresentado na equação (25). Neste exemplo, supõe-se que os valores futuros não são conhecidos, sendo estimados passo a passo e utilizando estas estimativas para prever os próximos. Usando o mesmo exemplo anterior, para prever o valor de x_{t+2} , são utilizados os valores verdadeiros de x_{t-4} até x_t e mais o valor predito no passo anterior, \hat{x}_{t+1} . De forma semelhante ocorre nos modelos Box-Jenkins, sendo utilizados os dados de acordo com as especificações do modelo. Por exemplo, um modelo AR(4) necessitaria dos últimos 4 valores, ao invés dos 6 últimos, conforme apresentado.

4 RESULTADOS

Este capítulo visa descrever e apresentar os resultados obtidos após a análise das duas séries de dados, índice de produção física e manchas solares, com base na metodologia apresentada no Capítulo 3. A série do índice de produção física, por se tratar de uma série ainda não trabalhada, será analisada tanto com seus valores absolutos quanto com o logaritmo dos valores, de forma a enriquecer a discussão dos resultados obtidos frente a outros trabalhos. A série das manchas solares, por sua vez, se trata de uma série conhecida e trabalhada na literatura, com características conhecidas, como, por exemplo, sua não linearidade.

4.1 Descrição das séries

Primeiramente é preciso conhecer as séries a serem trabalhadas, sendo assim estas séries de dados serão devidamente descritas. Nesta descrição também são apresentados os gráficos das séries sem nenhum dado retirado, buscando uma visão inicial da linearidade, contudo a sua verificação através dos testes de linearidade serão apresentados no passo seguinte.

4.1.1 Série do índice de produção física industrial mensal do RS

A série do índice de produção física industrial mensal para o Estado do Rio Grande do Sul foi obtida junto ao Instituto Brasileiro de Geografia e Estatística (IBGE) e corresponde aos dados primários da Pesquisa Industrial Mensal de Produção Física (PIM-PF). Segundo notas metodológicas do IBGE, o índice é obtido através de uma amostra intencional que representa cerca de 62% do valor da Produção Censo Industrial de 1985 e abrange 944 produtos e 6.200 empresas, obtendo um total de 13.000 informações mensais, desde janeiro de 1991. Para o cálculo do índice é utilizada uma versão adaptada de Laspeyres (base fixa em cadeia, com atualização de pesos).

O índice utilizado é o Índice de Base Fixa Mensal, que compara a produção do mês de referência do índice com a média mensal produzida no ano de 2002. Esta série foi escolhida por se tratar de um índice utilizado por diversos órgãos para apresentar um retrato industrial de sua região.

O interesse quanto a previsão do índice é oferecer uma ferramenta que auxilie na tomada de decisões que dependem deste índice. Por se tratar do índice que representa a posição industrial

atual (seja ela de crescimento ou decrescimento), a sua previsão pode auxiliar a prever certos cenários e, com isso, agilizar medidas cabíveis a estes cenários. Contudo, pode ser inadequado utilizar apenas o comportamento passado do índice, haja visto que diversos fatores externos (como crises internacionais) podem afetar a produção industrial. Porém, isso foge ao objetivo deste trabalho. A figura (19) apresenta a série obtida, sem nenhuma modificação ou valor extraído.

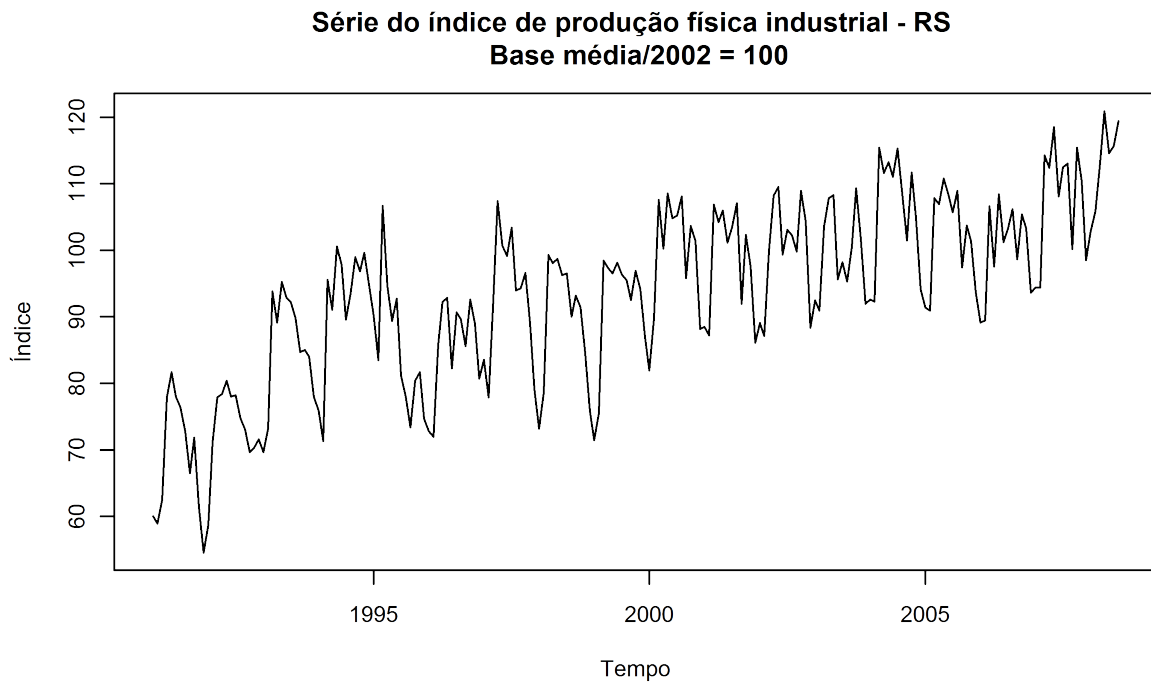


Figura 19: Série do índice de produção física industrial mensal do RS - base média/2002 = 100
Fonte: Instituto Brasileiro de Geografia e Estatística - IBGE

Analisando-se o gráfico da figura (19) pode-se perceber uma tendência de crescimento, o que indica uma série não estacionária. Esta tendência, porém, se assemelha a um crescimento maior até, aproximadamente, o ano de 1995 e então parece diminuir a taxa de crescimento, como uma curva logarítmica. Também é possível observar uma variabilidade maior até o ano de 2000, sendo posteriormente mais homogênea. Tanto a característica de tendência quanto da variabilidade podem representar uma série não linear. Além disso, devido as mudanças aparentes na variância, a aplicação do logaritmo aos resultados pode destacar estas características, o que é apresentado pelo gráfico da figura (20).

Apesar de sua semelhança com o gráfico da figura (19), no gráfico da figura (20), a tendência aparentemente não linear é mais destacada, assim como as diferenças com relação a variância, que ocorre em torno do ano de 2000.

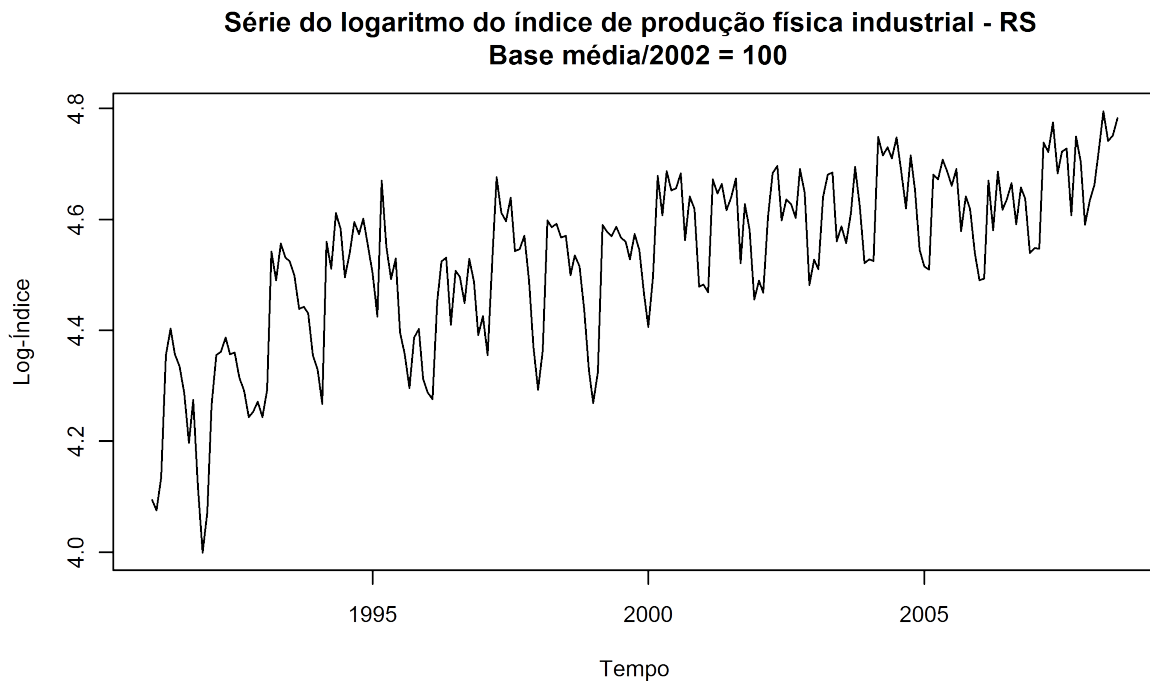


Figura 20: Série do logaritmo do índice de produção física industrial mensal do RS - base média/2002 = 100

Fonte: Instituto Brasileiro de Geografia e Estatística - IBGE

4.1.2 Série do número de manchas solares

A série do número de manchas solares, de período anual, é razoavelmente conhecida e foi utilizada em diversos trabalhos, como Hill et al. (1996), Lai e Wong (2001), Hamzaçebi et al. (2009). Do ponto de vista teórico, esta série apresenta fatores interessantes, como um ciclo aproximado de 11,3 anos, segundo (PRIESTLEY, 1991). Esta série está sendo utilizada de forma a comparar as duas diferentes metodologias utilizando-se não só uma série conhecida, como também já extensamente trabalhada, e com algumas propriedades bem conhecidas, como sua não linearidade, conforme Priestley (1991), Terui e van Dijk (2002). A figura (21) apresenta a série completa, sem nenhuma modificação ou valor extraído.

Analisando-se o gráfico da figura (21) pode-se notar, principalmente nos valores de picos, que a série possui uma variância não constante. Por se tratar de uma contagem, não são possíveis valores inferiores a zero o que, conjuntamente com os valores de picos, pode afetar a média, tornando-a, também, não constante. A série não aparenta nenhum tipo de tendência, assim como não são facilmente notados os ciclos descritos por Priestley (1991).

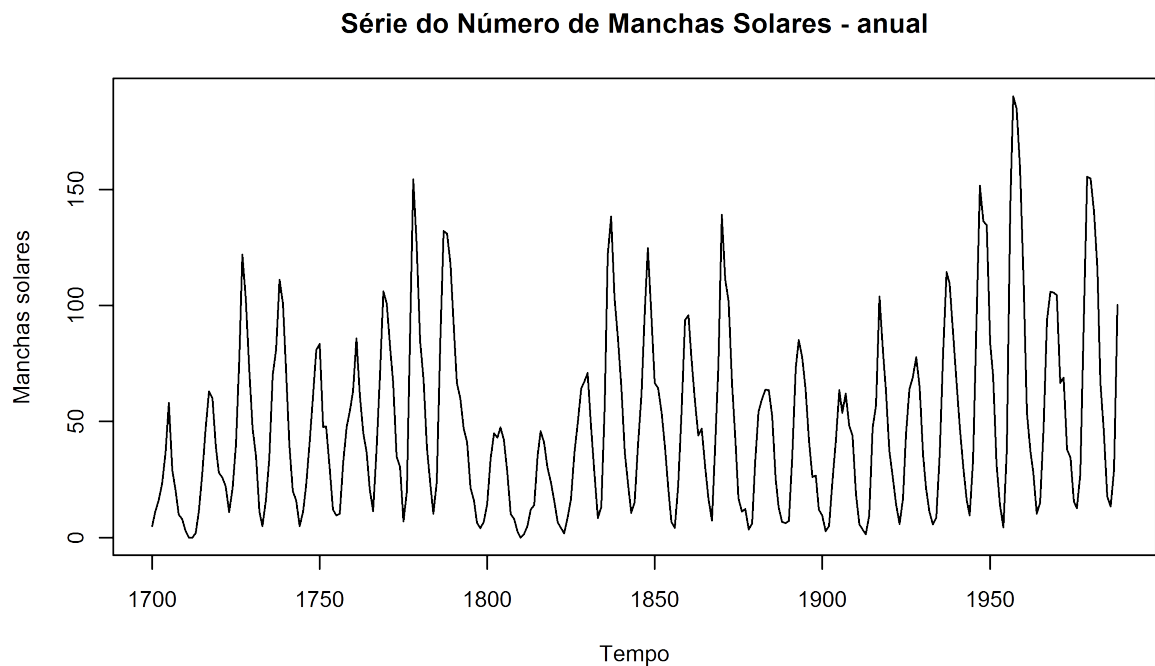


Figura 21: Série do número de manchas solares - periodicidade anual
 Fonte: pacote estatístico R (R Development Core Team, 2007)

4.2 Testes de verificação da linearidade das séries

Inicialmente serão realizados os testes quanto a linearidade das séries. Para isso, serão utilizados os testes descritos anteriormente. A tabela (1) apresenta os resultados dos testes de Tukey, enquanto a tabela (2) apresenta os resultados dos testes de White e Teräsvirta. Estas tabelas são apresentadas de forma a oferecer outras razões ao uso de modelos Box-Jenkins, que têm como, entre outras suposições, a de linearidade dos dados. Porém é importante deixar claro que a série do índice de produção física foi, primeiramente diferenciada (tomada a primeira diferença) de forma a tornar a série estacionária.

A série do índice de produção física, por se tratar de uma série não tanto trabalhada quanto a série das manchas solares, será tratada de duas formas. Como já explanado anteriormente, isso se deve a uma possível alta variabilidade encontrada na série do índice de produção física, podendo afetar os testes de linearidade. Esta possível alta variabilidade pode mascarar efeitos não lineares, fazendo com que os testes apresentem resultados que indiquem uma série não linear. Isso também poderá ser visto nos resultados apresentados pelas tabelas (1) e (2).

Os resultados que constam nas tabelas (1) e (2) apresentam resultados importantes sobre as duas séries. Analisando-se exclusivamente cada série de dados, pode-se perceber que, no caso da série do índice de produção física, o teste de Tukey apresentou como significativas

Teste de Tukey				
p-valor do teste nas séries de dados				
Série de dados	Ordem = 2	Ordem = 3	Ordem = 4	Ordem = 5
Índice de produção - log	0.6926	0.0519	<0.0001	<0.0001
Índice de produção	0.9101	0.2494	<0.0001	<0.0001
Manchas solares	<0.0001	<0.0001	<0.0001	<0.0001

Tabela 1: Resultados da aplicação do teste de Tukey nas séries de dados para diferentes ordens
Fonte: Elaborado pelo autor

Testes de Teräsvirta e White						
Séries de dados						
<i>lags</i>	Índice de produção - log		Índice de produção		Manchas solares	
	Teräsvirta	White	Teräsvirta	White	Teräsvirta	White
2	<0.0001	<0.0001	0.0485	0.8495	0.1917	0.9727
3	<0.0001	<0.0001	<0.0001	0.0521	0.0002	0.9108
4	<0.0001	<0.0001	<0.0001	0.9820	<0.0001	0.0004
5	<0.0001	<0.0001	<0.0001	0.5931	<0.0001	0.0932
6	<0.0001	<0.0001	<0.0001	0.1303	0.0004	0.4297
7	<0.0001	0.0182	0.0002	0.6174	0.0005	0.0001
8	<0.0001	0.0341	0.0017	0.3262	0.0010	0.7798
9	<0.0001	0.0213	NA ²	0.2974	NA ²	<0.0001
10	<0.0001	0.2592	NA ²	0.0992	NA ²	0.0071
11	NA ²	0.7711	NA ²	0.2499	NA ²	0.7552
12	NA ²	0.0560	NA ²	0.0909	NA ²	0.1559

Tabela 2: Resultados da aplicação dos testes de Teräsvirta e White nas séries de dados para diferentes *lags*

Fonte: Elaborado pelo autor

apenas as ordens 4 e 5, com p-valores inferiores a 0.0001, o que indica que a série deve ser não linear. Contudo, estes resultados podem ser devido a variabilidade dos dados. O que reforça esta hipótese são os resultados para a série do índice de produção física do teste de White. Observando-se os p-valores obtidos na tabela (2), nenhum destes valores foi inferior a 0.05. O que leva a crer que a série do índice de produção física possa ser linear, para qualquer *lag* utilizado. Porém, os resultados do teste de Teräsvirta indicam, para todos os *lags* em que o cálculo foi possível, que a série deve ser não linear. Como os testes de Teräsvirta e White possuem a mesma importância, e o teste de Tukey apresentou uma possível não linearidade, a série será tratada como uma série não linear.

Com relação aos resultados obtidos pela série do log-índice de produção, a hipótese de que uma possível alta variabilidade poderia afetar os testes pode ser vista nos resultados, principal-

²O cálculo do p-valor não foi possível.

mente, da tabela (2). Os resultados obtidos no teste de Tukey (tabela (1)), em ambas as séries, não apresentam conclusões distintas; a série do log-índice de produção também deve ser não linear nas ordens 4 e 5, com p-valores inferiores a 0.0001. Porém, nos resultados referentes ao teste de White, são muitas as diferenças. Com exceção dos *lags* 10, 11 e 12, todos os demais possuem valores estatisticamente significativos a um nível de significância de 0.05. O teste de Teräsvirta, por sua vez, possui todos os p-valores, dos *lags* que foram calculados, inferiores a 0.0001. Desta forma, conclui-se que a série do log-índice deve ser não linear.

A série das manchas solares obteve alguns resultados que não eram esperados, principalmente no teste de White, na tabela (2). Ela foi considerada uma série estatisticamente não linear apenas nos *lags* 4, 7, 9 e 10, tendo os p-valores, respectivamente, iguais a 0.0004, 0.0001, inferior a 0.0001 e 0.0071. Apesar de os resultados serem aparentemente suficientes para aceitar a série como não linear, eram esperados resultados mais próximos do que ocorreu com a série do log-índice de produção. O teste de Teräsvirta obteve uma aceitação quanto a linearidade da série apenas no *lag* 2, obtendo um p-valor igual a 0.1917. Os p-valores obtidos no teste de Tukey, na tabela (1), foram todos inferiores a 0.0001. Sendo assim, o que já era esperado, a série de manchas solares será tratada como uma série estatisticamente não linear.

4.3 Aplicação da metodologia à série do índice de produção

4.3.1 Modelagem ARIMA/SARIMA da série do índice de produção

A série do índice de produção física, analisando-se a figura (19), apresenta um comportamento não estacionário, com variância não homogênea e uma possível sazonalidade, conforme análise anterior. Desta forma, inicialmente são realizados os gráficos das *fac* e *facp* dos dados, para identificação dos modelos. As figuras (22) e (23) apresentam, respectivamente, a *fac* e *facp* da série de ajuste do índice de produção.

Observando-se as figuras (22) e (23), pode-se observar o componente sazonal, além do lento decréscimo, que condiciona uma série não estacionária. Também analisando-se os gráficos, a *facp* apresenta dois valores significativos, no *lag* no mês 1 e no 13, considerando-se a escala apresentada, em anos. Já a *fac*, devido a não estacionaridade, não é possível fazer nenhuma afirmação quanto aos *lags* significativos.

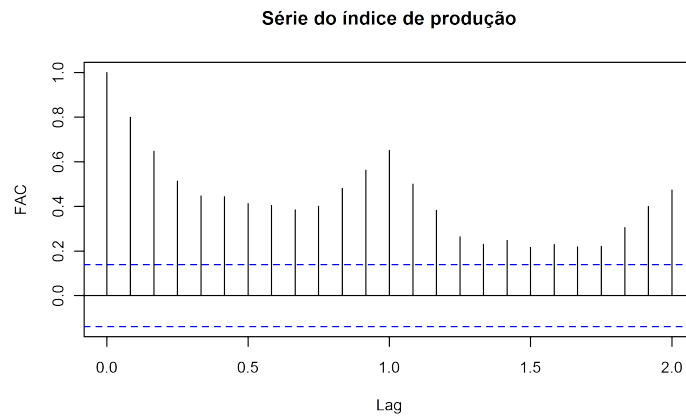


Figura 22: fac da série do índice de produção física - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

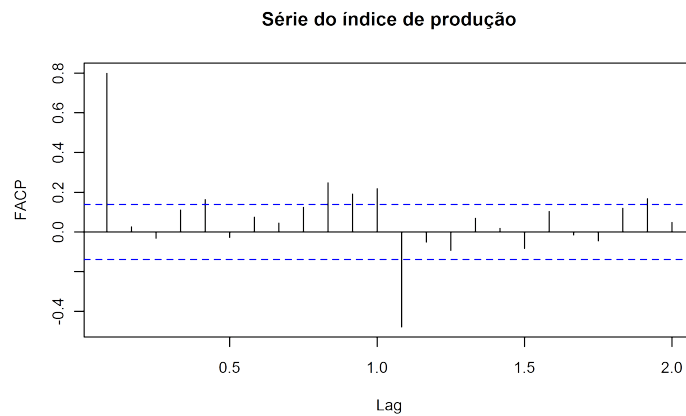


Figura 23: facp da série do índice de produção física - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

O próximo passo é analisar a série diferenciada. A partir deste ponto, os gráficos da fac e facp passam a auxiliar na construção dos modelos. Conforme visto nos capítulos 2 e 3, respectivamente, o algoritmo Box-Jenkins e a metodologia utilizada, foram obtidos 3 modelos considerados adequados a esta série de dados. Estes modelos, assim como seus critérios podem ser vistos na tabela (3).

Critérios dos modelos Box-Jenkins para o índice de produção			
Modelo	AIC	BIC	EQM
SARIMA(5,1,0)×(2,0,1) ₆	3,22	3,39	24,72
SARIMA(2,1,3)×(2,1,2) ₁₂	2,84	3,03	17,94
SARIMA(5,1,2)×(2,1,1) ₁₂	2,82	3,04	17,52

Tabela 3: Resultados dos critérios de modelagem nos modelos Box-Jenkins na série do índice de produção

Fonte: Elaborado pelo autor

4.3.2 Comparação dos modelos ARIMA/SARIMA na série do índice de produção

Com base na tabela (3), o modelo que apresenta os critérios de melhor ajuste é o modelo SARIMA(5, 1, 2)x(2, 1, 1)₁₂, pois, dentre os critérios apresentados, possui o menor AIC, sendo este igual a 2,82, o menor EQM, igual a 17,52 e o segundo menor BIC, sendo este igual a 3,04, e mesmo assim com uma pequena diferença em relação ao modelo com melhor BIC, o modelo SARIMA(2, 1, 3)x(2, 1, 2)₁₂, com um BIC igual a 3,03.

4.3.3 Ajuste do modelo ARIMA/SARIMA considerado melhor

As figuras (24) e (25) apresentam a fac e facp dos resíduos do modelo. Além disso, a figura (26) apresenta o ajuste do modelo a série de dados de ajuste, sem incluir as previsões.

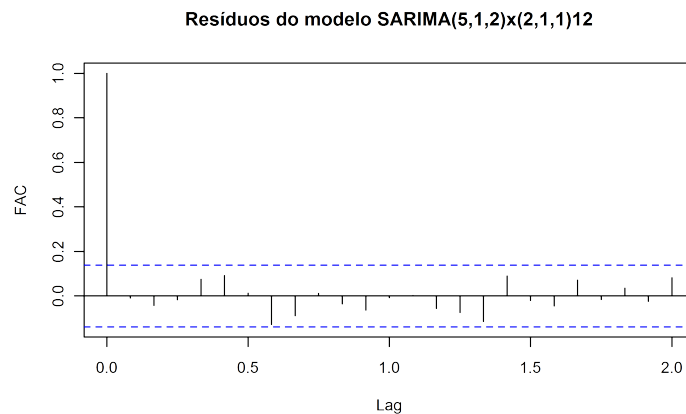


Figura 24: fac dos resíduos do modelo SARIMA(5, 1, 2)x(2, 1, 1)₁₂ - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

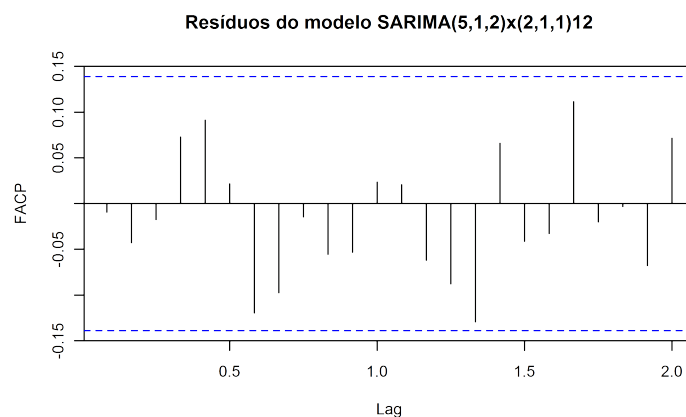


Figura 25: facp dos resíduos do modelo SARIMA(5, 1, 2)x(2, 1, 1)₁₂ - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

Observando-se os gráficos das figuras (24) e (25), pode-se perceber que nenhum valor dos

resíduos é estatisticamente significativo, não sendo, portanto, necessária a análise por testes estatísticos. Desta forma, pode-se concluir que o modelo adequa-se aos dados.

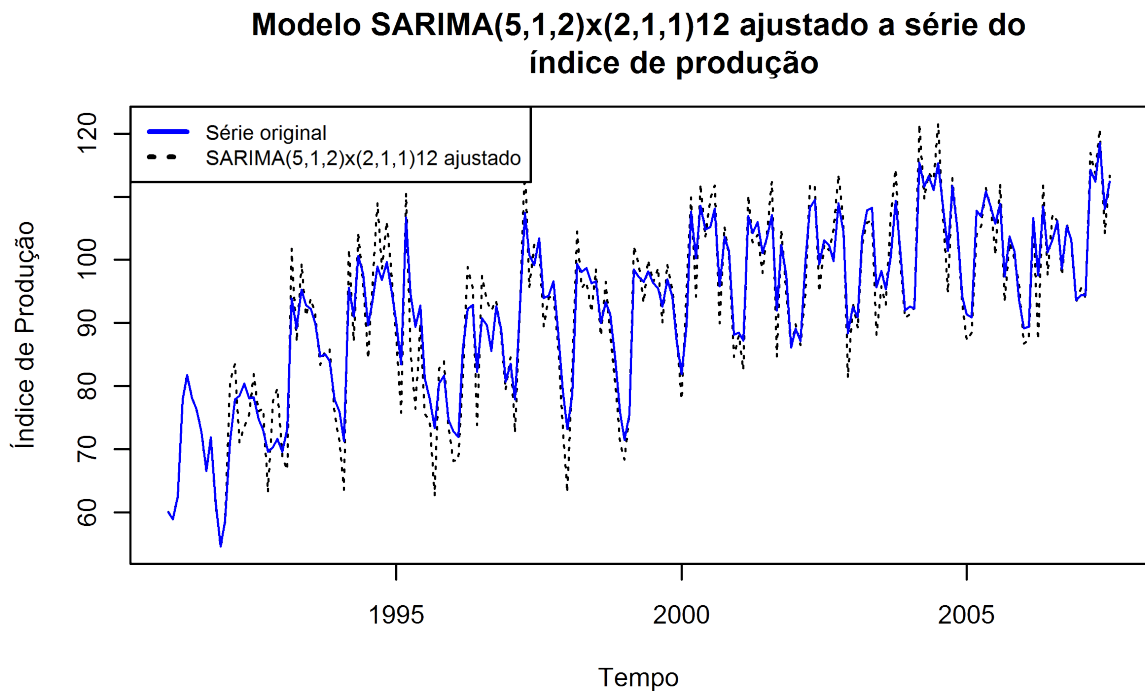


Figura 26: Gráfico da série do índice de produção e do modelo SARIMA(5, 1, 2)x(2, 1, 1)₁₂ ajustado
Fonte: Elaborado pelo autor

O gráfico apresentado na figura (26) apresenta o quão bem o modelo se ajusta aos dados. Como este gráfico não apresenta as previsões do modelo, não é possível verificar se o modelo está, ou não, sobre ajustado. Observa-se também, no início da série, que o modelo não consegue se ajustar bem aos dados. Aparentemente, também, o modelo apresenta dificuldades em prever os valor de pico e de vale em praticamente toda a série de dados.

4.3.4 Modelagem MLP/RNA na série do índice de produção

Modelos de RNA são considerados eficientes na análise de funções não lineares. Sendo assim, não é necessário que a série seja adaptada para análise, ou seja, a série de ajuste será usada normalmente tanto com os valores absolutos quanto no seu logaritmo, sem que seja necessário tornar a série não estacionária. Espera-se que os modelos MLP/RNA sejam capazes de modelar, inclusive, a não estacionaridade.

Seguindo a metodologia utilizada na obtenção de modelos MLP/RNA da figura (18), a tabela (4) apresenta os modelos destacados na configuração de 6 neurônios de entrada e seus critérios. A tabela (5), por sua vez, apresenta as mesmas informações, mas para os modelos com 12 neurônios na camada de entrada.

Critérios dos modelos MLP/RNA com 6 neurônios na camada de entrada					
Camada oculta	AIC	BIC	EQM Ajuste	EQM Previsão(12)	EQM Previsão(1)
10	2,89	5,05	40,41	58,98	82,10
18	3,61	7,52	50,29	80,31	80,98
28	3,40	9,48	46,84	67,18	77,21

Tabela 4: Resultados dos critérios nos modelos MLP/RNA com 6 neurônios na camada de entrada
Fonte: Elaborado pelo autor

Critérios dos modelos MLP/RNA com 12 neurônios na camada de entrada					
Camada oculta	AIC	BIC	EQM Ajuste	EQM Previsão(12)	EQM Previsão(1)
5	2,90	4,85	30,57	35,11	39,38
24	3,15	12,52	35,09	38,74	33,49
26	3,07	13,22	34,64	30,87	35,73

Tabela 5: Resultados dos critérios nos modelos MLP/RNA com 12 neurônios na camada de entrada
Fonte: Elaborado pelo autor

4.3.5 Comparação dos modelos MLP/RNA na série do índice de produção

Observando os resultados apresentados nas tabelas (4) e (5), pode-se notar uma melhor adequação dos modelos com 12 neurônios na camada de entrada. Apesar do critério BIC, o erro de previsão de modelos como o MLP(12,26,1) é inferior ao modelo MLP(12,5,1), respectivamente igual a 30,87 e 35,11. Porém, com relação ao ajuste, o modelo MLP(12,5,1) obteve um resultado melhor, de 30,57. Isso é um indicativo de um modelo com sobre ajuste, o que prejudicou seu poder preditivo. Apesar disso, quanto a comparação, o melhor modelo ficou sendo o MLP(12,5,1) que, de maneira geral, obteve melhores resultados.

4.3.6 Ajuste do modelo MLP/RNA considerado melhor

A figura (27) apresenta o modelo considerado melhor dentre os obtidos, ou seja, o modelo MLP(12,5,1), ajustado a série de validação e treino, sem incluir, obviamente, as previsões. Observa-se no gráfico da figura (27) que o modelo aparenta suavizar os dados. De maneira geral, o modelo aparenta se comportar de maneira adequada, falhando apenas nos pontos finais da série. Analisando-se apenas graficamente, o modelo parece deslocado dos dados do início e fim da série. Contudo, este modelo aparenta um ajuste melhor aos dados do que o modelo vista na figura (26). Porém, são necessárias as avaliações a respeito dos critérios utilizados com relação as previsões do modelo, o que será visto no próximo tópico.

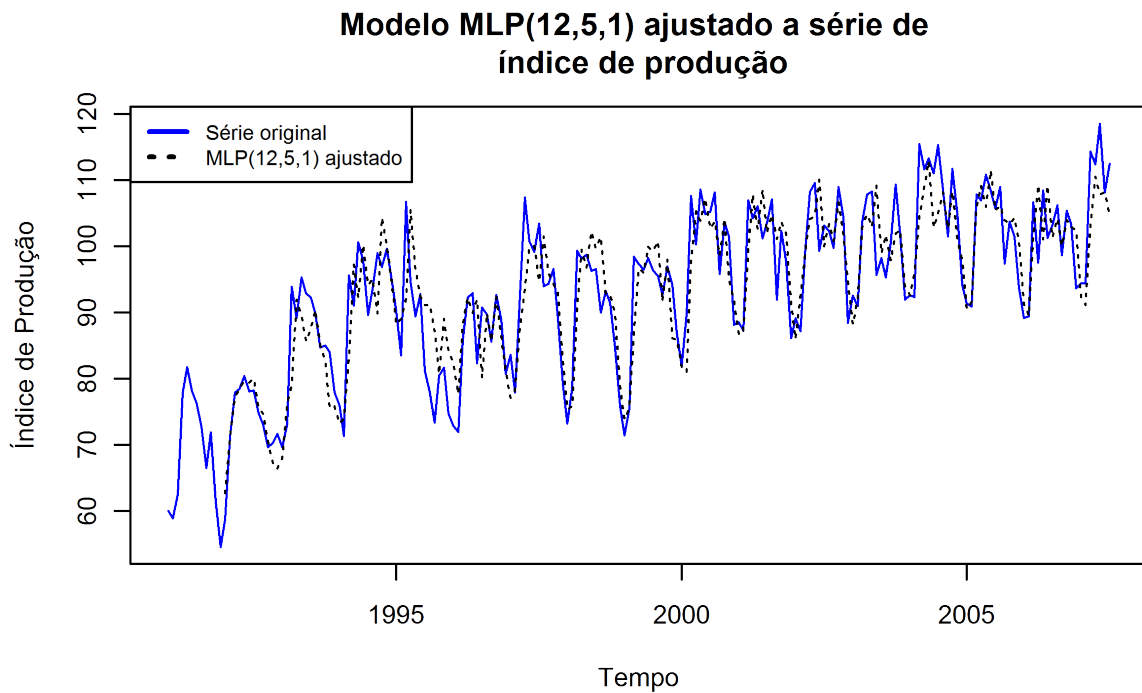


Figura 27: Gráfico da série do índice de produção e do modelo MLP(12,5,1) ajustado
 Fonte: Elaborado pelo autor

4.3.7 Comparação dos modelos MLP/RNA e ARIMA/SARIMA na série do índice de produção

A tabela (6) apresenta os resultados obtidos tanto pelos modelos clássicos como pelos modelos MLP/RNA na modelagem da série do índice de produção. Como citado no capítulo 3, os três modelos considerados melhores entre a modelagem ARIMA/SARIMA e os seis da modelagem por MLP/RNA são novamente comparados, com base agora apenas na sua qualidade preditiva.

Critérios de comparação de modelos			
Modelo	EQM Previsão	NMSE	MAPE
SARIMA(5,1,0)×(2,0,1) ₆	20,64	0,3799	3,1482%
SARIMA(2,1,3)×(2,1,2) ₁₂	14,15	0,2604	2,6530%
SARIMA(5,1,2)×(2,1,1) ₁₂	14,10	0,2594	2,8579%
MLP(6,10,1)	58,98	1,0856	6,1996%
MLP(6,18,1)	80,31	1,4781	6,8459%
MLP(6,28,1)	67,18	1,2365	6,3057%
MLP(12,5,1)	35,11	0,6463	4,8781%
MLP(12,24,1)	38,74	0,7131	4,9823%
MLP(12,26,1)	30,87	0,5682	4,3543%

Tabela 6: Resultados dos critérios para comparação de modelos clássicos e modelos MLP/RNA
 Fonte: Elaborado pelo autor

Com relação aos resultados apresentados na tabela (6), destaca-se o baixo EQM de previsão obtido pelo modelo $SARIMA(5,1,2) \times (2,1,1)_{12}$, que foi de 14,10, assim como o NMSE DE 0,2594 e o MAPE de 2,6530%. O modelo $SARIMA(2,1,3) \times (2,1,2)_{12}$ também obteve um EQM de previsão baixo, no valor de 14,15. Destaca-se também o baixo rendimento obtido com os modelos MLP/RNA. O menor EQM de previsão obtido pelos modelos de redes neurais, que é de 30,87 obtido pelo modelo MLP(12,26,1), é maior que o dobro do melhor modelo Box-Jenkins, o $SARIMA(5,1,2) \times (2,1,1)_{12}$, que obteve um EQM de previsão igual a 14,10. Desta forma, conclui-se que na comparação entre os modelos, dados os critérios utilizados, opta-se pela modelagem ARIMA/SARIMA, sendo o modelo $SARIMA(5,1,2) \times (2,1,1)_{12}$ o que obteve o melhor rendimento dentro os modelos apresentados.

Para finalizar, a figura (28) apresenta o modelo com menor EQM de previsão e menor NMSE, um $SARIMA(5,1,2) \times (2,1,1)_{12}$, ajustado a série completa, inclusive com as previsões.

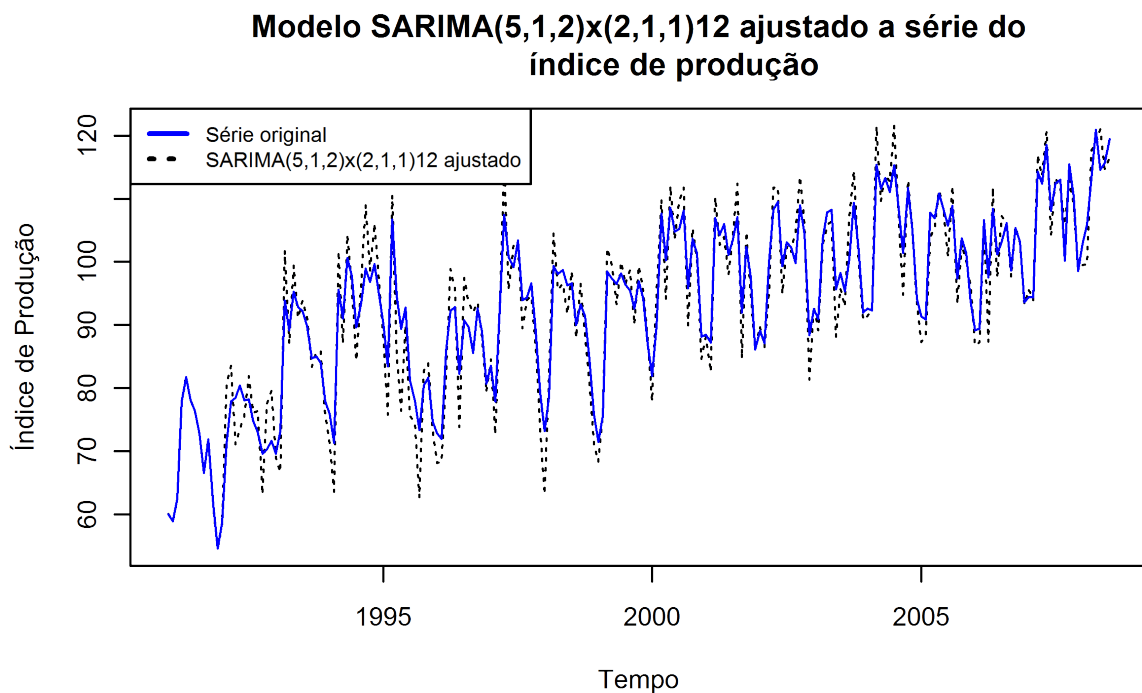


Figura 28: Gráfico da série do índice de produção e do modelo $SARIMA(5,1,2) \times (2,1,1)_{12}$ ajustado, com previsões

Fonte: Elaborado pelo autor

Como este modelo foi o mesmo escolhido na comparação feita apenas entre os modelos ARIMA/SARIMA, percebe-se que o problema de ajuste referente a estimação dos valores de pico e de vale, conforme o gráfico da figura (28), não aparece nas previsões. Até, aproximadamente, o ano de 2005 o modelo aparenta problemas com os valores extremos. Depois disso, o modelo se ajusta melhor aos resultados da série, inclusive nas previsões. Desta forma, conclui-se que os modelos ARIMA/SARIMA, com destaque para o modelo $SARIMA(5,1,2) \times (2,1,1)_{12}$,

se sobressaíram aos modelos MLP/RNA. Conclusão, esta, que é reforçada pelos resultados apresentados na tabela (6).

4.4 Aplicação da metodologia à série do log-índice de produção

4.4.1 Modelagem ARIMA/SARIMA da série do log-índice

Assim como na análise da série do índice de produção, as análises referentes a série do log-índice, apresentada na figura (20), iniciam pelos gráficos das fac e facp da série. As figuras (29) e (30), por sua vez, apresentam a fac e facp, respectivamente, com relação a série.

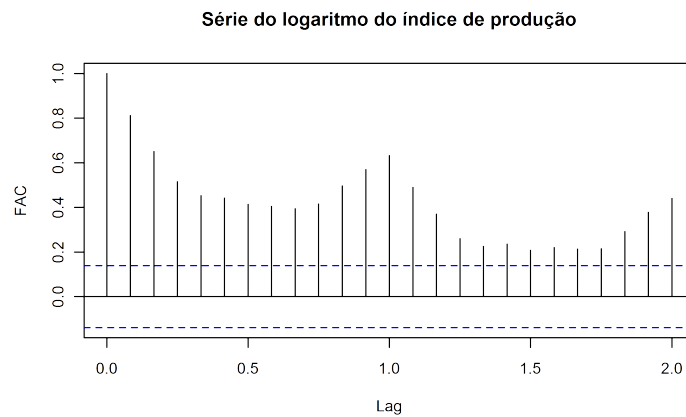


Figura 29: fac da série do log-índice de produção - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

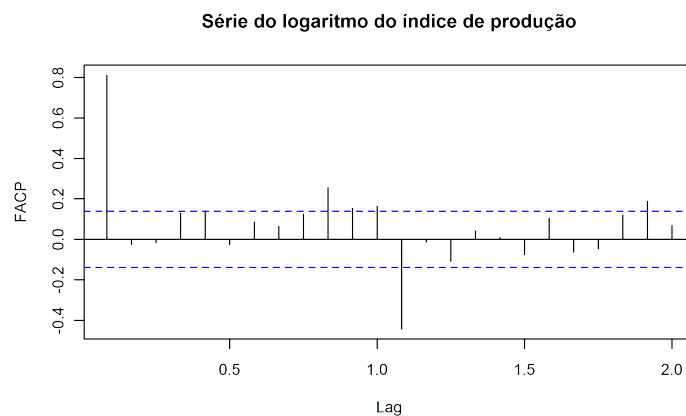


Figura 30: facp da série do log-índice de produção - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

Analisando-se o gráfico da figura (29) e o gráfico da figura (30), pode-se perceber as mesmas características da série do índice de produção, como a não estacionaridade apresentada pela lenta redução na fac, apresentada na figura (29). Nesta análise, porém, fica mais evidente

a não estacionaridade sazonal, que pode ser vista no gráfico da figura (29), nos *lags* 1 e 2, que representam, respectivamente, o período 12 e 24.

Desta forma, o próximo passo é realizar a diferenciação da série, na expectativa de torná-la não estacionária e então continuar com as análises das *fac* e *facp* até que os modelos sejam obtidos. Feito isso, três modelos se destacam. São eles: SARIMA(2, 1, 6) \times (1, 1, 0)₁₂, SARIMA(6, 1, 2) \times (1, 1, 0)₁₂ e SARIMA(3, 0, 0) \times (2, 1, 0)₆. Os critérios de modelagem destes modelos podem ser observados na tabela (7).

Critérios dos modelos Box-Jenkins para o log-índice de produção			
Modelo	AIC	BIC	EQM
SARIMA(3,0,0) \times (2, 1, 0) ₆	-5,69	-5,60	0,0034
SARIMA(2,1,6) \times (1, 1, 0) ₁₂	-5,87	-5,68	0,0030
SARIMA(6,1,2) \times (1, 1, 0) ₁₂	-5,88	-5,69	0,0029

Tabela 7: Resultados dos critérios de modelagem nos modelos Box-Jenkins na série do log-índice de produção

Fonte: Elaborado pelo autor

4.4.2 Comparação dos modelos ARIMA/SARIMA na série do log-índice

Observando-se a tabela (7), percebe-se que os critérios dos três modelos são muito semelhantes entre si. O modelo SARIMA(3, 0, 0) \times (2, 1, 0)₆ possui os critérios AIC, igual a -5,69 e BIC, com valor de -5,60, sendo o modelo que apresenta o maior EQM, de 0,0034, sendo considerado inferior aos outros dois modelos. Quanto aos modelos SARIMA(2, 1, 6) \times (1, 1, 0)₁₂ e SARIMA(6, 1, 2) \times (1, 1, 0)₁₂, seus valores são praticamente os mesmos. O modelo escolhido para ser ajustado aos dados foi o SARIMA(6, 1, 2) \times (1, 1, 0)₁₂, pois este modelo é superior aos outros modelos apresentados nos três critérios utilizados.

4.4.3 Ajuste do modelo ARIMA/SARIMA considerado melhor

As figuras (31) e (32) apresentam, respectivamente, os gráficos da *fac* e da *facp* dos resíduos do modelo SARIMA(6, 1, 2) \times (1, 1, 0)₁₂. A figura (33) apresenta o ajuste do modelo a série de dados de ajuste, ainda sem incluir as previsões.

Observando-se os gráficos das figuras (31) e (32), percebe-se que nenhum valor é apresentado como significativo na *fac* ou *facp* dos resíduos do modelo. Devido a isso, não é necessário a realização de testes estatísticos para verificação dos resíduos.

Com relação ao gráfico da série do log-índice e do modelo ajustado apresentado na figura (33), nota-se que o modelo está bem ajustado aos dados. Ainda não é possível verificar suas

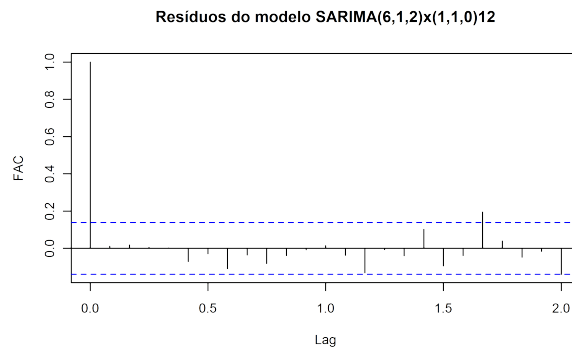


Figura 31: fac dos resíduos do modelo SARIMA(6,1,2)x(1,1,0)₁₂ - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

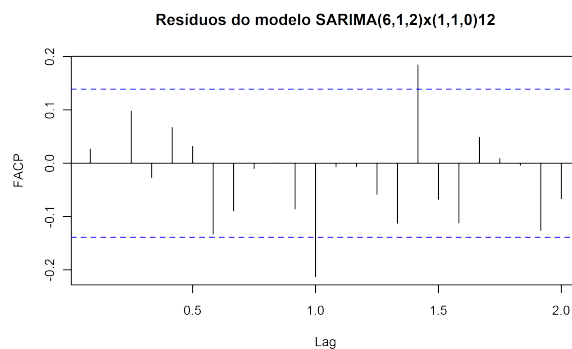


Figura 32: facp dos resíduos do modelo SARIMA(6,1,2)x(1,1,0)₁₂ - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

previsões, mas graficamente, o modelo aparenta estar bem ajustado, sem grandes desvios de estimação, o que reforça os resultados apresentados na tabela (7).

4.4.4 Modelagem MLP/RNA na série do log-índice

Assim como na análise dos modelos ARIMA/SARIMA, a modelagem por RNA também será feita tanto na série de dados quanto na série do log-índice. Para proceder com a modelagem, foram executadas todas as etapas apresentadas na figura (18). As tabelas (8) e (9) apresentam os modelos, considerados melhores, com seus critérios de ajuste e o EQM de previsão, nas configurações de 6 e 12 neurônios na camada de entrada, respectivamente. Com relação a camada oculta, os modelos continuam tendo um mínimo de 5 neurônios e um máximo de 30.

4.4.5 Comparação dos modelos MLP/RNA na série do log-índice

Observando-se as tabelas (8) e (9), pode-se observar que, de maneira geral, os modelos com a configuração de 6 neurônios na camada de entrada se sobressaíram aos modelos com a configuração de 12 neurônios, nos três critérios apresentados. Devido a utilização do log-índice,

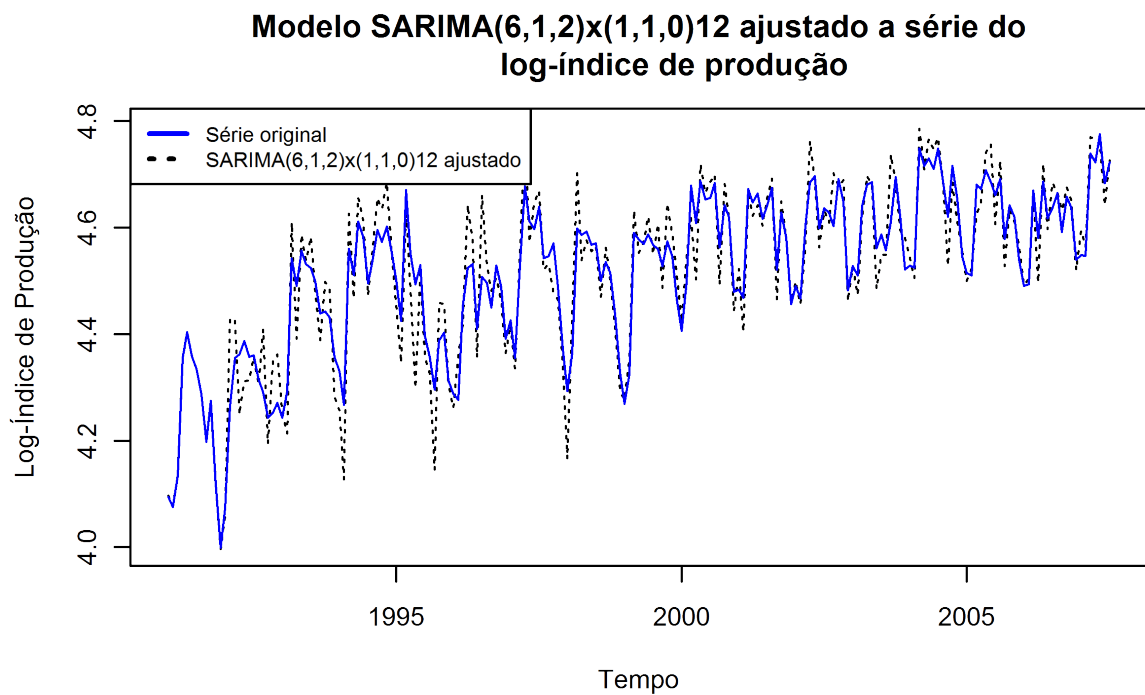


Figura 33: Gráfico da série do log-índice de produção e do modelo SARIMA(6, 1, 2)x(1, 1, 0)₁₂ ajustado
Fonte: Elaborado pelo autor

Critérios dos modelos MLP/RNA com 6 neurônios na camada de entrada

Camada oculta	AIC	BIC	EQM Ajuste	EQM Previsão(12)	EQM Previsão(1)
14	-6,41	-3,38	0,0042	0,0024	0,0049
15	-6,88	-3,63	0,0040	0,0024	0,0017
26	-6,54	-0,89	0,0037	0,0023	0,0021

Tabela 8: Resultados dos critérios nos modelos MLP/RNA com 6 neurônios na camada de entrada
Fonte: Elaborado pelo autor

Critérios dos modelos MLP/RNA com 12 neurônios na camada de entrada

Camada oculta	AIC	BIC	EQM Ajuste	EQM Previsão(12)	EQM Previsão(1)
9	-12,64	-9,13	0,0219	0,0080	0,0117
10	-11,53	-7,64	0,0218	0,0078	0,0112
11	-14,43	-10,15	0,0218	0,0081	0,0118

Tabela 9: Resultados dos critérios nos modelos MLP/RNA com 12 neurônios na camada de entrada
Fonte: Elaborado pelo autor

os EQM de previsão são muito baixos e aparentemente com diferenças pouco relevantes. Apesar do modelo MLP(6,26,1) possuir o menor EQM de previsão, de 0,0023, os critérios AIC e BIC são muito superiores, sendo, respectivamente, iguais a -6,54 e -0,89. Desta forma, levando-se em consideração também os critérios AIC e BIC, o modelo com melhor rendimento é o modelo MLP(6,15,1) que possui o menor AIC, igual a -6,88, o menor BIC, com valor igual a -3,63 e

um dos menores EQM de ajuste, de 0,0218, apesar do seu EQM de previsão, com valor igual a 0,0024 não ser o menor.

4.4.6 Ajuste do modelo MLP/RNA considerado melhor

Tendo sido escolhido o modelo MLP(6,14,1), a figura (34) apresenta o modelo ajustado a série de validação e treino, sem incluir, obviamente, as previsões. Observa-se no gráfico da figura (34), que o modelo não se ajusta bem aos dados do início da série, quando a variabilidade aparenta ser maior. A partir do ano 2000, a variabilidade dos dados parece ser menor e o modelo se ajusta melhor aos dados. Apesar do gráfico indicar que o modelo possa ter boas previsões, dado o comportamento do mesmo na parte final da série, ainda restam as análises a respeito dos critérios de comparação para este modelo.

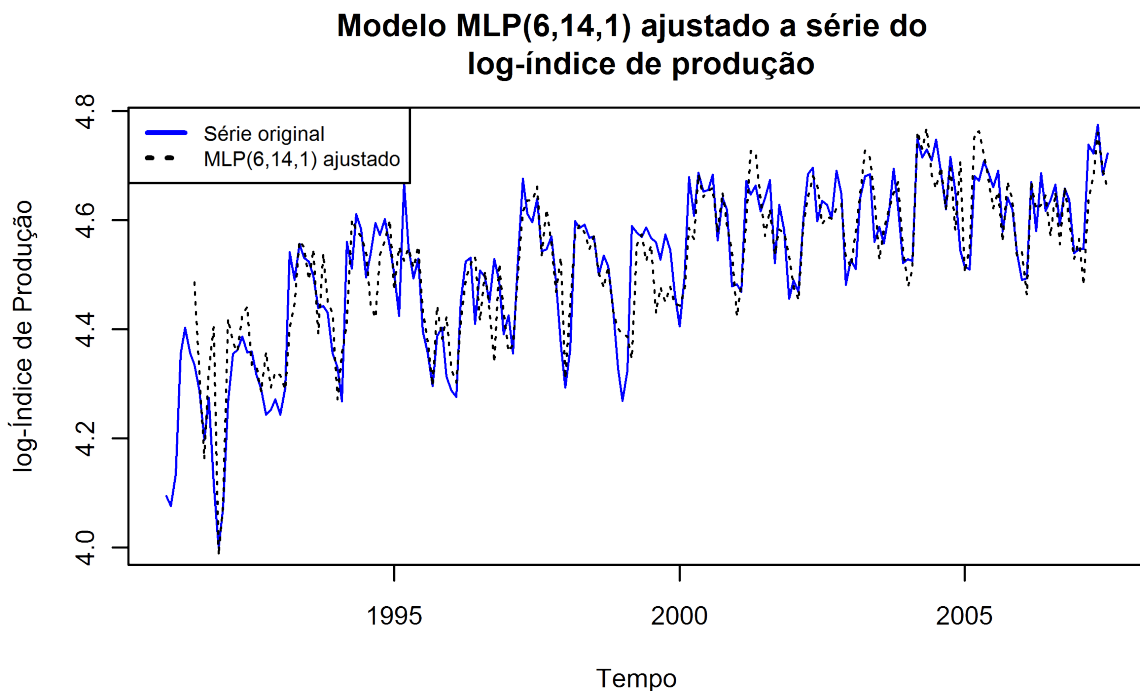


Figura 34: Gráfico da série do log-índice de produção e do modelo MLP(6,14,1) ajustado
Fonte: Elaborado pelo autor

4.4.7 Comparação dos modelos MLP/RNA e ARIMA/SARIMA na série do log-índice

A tabela (10) apresenta os resultados obtidos tanto pelos modelos clássicos como pelos modelos MLP/RNA na modelagem da série do log-índice de produção. Aqui, como no caso da série do índice de produção, destaca-se o fato de todos os modelos ARIMA/SARIMA terem obtido rendimento superior nos critérios utilizados. Apesar das diferenças parecerem pequenas,

isso se deve ao fato da série do log-índice apresentar os resultados em uma escala reduzida, devido ao logaritmo.

Critérios de comparação de modelos			
Modelo	EQM Previsão	NMSE	MAPE
SARIMA(3,0,0) \times (2, 1, 0) ₆	0,0017	0,3674	0,7483%
SARIMA(2,1,6) \times (1, 1, 0) ₁₂	0,0016	0,3410	0,7001%
SARIMA(6,1,2) \times (1, 1, 0) ₁₂	0,0015	0,3279	0,6632%
MLP(6,14,1)	0,0024	0,5303	0,8982%
MLP(6,14,1)LM	0,1118	0,5303	0,8982%
MLP(6,15,1)	0,0024	0,5296	0,8328%
MLP(6,26,1)	0,0023	0,5068	0,7982%

Tabela 10: Resultados dos critérios para comparação de modelos clássicos e modelos MLP/RNA
Fonte: Elaborado pelo autor

Aqui, assim como na modelagem da série do índice de produção física, os resultados da tabela (10) indicam uma superioridade dos modelos ARIMA/SARIMA frente aos modelos MLP/RNA, com base nos critérios utilizados. Apesar da relação das diferenças não ser a mesma, nota-se que, mesmo o modelo ARIMA/SARIMA com os maiores critérios ainda possui rendimento superior ao modelo MLP/RNA com os menores critérios. Com base nestes resultados, conclui-se que a metodologia Box-Jenkins obteve um rendimento superior aos modelos MLP/RNA, reforçando o resultado da comparação na série do índice de produção com valores absolutos.

A figura (35) apresenta o modelo que obteve os melhores critérios, menor EQM de previsão, igual a 0,0015, menor NMSE, igual a 0,3279 e o menor MAPE, de 0,6632%: um SARIMA(6, 1, 2) \times (1, 1, 0)₁₂, ajustado a série completa, inclusive com as previsões.

Analisando-se o gráfico do modelo ajustado com previsões a série do log-índice, apresentado na figura (35), pode-se perceber que o modelo escolhido, o SARIMA(6, 1, 2) \times (1, 1, 0)₁₂, apresenta problemas de ajuste no início da série, indo até a proximidade do ano 2000. A partir deste ponto o modelo parece se ajustar melhor aos dados, inclusive na parte das previsões. Desta forma, a análise gráfica reforça a conclusão obtida a partir da análise dos critérios apresentados na tabela (10), apontando o modelo SARIMA(6, 1, 2) \times (1, 1, 0)₁₂ como o modelo com os melhores critérios dentre os utilizados nesta dissertação.

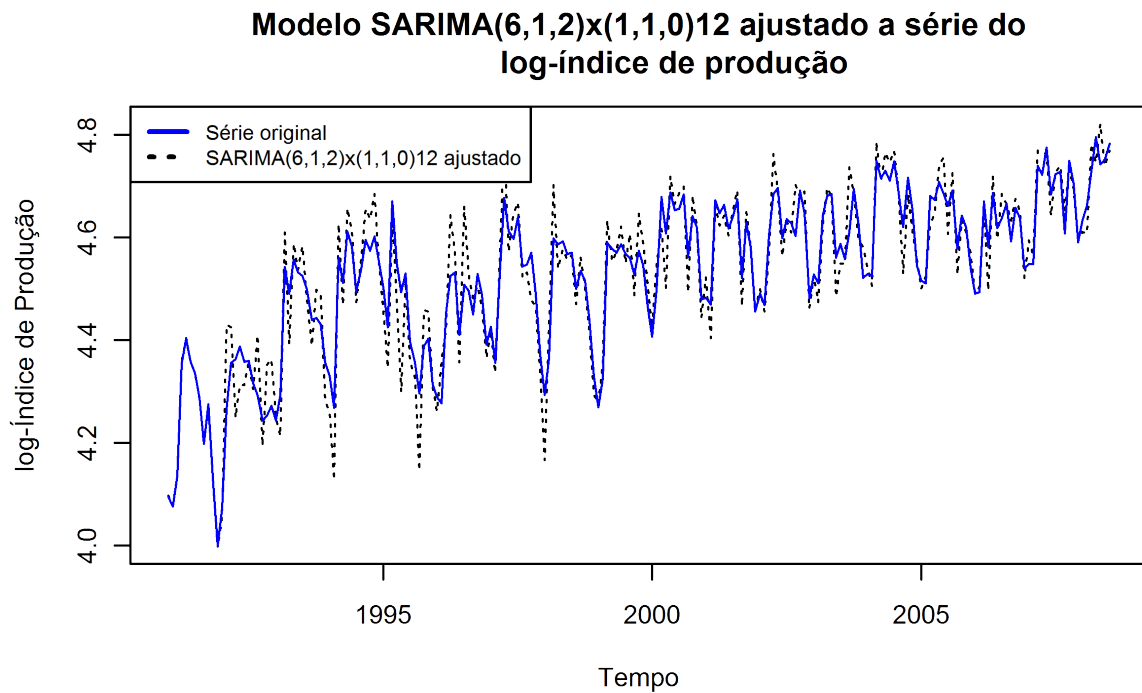


Figura 35: Gráfico da série do log-índice de produção e do modelo SARIMA(6, 1, 2)x(1, 1, 0)₁₂ ajustado, com previsões

Fonte: Elaborado pelo autor

4.5 Aplicação da metodologia a série das manchas solares

4.5.1 Modelagem ARIMA/SARIMA na série das manchas solares

Iniciando-se então a análise da série, segundo a figura (21), a série das manchas solares não aparenta nenhuma falha de estacionaridade perceptível. Porém, ao se observar os gráficos da *fac* e *facp* da série, apresentados nas figuras (36) e (37), respectivamente, o decaimento da *fac* é relativamente lento. Pode-se concluir, também, que existe uma sazonalidade na série, devido aos *lags* significativos em 6 e 12. Contudo, existe um termo em destaque junto a *facp*, no *lag* 2, que pode ser visto na figura (37), indicando um modelo AR(2).

A partir da análise gráfica da série, incluindo os gráficos apresentados nas figuras (21), (36) e (37) e da metodologia Box-Jenkins, três diferentes modelos foram obtidos: um modelo ARMA(4,4), um ARMA(3,5) e um SARIMA(4, 0, 3)x(2, 0, 0)₅. Foi utilizado nas comparações também o modelo citado por Priestley (1991), um ARMA(6,6). Neste caso específico, além dos três diferentes modelos obtidos neste trabalho, conforme a metodologia apresentada no capítulo 3 e na figura (17), também será visto e analisado um modelo apresentado pela literatura.

A partir da análise da gráfica da série, incluindo os gráficos apresentados nas figuras (21), (36) e (37) e da metodologia Box-Jenkins, três diferentes modelos foram obtidos: um modelo

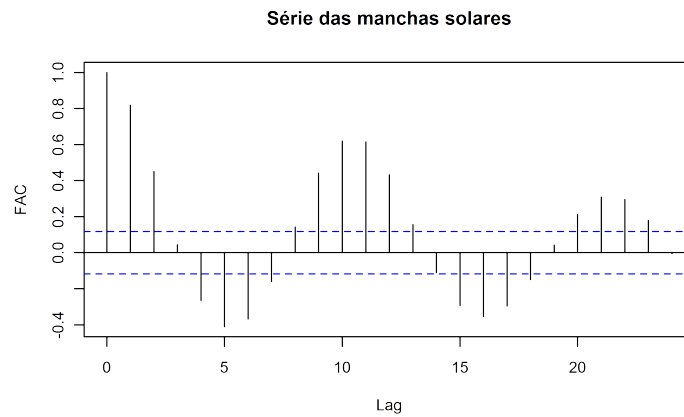


Figura 36: fac da série das manchas solares - lag em anos
Fonte: Elaborado pelo autor

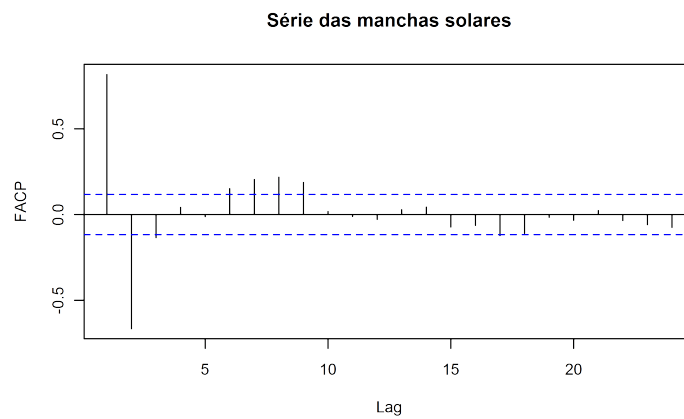


Figura 37: facp da série das manchas solares - lag em anos
Fonte: Elaborado pelo autor

ARMA(4,4), um ARMA(3,5) e um SARIMA(4,0,3) \times (2,0,0)₅. Foi utilizado nas comparações também o modelo citado por Priestley (1991), um ARMA(6,6). Neste caso específico, além dos três diferentes modelos obtidos neste trabalho, conforme a metodologia apresentada no capítulo 3 e na figura (17), também será visto e analisado um modelo apresentado pela literatura.

4.5.2 Comparação dos modelos ARIMA/SARIMA na série de manchas solares

Os valores para os critérios de comparação obtidos por estes modelos são apresentados na tabela (11).

Com base nos resultados da tabela (11) dois modelos se destacam: ARMA(6,6) e o modelo ARMA(3,5). Como destaque também, o fato de três dos quatro modelos não possuírem nenhum parâmetro sazonal, apesar de análises, como a vista em Priestley (1991), indicarem um ciclo de, aproximadamente, 11,3 anos. Quanto aos modelos ARMA(6,6) e ARMA(3,5), ambos se

Critérios dos modelos Box-Jenkins

Modelo	AIC	BIC	EQM
ARMA(4,4)	5,3924	5,5398	216,4476
ARMA(3,5)	5,3879	5,5353	215,4857
ARMA(6,6)	5,3519	5,5776	207,2638
SARIMA(4,0,3) \times (2,0,0) ₅	5,4450	5,6119	227,9544

Tabela 11: Resultados dos critérios nos modelos Box-Jenkins

Fonte: Elaborado pelo autor

destacam frente aos demais nos critérios apresentados. O ARMA(6,6) possui o menor critério AIC dentre os modelos, igual a 5,3519, enquanto o ARMA(3,5) possui o menor critério BIC, igual a 5,5353. Quanto ao EQM, o ARMA(6,6) é um tanto menor que o ARMA(3,5), sendo os valores, respectivamente, iguais a 207,2638 e 215,4857. Entretanto, o critério BIC penaliza mais modelos com muitos parâmetros e, dada a, relativamente, pequena diferença entre o EQM dos dois modelos, o escolhido é o modelo ARMA(3,5). Contudo, o fato do modelo ARMA(6,6) ter sido utilizado por outros autores, os valores dos critérios obtidos também serão utilizados neste trabalho como base de comparação.

4.5.3 Ajuste do modelo ARIMA/SARIMA considerado melhor

Sendo o melhor modelo ARIMA/SARIMA o modelo ARMA(3,5), os gráficos das figuras (38), (39) e (40) são das funções de autocorrelação, autocorrelação parcial e o mesmo ajustado a série de dados, respectivamente.

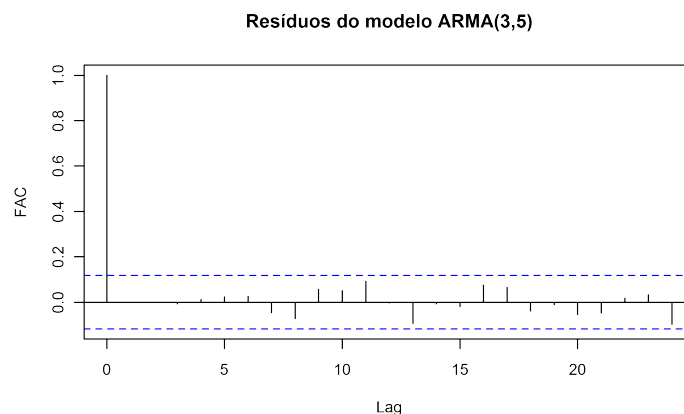


Figura 38: fac dos resíduos do modelo ARMA(3,5) - lag em anos (1 = 12 meses)

Fonte: Elaborado pelo autor

Com relação aos gráficos da fac e facp, apresentados, respectivamente, pelas figuras (38) e (39), pode se notar que não há nenhum valor significativo. Desta forma, o modelo está aparen-

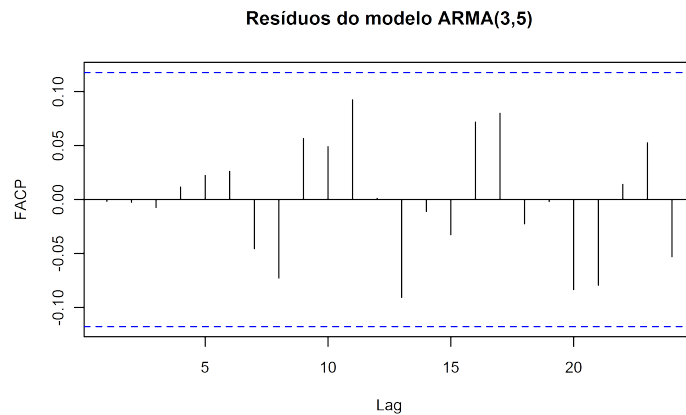


Figura 39: facp dos resíduos do modelo ARMA(3,5) - lag em anos (1 = 12 meses)
Fonte: Elaborado pelo autor

temente bem ajustado e não há necessidade de realizar um teste estatístico para verificação da fac dos resíduos.

Observando-se o gráfico da figura (40), pode-se perceber que o modelo está bem ajustado, acompanhando a série de dados. Porém, com relação aos valores de pico o modelo acaba superestimando e subestimando os valores de vale. Além disso, em alguns pontos, como nas proximidades do ano de 1900 e 1950, o modelo acaba não estimando muito bem os valores.

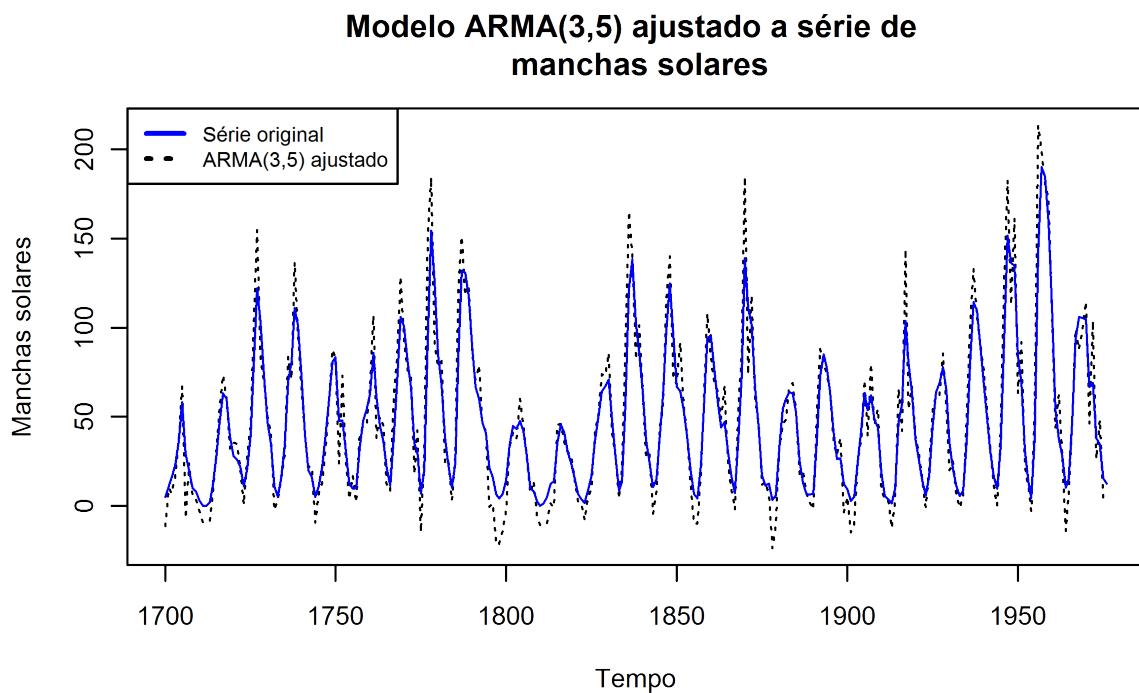


Figura 40: Gráfico da série das manchas solares e do modelo ARMA(3,5) ajustado
Fonte: Elaborado pelo autor

4.5.4 Modelagem MLP/RNA na série das manchas solares

De forma a simplificar a posterior comparação entre os modelos clássicos e os modelos MLP/RNA, a série será utilizada sem nenhum ajuste nos dados. A tabela (12) apresenta os critérios de alguns dos melhores modelos treinados com 6 neurônios na camada de entrada, inclusive o EQM de previsão. A tabela (13), por sua vez, apresenta os resultados obtidos pelos melhores modelos com 12 neurônios na camada de entrada.

Critérios dos modelos MLP/RNA com 6 neurônios na camada de entrada					
Camada oculta	AIC	BIC	EQM Ajuste	EQM Previsão(12)	EQM Previsão(1)
20	5,16	8,45	269,24	875,33	581,34
25	4,88	8,99	220,05	604,03	327,13
27	4,99	9,44	239,92	898,21	742,70

Tabela 12: Resultados dos critérios nos modelos MLP/RNA com 6 neurônios na camada de entrada
Fonte: Elaborado pelo autor

Critérios dos modelos MLP/RNA com 12 neurônios na camada de entrada					
Camada oculta	AIC	BIC	EQM Ajuste	EQM Previsão(12)	EQM Previsão(1)
12	6,40	9,92	882,87	1400,25	6029,12
16	6,05	10,74	702,07	1352,35	6920,01
28	6,35	14,58	777,44	1115,37	5425,14

Tabela 13: Resultados dos critérios nos modelos MLP/RNA com 12 neurônios na camada de entrada
Fonte: Elaborado pelo autor

4.5.5 Comparação dos modelos MLP/RNA na série de manchas solares

Com base nos dados das tabelas (12) e (13) destaca-se o fato da diferença obtida pelos modelos de 6 e 12 neurônios na camada de entrada. Entretanto, dois destes modelos se destacam dos demais, o MLP(6,20,1) e o MLP(6,25,1) devido aos critérios de seleção AIC e BIC. Isto é, são modelos que conseguiram obter um baixo EQM e com, relativamente, poucos parâmetros. No caso dos modelos MLP, é dado um maior destaque ao critério BIC, devido ao fato de ser mais punitivo quanto ao número de parâmetros do que o critério AIC. Como exemplo, pode-se citar o modelo MLP(12,28,1) que possui um AIC de 6,35, inferior ao AIC do modelo MLP(12,12,1). O BIC, entretanto, do modelo MLP(12,28,1) é cerca de 45% maior que o do modelo MLP(12,12,1), 14,58 contra 9,92, respectivamente. Contudo, o modelo com os melhores critérios de comparação é o modelo MLP(6,25,1). Apesar de seu BIC, que é igual a 8,99, não ser o menor dentre os modelos MLP/RNA das tabelas (12) e (13), o critério AIC igual a 4,88, o EQM de ajuste igual a 220,05 e o EQM de previsão igual a 604,03 são os melhores

dentre os modelos apresentados. Desta forma, o modelo escolhido é o modelo MLP(6,25,1), que obteve, de maneira mais ampla, melhores critérios que os demais modelos apresentados.

4.5.6 Ajuste do modelo MLP/RNA considerado melhor

A figura (41) apresenta o gráfico do modelo escolhido como melhor dentro os apresentados, o MLP(6,25,1). No gráfico é apresentada somente a série e o modelo ajustado, sem incluir as previsões.

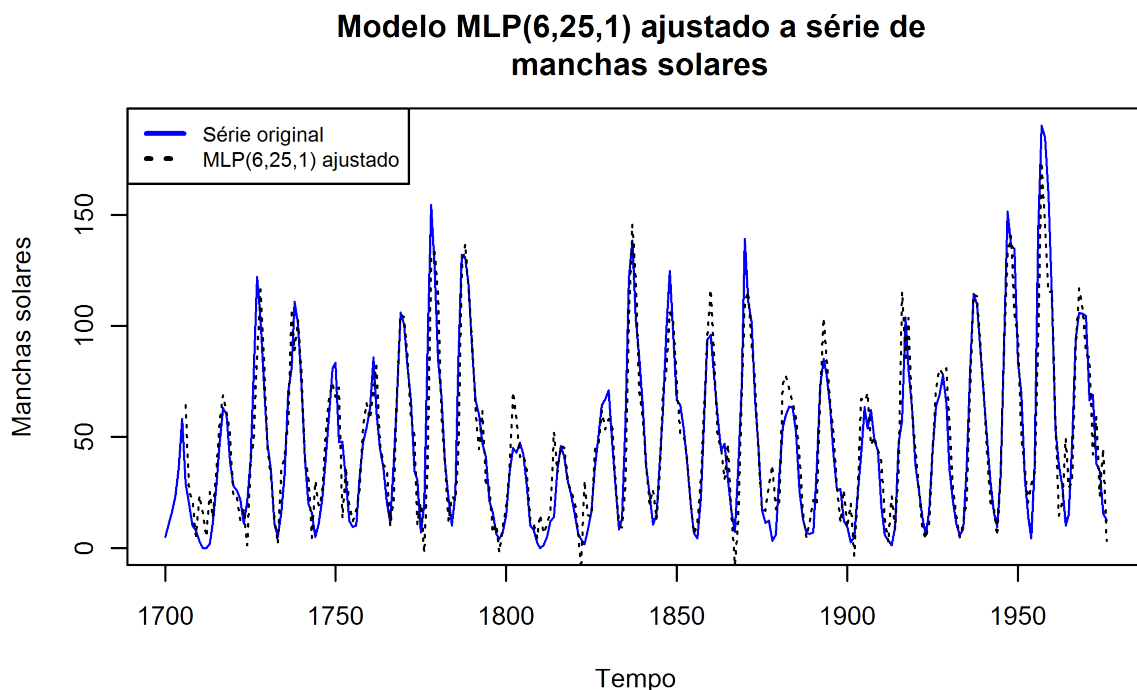


Figura 41: Gráfico da série das manchas solares e do modelo MLP(6,25,1) ajustado
Fonte: Elaborado pelo autor

Observando o gráfico da figura (12), percebe-se que o modelo está bem ajustado aos dados, apenas com algumas diferenças em alguns pontos da série. Como o gráfico não inclui as previsões, ainda não é possível afirmar nada a respeito de um possível sobre ajuste do modelo.

4.5.7 Comparação dos modelos MLP/RNA e ARIMA/SARIMA na série de manchas solares

A tabela (14) apresenta os resultados obtidos tanto pelos modelos clássicos como pelos modelos MLP/RNA. Destacam-se os resultados obtidos pelos modelos ARMA(3,5), MLP(6,20,1) e MLP(6,25,1), sendo este último o que obteve menor EQM de previsão dentre os modelos para a série de manchas solares, igual a 604,03.

Critérios de comparação de modelos			
Modelo	EQM Previsão	NMSE	MAPE
ARMA(3,5)	1812,26	0,6279	32,34%
ARMA(6,6)	1940,22	0,6722	35,48%
MLP(6,20,1)	875,33	0,3033	29,77%
MLP(6,25,1)	604,03	0,2093	28,45%
MLP(12,16,1)	1352,35	0,4671	59,65%
MLP(12,28,1)	1115,37	0,3852	42,70%

Tabela 14: Resultados dos critérios para comparação de modelos clássicos e modelos MLP/RNA
Fonte: Elaborado pelo autor

Observando-se os resultados apresentados na tabela (14), percebe-se uma situação bem distinta do que ocorreu na análise da série do índice do produção. Neste caso, a metodologia RNA obteve melhores resultados com base nos critérios utilizados, do que a metodologia Box-Jenkins, com exceção dos modelos com a configuração de 12 neurônios na camada oculta, que obtiveram MAPE superior aos modelos Box-Jenkins.

Com relação a comparação direta entre os modelos, o modelo MLP(6,25,1) obteve um excelente rendimento nos critérios utilizados, tendo os menores EQM de previsão, igual a 604,03, NMSE, igual a 0,2093, e MAPE, igual a 28,45%. Exceto pelo critério MAPE, os outros modelos MLP/RNA também obtiveram bons resultados nos critérios, com destaque ao modelo MLP(6,20,1), que apesar do EQM de previsão igual a 875,33, obteve um NMSE relativamente baixo de 0,3033 e um MAPE de 29,77%, bem próximo do modelo escolhido como melhor, o MLP(6,25,1). Para a análise gráfica, a figura (42) apresenta o modelo MLP(6,25,1) ajustado a série completa, inclusive com as previsões.

Analisando-se o gráfico apresentado pela figura (42), nota-se que o modelo não só teve um bom ajuste aos dados de treinamento e ajuste como o mesmo se repetiu nas previsões. O modelo está bem adequado à série de dados, sem, aparentemente, nenhum erro de estimação que destoe dos demais, acompanhando o comportamento da série em, praticamente, todo o período apresentado.

4.6 Discussão

A primeira etapa da metodologia utilizada nesta dissertação era a verificação de linearidade da série de dados. A série das manchas solares é uma série mais trabalhada na literatura, como em Priestley (1991), e os resultados a serem obtidos pelos testes de linearidade aplicados à série eram esperados que apresentassem de forma clara a não linearidade da série.

Modelo MLP(6,25,1) ajustado a série de manchas solares

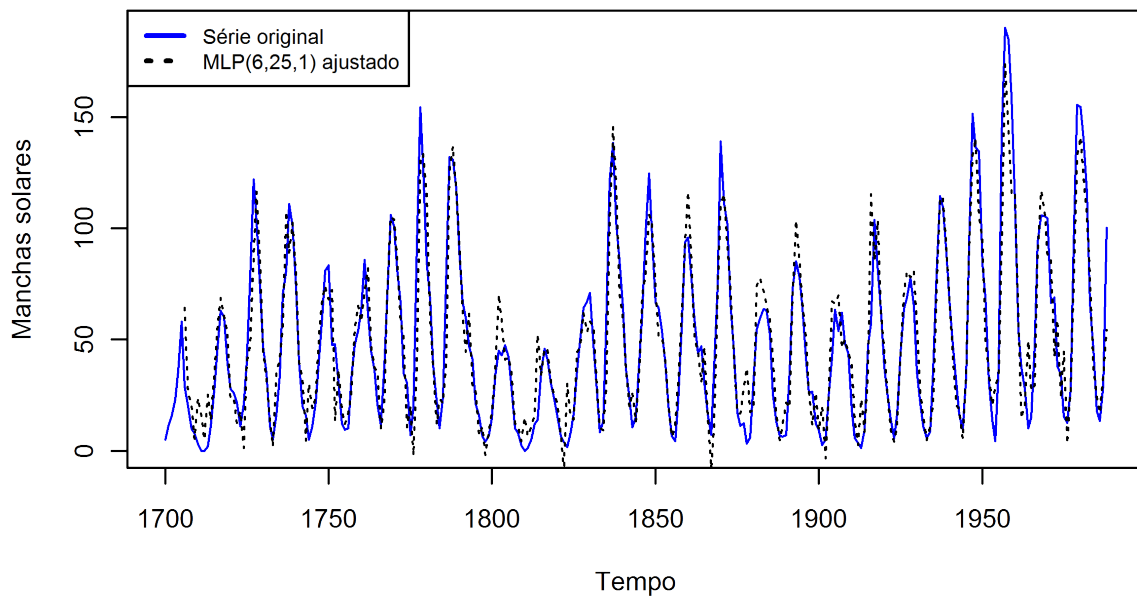


Figura 42: Gráfico da série das manchas solares e do modelo MLP(6,25,1) ajustado, com previsões
Fonte: Elaborado pelo autor

Contudo, não foi isso que ocorreu. Os resultados obtidos pelos testes apresentados nas tabelas (1) e (2) diferem do que era esperado em relação, principalmente, a série de manchas solares. Esta série é uma série de dados reconhecidamente não linear. Porém, com relação ao teste de White, apenas 4 dos 11 *lags* testados apresentaram resultados que indicasse que a série deveria ser não linear. Estes resultados podem levar a crer que os resultados obtidos pelo teste de White não deveriam ter a mesma importância do teste de Teräsvirta. O teste de Teräsvirta, por sua vez, obteve em todos os *lags*, exceto pelo *lag* 2, resultados que indicam a série como sendo não linear.

Estes resultados também são reforçados com relação as diferenças obtidas no teste de White para a séries do índice de produção física e do log-índice de produção. No teste de White, enquanto a série do log-índice obteve resultados significativos até o *lag* 9, a série do índice de produção não obteve a mesma quantidade de resultados significativos. Na série do índice de produção física nenhum *lag* foi significativo no teste de White, indicando que a série pudesse ser linear. Desta forma, deu-se maior prioridade ao teste de Teräsvirta, que, aparentemente, detectou de forma mais clara uma possível não linearidade das séries de dados. Portanto, admitindo o teste de Teräsvirta como prioritário, nas duas configurações da série do índice de produção física (com e sem o logaritmo) e na série de manchas solares, os resultados das tabelas (1) e (2) indicam que os dados devem ser não lineares.

Na comparação de modelos, afirmações feitas em Haykin (2001), Cheng e Titterington (1994) e Hill et al. (1996) dizem que os modelos RNA são mais adequados e possuem melhores resultados frente a sistemas não lineares, principalmente no caso de séries temporais para previsões de horizontes menores. Para horizontes de previsão maiores, os resultados obtidos não são claros. Isso ocorreu na série de manchas solares. Os modelos RNA foram superiores - em termos do EQM de previsão e também nos critérios NMSE e MAPE - aos modelos Box-Jenkins. Porém o mesmo não ocorreu na série do índice de produção física industrial. Nesta série os modelos RNA obtiveram resultados muito aquém dos modelos Box-Jenkins. Desta forma, conclui-se que não é possível assumir que um modelo seja superior ao outro de maneira definitiva. Porém esse resultado coloca em dúvida as hipóteses de que: (i) série do índice de produção física industrial é não linear e (ii) de que os modelos RNA possuem rendimento superior aos modelos Box-Jenkins em séries não lineares com horizontes de previsão maiores.

A hipótese de que a série seja linear deve ser descartada dados os resultados apresentados nas tabelas (1) e (2), que apresentam, respectivamente, os testes de Tukey e Teräsvirta e White. Contudo, mesmo que seja considerada uma falha generalizada nos testes, de tal forma que a série venha a ser uma série linear, Zhang (2001) afirma que os modelos RNA possuem boa acurácia, mesmo em sistemas lineares. Já Pinto e Medeiros (2005) afirma que os modelos RNA podem ter problemas com sistemas lineares, quando comparados a modelos específicos para este tipo de situação, como os modelos Box-Jenkins utilizados nesta dissertação. De qualquer forma, mesmo que a hipótese de não linearidade da série do índice de produção física seja ou não aceita, o rendimento dos modelos MLP/RNA não deveria ter sido tão aquém, conforme visto nas tabelas (6) e (10), apesar de nesta última o rendimento nos critérios NMSE e MAPE, principalmente, não terem obtido uma diferença relativa de mesma magnitude que vista na série do índice de produção física sem o uso do logaritmo.

Com relação a hipótese de superioridade dos modelos MLP/RNA frente aos modelos Box-Jenkins em séries não lineares, alguns cuidados devem ser tomados. Como este trabalho comparou os modelos com um horizonte maior de previsão, os modelos Box-Jenkins possuem certa vantagem. Em trabalhos como Chen (2006), Suarez-Farinas et al. (2004), Hamzaçebi et al. (2009), o horizonte de previsão é de apenas 1 passo, obtendo EQM de previsão e MAPE bem abaixo dos apresentados neste trabalho. Contudo, conforme visto em Haykin (2001), com referência a modelos MLP/RNA, e Morettin e Toloí (2006), com referência a modelos Box-Jenkins, modelos com excessos de parâmetros podem ter problemas quanto as previsões. No caso específico de modelos MLP/RNA isso ocorre tanto devido ao objetivo do treinamento quanto a sensibilidade dos modelos. O que pode ser visto nas tabelas (4), (5), (8) e (5) são modelos que, na maior parte dos casos, o EQM de previsão a 12 passos é inferior ao EQM

de previsão a 1 passo. Como os modelos foram treinados segundo a equação (15) e segundo a figura (18) o que ocorreu foi que modelos que tivessem baixo EQM de previsão a 1 passo, como nos trabalhos citados, fossem preteridos por modelos que conseguissem maior estabilidade de previsão. Com relação a sensibilidade, modelos que possuem EQM de previsão a 1 passo muito baixo tendem a ser extremamente sensíveis a modificações dos valores de entrada. De certa forma, o erro de previsão tende a crescer exponencialmente nestes casos, enquanto que nos modelos lineares o erro de previsão cresce mais lentamente. Desta forma, no horizonte de previsão utilizado neste trabalho, para as séries de produção física industrial os modelos MLP/RNA não foram tão estáveis quanto os modelos Box-Jenkins.

Ainda com relação a hipótese de que os modelos MLP/RNA são superiores aos modelos Box-Jenkins em séries não lineares, a mesma não pode ser completamente rejeitada dados os resultados apresentados na série do índice de produção física industrial mensal para o Estado do Rio Grande do Sul, onde o modelo Box-Jenkins apresentou melhores critérios de acurácia. O modelo obteve um EQM de previsão menor que a metade do EQM de previsão do considerado melhor modelo RNA ajustado. No entanto, existem, pelo menos, duas possíveis razões para este fato: ou a suposição de não linearidade da série é falha ou os modelos MLP foram muito limitados para tratar da série, isto é, simular o sistema gerador dos dados de maneira adequada. Como a hipótese de a série do índice de produção física se tratar de uma série linear foi rejeitada estatisticamente, resta a possibilidade dos modelos MLP/RNA terem sido mal dimensionados com relação ao número de neurônios em cada camada e, possível também, o número de camadas escolhidas.

Esta possibilidade é reforçada com base nos resultados apresentados nas comparações feitas entre os modelos MLP/RNA e os modelos Box-Jenkins, na tabela (6), referente a série do índice de produção física, e na tabela (10), referente a série do log-índice. Na série do índice de produção física, por exemplo, o modelo com os melhores critérios de acurácia dentre os modelos MLP/RNA, foi um dos modelos com maior número de parâmetros, o modelo MLP(12,26,1), enquanto modelos mais simples, com menor número de parâmetros obtiveram pior rendimento, como os modelos MLP(6,10,1), MLP(6,18,1) e MLP(6,28,1). Nesta série, provavelmente, modelos com maior número de camadas ocultas, ou maior número de neurônios, pudessem obter um rendimento melhor.

Contudo, esta afirmação encontra resistência pelos resultados obtidos tanto na análise da série do log-índice, quando em Chen (2006). No estudo de Chen (2006), o autor comparou modelos Box-Jenkins e modelos MLP/RNA em uma série semelhante e concluiu que os modelos MLP/RNA obtiveram melhores resultados, utilizando para tanto um horizonte de previsão de apenas 1 passo. Além disso, os modelos utilizados tinham menos parâmetros. No estudo de

Chen (2006), o modelo escolhido com o melhor rendimento, foi um MLP(2,4,1), fora do escopo de modelos utilizados nesta dissertação. Ainda relativo a modelos com maior número de parâmetros, a análise da série do log-índice, assim como a série de manchas solares, os modelos com melhor rendimento dentre os modelos MLP/RNA foram modelos com apenas 6 neurônios na camada de entrada, com reduzido o número de parâmetros com relação aos modelos com 12 neurônios na camada de entrada. Sendo assim, reforça-se a hipótese de que os modelos possam ter sido mal dimensionados, não em relação a modelos com maior número de parâmetros, mas a modelos com menor número de parâmetros.

Ainda com relação a hipótese de os modelos MLP/RNA terem sido mal dimensionados, não necessariamente a causa é, unicamente, a quantidade de parâmetros. No estudo de Chen (2006), o autor compara não só modelos com diferentes configurações de neurônios na camada oculta, como também compara as taxas de aprendizagem utilizadas. Conforme visto na seção 2.3, são necessárias outras configurações que não apenas a quantidade de neurônios em cada camada e o número de camadas. É necessário configurar uma ou mais funções de ativação dos neurônios, taxa de aprendizagem e taxas de parada do algoritmo de obtenção dos pesos sinápticos como número máximo de interações e erro mínimo aceitável, apenas para citar algumas. As configurações utilizadas nesta dissertação contribuíram para os resultados obtidos pelos modelos MLP/RNA.

Já no estudo de caso envolvendo a série de dados das manchas solares, os modelos de RNA obtiveram resultados superiores aos modelos Box-Jenkins, inclusive a modelos apresentados na literatura, em especial o modelo apresentado em Priestley (1991). Além disso, observando-se a tabela (12), referente a modelos com 6 neurônios na camada de entrada, percebe-se que o EQM de previsão a 1 passo é menor nos modelos apresentados, enquanto o mesmo não ocorre na tabela (13), referente a modelos com 12 neurônios na camada de entrada. Isso, provavelmente, se deve a maior complexidade desta série, assim como o fato de ser reconhecidamente uma série não linear (HAMZAÇEBI et al., 2009). Além disso, ainda conforme consta em Priestley (1991), esta série não possui um ciclo fixo, prejudicando a modelagem tanto por modelos Box-Jenkins, quanto por modelos MLP/RNA. Também vale ressaltar os resultados obtidos pelos modelos ajustados neste trabalho, com relação aos critérios utilizados, tendo sido semelhantes aos apresentados em Priestley (1991). Ainda com relação a série das manchas solares, trabalhos como Suarez-Farinas et al. (2004), Hamzaçebi et al. (2009), o rendimento dos modelos RNA são considerados melhores quando comparados a outros modelos, resultados estes, reforçados pelos apresentados nesta dissertação. Especificamente com relação aos resultados apresentados pelos modelos com 12 neurônios na camada de entrada, isso pode ser devido ao ciclo variável da série, fazendo com que o modelo preveja um valor de pico quando o valor verdadeiro seria

um valor de vale, o que acaba por aumentar demasiadamente o EQM de previsão. Isso pode ser visto nos resultados apresentados na tabela (13).

Apesar de trabalhos como Fernandes et al. (1996), Hill et al. (1996), Hoo et al. (2002), Dellana e West (2009) apenas para citar alguns, apresentarem grandes vantagens no uso de modelos RNA, inclusive os modelos MLP/RNA, os resultados apresentados por estes modelos na série do índice de produção física e do log-índice indicam a robustez dos modelos Box-Jenkins, citada por Hamilton (1994), Enders (1995), Morettin e Tolo (2006), permite a estes modelos uma modelagem relativamente adequada mesmo em sistemas não lineares. Mesmo na série das manchas solares, o NMSE e MAPE dos modelos ARIMA/SARIMA, apresentados na tabela (14), não podem ser considerados inadequados, quando comparados aos resultados dos modelos MLP/RNA nesta mesma série. Enders (1995), inclusive, cita que muitas vezes é melhor ignorar o tipo de não linearidade existente do que tentar ajustar um modelo específico a esta não linearidade.

Outro ponto a ser destacado é quanto ao tempo de modelagem. Os modelos Box-Jenkins, apesar do seu algoritmo, dependem demasiadamente da observação e análise dos gráficos da fac e facp. Isso torna sua modelagem, de certa forma, lenta. A construção dos modelos Box-Jenkins deste trabalho, excetuando-se o tempo computacional, foi de aproximadamente 3 horas. Já os modelos MLP/RNA podem ser automatizados, conforme visto neste trabalho. Algumas análises foram realizadas, mas, novamente sem contar o tempo computacional para obtenção dos modelos, estas análises tomaram um tempo inferior a 1 hora. O tempo computacional gasto na modelagem MLP/RNA é muito superior ao gasto em modelos Box-Jenkins. O cálculo dos parâmetros dos modelos apresentados demora, em média, 5 minutos, enquanto os modelos MLP/RNA demoraram, em média, 2 horas.

5 CONSIDERAÇÕES FINAIS

Nesta dissertação foram comparadas duas diferentes metodologias para prever resultados a partir de uma série de dados: a metodologia clássica dos modelos Box-Jenkins e a metodologia de redes neurais artificiais (RNA). Ambas são demasiadamente extensas e foram selecionados modelos dentro de cada metodologia. Na metodologia clássica, foram selecionados os modelos Box-Jenkins, ou modelos da família ARIMA/SARIMA por se tratarem dos modelos mais usuais e já implementados em diversos pacotes estatísticos e programas comerciais. Na metodologia de RNA, foram selecionados os modelos MLP, de *perceptron* de múltiplas camadas, também por se tratarem de modelos comuns dentro da utilização de RNA, apesar de sua implementação ainda não possuir o mesmo alcance dos modelos ARIMA/SARIMA.

Para comparar formas de estabelecer modelos de séries temporais, foram selecionadas duas séries de dados: a série do índice de produção física industrial do Estado do Rio Grande do Sul e a série das manchas solares. A primeira foi escolhida por se tratar de uma série muito utilizada para previsão de produção e planejamento de indústrias e governos. A segunda foi escolhida por se tratar de uma série com características conhecidas e de interesse desta dissertação, como a não linearidade em relação a média, sendo utilizada como caso-controle.

A comparação de modelos Box-Jenkins e MLP/RNA utilizando-se a série de índice de produção física industrial com valores absolutos e a mesma com o uso da função logaritmo, a série do log-índice de produção, apresentada no capítulo 4, representa o principal resultado deste trabalho. Para tanto, tem-se a hipótese inicial de que esta série seja não linear, o que foi verificado com testes de linearidade. Sendo assim, dados os critérios utilizados, os resultados obtidos e dentre os modelos apresentados neste trabalho, o modelo com melhor rendimento em ambas as séries foi da metodologia Box-Jenkins. Na série do índice de produção em valores absolutos, o modelo escolhido foi um SARIMA(5, 1, 2) \times (2, 1, 1)₁₂, enquanto que na série do log-índice de produção, o modelo escolhido foi um SARIMA(6, 1, 2) \times (1, 1, 0)₁₂. Estes resultados, por sua vez, contrariam os resultados obtidos por Chen (2006) que, em comparação semelhante, obteve um modelo MLP/RNA com rendimento superior aos modelos Box-Jenkins utilizando como critérios de comparação o MAPE e o NMSE.

Quanto a comparação dos modelos utilizando-se a série de manchas solares, o modelo escolhido foi um MLP(6, 25, 1), dos modelos MLP/RNA, o que era esperado, tendo em vista as características da série. Estes resultados também foram apresentados no capítulo 4 deste trabalho. A série das manchas solares é uma série muito utilizada quando se refere a séries temporais não

lineares, tendo sido utilizada em diversos trabalhos como Priestley (1991), Müller et al. (1995), De Wilde (1997) e Hamzaçebi et al. (2009). O estudo realizado chegou as mesmas conclusões que trabalhos como Suarez-Farinas et al. (2004), Kuo e Reitsch (1995/1996), Hamzaçebi et al. (2009) para citar alguns, que afirmam que os modelos MLP/RNA possuem previsões mais acuradas em sistemas não lineares, utilizando séries de dados iguais ou semelhantes, o que reforça os resultados obtidos nesta dissertação.

Com relação a metodologia para as comparações, a mesma é apresentada no capítulo 3 deste trabalho, sendo detalhadamente descrita. Também é apresentada a figura (17), que apresenta de forma gráfica as etapas realizadas para as comparações dos modelos. Também são apresentados os critérios utilizados para as comparações, assim como suas formulações.

Dados os resultados das comparações, não é possível afirmar, de forma categórica, que uma das duas metodologias seja superior a outra em modelagens de séries temporais não lineares. Enquanto que, na comparação utilizando a série de manchas solares, o modelo MLP/RNA selecionado era esperado, devido a resultados semelhantes na literatura, os modelos Box-Jenkins também obtiveram bons resultados, com relação aos critérios utilizados. Na comparação utilizando ambas as configurações da série do índice de produção física industrial, os resultados obtidos pelos modelos MLP/RNA, podem ter sido prejudicados dadas as limitações do trabalho. De qualquer forma, aqui também os modelos Box-Jenkins obtiveram um excelente rendimento nos critérios utilizados.

5.1 Sugestões para trabalhos futuros

O uso de modelos de RNA com maior número de camadas seria uma sugestão para futuros trabalhos. Na série das manchas solares os modelos de RNA obtiveram resultados satisfatórios. Contudo, na série do índice de produção física, o resultado foi muito aquém dos modelos Box e Jenkins. Os modelos RNA não necessariamente precisam ser superiores aos modelos Box e Jenkins, mas tampouco obter EQM de previsão relativamente altos. Outras sugestões, ainda relativas ao uso de modelos RNA é a utilização de outros modelos, como Funções de Base Radial (RBF) ou ainda modelos dinâmicos como Redes Recorrentes. Ainda referente a modelos de RNA, outras alternativas de treinamento, ou até algoritmos de treinamento também poderiam ser utilizados.

Com relação aos modelos clássicos, sugere-se o uso de outras técnicas de análise de séries temporais, como decomposição e ajustamento exponencial. E ainda os modelos específicos de não linearidade, como os modelos TAR e STAR, não abordados nesta dissertação, também poderiam ser comparados.

REFERÊNCIAS

- ABRAHAM, B.; LEDOLTER, J. **Statistical methods for forecasting**. New York: Wiley, 2005. 445 p.
- AUER, P.; BURGSTEINER, H.; MAASS, W. A learning rule for very simple universal approximators consisting of a single layer of perceptrons. **Neural Networks**, v. 21, p. 786–795, 2008.
- BALKIN, S. D.; ORD, J. K. Automatic neural network modeling for univariate time series. **International Journal of Forecasting**, v. 16, p. 509–515, 2000.
- BIGUS, J. P. **Data mining with neural networks: Solving business problems from application development to decision support**. New York: McGraw-Hill, 1996. 220 p.
- BOX, G.; JENKINS, G. M. **Time series analysis**. San Francisco: Holden-Day, 1976.
- BOX, G.; JENKINS, G. M.; REINSEL, G. **Time series analysis: Forecasting and control**. 3. ed. New York: Prentice Hall, 1994. 592 p.
- BUSSAB, W. O.; MORETTIN, P. A. **Estatística Básica**. 5. ed. Rio de Janeiro: Saraiva, 2006. 540 p.
- CHAO, J. C.; PHILLIPS, P. C. B. Model selection in partially nonstationary vector autoregressive processes with reduced rank structure. **Journal of Econometrics**, v. 91, p. 227–271, 1999.
- CHEN, K.-Y. Using back-propagation neural networks to forecast the production values of the machinery industry in Taiwan. **Journal of American Academy of Business**, v. 9, n. 1, p. 183–190, 2006.
- CHENG, B.; TITTERINGTON, D. Neural networks: a review from a statistical perspective. **Statistical Science**, v. 9, n. 1, p. 2–30, 1994.
- CONGDON, P. **Applied bayesian modelling**. London: Wiley, 2003. 457 p.
- DE WILDE, P. **Neural networks models: theory and projects**. 2. ed. New York: Springer-Verlag, 1997. 174 p.
- DELLANA, S. A.; WEST, D. Predictive modeling for wastewater applications: linear and nonlinear approaches. **Environmental Modelling & Software**, v. 24, p. 96–106, 2009.
- ENDERS, W. **Applied econometric time series**. New York: Wiley, 1995. 433 p.
- ENGLE, R. F. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. **Econometrica**, v. 50, p. 987–1007, 1982.

- FERNANDES, L. G. L.; PORTUGAL, M. S.; NAVAUX, P. O. A. Previsão de séries de tempo: redes neurais artificiais e modelos estruturais. **Pesquisa e Planejamento Econômico**, v. 26, n. 2, p. 253–276, 1996.
- FLORES, J. H. F. Graduação, **Aplicação de redes neurais artificiais à previsão de vendas de máquinas agrícolas**: Um estudo de caso junto a AGCO do Brasil Ltda. Porto Alegre: [s.n.], 2006. 60 f.
- GAURI, S. K.; CHAKRABORTY, S. Feature-based recognition of control charts patterns. **Computers and Industrial Engineering**, v. 51, p. 726–742, 2006.
- GIANNELLIS, N.; PAPADOPOULOS, A. P. Testing for efficiency in selected developing foreign exchange markets: an equilibrium-based approach. **Economic Modelling**, v. 26, p. 155–166, 2009.
- GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 175 p.
- GRANGER, C. W. J.; KING, M. L.; WHITE, H. Comments on testing economic theories and the use of model selection criteria. **Journal of Econometrics**, v. 67, p. 173–187, 1995.
- HAIR, J. J. F. et al. **Análise multivariada de dados**. 5. ed. Porto Alegre: Bookman, 2005. 593 p.
- HAMILTON, J. D. **Time series analysis**. Princeton: Princeton University Press, 1994. 799 p.
- HAMZAÇEBI, C.; AKAY, D.; KUTAY, F. Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. **Expert Systems with Applications**, v. 36, p. 3839–3844, 2009.
- HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001. 900 p.
- HILL, T.; O’CONNOR, M.; REMUS, W. Neural network models for time series forecasts. **Management Science**, v. 42, n. 7, p. 1082–1092, Julho 1996.
- HOO, K. A.; SINZINGER, E. D.; PIOVOSO, M. J. Improvements in the predictive capability of neural networks. **Journal of Process Control**, v. 12, p. 193–202, 2002.
- INOUE, A.; KILIAN, L. On the selection of forecasting models. **Journal of Econometrics**, v. 130, p. 273–306, 2006.
- JAKUBEK, S.; STRASSER, T. Artificial neural networks for fault detection in large-scale data acquisition systems. **Engineering Applications of Artificial Intelligence**, v. 17, p. 233–248, 2004.
- KAASTRA, I.; BOYD, M. Designing a neural network for forecasting financial and economic time series. **Neurocomputing**, v. 10, p. 215–236, 1996.
- KIM, C.-J.; NELSON, C. R. **State-space models with regime switching**: Classical and Gibbs-sampling approaches with applications. Massachusetts: MIT press, 1999. 250 p.
- KRÖSE, B.; SMAGT, P. van der. **An introduction to neural networks**. The University of Amsterdam: URL ftp://ftp.informatik.uni-freiburg.de/papers/neuro/ann_intro_smag.ps.gz, 1996. Disponível em: <citeseer.ist.psu.edu/ose96introduction.html>.

- KUO, C.; REITSCH, A. Neural networks vs. conventional methods of forecasting. **The Journal of Business Forecasting Methods & Systems**, v. 14, n. 4, p. 17–22, 1995/1996.
- LAI, T. L.; WONG, S. P.-S. Stochastic neural networks with applications to nonlinear time series. **Journal of the American Statistical Association**, v. 96, n. 455, p. 968–981, 2001.
- LEE, T. H.; WHITE, H.; GRANGER, C. W. J. Testing for neglected nonlinearity in time series models. **Journal of Econometrics**, v. 56, p. 269–290, 1993.
- LIGHTBODY, G.; IRWIN, G. W. Multi-layer perceptron based modelling of nonlinear systems. **Fuzzy Sets and Systems**, n. 79, p. 93–112, 1996.
- MAKRIDAKIS, S.; WHEELWRIGHT, S. C.; HYNDMAN, R. J. **Forecasting Methods and Applications**. 3. ed. New York: Wiley, 1998. 642 p.
- MINGOTI, S. A. **Análise de dados através de métodos de estatística multivariada**: Uma abordagem aplicada. Belo Horizonte: UFMG, 2005. 295 p.
- MÜLLER, B.; REINHARDT, J.; STRICKLAND, M. T. **Neural networks**: an introduction. 2. ed. New York: Springer-Verlag, 1995. 330 p.
- MORETTIN, P. A.; TOLOI, C. M. C. **Análise de séries temporais**. 2. ed. São Paulo: Edgard Blücher, 2006. 538 p.
- NETER, J. et al. **Applied Linear Statistical Models**. 4. ed. New York: McGraw-Hill/Irwin, 1996. 1408 p.
- NIKOLAEV, N. Y.; IBA, H. Polynomial harmonic gmdh learning networks for time series modeling. **Neural Networks**, v. 16, p. 1527–1540, 2003.
- OH, S.-K.; PEDRYCZ, W. Genetic optimization-driven multi-layer hybrid fuzzy neural networks. **Simulation Modelling Practice and Theory**, v. 14, p. 597–613, 2006.
- OZCELIK, B.; ERZURUMLU, T. Comparison of the warpage optimization in the plastic injection molding using ANOVA, neural network model and genetic algorithm. **Journal of Materials Processing Technology**, v. 171, p. 437–445, 2006.
- PINTO, F. C. A.; MEDEIROS, M. C. Misspecified neural network models and linear time series forecasting. **Revista da Sociedade Brasileira de Redes Neurais**, Sociedade Brasileira de Redes Neurais, v. 1, n. 5, p. 317–326, 2005.
- PRIESTLEY, M. B. **Non-linear and non-stationary time series analysis**. London: Academic Press, 1991. 238 p.
- QI, M.; ZHANG, G. P. An investigation of model selection criteria for neural network time series forecasting. **European Journal of Operational Research**, v. 132, p. 666–680, 2001.
- R Development Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2007. ISBN 3-900051-07-0. Disponível em: <<http://www.R-project.org>>. Acessado em 7 de junho de 2009.
- RATHBUN, T. F. et al. MLP iterative construction algorithm. **Neurocomputing**, v. 17, p. 195–216, 1997.

- RIPLEY, B. D. **Pattern recognition and neural networks**. Cambridge: Cambridge University Press, 1996. 415 p.
- SHUMWAY, R. H.; STOFFER, D. S. **Time series analysis and its applications**. New York: Springer-Verlag, 2000. 549 p.
- SIN, C.-Y.; WHITE, H. Information criteria for selecting possibly misspecified models. **Journal of Econometrics**, v. 71, p. 207–225, 1996.
- SMITH, B. L.; WILLIAMS, B. M.; OSWALD, R. K. Comparison of parametric and nonparametric models for traffic flow forecasting. **Transportation Research Part C**, v. 10, p. 303–321, 2002.
- SUAREZ-FARINAS, M.; PEDREIRA, C. E.; MEDEIROS, M. C. Local global neural networks: a new approach for nonlinear time series modeling. **Journal of the American Statistical Association**, v. 99, p. 1092–2007, 2004.
- TABAK, B. M.; LIMA, E. J. Market efficiency of brazilian exchange rate: evidence from variance ratio statistics and technical trading rules. **European Journal of Operational Research**, v. 194, p. 814–820, 2009.
- TERÄSVIRTA, T.; LIN, C. F.; GRANGER, C. W. J. Power of the neural network linearity test. **Journal of Time Series Analysis**, v. 14, n. 2, p. 209–220, 1993.
- TERUI, N.; van Dijk, H. K. Combined forecasts from linear and nonlinear time series models. **International Journal of Forecasting**, v. 18, p. 421–438, 2002.
- TSAI, T.-I.; LI, D.-C. Approximate modeling for high order non-linear functions using small sample sets. **Expert Systems with Applications**, v. 34, p. 564–569, 2008.
- TSAY, R. S. Nonlinearity tests for time series. **Biometrika**, v. 73, n. 2, p. 461–466, 1986.
- VAPNIK, V. N.; CHERVONENKIS, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. **Theoretical Probability and Its Applications**, v. 17, p. 264–280, 1971.
- WANG, D.; LU, W.-Z. Forecasting of ozone level in time series using MLP model with a novel hybrid training algorithm. **Atmospheric Environment**, v. 40, p. 913–924, 2006.
- WANG, D.; LU, W.-Z. Forecasting of ozone level in time series using mlp model with a novel hybrid training algorithm. **Atmospheric Environment**, v. 40, p. 913–924, 2006.
- WANG, Z.; BESSLER, D. A. The homogeneity restriction and forecasting performance of VAR-type demand systems: an empirical examination of US meat consumption. **Journal of Forecasting**, v. 21, p. 193–206, 2002.
- ZHANG, G. P. An investigation of neural networks for linear time series forecasting. **Computers & Operations Research**, v. 28, p. 1183–1202, 2001.
- ZHANG, G. P.; BERARDI, V. L. Time series forecasting with neural network ensembles: an application for exchange rate prediction. **Journal of Operational Research Society**, v. 52, p. 652–664, 2001.

APÊNDICE A – CÓDIGO DO PROGRAMA R UTILIZADO NAS MO- DELAGENS

```

#! -----> SCRIPT PRINCIPAL <-----
# Carrega todos os scripts necessarios, alem das funcoes e
# dos bancos de dados.
# Carrega os bancos de dados
solar <- sunspot.year #serie das manchas solares
industria <- read.csv2("C:/DISSERTACAO/R/bd.csv") #serie industrial
producao <- ts(industria$prod, frequency=12, start=c(1991,1)) #producao mensal
logproducao <- ts(log(industria$prod), frequency=12, start=c(1991,1)) #venda mensal
# Ajusta variaveis globais
passos <- 12 #numero de passos para previsao
roda <- 100 #quant modelos gerados (mlp/rna)
# Ajusta os limites para as series de dados
ajuste.solar <- length(solar)-passos
ajuste.prod <- length(producao)-passos
ajuste.logprod <- length(producao)-passos
# Separa as séries para análise (ajuste e previsao)
solar.aju <- ts(solar[1:ajuste.solar],
               frequency = 1, start=1700) #manchas solares
prod.aju <- ts(producao[1:ajuste.prod],
              frequency=12, start=c(1991,1)) #ind producao
logprod.aju <- ts(log(producao[1:ajuste.prod]),
                 frequency=12, start=c(1991,1)) #ind vendas
# Inicializa uma semente para posterior reprodução
set.seed(21)
# Carrega os pacotes necessarios (ou que serao necessarios)
library(AMORE)
library(tseries)
library(nnet)
library(nlts)
library(xtable)
# Testa as series quanto a sua linearidade (Tukey, Terasvirta e White)
source("C:/DISSERTACAO/R/testes_series.r")
# Modelos ARIMA/SARIMA na serie de manchas solares
source("C:/DISSERTACAO/R/arima_solar.r")
# Saida dos resultados em latex
print(xtable(result.arima.solar, digits=5), type="latex",
      file="C:/DISSERTACAO/R/arima.solar.tex")
# Modelos ARIMA/SARIMA na serie de indice de producao
source("C:/DISSERTACAO/R/arima_prod.r")
# Saida dos resultados em latex
print(xtable(result.arima.prod, digits=5), type="latex",
      file="C:/DISSERTACAO/R/arima.prod.tex")
# Modelos ARIMA/SARIMA na serie do log do indice de producao
source("C:/DISSERTACAO/R/arima_logprod.r")
# Saida dos resultados em latex
print(xtable(result.arima.logprod, digits=5), type="latex",
      file="C:/DISSERTACAO/R/arima.logprod.tex")
# Modelos MLP/RNA com 6 neuronios - serie manchas solares
source("C:/DISSERTACAO/R/mlp_rna_solar_6.r")

```

```

mod.final.6.solar <- result.mod.6
matriz.final.6.solar <- matriz.final.6
# Saida dos resultados em latex
print(xtable(matriz.final.6.solar, digits=5), type="latex",
  file="C:/DISSERTACAO/R/matriz.solar.6.tex")
# Modelos MLP/RNA com 12 neuronios - serie manchas solares
source("C:/DISSERTACAO/R/mlp_rna_solar_12.r")
mod.final.12.solar <- result.mod.12
matriz.final.12.solar <- matriz.final.12
# Saida dos resultados em latex
print(xtable(matriz.final.12.solar, digits=5), type="latex",
  file="C:/DISSERTACAO/R/matriz.solar.12.tex")
# Modelos MLP/RNA com 6 neuronios - serie indice producao
source("C:/DISSERTACAO/R/mlp_rna_prod_6.r")
mod.final.6.prod <- result.mod.6
matriz.final.6.prod <- matriz.final.6
# Saida dos resultados em latex
print(xtable(matriz.final.6.prod, digits=5), type="latex",
  file="C:/DISSERTACAO/R/matriz.prod.6.tex")
# Modelos MLP/RNA com 12 neuronios - serie indice producao
source("C:/DISSERTACAO/R/mlp_rna_prod_12.r")
mod.final.12.prod <- result.mod.12
matriz.final.12.prod <- matriz.final.12
# Saida dos resultados em latex
print(xtable(matriz.final.12.prod, digits=5), type="latex",
  file="C:/DISSERTACAO/R/matriz.prod.12.tex")
# Modelos MLP/RNA com 6 neuronios - serie log indice producao
source("C:/DISSERTACAO/R/mlp_rna_logprod_6.r")
mod.final.6.logprod <- result.mod.6
matriz.final.6.logprod <- matriz.final.6
# Saida dos resultados em latex
print(xtable(matriz.final.6.logprod, digits=5), type="latex",
  file="C:/DISSERTACAO/R/matriz.logprod.6.tex")
# Modelos MLP/RNA com 12 neuronios - serie log indice producao
source("C:/DISSERTACAO/R/mlp_rna_logprod_12.r")
mod.final.12.logprod <- result.mod.12
matriz.final.12.logprod <- matriz.final.12
# Saida dos resultados em latex
print(xtable(matriz.final.12.logprod, digits=5), type="latex",
  file="C:/DISSERTACAO/R/matriz.logprod.12.tex")
==== TESTES DE LINEARIDADE APLICADOS AS SERIES
# Resultados dos testes de Tukey, Terasvirta e White
tukey.solar <- vector(length=4)
terasvirta.solar <- vector(length=11)
white.solar <- vector(length=11)
tukey.prod <- vector(length=4)
terasvirta.prod <- vector(length=11)
white.prod <- vector(length=11)
tukey.logprod <- vector(length=4)
terasvirta.logprod <- vector(length=11)
white.logprod <- vector(length=11)
# trabalhar com serie diferenciada (estacionaria)
mod.ind <- residuals(arima(producao, order=c(0,1,0)))
mod.log <- residuals(arima(log(producao), order=c(0,1,0)))

```

```

for (i in 1:4)
{
  tukey.solar[i] <- lin.test(solar, order=i+1)[[5]]
  tukey.prod[i] <- lin.test(mod.ind, order=i+1)[[5]]
  tukey.logprod[i] <- lin.test(mod.log, order=i+1)[[5]]
}
for (i in 1:11)
{
  terasvirta.solar[i] <- terasvirta.test(solar, lag=i+1, type = "F")$p.value
  terasvirta.prod[i] <- terasvirta.test(mod.ind, lag=i+1, type = "F")$p.value
  terasvirta.logprod[i] <- terasvirta.test(mod.log, lag=i+1, type = "F")$p.value
  white.solar[i] <- white.test(solar, lag=i+1, type="F")$p.value
  white.prod[i] <- white.test(mod.ind, lag=i+1, type="F")$p.value
  white.logprod[i] <- white.test(mod.log, lag=i+1, type="F")$p.value
}
#=== MODELOS ARIMA/SARIMA
# Modelo SARIMA(4,0,3)x(2,0,0)_5
mod.sarima <- arima(solar.aju, order=c(4,0,3), seasonal=list(order=c(2,0,0),
  period=5))
# Modelo ARMA(4,4)
mod.arma44 <- arima(solar.aju, order=c(4,0,4))
# Modelo ARMA(3,5)
mod.arma35 <- arima(solar.aju, order=c(3,0,5))
# Modelo ARMA(6,6)
mod.arma66 <- arima(solar.aju, order=c(6,0,6))
# Define listas para resultados
crit.mod.solar <- list()
# criterios AIC, BIC e EQM
crit.mod.solar[[1]] <- mod.sarima
crit.mod.solar[[1]]$bic <- bic(mod.sarima)
crit.mod.solar[[1]]$aic <- bic(mod.sarima, tipo="aic")
crit.mod.solar[[1]]$eqmaju <- mod.sarima$sigma2
crit.mod.solar[[2]] <- mod.arma44
crit.mod.solar[[2]]$bic <- bic(mod.arma44)
crit.mod.solar[[2]]$aic <- bic(mod.arma44, tipo="aic")
crit.mod.solar[[2]]$eqmaju <- mod.arma44$sigma2
crit.mod.solar[[3]] <- mod.arma35
crit.mod.solar[[3]]$bic <- bic(mod.arma35)
crit.mod.solar[[3]]$aic <- bic(mod.arma35, tipo="aic")
crit.mod.solar[[3]]$eqmaju <- mod.arma35$sigma2
crit.mod.solar[[4]] <- mod.arma66
crit.mod.solar[[4]]$bic <- bic(mod.arma66)
crit.mod.solar[[4]]$aic <- bic(mod.arma66, tipo="aic")
crit.mod.solar[[4]]$eqmaju <- mod.arma66$sigma2
# Testes dos parâmetros (caso seja necessário)
#coef(mod.1)/diag(vcov(mod.1)^.5)
# EQM de previsao dos modelos ARIMA/SARIMA
#variaveis temporarias
inicio <- length(solar)-(passos-1)
fim <- length(solar)
tam <- fim-inicio+1
var.dados <- sum((solar[inicio:fim] - mean(solar[inicio:fim]))^2)/
  (tam-1)
crit.mod.solar[[1]]$eqmprev <- sum((predict(crit.mod.solar[[1]], passos)$pred-

```

```

                                solar[inicio:fim])^2)/passos
crit.mod.solar[[1]]$nmse <- sum((predict(crit.mod.solar[[1]], passos)$pred-
                                solar[inicio:fim])^2)/(var.dados*tam)
crit.mod.solar[[1]]$mape <- 100 * sum(abs((predict(crit.mod.solar[[1]],
                                passos)$pred - solar[inicio:fim])/
                                solar[inicio:fim]))/tam
crit.mod.solar[[2]]$eqmprev <- sum((predict(crit.mod.solar[[2]], passos)$pred-
                                solar[inicio:fim])^2)/passos
crit.mod.solar[[2]]$nmse <- sum((predict(crit.mod.solar[[2]], passos)$pred-
                                solar[inicio:fim])^2)/(var.dados*tam)
crit.mod.solar[[2]]$mape <- 100 * sum(abs((predict(crit.mod.solar[[2]],
                                passos)$pred - solar[inicio:fim])/
                                solar[inicio:fim]))/tam
crit.mod.solar[[3]]$eqmprev <- sum((predict(crit.mod.solar[[3]], passos)$pred-
                                solar[inicio:fim])^2)/passos
crit.mod.solar[[3]]$nmse <- sum((predict(crit.mod.solar[[3]], passos)$pred-
                                solar[inicio:fim])^2)/(var.dados*tam)
crit.mod.solar[[3]]$mape <- 100 * sum(abs((predict(crit.mod.solar[[3]],
                                passos)$pred - solar[inicio:fim])/
                                solar[inicio:fim]))/tam
crit.mod.solar[[4]]$eqmprev <- sum((predict(crit.mod.solar[[4]], passos)$pred-
                                solar[inicio:fim])^2)/passos
crit.mod.solar[[4]]$nmse <- sum((predict(crit.mod.solar[[4]], passos)$pred-
                                solar[inicio:fim])^2)/(var.dados*tam)
crit.mod.solar[[4]]$mape <- 100 * sum(abs((predict(crit.mod.solar[[4]],
                                passos)$pred - solar[inicio:fim])/
                                solar[inicio:fim]))/tam

# Prepara exportacao de resultados em latex
matriz.arima.solar <- matrix(ncol=4, nrow=6)
# ciclo de resultados
for (i in 1:4)
{
  matriz.arima.solar[1,i] <- crit.mod.solar[[i]]$aic
  matriz.arima.solar[2,i] <- crit.mod.solar[[i]]$bic
  matriz.arima.solar[3,i] <- crit.mod.solar[[i]]$eqmaju
  matriz.arima.solar[4,i] <- crit.mod.solar[[i]]$eqmprev
  matriz.arima.solar[5,i] <- crit.mod.solar[[i]]$nmse
  matriz.arima.solar[6,i] <- crit.mod.solar[[i]]$mape
}
# data.frame
result.arima.solar <- as.data.frame(matriz.arima.solar)
# ajusta nomes dos modelos
names(result.arima.solar) <- c("SARIMA", "ARMA44", "ARMA35", "ARMA66")
# ajusta nomes dos criterios
rownames(result.arima.solar) <- c("AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",
                                "MAPE")

=== MODELOS ARIMA/SARIMA
# Modelo SARIMA(5,1,0)x(2,0,1)_6
mod.sarima.201 <- arima(prod.aju, order=c(5,1,0), seasonal=list(order=c(2,0,1),
                                period=6))
# Modelo SARIMA(2,1,3)x(2,1,2)_12
mod.sarima.212 <- arima(prod.aju, order=c(2,1,3), seasonal=list(order=c(2,1,2),
                                period=12))
# Modelo SARIMA(5,1,2)x(2,1,1)_12

```



```

mod.sarima.211 <- arima(prod.aju, order=c(5,1,2), seasonal=list(order=c(2,1,1),
  period=12))
# Define listas para resultados
crit.mod.prod <- list()
# criterios AIC, BIC e EQM
crit.mod.prod[[1]] <- mod.sarima.201
crit.mod.prod[[1]]$bic <- bic(mod.sarima.201)
crit.mod.prod[[1]]$aic <- bic(mod.sarima.201, tipo="aic")
crit.mod.prod[[1]]$eqmaj <- mod.sarima.201$sigma2
crit.mod.prod[[2]] <- mod.sarima.212
crit.mod.prod[[2]]$bic <- bic(mod.sarima.212)
crit.mod.prod[[2]]$aic <- bic(mod.sarima.212, tipo="aic")
crit.mod.prod[[2]]$eqmaj <- mod.sarima.212$sigma2
crit.mod.prod[[3]] <- mod.sarima.211
crit.mod.prod[[3]]$bic <- bic(mod.sarima.211)
crit.mod.prod[[3]]$aic <- bic(mod.sarima.211, tipo="aic")
crit.mod.prod[[3]]$eqmaj <- mod.sarima.211$sigma2
# Testes dos parâmetros (caso seja necessário)
#coef(mod.1)/diag(vcov(mod.1)^.5)
# EQM de previsao dos modelos ARIMA/SARIMA
#variaveis temporarias
inicio <- length(producao)-(passos-1)
fim <- length(producao)
tam <- fim-inicio+1
var.dados <- sum((producao[inicio:fim] - mean(producao[inicio:fim]))^2)/
  (tam-1)
crit.mod.prod[[1]]$eqmprev <- sum((predict(crit.mod.prod[[1]], passos)$pred-
  producao[inicio:fim])^2)/passos
crit.mod.prod[[1]]$nmse <- sum((predict(crit.mod.prod[[1]], passos)$pred-
  producao[inicio:fim])^2)/(var.dados*tam)
crit.mod.prod[[1]]$mape <- 100 * sum(abs((predict(crit.mod.prod[[1]],
  passos)$pred - producao[inicio:fim])/
  producao[inicio:fim]))/tam
crit.mod.prod[[2]]$eqmprev <- sum((predict(crit.mod.prod[[2]], passos)$pred-
  producao[inicio:fim])^2)/passos
crit.mod.prod[[2]]$nmse <- sum((predict(crit.mod.prod[[2]], passos)$pred-
  producao[inicio:fim])^2)/(var.dados*tam)
crit.mod.prod[[2]]$mape <- 100 * sum(abs((predict(crit.mod.prod[[2]],
  passos)$pred - producao[inicio:fim])/
  producao[inicio:fim]))/tam
crit.mod.prod[[3]]$eqmprev <- sum((predict(crit.mod.prod[[3]], passos)$pred-
  producao[inicio:fim])^2)/passos
crit.mod.prod[[3]]$nmse <- sum((predict(crit.mod.prod[[3]], passos)$pred-
  producao[inicio:fim])^2)/(var.dados*tam)
crit.mod.prod[[3]]$mape <- 100 * sum(abs((predict(crit.mod.prod[[3]],
  passos)$pred - producao[inicio:fim])/
  producao[inicio:fim]))/tam
# Prepara exportacao de resultados em latex
matriz.arima.prod <- matrix(ncol=3, nrow=6)
# ciclo de resultados
for (i in 1:3)
{
  matriz.arima.prod[1,i] <- crit.mod.prod[[i]]$aic
  matriz.arima.prod[2,i] <- crit.mod.prod[[i]]$bic

```

```

matriz.arima.prod[3,i] <- crit.mod.prod[[i]]$eqmaju
matriz.arima.prod[4,i] <- crit.mod.prod[[i]]$eqmprev
matriz.arima.prod[5,i] <- crit.mod.prod[[i]]$nmse
matriz.arima.prod[6,i] <- crit.mod.prod[[i]]$mape
}

# data.frame
result.arima.prod <- as.data.frame(matriz.arima.prod)
# ajusta nomes dos modelos
names(result.arima.prod) <- c("SARIMA.201", "SARIMA.212", "SARIMA.211")
# ajusta nomes dos criterios
rownames(result.arima.prod) <- c("AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",
                                "MAPE")

#### MODELOS ARIMA/SARIMA
# Modelo SARIMA(2,1,6)x(1,1,0)_12
mod.sarima.216 <- arima(logprod.aju, order=c(2,1,6),
                      seasonal=list(order=c(1,1,0), period=12))
# Modelo SARIMA(6,1,2)x(1,1,0)_12
mod.sarima.612 <- arima(logprod.aju, order=c(6,1,2),
                      seasonal=list(order=c(1,1,0), period=12))
# Modelo SARIMA(3,0,0)x(2,1,0)_6
mod.sarima.300 <- arima(logprod.aju, order=c(3,0,0),
                      seasonal=list(order=c(2,1,0), period=6))
# Define listas para resultados
crit.mod.logprod <- list()
# criterios AIC, BIC e EQM
crit.mod.logprod[[1]] <- mod.sarima.216
crit.mod.logprod[[1]]$bic <- bic(mod.sarima.216)
crit.mod.logprod[[1]]$aic <- bic(mod.sarima.216, tipo="aic")
crit.mod.logprod[[1]]$eqmaju <- mod.sarima.216$sigma2
crit.mod.logprod[[2]] <- mod.sarima.612
crit.mod.logprod[[2]]$bic <- bic(mod.sarima.612)
crit.mod.logprod[[2]]$aic <- bic(mod.sarima.612, tipo="aic")
crit.mod.logprod[[2]]$eqmaju <- mod.sarima.612$sigma2
crit.mod.logprod[[3]] <- mod.sarima.300
crit.mod.logprod[[3]]$bic <- bic(mod.sarima.300)
crit.mod.logprod[[3]]$aic <- bic(mod.sarima.300, tipo="aic")
crit.mod.logprod[[3]]$eqmaju <- mod.sarima.300$sigma2
# Testes dos parâmetros (caso seja necessário)
#coef(mod.1)/diag(vcov(mod.1)^.5)
# EQM de previsao dos modelos ARIMA/SARIMA
#variaveis temporarias
inicio <- length(logproducao)-(passos-1)
fim <- length(logproducao)
tam <- fim-inicio+1
var.dados <- sum((logproducao[inicio:fim] - mean(logproducao[inicio:fim]))^2)/
              (tam-1)
crit.mod.logprod[[1]]$eqmprev <- sum((predict(crit.mod.logprod[[1]], passos)$pred-
logproducao[inicio:fim])^2)/passos
crit.mod.logprod[[1]]$nmse <- sum((predict(crit.mod.logprod[[1]], passos)$pred-
logproducao[inicio:fim])^2)/(var.dados*tam)
crit.mod.logprod[[1]]$mape <- 100 * sum(abs((predict(crit.mod.logprod[[1]],
passos)$pred - logproducao[inicio:fim])/
logproducao[inicio:fim]))/tam
crit.mod.logprod[[2]]$eqmprev <- sum((predict(crit.mod.logprod[[2]], passos)$pred-
```

```

                                logproducao[inicio:fim])^2)/passos
crit.mod.logprod[[2]]$nmse <- sum((predict(crit.mod.logprod[[2]], passos)$pred-
                                logproducao[inicio:fim])^2)/(var.dados*tam)
crit.mod.logprod[[2]]$mape <- 100 * sum(abs((predict(crit.mod.logprod[[2]],
                                passos)$pred - logproducao[inicio:fim])/
                                logproducao[inicio:fim]))/tam
crit.mod.logprod[[3]]$eqmprev <- sum((predict(crit.mod.logprod[[3]], passos)$pred-
                                logproducao[inicio:fim])^2)/passos
crit.mod.logprod[[3]]$nmse <- sum((predict(crit.mod.logprod[[3]], passos)$pred-
                                logproducao[inicio:fim])^2)/(var.dados*tam)
crit.mod.logprod[[3]]$mape <- 100 * sum(abs((predict(crit.mod.logprod[[3]],
                                passos)$pred - logproducao[inicio:fim])/
                                logproducao[inicio:fim]))/tam

# Prepara exportacao de resultados em latex
matriz.arima.logprod <- matrix(ncol=3, nrow=6)
# ciclo de resultados
for (i in 1:3)
{
  matriz.arima.logprod[1,i] <- crit.mod.logprod[[i]]$aic
  matriz.arima.logprod[2,i] <- crit.mod.logprod[[i]]$bic
  matriz.arima.logprod[3,i] <- crit.mod.logprod[[i]]$eqmaju
  matriz.arima.logprod[4,i] <- crit.mod.logprod[[i]]$eqmprev
  matriz.arima.logprod[5,i] <- crit.mod.logprod[[i]]$nmse
  matriz.arima.logprod[6,i] <- crit.mod.logprod[[i]]$mape
}

# data.frame
result.arima.logprod <- as.data.frame(matriz.arima.logprod)
# ajusta nomes dos modelos
names(result.arima.logprod) <- c("SARIMA.216", "SARIMA.612", "SARIMA.300")
# ajusta nomes dos criterios
rownames(result.arima.logprod) <- c("AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",
                                "MAPE")

#### MODELOS MLP/RNA COM 6 NEURONIOS NA CAMADA DE ENTRADA
# Semente aleatória para reprodutibilidade
set.seed(21)
# Ajusta as matrizes de treino/ajuste e previsao
prod.rna.6 <- gera.rna(prod.aju, 6)
prod.teste.6 <- gera.rna(producao,6)
prod.teste.6.1 <- prod.teste.6[(nrow(prod.teste.6)-(passos-1)):
                                nrow(prod.teste.6),1:6]
prod.teste.6 <- prod.teste.6[(nrow(prod.teste.6)-(passos-1)):
                                nrow(prod.teste.6),7]

# Para avaliar o tempo necessario
tempo.6.prod <- Sys.time()
# Prepara vetores de resultados
matriz.res <- matrix(ncol=7, nrow=roda)
temp.mod <- list()
result.mod.6 <- list()
matriz.final.6 <- matrix(ncol=8, nrow=26)
for (j in 5:30)
{
  for (i in 1:roda)
  {
    amostra <- sample(1:nrow(prod.rna.6), round(nrow(prod.rna.6)*.7))

```

```

mod <- nnet(prod.rna.6[amostra,1:6], prod.rna.6[amostra,7], size=j,
           linout = TRUE, rang = 0.008, decay = 0, maxit = 10000,
           trace = TRUE, MaxNWts = 1000, abstol = 1.0e-5,
           reltol = 1.0e-5)
res.mod <- predict(mod, prod.rna.6[,1:6], type="raw")-prod.rna.6[,7]
tam <- length(res.mod)
param <- length(mod$wts)
num <- log(sum(residuals(mod)^2)/tam)
matriz.res[i,1] <- num + (2*log(param))/tam #AIC
matriz.res[i,2] <- num + (param*log(tam))/tam #BIC
matriz.res[i,3] <- sum(res.mod^2)/tam #EQM de ajuste
#EQM de previsao
matriz.res[i,4] <- sum((prod.teste.6-
                    pred.nnet(prod.rna.6, passos, mod)[,7])^2)/passos
#NMSE
var.dados.6 <- sum((prod.teste.6 - mean(prod.teste.6))^2)/
              (passos-1)
matriz.res[i,5] <- sum((prod.teste.6 -
                    pred.nnet(prod.rna.6, passos, mod)[,7])^2)/
              (var.dados.6*passos)

#MAPE
peres.mod <- (prod.teste.6 -
             pred.nnet(prod.rna.6, passos, mod)[,7])/prod.teste.6
matriz.res[i,6] <- sum(abs(peres.mod))*100/passos
#EQM de 1 passo
matriz.res[i,7] <- sum((prod.teste.6 -
                    predict(mod, prod.teste.6.1, type="raw"))^2)/passos
#SALVA O MODELO
temp.mod[[i]] <- mod #salva o modelo treinado
}

# Encontra o modelo com menor EQM de previsao dentre os 100 gerados
max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
temp <- max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
# Grava os resultados do melhor modelo entre os 100 gerados
matriz.final.6[(j-4),1] <- j
matriz.final.6[(j-4),2] <- matriz.res[temp,1]
matriz.final.6[(j-4),3] <- matriz.res[temp,2]
matriz.final.6[(j-4),4] <- matriz.res[temp,3]
matriz.final.6[(j-4),5] <- matriz.res[temp,4]
matriz.final.6[(j-4),6] <- matriz.res[temp,5]
matriz.final.6[(j-4),7] <- matriz.res[temp,6]
matriz.final.6[(j-4),8] <- matriz.res[temp,7]
# Grava o melhor modelo entre os 100
result.mod.6[[j-4]] <- temp.mod[[temp]]
}

# Tempo final de computacao
tempo.6.prod <- Sys.time()-tempo.6.prod
# Ajusta os nomes na matrix final
matriz.final.6 <- as.data.frame(matriz.final.6)
names(matriz.final.6) <- c("N", "AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",
                        "MAPE", "EQM1")

=== MODELOS MLP/RNA COM 12 NEURONIOS NA CAMADA DE ENTRADA
# Semente aleatória para reprodutibilidade
set.seed(21)

```

```

# Ajusta as matrizes de treino/ajuste e previsao
prod.rna.12 <- gera.rna(prod.aju, 12)
prod.teste.12 <- gera.rna(producao,12)
prod.teste.12.1 <- prod.teste.12[(nrow(prod.teste.12)-(passos-1)):
                               nrow(prod.teste.12),1:12]
prod.teste.12 <- prod.teste.12[(nrow(prod.teste.12)-(passos-1)):
                               nrow(prod.teste.12),13]
# Para avaliar o tempo necessario
tempo.12.prod <- Sys.time()
# Prepara vetores de resultados
matriz.res <- matrix(ncol=7, nrow=roda)
temp.mod <- list()
result.mod.12 <- list()
matriz.final.12 <- matrix(ncol=8, nrow=26)
for (j in 5:30)
{
  for (i in 1:roda)
  {
    amostra <- sample(1:nrow(prod.rna.12), round(nrow(prod.rna.12)*.7))
    mod <- nnet(prod.rna.12[amostra,1:12], prod.rna.12[amostra,13], size=j,
               linout = TRUE, rang = 0.008, decay = 0, maxit = 10000,
               trace = TRUE, MaxNWts = 1000, abstol = 1.0e-5,
               reltol = 1.0e-5)
    res.mod <- predict(mod, prod.rna.12[,1:12], type="raw")-prod.rna.12[,13]
    tam <- length(res.mod)
    param <- length(mod$wts)
    num <- log(sum(residuals(mod)^2)/tam)
    matriz.res[i,1] <- num + (2*log(param))/tam #AIC
    matriz.res[i,2] <- num + (param*log(tam))/tam #BIC
    matriz.res[i,3] <- sum(res.mod^2)/tam #EQM de ajuste
    #EQM de previsao
    matriz.res[i,4] <- sum((prod.teste.12-
                          pred.nnet(prod.rna.12, passos, mod)[,13])^2)/passos
    #NMSE
    var.dados.12 <- sum((prod.teste.12 - mean(prod.teste.12))^2)/
                    (passos-1)
    matriz.res[i,5] <- sum((prod.teste.12 -
                          pred.nnet(prod.rna.12, passos, mod)[,13])^2)/
                    (var.dados.12*passos)
    #MAPE
    peres.mod <- (prod.teste.12 -
                 pred.nnet(prod.rna.12, passos, mod)[,13])/prod.teste.12
    matriz.res[i,6] <- sum(abs(peres.mod))*100/passos
    #EQM de 1 passo
    matriz.res[i,7] <- sum((prod.teste.12 -
                          predict(mod, prod.teste.12.1, type="raw"))^2)/passos
    #SALVA O MODELO
    temp.mod[[i]] <- mod #salva o modelo treinado
  }
}
# Encontra o modelo com menor EQM de previsao dentre os 100 gerados
max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
temp <- max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
# Grava os resultados do melhor modelo entre os 100 gerados
matriz.final.12[(j-4),1] <- j

```

```

matriz.final.12[(j-4),2] <- matriz.res[temp,1]
matriz.final.12[(j-4),3] <- matriz.res[temp,2]
matriz.final.12[(j-4),4] <- matriz.res[temp,3]
matriz.final.12[(j-4),5] <- matriz.res[temp,4]
matriz.final.12[(j-4),6] <- matriz.res[temp,5]
matriz.final.12[(j-4),7] <- matriz.res[temp,6]
matriz.final.12[(j-4),8] <- matriz.res[temp,7]
# Grava o melhor modelo entre os 100
result.mod.12[[j-4]] <- temp.mod[[temp]]
}
# Tempo final de computacao
tempo.12.prod <- Sys.time()-tempo.12.prod
# Ajusta os nomes na matrix final
matriz.final.12 <- as.data.frame(matriz.final.12)
names(matriz.final.12) <- c("N", "AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",
                          "MAPE", "EQM1")
#### MODELOS MLP/RNA COM 6 NEURONIOS NA CAMADA DE ENTRADA
# Semente aleatória para reprodutibilidade
set.seed(21)
# Ajusta as matrizes de treino/ajuste e previsao
logprod.rna.6 <- gera.rna(logprod.aju, 6)
logprod.teste.6 <- gera.rna(logproducao,6)
logprod.teste.6.1 <- logprod.teste.6[(nrow(logprod.teste.6)-(passos-1)):
                                     nrow(logprod.teste.6),1:6]
logprod.teste.6 <- logprod.teste.6[(nrow(logprod.teste.6)-(passos-1)):
                                     nrow(logprod.teste.6),7]
# Para avaliar o tempo necessario
tempo.6.logprod <- Sys.time()
# Prepara vetores de resultados
matriz.res <- matrix(ncol=7, nrow=roda)
temp.mod <- list()
result.mod.6 <- list()
matriz.final.6 <- matrix(ncol=8, nrow=26)
for (j in 5:30)
{
  for (i in 1:roda)
  {
    amostra <- sample(1:nrow(logprod.rna.6), round(nrow(logprod.rna.6)*.7))
    mod <- nnet(logprod.rna.6[amostra,1:6], logprod.rna.6[amostra,7], size=j,
               linout = TRUE, rang = 0.008, decay = 0, maxit = 10000,
               trace = TRUE, MaxNWts = 1000, abstol = 1.0e-5,
               reltol = 1.0e-5)
    res.mod <- predict(mod, logprod.rna.6[,1:6], type="raw")-logprod.rna.6[,7]
    tam <- length(res.mod)
    param <- length(mod$wts)
    num <- log(sum(residuals(mod)^2)/tam)
    matriz.res[i,1] <- num + (2*log(param))/tam #AIC
    matriz.res[i,2] <- num + (param*log(tam))/tam #BIC
    matriz.res[i,3] <- sum(res.mod^2)/tam #EQM de ajuste
    #MEDIDAS DE PREVISAO
    #EQM de previsao
    matriz.res[i,4] <- sum((logprod.teste.6-
                          pred.nnet(logprod.rna.6, passos, mod)[,7])^2)/passos
    #NMSE

```

```

var.dados.6 <- sum((logprod.teste.6 - mean(logprod.teste.6))^2)/
  (passos-1)
matriz.res[i,5] <- sum((logprod.teste.6 -
  pred.nnet(logprod.rna.6, passos, mod)[,7])^2)/
  (var.dados.6*passos)

#MAPE
peres.mod <- (logprod.teste.6 -
  pred.nnet(logprod.rna.6, passos, mod)[,7])/logprod.teste.6
matriz.res[i,6] <- sum(abs(peres.mod))*100/passos
#EQM de 1 passo
matriz.res[i,7] <- sum((logprod.teste.6 -
  predict(mod, logprod.teste.6.1, type="raw"))^2)/passos
#SALVA O MODELO
temp.mod[[i]] <- mod #salva o modelo treinado
}

# Encontra o modelo com menor EQM de previsao dentre os 100 gerados
max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
temp <- max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
# Grava os resultados do melhor modelo entre os 100 gerados
matriz.final.6[(j-4),1] <- j
matriz.final.6[(j-4),2] <- matriz.res[temp,1]
matriz.final.6[(j-4),3] <- matriz.res[temp,2]
matriz.final.6[(j-4),4] <- matriz.res[temp,3]
matriz.final.6[(j-4),5] <- matriz.res[temp,4]
matriz.final.6[(j-4),6] <- matriz.res[temp,5]
matriz.final.6[(j-4),7] <- matriz.res[temp,6]
matriz.final.6[(j-4),8] <- matriz.res[temp,7]
# Grava o melhor modelo entre os 100
result.mod.6[[j-4]] <- temp.mod[[temp]]
}

# Tempo final de computacao
tempo.6.logprod <- Sys.time()-tempo.6.logprod
# Ajusta os nomes na matrix final
matriz.final.6 <- as.data.frame(matriz.final.6)
names(matriz.final.6) <- c("N", "AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",
  "MAPE", "EQM1")

=== MODELOS MLP/RNA COM 12 NEURONIOS NA CAMADA DE ENTRADA
# Semente aleatória para reprodutibilidade
set.seed(21)
# Ajusta as matrizes de treino/ajuste e previsao
logprod.rna.12 <- gera.rna(logprod.aju, 12)
logprod.teste.12 <- gera.rna(logproducao,12)
logprod.teste.12.1 <- logprod.teste.12[(nrow(logprod.teste.12)-(passos-1)):
  nrow(logprod.teste.12),1:12]
logprod.teste.12 <- logprod.teste.12[(nrow(logprod.teste.12)-(passos-1)):
  nrow(logprod.teste.12),13]

# Para avaliar o tempo necessario
tempo.12.logprod <- Sys.time()
# Prepara vetores de resultados
matriz.res <- matrix(ncol=7, nrow=roda)
temp.mod <- list()
result.mod.12 <- list()
matriz.final.12 <- matrix(ncol=8, nrow=26)
for (j in 5:30)

```

```

{
for (i in 1:roda)
{
  amostra <- sample(1:nrow(logprod.rna.12), round(nrow(logprod.rna.12)*.7))
  mod <- nnet(logprod.rna.12[amostra,1:12], logprod.rna.12[amostra,7], size=j,
             linout = TRUE, rang = 0.008, decay = 0, maxit = 10000,
             trace = TRUE, MaxNWts = 1000, abstol = 1.0e-5,
             reltol = 1.0e-5)
  res.mod <- predict(mod, logprod.rna.12[,1:12], type="raw")-logprod.rna.12[,13]
  tam <- length(res.mod)
  param <- length(mod$wts)
  num <- log(sum(residuals(mod)^2)/tam)
  matriz.res[i,1] <- num + (2*log(param))/tam #AIC
  matriz.res[i,2] <- num + (param*log(tam))/tam #BIC
  matriz.res[i,3] <- sum(res.mod^2)/tam #EQM de ajuste
  #EQM de previsao
  matriz.res[i,4] <- sum((logprod.teste.12-
                       pred.nnet(logprod.rna.12, passos, mod)[,13])^2)/passos
  #NMSE
  var.dados.12 <- sum((logprod.teste.12 - mean(logprod.teste.12))^2)/
                (passos-1)
  matriz.res[i,5] <- sum((logprod.teste.12 -
                       pred.nnet(logprod.rna.12, passos, mod)[,13])^2)/
                (var.dados.12*passos)
  #MAPE
  peres.mod <- (logprod.teste.12 -
               pred.nnet(logprod.rna.12, passos, mod)[,13])/logprod.teste.12
  matriz.res[i,6] <- sum(abs(peres.mod))*100/passos
  #EQM de 1 passo
  matriz.res[i,7] <- sum((logprod.teste.12 -
                       predict(mod, logprod.teste.12.1, type="raw"))^2)/passos
  #SALVA O MODELO
  temp.mod[[i]] <- mod #salva o modelo treinado
}
}

# Encontra o modelo com menor EQM de previsao dentre os 100 gerados
max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
temp <- max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
# Grava os resultados do melhor modelo entre os 100 gerados
matriz.final.12[(j-4),1] <- j
matriz.final.12[(j-4),2] <- matriz.res[temp,1]
matriz.final.12[(j-4),3] <- matriz.res[temp,2]
matriz.final.12[(j-4),4] <- matriz.res[temp,3]
matriz.final.12[(j-4),5] <- matriz.res[temp,4]
matriz.final.12[(j-4),6] <- matriz.res[temp,5]
matriz.final.12[(j-4),7] <- matriz.res[temp,6]
matriz.final.12[(j-4),8] <- matriz.res[temp,7]
# Grava o melhor modelo entre os 100
result.mod.12[(j-4)] <- temp.mod[[temp]]
}

# Tempo final de computacao
tempo.12.logprod <- Sys.time()-tempo.12.logprod
# Ajusta os nomes na matrix final
matriz.final.12 <- as.data.frame(matriz.final.12)
names(matriz.final.12) <- c("N", "AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",

```



```

"MAPE", "EQM1")
#### MODELOS MLP/RNA COM 6 NEURONIOS NA CAMADA DE ENTRADA
# Semente aleatória para reprodutibilidade
set.seed(21)
# Ajusta as matrizes de treino/ajuste e previsao
solar.rna.6 <- gera.rna(solar.aju, 6)
solar.teste.6 <- gera.rna(solar,6)
solar.teste.6.1 <- solar.teste.6[(nrow(solar.teste.6)-(passos-1)):
                               nrow(solar.teste.6),1:6]
solar.teste.6 <- solar.teste.6[(nrow(solar.teste.6)-(passos-1)):
                               nrow(solar.teste.6),7]
# Para avaliar o tempo necessario
tempo.6.solar <- Sys.time()
# Prepara vetores de resultados
matriz.res <- matrix(ncol=7, nrow=roda)
temp.mod <- list()
result.mod.6 <- list()
matriz.final.6 <- matrix(ncol=8, nrow=26)
for (j in 5:30)
{
  for (i in 1:roda)
  {
    amostra <- sample(1:nrow(solar.rna.6), round(nrow(solar.rna.6)*.7))
    amostra <- amostra[order(amostra)]
    mod <- nnet(solar.rna.6[amostra,1:6], solar.rna.6[amostra,7], size=j,
               linout = TRUE, rang = 0.003, decay = 0, maxit = 10000,
               trace = TRUE, MaxNWts = 1000, abstol = 1.0e-5,
               reltol = 1.0e-5, control=list(trace=-1))
    res.mod <- predict(mod, solar.rna.6[,1:6], type="raw")-solar.rna.6[,7]
    tam <- length(res.mod)
    param <- length(mod$wts)
    num <- log(sum(residuals(mod)^2)/tam)
    matriz.res[i,1] <- num + (2*log(param))/tam #AIC
    matriz.res[i,2] <- num + (param*log(tam))/tam #BIC
    matriz.res[i,3] <- sum(res.mod^2)/tam #EQM de ajuste
    #EQM de previsao
    matriz.res[i,4] <- sum((solar.teste.6-
                          pred.nnet(solar.rna.6, passos, mod)[,7])^2)/passos
    #NMSE
    var.dados.6 <- sum((solar.teste.6 - mean(solar.teste.6))^2)/
                  (passos-1)
    matriz.res[i,5] <- sum((solar.teste.6 -
                          pred.nnet(solar.rna.6, passos, mod)[,7])^2)/
                  (var.dados.6*passos)
    #MAPE
    peres.mod <- (solar.teste.6 -
                  pred.nnet(solar.rna.6, passos, mod)[,7])/solar.teste.6
    matriz.res[i,6] <- sum(abs(peres.mod))*100/passos
    #EQM de 1 passo
    matriz.res[i,7] <- sum((solar.teste.6 -
                          predict(mod, solar.teste.6.1, type="raw"))^2)/passos
    #SALVA O MODELO
    temp.mod[[i]] <- mod #salva o modelo treinado
  }
}

```

```

# Encontra o modelo com menor EQM de previsao dentre os 100 gerados
max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
temp <- max((matriz.res[,] %in% min(matriz.res[,4]))*row(matriz.res))
# Grava os resultados do melhor modelo entre os 100 gerados
matriz.final.6[(j-4),1] <- j
matriz.final.6[(j-4),2] <- matriz.res[temp,1]
matriz.final.6[(j-4),3] <- matriz.res[temp,2]
matriz.final.6[(j-4),4] <- matriz.res[temp,3]
matriz.final.6[(j-4),5] <- matriz.res[temp,4]
matriz.final.6[(j-4),6] <- matriz.res[temp,5]
matriz.final.6[(j-4),7] <- matriz.res[temp,6]
matriz.final.6[(j-4),8] <- matriz.res[temp,7]
# Grava o melhor modelo entre os 100
result.mod.6[[j-4]] <- temp.mod[[temp]]
}
# Tempo final de computacao
tempo.6.solar <- Sys.time()-tempo.6.solar
# Ajusta os nomes na matrix final
matriz.final.6 <- as.data.frame(matriz.final.6)
names(matriz.final.6) <- c("N", "AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",
                          "MAPE", "EQM1")
#### MODELOS MLP/RNA COM 12 NEURONIOS NA CAMADA DE ENTRADA
# Semente aleatória para reprodutibilidade
set.seed(21)
# Ajusta as matrizes de treino/ajuste e previsao
solar.rna.12 <- gera.rna(solar.aju, 12)
solar.teste.12 <- gera.rna(solar,12)
solar.teste.12.1 <- solar.teste.12[(nrow(solar.teste.12)-(passos-1)):
                                   nrow(solar.teste.12),1:12]
solar.teste.12 <- solar.teste.12[(nrow(solar.teste.12)-(passos-1)):
                                   nrow(solar.teste.12),7]
# Para avaliar o tempo necessario
tempo.12.solar <- Sys.time()
# Prepara vetores de resultados
matriz.res <- matrix(ncol=7, nrow=roda)
temp.mod <- list()
result.mod.12 <- list()
matriz.final.12 <- matrix(ncol=8, nrow=26)
for (j in 5:30)
{
  for (i in 1:roda)
  {
    amostra <- sample(1:nrow(solar.rna.12), round(nrow(solar.rna.12)*.7))
    mod <- nnet(solar.rna.12[amostra,1:12], solar.rna.12[amostra,13], size=j,
               linout = TRUE, rang = 0.008, decay = 0, maxit = 10000,
               trace = TRUE, MaxNWts = 1000, abstol = 1.0e-5,
               reltol = 1.0e-5)
    res.mod <- predict(mod, solar.rna.12[,1:12], type="raw")-solar.rna.12[,13]
    tam <- length(res.mod)
    param <- length(mod$wts)
    num <- log(sum(residuals(mod)^2)/tam)
    matriz.res[i,1] <- num + (2*log(param))/tam #AIC
    matriz.res[i,2] <- num + (param*log(tam))/tam #BIC
    matriz.res[i,3] <- sum(res.mod^2)/tam #EQM de ajuste
  }
}

```

```

#EQM de previsao
matriz.res[i,4] <- sum((solar.teste.12-
                    pred.nnet(solar.rna.12, passos, mod)[,13])^2)/passos

#NMSE
var.dados.12 <- sum((solar.teste.12 - mean(solar.teste.12))^2)/
                (passos-1)
matriz.res[i,5] <- sum((solar.teste.12 -
                    pred.nnet(solar.rna.12, passos, mod)[,13])^2)/
                (var.dados.12*passos)

#MAPE
peres.mod <- (solar.teste.12 -
              pred.nnet(solar.rna.12, passos, mod)[,13])/solar.teste.12
matriz.res[i,6] <- sum(abs(peres.mod))*100/passos
#EQM de 1 passo
matriz.res[i,7] <- sum((solar.teste.12 -
                    predict(mod, solar.teste.12.1, type="raw"))^2)/passos
#SALVA O MODELO
temp.mod[[i]] <- mod #salva o modelo treinado
}

# Encontra o modelo com menor EQM de previsao dentre os 100 gerados
max((matriz.res[,j] %in% min(matriz.res[,4]))*row(matriz.res))
temp <- max((matriz.res[,j] %in% min(matriz.res[,4]))*row(matriz.res))
# Grava os resultados do melhor modelo entre os 100 gerados
matriz.final.12[(j-4),1] <- j
matriz.final.12[(j-4),2] <- matriz.res[temp,1]
matriz.final.12[(j-4),3] <- matriz.res[temp,2]
matriz.final.12[(j-4),4] <- matriz.res[temp,3]
matriz.final.12[(j-4),5] <- matriz.res[temp,4]
matriz.final.12[(j-4),6] <- matriz.res[temp,5]
matriz.final.12[(j-4),7] <- matriz.res[temp,6]
matriz.final.12[(j-4),8] <- matriz.res[temp,7]
# Grava o melhor modelo entre os 100
result.mod.12[(j-4)] <- temp.mod[[temp]]
}

# Tempo final de computacao
tempo.12.solar <- Sys.time()-tempo.12.solar
# Ajusta os nomes na matrix final
matriz.final.12 <- as.data.frame(matriz.final.12)
names(matriz.final.12) <- c("N", "AIC", "BIC", "EQMAju", "EQMPrev", "NMSE",
                          "MAPE", "EQM1")

```

APÊNDICE B – CRITÉRIOS DOS MODELOS MLP/RNA

B.1 Critérios dos modelos MLP/RNA gerados com 6 neurônios na camada de entrada, aplicados a série do índice de produção

Oculto	AIC	BIC	EQM Aju	EQM Prev (12)	NMSE	MAPE	EQM Prev (1)
5	4.66983	5.74933	157.23644	311.06193	5.72520	14.22359	311.06193
6	4.73767	6.03346	156.77272	316.57914	5.82675	14.37740	316.57914
7	3.61713	5.12950	52.49902	115.11038	2.11865	8.01579	63.38549
8	3.63641	5.36556	52.25326	164.54591	3.02852	9.54379	81.78444
9	3.73279	5.67888	54.05271	122.32536	2.25144	8.29690	96.72306
10	2.88794	5.05110	40.41348	58.98116	1.08557	6.19965	82.09614
11	4.66132	7.04164	157.56766	307.51806	5.65997	14.12394	307.51806
12	3.64132	6.23889	51.97364	145.92183	2.68574	9.05257	77.60009
13	3.58250	6.39740	48.80174	177.11385	3.25984	9.87451	65.19507
14	4.68819	7.72046	156.42408	321.31118	5.91384	14.50806	321.31118
15	3.69421	6.94392	51.91694	142.07782	2.61499	8.93672	79.83954
16	3.52489	6.99208	52.08022	122.35866	2.25205	8.25370	63.34307
17	3.77555	7.46026	62.85313	132.47712	2.43829	8.62129	131.65186
18	3.61389	7.51615	50.29490	80.30936	1.47812	6.84586	80.98346
19	3.66528	7.78513	52.25326	125.61222	2.31194	8.33700	68.12455
20	3.13577	7.47323	100.42232	105.16984	1.93569	7.76326	72.79821
21	3.61169	8.16679	52.75123	117.32027	2.15932	8.09237	61.91553
22	3.64002	8.41279	52.91647	110.75763	2.03853	7.97828	73.79031
23	3.57790	8.56835	52.52973	109.83628	2.02157	7.94282	73.53245
24	3.54912	8.75727	50.24793	96.27021	1.77189	7.32625	85.62350
25	3.63056	9.05644	53.16295	129.30732	2.37995	8.46943	70.60532
26	3.59688	9.24049	56.08805	96.14020	1.76949	7.52349	79.09834
27	3.64272	9.50409	52.47427	108.32981	1.99385	7.83487	59.54672
28	3.39598	9.47512	46.83962	67.17987	1.23647	6.30572	77.20757
29	3.53799	9.83491	53.13336	108.62128	1.99921	7.79490	56.45801
30	3.54583	10.06054	52.57433	90.94407	1.67386	7.28259	59.13570

B.2 Critérios dos modelos MLP/RNA gerados com 12 neurônios na camada de entrada, aplicados a série do índice de produção

Oculto	AIC	BIC	EQM Aju	EQM Prev (12)	NMSE	MAPE	EQM Prev (1)
5	2.90473	4.84528	30.57371	35.11332	0.64627	4.87810	39.38175
6	3.11086	5.44112	34.97383	49.24989	0.90646	5.55627	37.46754
7	3.13500	5.85526	34.98943	82.26467	1.51411	7.19233	44.87386
8	3.14584	6.25632	35.46915	65.60068	1.20740	6.49268	44.42414
9	1.54243	5.04329	23.27552	56.02847	1.03122	5.15366	34.14606
10	3.10584	6.99722	35.70004	46.52482	0.85631	5.40332	38.69717
11	3.34828	7.63028	34.07366	47.71249	0.87816	5.51602	34.74182
12	4.55430	9.22701	137.55006	290.72317	5.35086	13.64207	290.72317
13	3.11530	8.17879	34.08764	58.23683	1.07187	6.13597	41.52556
14	1.49090	6.94525	2622.80647	47.60958	0.87627	4.90128	1632.76906
15	4.43477	10.28001	138.42247	281.23306	5.17619	13.36233	281.23306
16	3.19887	9.43506	34.86981	67.68490	1.24576	6.83732	51.12301
17	3.10226	9.72944	35.24454	62.00559	1.14123	6.18996	25.04870
18	3.15087	10.16907	34.52354	60.11701	1.10647	6.09057	48.36792
19	3.07053	10.47979	35.51657	50.06502	0.92146	5.68428	39.32444
20	3.21321	11.01355	34.83285	45.19817	0.83189	5.35691	36.68421
21	3.12774	11.31920	34.80099	62.01506	1.14141	6.41214	47.33216
22	3.13258	11.71517	35.85547	48.35569	0.89000	5.60701	43.94789
23	3.03483	12.00859	34.82417	58.21621	1.07149	5.91895	37.34422
24	3.15385	12.51879	35.08536	38.74216	0.71306	4.98232	33.49320
25	3.24237	12.99851	34.89096	53.74530	0.98920	5.73822	45.38408
26	3.06979	13.21714	34.64242	30.86886	0.56815	4.35428	35.73175
27	3.17615	13.71474	34.47660	44.95541	0.82742	5.30118	36.96202
28	3.12410	14.05393	34.63993	48.13395	0.88592	5.54596	36.39082
29	3.28342	14.60451	34.27275	47.02738	0.86555	5.30059	37.83076
30	2.83082	14.54318	33.34951	59.16186	1.08889	5.83174	41.62712

B.3 Critérios dos modelos MLP/RNA gerados com 6 neurônios na camada de entrada, aplicados a série do log-índice de produção

Oculto	AIC	BIC	EQM Aju	EQM Prev (12)	NMSE	MAPE	EQM Prev (1)
5	-5.48889	-4.40939	0.00589	0.00734	1.60670	1.58863	0.00603
6	-5.88444	-4.58864	0.00698	0.00313	0.68403	0.88978	0.00308
7	-5.46368	-3.95131	0.00619	0.00424	0.92700	1.22153	0.00736
8	-5.98672	-4.25757	0.00431	0.01328	2.90582	1.97178	0.00248
9	-4.38480	-2.43871	0.02038	0.03004	6.57536	3.39816	0.03004
10	-5.52640	-3.36324	0.00698	0.00774	1.69423	1.63205	0.00703
11	-5.80642	-3.42610	0.00516	0.00606	1.32714	1.45061	0.00389
12	-6.32984	-3.73227	0.00503	0.00491	1.07459	1.25590	0.00317
13	-6.20647	-3.39157	0.00406	0.00274	0.59876	0.85570	0.00343
14	-6.41220	-3.37992	0.00417	0.00242	0.53027	0.89821	0.00493
15	-6.88015	-3.63044	0.00397	0.00242	0.52955	0.83276	0.00171
16	-7.32459	-3.85740	0.00769	0.00595	1.30190	1.35440	0.00629
17	-6.16365	-2.47894	0.00399	0.00326	0.71316	0.94335	0.00222
18	-6.61488	-2.71262	0.00569	0.00438	0.95934	1.23754	0.00343
19	-7.27125	-3.15140	0.00325	0.00387	0.84738	0.89851	0.00263
20	-7.15519	-2.81772	0.02723	0.00639	1.39839	1.39454	0.00286
21	-6.96751	-2.41241	0.00319	0.00342	0.74905	0.99208	0.00217
22	-7.01926	-2.24650	0.00661	0.00532	1.16406	1.01586	0.00551
23	-7.01990	-2.02945	0.00442	0.00326	0.71258	0.87723	0.00690
24	-7.68430	-2.47614	0.00317	0.00353	0.77355	1.11496	0.00385
25	-7.39256	-1.96668	0.03402	0.00378	0.82842	1.16878	0.00205
26	-6.53575	-0.89214	0.00373	0.00232	0.50680	0.79817	0.00214
27	-7.68272	-1.82136	0.00307	0.00440	0.96216	1.23065	0.00319
28	-6.34605	-0.26692	0.00391	0.00442	0.96840	1.19756	0.00355
29	-7.32252	-1.02560	0.00626	0.00585	1.28119	1.15207	0.01956
30	-7.67788	-1.16317	0.00718	0.00378	0.82715	1.12976	0.00444

B.4 Critérios dos modelos MLP/RNA gerados com 12 neurônios na camada de entrada, aplicados a série do log-índice de produção

Oculto	AIC	BIC	EQM Aju	EQM Prev (12)	NMSE	MAPE	EQM Prev (1)
5	-14.90402	-12.96347	0.02183	0.00824	1.80344	1.67136	0.01176
6	-14.24399	-11.91373	0.02183	0.00811	1.77552	1.64473	0.01169
7	-16.48110	-13.76084	0.02184	0.00822	1.79898	1.66165	0.01180
8	-16.06723	-12.95675	0.02184	0.00826	1.80782	1.67000	0.01181
9	-12.63585	-9.13498	0.02185	0.00804	1.76007	1.63264	0.01166
10	-11.53201	-7.64062	0.02184	0.00781	1.71010	1.60941	0.01118
11	-14.43245	-10.15044	0.02178	0.00810	1.77324	1.65476	0.01166
12	-15.22558	-10.55287	0.02185	0.00819	1.79326	1.66737	0.01174
13	-15.07690	-10.01340	0.02175	0.00817	1.78937	1.65537	0.01172
14	-16.69142	-11.23708	0.02185	0.00827	1.80932	1.67401	0.01187
15	-15.37474	-9.52950	0.02182	0.00827	1.80943	1.67974	0.01178
16	-16.71253	-10.47634	0.02185	0.00827	1.80922	1.67922	0.01185
17	-16.70273	-10.07555	0.02186	0.00829	1.81436	1.68165	0.01186
18	-16.69326	-9.67505	0.02185	0.00828	1.81145	1.67805	0.01188
19	-16.70165	-9.29239	0.02185	0.00826	1.80815	1.67232	0.01184
20	-15.21266	-7.41231	0.02171	0.00824	1.80457	1.66425	0.01180
21	-16.16437	-7.97291	0.02184	0.00823	1.80175	1.66618	0.01183
22	-13.64199	-5.05939	0.02187	0.00818	1.79142	1.65659	0.01172
23	-16.07159	-7.09783	0.02180	0.00825	1.80515	1.66586	0.01183
24	-16.69544	-7.33050	0.02186	0.00827	1.80971	1.67833	0.01183
25	-16.74014	-6.98400	0.02185	0.00823	1.80183	1.66774	0.01182
26	-13.71297	-3.56562	0.02179	0.00814	1.78124	1.65509	0.01167
27	-16.75294	-6.21435	0.02186	0.00827	1.80990	1.68078	0.01183
28	-16.69658	-5.76675	0.02185	0.00826	1.80693	1.67193	0.01185
29	-16.68539	-5.36430	0.02179	0.00827	1.81074	1.67540	0.01187
30	-16.79308	-5.08072	0.02183	0.00823	1.80178	1.66670	0.01184

B.5 Critérios dos modelos MLP/RNA gerados com 6 neurônios na camada de entrada, aplicados a série das manchas solares

Oculto	AIC	BIC	EQM Aju	EQM Prev (12)	NMSE	MAPE	EQM Prev (1)
5	6.27460	7.09475	756.04262	1632.32380	0.56553	38.28739	2093.63612
6	5.71776	6.70197	526.84283	1331.53543	0.46132	29.88434	1059.56745
7	6.05565	7.20412	619.79758	1100.06184	0.38113	36.36252	1100.06184
8	6.14703	7.45990	608.68745	1963.05420	0.68012	35.38916	1791.09627
9	5.85045	7.32784	561.01183	1668.07946	0.57792	43.50721	1748.16330
10	6.10899	7.75099	636.76254	1381.16235	0.47852	32.92682	1647.22265
11	6.30817	8.11486	805.82425	1547.16330	0.53603	38.91257	1953.01888
12	5.13102	7.10244	300.69023	1803.65661	0.62489	42.44380	810.83799
13	4.57392	6.71013	172.29690	1058.43552	0.36670	54.01259	445.13853
14	4.92858	7.22963	212.38472	1134.81988	0.39317	27.57182	371.34510
15	4.53821	7.00413	219.45324	1344.29822	0.46574	46.27304	440.43794
16	4.64262	7.27344	287.28620	1009.36924	0.34970	29.94944	667.94896
17	5.12271	7.91847	243.98117	949.50327	0.32896	26.54647	398.40040
18	5.08681	8.04752	301.14228	1453.25473	0.50349	34.91629	872.29252
19	5.51464	8.64033	419.47466	1050.10409	0.36382	29.88872	1051.70700
20	5.15650	8.44720	269.24039	875.33393	0.30327	29.77126	581.34337
21	4.70473	8.16044	180.40128	1234.67116	0.42776	34.49101	670.69816
22	4.03611	7.65686	157.08481	920.78818	0.31902	28.49488	860.42129
23	5.64939	9.43519	555.74911	1294.63947	0.44854	39.30523	1423.89295
24	6.07583	10.02669	671.42226	994.83852	0.34467	29.63508	2112.92102
25	4.87809	8.99403	220.04786	604.02977	0.20927	28.44793	327.12889
26	5.19527	9.47629	288.98834	1062.01618	0.36794	47.05179	848.17373
27	4.99274	9.43887	239.91555	898.21475	0.31119	61.82467	742.69739
28	5.48799	10.09923	367.66226	998.13191	0.34581	29.54300	439.10726
29	5.25802	10.03438	273.44570	1316.83872	0.45623	26.63696	509.06968
30	5.89442	10.83590	511.56454	1251.31318	0.43353	43.28403	1315.16086

B.6 Critérios dos modelos MLP/RNA gerados com 12 neurônios na camada de entrada, aplicados a série das manchas solares

Oculto	AIC	BIC	EQM Aju	EQM Prev (12)	NMSE	MAPE	EQM Prev (1)
5	6.43906	7.90184	837.99988	2207.30617	0.76233	53.71890	6348.15788
6	6.69801	8.45421	1065.92104	2826.08513	0.97603	105.18379	5251.46260
7	6.70318	8.75300	1023.92462	2932.82239	1.01290	113.96570	5778.62524
8	3.73117	6.07477	220.26452	2939.24334	1.01511	116.52219	7901.50668
9	5.71098	8.34848	495.25521	2092.96476	0.72284	58.43762	7197.23422
10	6.40135	9.33284	823.76331	2494.66876	0.86157	122.02103	6342.72969
11	6.33804	9.56359	742.20686	1691.92029	0.58433	49.44481	6255.18410
12	6.39649	9.91617	882.86499	1400.24579	0.48360	45.98515	6029.11583
13	6.27803	10.09188	758.86456	1846.82104	0.63783	51.49264	6328.53395
14	6.47895	10.58703	841.74755	2429.30173	0.83900	87.67074	6605.84896
15	6.27431	10.67665	769.03499	1774.82079	0.61296	45.54352	6305.11992
16	6.04585	10.74248	702.07451	1352.34591	0.46705	59.65263	6920.00956
17	5.75476	10.74572	549.99932	2008.29193	0.69360	81.12047	7358.82805
18	6.10908	11.39438	711.88562	1774.16480	0.61274	86.46458	7110.27902
19	5.80569	11.38536	523.13882	1644.39627	0.56792	100.70959	5810.54553
20	6.11053	11.98460	662.41784	1973.12444	0.68145	49.00735	6665.71794
21	5.81465	11.98313	594.85549	2024.12546	0.69906	85.29284	7323.14440
22	6.16853	12.63144	711.78790	1521.55892	0.52549	42.88905	6784.06756
23	6.21941	12.97676	673.94550	1785.19104	0.61654	48.03147	6748.23626
24	5.95950	13.01131	592.54521	1520.04293	0.52497	65.28888	7125.48319
25	6.21005	13.55633	790.68509	1688.27607	0.58307	44.08893	6562.10227
26	6.16616	13.80692	641.84789	1561.28837	0.53922	63.84311	6739.82801
27	6.00340	13.93865	714.45749	1940.79016	0.67028	50.45832	6731.06650
28	6.34852	14.57828	777.44413	1115.36559	0.38521	42.70185	5425.13905
29	6.67744	15.20171	1115.27024	2142.42109	0.73992	52.33541	4781.55673
30	6.32607	15.14487	825.78863	1796.37630	0.62041	80.87845	5695.50803

ANEXO A – TABELAS DAS SÉRIES

A.1 Série mensal do índice de produção física industrial do Estado do Rio Grande do Sul

Ano	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1991	60.0	58.9	62.4	77.8	81.7	78.0	76.3	72.7	66.5	71.8	61.4	54.6
1992	58.7	71.2	77.9	78.4	80.4	78.0	78.2	74.8	73.1	69.6	70.3	71.6
1993	69.6	73.2	93.8	89.1	95.3	92.9	92.2	89.8	84.7	85.0	84.0	77.9
1994	76.0	71.3	95.6	91.0	100.6	97.8	89.6	93.5	99.0	96.9	99.6	94.8
1995	90.2	83.5	106.7	94.7	89.4	92.7	81.2	78.0	73.4	80.4	81.7	74.7
1996	72.8	72.0	85.8	92.2	92.9	82.3	90.7	89.7	85.6	92.6	88.9	80.7
1997	83.5	77.9	92.6	107.4	100.7	99.2	103.4	93.9	94.3	96.6	89.0	79.2
1998	73.2	78.6	99.3	98.1	98.7	96.3	96.5	90.0	93.2	91.4	84.5	75.8
1999	71.4	75.5	98.4	97.3	96.5	98.2	96.3	95.5	92.6	96.9	94.2	86.9
2000	81.9	89.7	107.6	100.3	108.5	104.8	105.2	108.1	95.8	103.7	101.4	88.1
2001	88.5	87.2	106.9	104.2	106.0	101.2	103.5	107.1	91.9	102.3	97.6	86.1
2002	89.1	87.1	99.8	108.2	109.5	99.3	103.1	102.3	99.8	108.9	104.4	88.4
2003	92.5	91.0	103.6	107.8	108.3	95.6	98.2	95.3	100.5	109.3	101.7	91.9
2004	92.6	92.3	115.4	111.6	113.3	111.0	115.3	108.5	101.5	111.7	105.0	94.2
2005	91.4	90.9	107.8	107.0	110.8	108.3	105.7	108.9	97.4	103.7	101.4	93.7
2006	89.2	89.4	106.6	97.5	108.4	101.2	103.2	106.2	98.6	105.4	103.3	93.6
2007	94.4	94.4	114.3	112.4	118.5	108.1	112.4	113.0	100.2	115.5	110.4	98.5
2008	102.9	105.9	113.2	120.9	114.6	115.7	119.4					

Fonte: Instituto Brasileiro de Geografia e Estatística - IBGE

A.2 Série mensal do logaritmo do índice de produção física industrial do Estado do Rio Grande do Sul

Ano	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1991	4.09	4.08	4.13	4.35	4.40	4.36	4.34	4.29	4.20	4.27	4.12	4.00
1992	4.07	4.26	4.36	4.36	4.39	4.36	4.36	4.31	4.29	4.24	4.25	4.27
1993	4.24	4.29	4.54	4.49	4.56	4.53	4.52	4.50	4.44	4.44	4.43	4.36
1994	4.33	4.27	4.56	4.51	4.61	4.58	4.50	4.54	4.59	4.57	4.60	4.55
1995	4.50	4.42	4.67	4.55	4.49	4.53	4.40	4.36	4.30	4.39	4.40	4.31
1996	4.29	4.28	4.45	4.52	4.53	4.41	4.51	4.50	4.45	4.53	4.49	4.39
1997	4.43	4.36	4.53	4.68	4.61	4.60	4.64	4.54	4.55	4.57	4.49	4.37
1998	4.29	4.36	4.60	4.59	4.59	4.57	4.57	4.50	4.53	4.52	4.44	4.33
1999	4.27	4.32	4.59	4.58	4.57	4.59	4.57	4.56	4.53	4.57	4.54	4.47
2000	4.41	4.50	4.68	4.61	4.69	4.65	4.66	4.68	4.56	4.64	4.62	4.48
2001	4.48	4.47	4.67	4.65	4.66	4.62	4.64	4.67	4.52	4.63	4.58	4.46
2002	4.49	4.47	4.60	4.68	4.70	4.60	4.64	4.63	4.60	4.69	4.65	4.48
2003	4.53	4.51	4.64	4.68	4.68	4.56	4.59	4.56	4.61	4.69	4.62	4.52
2004	4.53	4.52	4.75	4.72	4.73	4.71	4.75	4.69	4.62	4.72	4.65	4.54
2005	4.51	4.51	4.68	4.67	4.71	4.69	4.66	4.69	4.58	4.64	4.62	4.54
2006	4.49	4.49	4.67	4.58	4.69	4.62	4.64	4.66	4.59	4.66	4.64	4.54
2007	4.55	4.55	4.74	4.72	4.77	4.68	4.72	4.73	4.61	4.75	4.70	4.59
2008	4.63	4.66	4.73	4.80	4.74	4.75	4.78					

Fonte: Instituto Brasileiro de Geografia e Estatística - IBGE

A.3 Série do número de manchas solares - periodicidade anual - do ano de 1700 ao ano de 1988

Ano	Valor	Ano	Valor	Ano	Valor	Ano	Valor
1700	5	1741	40	1782	38	1823	2
1701	11	1742	20	1783	23	1824	8
1702	16	1743	16	1784	10	1825	17
1703	23	1744	5	1785	24	1826	36
1704	36	1745	11	1786	83	1827	50
1705	58	1746	22	1787	132	1828	64
1706	29	1747	40	1788	131	1829	67
1707	20	1748	60	1789	118	1830	71
1708	10	1749	81	1790	90	1831	48
1709	8	1750	83	1791	67	1832	28
1710	3	1751	48	1792	60	1833	8
1711	0	1752	48	1793	47	1834	13
1712	0	1753	31	1794	41	1835	57
1713	2	1754	12	1795	21	1836	122
1714	11	1755	10	1796	16	1837	138
1715	27	1756	10	1797	6	1838	103
1716	47	1757	32	1798	4	1839	86
1717	63	1758	48	1799	7	1840	65
1718	60	1759	54	1800	14	1841	37
1719	39	1760	63	1801	34	1842	24
1720	28	1761	86	1802	45	1843	11
1721	26	1762	61	1803	43	1844	15
1722	22	1763	45	1804	48	1845	40
1723	11	1764	36	1805	42	1846	62
1724	21	1765	21	1806	28	1847	98
1725	40	1766	11	1807	10	1848	125
1726	78	1767	38	1808	8	1849	96
1727	122	1768	70	1809	2	1850	67
1728	103	1769	106	1810	0	1851	64
1729	73	1770	101	1811	1	1852	54
1730	47	1771	82	1812	5	1853	39
1731	35	1772	66	1813	12	1854	21
1732	11	1773	35	1814	14	1855	7
1733	5	1774	31	1815	35	1856	4
1734	16	1775	7	1816	46	1857	23
1735	34	1776	20	1817	41	1858	55
1736	70	1777	92	1818	30	1859	94
1737	81	1778	154	1819	24	1860	96
1738	111	1779	126	1820	16	1861	77
1739	101	1780	85	1821	7	1862	59
1740	73	1781	68	1822	4	1863	44

Série do número de manchas solares - periodicidade anual - do ano de 1700 ao ano de 1988 - continuação

Ano	Valor	Ano	Valor	Ano	Valor	Ano	Valor
1864	47	1905	64	1946	93	1987	29
1865	30	1906	54	1947	152	1988	100
1866	16	1907	62	1948	136		
1867	7	1908	48	1949	135		
1868	38	1909	44	1950	84		
1869	74	1910	19	1951	69		
1870	139	1911	6	1952	32		
1871	111	1912	4	1953	14		
1872	102	1913	1	1954	4		
1873	66	1914	10	1955	38		
1874	45	1915	47	1956	142		
1875	17	1916	57	1957	190		
1876	11	1917	104	1958	185		
1877	12	1918	81	1959	159		
1878	3	1919	64	1960	112		
1879	6	1920	38	1961	54		
1880	32	1921	26	1962	38		
1881	54	1922	14	1963	28		
1882	60	1923	6	1964	10		
1883	64	1924	17	1965	15		
1884	64	1925	44	1966	47		
1885	52	1926	64	1967	94		
1886	25	1927	69	1968	106		
1887	13	1928	78	1969	106		
1888	7	1929	65	1970	104		
1889	6	1930	36	1971	67		
1890	7	1931	21	1972	69		
1891	36	1932	11	1973	38		
1892	73	1933	6	1974	34		
1893	85	1934	9	1975	16		
1894	78	1935	36	1976	13		
1895	64	1936	80	1977	28		
1896	42	1937	114	1978	92		
1897	26	1938	110	1979	155		
1898	27	1939	89	1980	155		
1899	12	1940	68	1981	140		
1900	10	1941	48	1982	116		
1901	3	1942	31	1983	67		
1902	5	1943	16	1984	46		
1903	24	1944	10	1985	18		
1904	42	1945	33	1986	13		

Fonte: pacote estatístico R (R Development Core Team, 2007)